This book is dedicated to OpenNA staff. Thanks, guys (no-gender)!!

--Gerhard Mourani

Title: Securing and Optimizing Linux: The Ultimate Solution
Page Count: 855
Version: 2.0
Last Revised: 2001-06-10
Publisher: Open Network Architecture, Inc.
Editor: Ted Nackad
Text Design & Drawings (Graphics): Bruno Mourani
Printing History: June 2000: First Publication.

Author's: Gerhard Mourani
Mail: **gmourani@openna.com**
Website: **http://www.openna.com/**

National Library Act. R.S., c. N-11, s. 1.

Legal Deposit, 2001

Securing and Optimizing Linux: The Ultimate Solution / Open Network Architecture.
Published by Open Network Architecture, Inc., 11090 Drouart, Montreal, H3M 2S3, Canada.

Includes Index.
ISBN 0-9688793-0-6

Latest version of this book

New version of this book (version 3.0 title "Securing & Optimizing Linux: The Hacking Solution") is available on our website but not as a free document. If you like this book and are interested to get the latest version, then go to http://www.openna.com/.

Overview

Contents

Organization of the Book

Securing and Optimizing Linux: Red Hat Edition has 31 chapters, organized into thirteen parts and four appendixes:

- **Part I: Installation Related Reference** includes two chapters; the first chapter introduces Linux in general and gives some basic information to the new Linux reader who is not familiar with this operating system. The second chapter guides you through the steps of installing Linux (from CD) in the most secure manner, with only the essential and critical software for a clean and secure installation.

- **Part II: Security and Optimization Related Reference** focuses on how to secure and tune Linux after it has been installed. Part II includes four chapters that explain how to protect your Linux system, how to use and apply Pluggable Authentication Modules (`PAM`), how to optimize your system for your specific processor, and memory. Finally, the last chapter describes how to install, optimize, protect and customize the Kernel. All information in part II of the book applies to the whole system.

- **Part III: Networking Related Reference** contains three chapters, where the first chapter answers fundamental questions about network devices, network configuration files, and network security as well as essential networking commands. The second and third chapters provide information about firewalls as well as the popular masquerading feature of Linux and how to configure and customize the new powerful `IPTABLES` tool of this system to fit your personal needs.

- **Part IV: Cryptography & Authentication Related Reference** handle three chapters which talk about essential security tools needed to secure network communication. These tools are the minimum that should be installed on any type of Linux server.

- **Part V: Monitoring & System Integrity Related Reference** provides five chapters which help you to tighten security in your server by the use of some powerful security software.

- **Part VI: Management & Limitation Related Reference** presently includes just one chapter which is about limiting users space usage on the server.

- **Part VII: Domain Name System Related Reference** will discuss the Domain Name System, which is an essential service to install in all Linux servers you want on the network. This part of the book is important and must be read by everyone.

- **Part VIII: Mail Transfer Agent Related Reference** will explain everything about installing and configuring a Mail Server and the minimum mail software to install. It is one of the most important parts of the book.

- **Part IX: Internet Message Access Protocol Related Reference** is the last required part to read before going into installation of specific services in your Linux system. It discusses the mail software required to allow your users to get and read their electronic mail.

- **Part X: Database Server Related Reference** contains three chapters about the most commonly used and powerful databases on *NIX systems.

- **Part XI: Gateway Server Related Reference** discusses installing a powerful proxy server and configuring encrypted network services.

- **Part XII: Other Server Related Reference** shows you how to use Linux for specific purposes such as setting up a customized FTP server, running a World Wide Web server and sharing files between different systems, all in a secure and optimized manner.

- **Part XIII: Backup Related reference** describes how to make a reliable backup of your valuable files in a convenient way. This part includes a chapter that explains how to perform backups with the traditional and universal UNIX tools "`tar`", and "`dump`", which enables you to use the same procedures, without any modification, with the other Unix family platforms.

- **Appendixes** is as follow:

    - **Appendix A: Tweaks, Tips and Administration Tasks** has several useful Linux tips on administration, networking and shell commands.

    - **Appendix B: Contributor Users** lists Linux users around the world who have participated in a voluntary basis by providing good suggestions, recommendations, help, tips, corrections, ideas and other information to help in the development of this book. Thanks to all of you.

    - **Appendix C: Obtaining Requests for Comments (RFCs)** provides an alphabetical reference for important RFCs related to the software or protocols described in the book.

Steps of installation

Depending of your level of knowledge in Linux, you can read this book from the beginning through to the end or the chapters that interest you. Each chapter and section of this book appears in a manner that lets you read only the parts of your interest without the need to schedule one day of reading. Too many books on the market take myriad pages to explain something that can be explained in two lines, I'm sure that a lot of you agree with my opinion. This book tries to be different by talking about only the essential and important information that the readers want to know by eliminating all the nonsense.

Although you can read this book in the order you want, there is a particular order that you could follow if something seems to be confusing you. The steps shown below are what I recommend :

- ✓ Setup Linux in your computer.
- ✓ Remove all the unnecessary RPM's packages.
- ✓ Install the necessary RPM's packages for compilation of software (if needed).
- ✓ Secure the system in general.
- ✓ Optimize the system in general.
- ✓ Reinstall, recompile and customize the Kernel to fit your specific system.
- ✓ Configure firewall script according to which services will be installed in your system.
- ✓ Install `OpenSSL` to be able to use encryption with the Linux server.
- ✓ Install `OpenSSH` to be able to make secure remote administration tasks.
- ✓ Install `sXid`.
- ✓ Install `Logcheck`.
- ✓ Install `PortSentry`.
- ✓ Install `Tripwire`.
- ✓ Install `ICS BIND/DNS`.
- ✓ Install `Sendmail` or `qmail`.
- ✓ Install any software you need after to enable specific services into the server.

Author note

According to some surveys on the Internet, Linux will be the number one operating system for a server platform in year 2003. Presently it is number two and no one at one time thought that it would be in this second place. Many organizations, companies, universities, governments, and the military, etc, kept quiet about it. Crackers use it as the operating system by excellence to crack computers around the world. Why do so many people use it instead of other well know operating systems? The answer is simple, Linux is free and the most powerful, reliable, and secure operating system in the world, providing it is well configured. Millions of programmers, home users, hackers, developers, etc work to develop, on a voluntary basis, different programs related to security, services, and share their work with other people to improve it without expecting anything in return. This is the revolution of the Open Source movement that we see and hear about so often on the Internet and in the media.

If crackers can use Linux to penetrate servers, security specialists can use the same means to protect servers (to win a war, you should at least have equivalent weapons to what your enemy may be using). When security holes are encountered, Linux is the one operating system that has a solution and that is not by chance. Now someone may say: with all these beautiful features why is Linux not as popular as other well know operating system? There are many reasons and different answers on the Internet. I would just say that like everything else in life, anything that we are to expect the most of, is more difficult to get than the average and easier to acquire. Linux and *NIX are more difficult to learn than any other operating system. It is only for those who want to know computers in depth and know what they doing. People prefer to use other OS’s, which are easy to operate but hard to understand what is happening in the background since they only have to click on a button without really knowing what their actions imply. Every UNIX operating system like Linux will lead you unconsciously to know exactly what you are doing because if you pursue without understanding what is happening by the decision you made, then nothing will surely work as expected. This is why with Linux, you will know the real meaning of a computer and especially a server environment where every decision warrants an action which will closely impact on the security of your organization and employees.

Many Web sites are open to all sorts of "web hacking." According to the Computer Security Institute and the FBI's joint survey, 90% of 643 computer security practitioners from government agencies, private corporations, and universities detected cyber attacks last year. Over $265,589,940 in financial losses was reported by 273 organizations.

Many readers of the previous version of this book told me that the book was an easy step by step guide for newbies, I am flattered but I prefer to admit that it was targeting for a technical audience and I assumed the reader had some background in Linux, UNIX systems. If this is not true in your case, I highly recommend you to read some good books in network administration related to UNIX and especially to Linux before venturing into this book. Remember talking about security and optimization is a very serious endeavor. It is very important to be attentive and understand every detail in this book and if difficulties arise, try to go back and reread the explanation will save a lot of frustration. Once again, security is not a game and crackers await only one single error from your part to enter your system. A castle has many doors and if just one stays open, will be enough to let intruders into your fortress. You have been warned.

Many efforts went into the making of this book, making sure that the results were as accurate as possible. If you find any abnormalities, inconsistent results, errors, omissions or anything else that doesn't look right, please let me know so I can investigate the problem and/or correct the error. Suggestions for future versions are also welcome and appreciated. A web site dedicated to this book is available on the Internet for your convenience. If you any have problem, question, recommendation, etc, please go to the following URL: http://www.openna.com/ We made this site for you.

Audience

This book is intended for a technical audience and system administrators who manage Linux servers, but it also includes material for home users and others. It discusses how to install and setup a Linux Server with all the necessary security and optimization for a high performance Linux specific machine. It can also be applied with some minor changes to other Linux variants without difficulty. Since we speak of optimization and security configuration, we will use a source distribution (`tar.gz`) program for critical server software like `Apache`, `ISC BIND/DNS`, `Samba`, `Squid`, `OpenSSL` etc. Source packages give us fast upgrades, security updates when necessary, and better compilation, customization, and optimization options for specific machines that often aren't available with RPM packages.

These installation instructions assume

You have a CD-ROM drive on your computer and the Official Red Hat Linux CD-ROM. Installations were tested on the Official Red Hat Linux version 7.1.

You should familiarize yourself with the hardware on which the operating system will be installed. After examining the hardware, the rest of this document guides you, step-by-step, through the installation process.

About products mentioned in this book

Many products will be mentioned in this book— some commercial, but most are not, cost nothing and can be freely used or distributed. It is also important to say that I'm not affiliated with any specific brand and if I mention a tool, it's because it is useful. You will find that a lot of big companies in their daily tasks, use most of them.

Obtaining the example configuration files

In a true server environment and especially when Graphical User Interface is not installed, we will often use text files, scripts, shell, etc. Throughout this book we will see shell commands, script files, configuration files and many other actions to execute on the terminal of the server. You can enter them manually or use the compressed archive file that I made which contains all configuration examples and paste them directly to your terminal. This seems to be useful in many cases to save time.

The example configuration files in this book are available electronically via HTTP from this URL:

ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz

- In either case, extract the files into your Linux server from the archive by typing:

```
[root@deep /]# cd /var/tmp
[root@deep tmp]# tar xzpf floppy-2.0.tgz
```

If you cannot get the examples from the Internet, please contact the author at this email address:

gmourani@openna.com

Problem with Securing & Optimizing Linux

When you encounter a problem in "Securing & Optimizing Linux" we want to hear about it. Your reports are an important part in making the book more reliable, because even with the utmost care we cannot guarantee that every part of the book will work on every platform under every circumstance.

We cannot promise to fix every error right away. If the problem is obvious, critical, or affects a lot of users, chances are that someone will look into it. It could also happen that we tell you to update to a newer version to see if the problem persists there. Or we might decide that the problem cannot be fixed until some major rewriting has been done. If you need help immediately, consider obtaining a commercial support contract or try our Q&A archive from the mailing list for an answer.

Below are some important links:

OpenNA.com web site: http://www.openna.com/
Mailing list: http://www.openna.com/support/mailing/mailing.php
Errata: http://www.openna.com/products/books/errata/errata.php
Support: http://www.openna.com/support/support.php
RPM Download: http://www.openna.com/downloads/downloads.php

Acknowledgments

First of all, I would like to thank my younger brother Bruno Mourani for his valuable help that he brought by drawing all the networking drafts shown in this book. For your information he has made all the schemas by hand and without any special diagram software. Yes, he is a natural better than me in many computer areas but don't take the time to profit of his skill.

A special gratitude and many thanks to Colin Henry who made tremendous efforts to make this book grammatically and orthographically sound in a professional manner. Gregory A Lundberg and the WU-FTPD Development Group for their help, recommendations on the `FTP` chapter in this book. Werner Puschitz for his help in the `PAM` chapter of this book and his recommendation with `SSH` software (thanks Werner). OpenNA who has decided to publish my book and all Linux users around the world who have participated by providing good comments, ideas, recommendations and suggestions (a dedicated section has been made for them at the end of this book).

Part I Installation Related Reference

In this Part

Installation - Introduction
Installation - Installing a Linux Server

This part of the book deals with all the basic knowledge required to properly install a Linux OS, in our case a Red Hat Linux on your system in the most secure and clean manner available.

1 Installation - Introduction

In this Chapter

Introduction

What is Linux?

Linux is an operating system that was first created at the University of Helsinki in Finland by a young student named Linus Torvalds. At this time the student was working on a UNIX system that was running on an expensive platform. Because of his low budget, and his need to work at home, he decided to create a copy of the UNIX system in order to run it on a less expensive platform, such as an IBM PC. He began his work in 1991 when he released version 0.02 and worked steadily until 1994 when version 1.0 of the Linux Kernel was released. The current full-featured version at this time is 2.2.X (released January 25, 1999), and development continues.

The Linux operating system is developed under the GNU General Public License (also known as GNU GPL) and its source code is freely available to everyone who downloads it via the Internet. The CD-ROM version of Linux is also available in many stores, and companies that provide it will charge you for the cost of the media and support. Linux may be used for a wide variety of purposes including networking, software development, and as an end-user platform. Linux is often considered an excellent, low-cost alternative to other more expensive operating systems because you can install it on multiple computers without paying more.

Some good reasons to use Linux

There are no royalty or licensing fees for using Linux, and the source code can be modified to fit your needs. The results can be sold for profit, but the original authors retain copyright and you must provide the source to your modifications.

Because it comes with source code to the kernel, it is quite portable. Linux runs on more CPUs and platforms than any other computer operating system.

The recent direction of the software and hardware industry is to push consumers to purchase faster computers with more system memory and hard drive storage. Linux systems are not affected by those industries' orientation because of it's capacity to run on any kind of computer, even aging x486-based computers with limited amounts of RAM.

Linux is a true multi-tasking operating system similar to it's brother, UNIX. It uses sophisticated, state-of-the-art memory management to control all system processes. That means that if a program crashes you can kill it and continue working with confidence.

Another benefit is that Linux is practically immunized against all kinds of viruses that we find in other operating systems. To date we have found only two viruses that were effective on Linux systems.

Let's dispel some of the fear, uncertainty, and doubt about Linux

It's a toy operating system.
Fortune 500 companies, governments, and consumers more and more use Linux as a cost-effective computing solution. It has been used and is still used by big companies like IBM, Amtrak, NASA, and others.

There's no support.
Every Linux distribution comes with more than 12,000 pages of documentation. Commercial Linux distributions such as Red Hat Linux, Caldera, SuSE, Mandrake, Turbo Linux and OpenLinux offer initial support for registered users, and small business and corporate accounts can get 24/7 supports through a number of commercial support companies. As an Open Source operating system, there's no six-month wait for a service release, plus the online Linux community fixes many serious bugs within hours.

Why choose Pristine source?

All the programs in Red Hat distributions of Linux are provided as RPM files. An RPM file, also known, as a “package”, is a way of distributing software so that it can be easily installed, upgraded, queried, and deleted. However, in the Unix world, the defacto-standard for package distribution continues to be by way of so-called “tarballs”. Tarballs are simply compressed files that can be readable and uncompressed with the “`tar`” utility. Installing from `tar` is usually significantly more tedious than using RPM. So why would we choose to do so?

1) Unfortunately, it takes a few weeks for developers and helpers to get the latest version of a package converted to RPM’s because many developers first release them as tarballs.

2) When developers and vendors release a new RPM, they include a lot of options that often are not necessary. Those organization and companies don’t know what options you will need and what you will not, so they include the most used to fit the needs of everyone.

3) Often RPMs are not optimized for your specific processors; companies like Red Hat Linux build RPM’s based on a standard PC. This permits their RPM packages to be installed on all sorts of computers since compiling a program for an i386 machine means it will work on all systems.

4) Sometimes you download and install RPM’s, which other people around the world are building and make available for you to use. This can pose conflicts in certain cases depending how this individual built the package, such as errors, security and all the other problems described above.

Compiling software on your system

A program is something a computer can execute. Originally, somebody wrote the "source code" in a programming language he/she could understand (e.g., C, C++). The program "source code" also makes sense to a compiler that converts the instructions into a binary file suited to whatever processor is wanted (e.g. a 386 or similar). A modern file format for these "executable" programs isELF. The programmer compiles his source code on the compiler and gets a result of some sort. It's not at all uncommon that early attempts fail to compile, or having compiled, fail to act as expected. Half of programming is tracking down and fixing these problems (debugging).

For the beginners there are more aspect and new words relating to the compilation of source code that you must know, these include but are not limited to:

Multiple Files (Linking)
One-file programs are quite rare. Usually there are a number of files (say `*.c`, `*.cpp`, etc) that are each compiled into object files (`*.o`) and then linked into an executable. The compiler is usually used to perform the linking and calls the '`ld`' program behind the scenes.

Makefiles
Makefiles are intended to aid you in building your program the same way each time. They also often help with increasing the speed of a program. The "`make`" program uses "dependencies" in the Makefile to decide what parts of the program need to be recompiled. If you change one source file out of fifty you hope to get away with one compile and one link step, instead of starting from scratch.

Libraries
Programs can be linked not only to object files (`*.o`) but also to libraries that are collections of object files. There are two forms of linking to libraries: static, where the code goes in the executable file, and dynamic, where the code is collected when the program starts to run.

Patches
It was common for executable files to be given corrections without recompiling them. Now this practice has died out; in modern days, people change a small portion of the source code, putting a change into a file called a "patch". Where different versions of a program are required, small changes to code can be released this way, saving the trouble of having two large distributions.

Errors in Compilation and Linking
Errors in compilation and linking are often due to typos, omissions, or misuse of the language. You have to check that the right "includes file" is used for the functions you are calling. Unreferenced symbols are the sign of an incomplete link step. Also check if the necessary development libraries (`GLIBC`) or tools (`GCC`, `DEV86`, `MAKE`, etc) are installed on your system.

Debugging
Debugging is a large topic. It usually helps to have statements in the code that inform you of what is happening. To avoid drowning in output you might sometimes get them to print out only the first 3 passes in a loop. Checking that variables have passed correctly between modules often helps. Get familiar with your debugging tools.

Build & install software on your system

You will see in this book that we use many different compile commands to build and install programs on the server. These commands are UNIX compatible and are used on all variants of *NIX machines to compile and install software.

The procedure to compile and install software tarballs on your server are as follows:

1. First of all, you must download the tarball from your trusted software archive site. Usually from the main site of the software you hope to install.

2. After downloading the tarball change to the `/var/tmp` directory (note that other paths are possible, as personal discretion) and untar the archive by typing the commands (as root) as in the following example:

```
[root@deep /]# tar xzpf foo.tar.gz
```

The above command will extract all files from the example `foo.tar.gz` compressed archive and will create a new directory with the name of the software from the path where you executed the command.

The "x" option tells tar to extract all files from the archive.
The "z" option tells tar that the archive is compressed with gzip utility.
The "p" option maintains the original permissions the files had when the archive was created.
The "f" option tells tar that the very next argument is the file name.

Once the tarball has been decompressed into the appropriate directory, you will almost certainly find a "README" and/or an "INSTALL" file included with the newly decompressed files, with further instructions on how to prepare the software package for use. Likely, you will need to enter commands similar to the following example:

```
./configure
make
make install
```

The above commands **./configure** will configure the software to ensure your system has the necessary libraries to successfully compile the package, **make** will compile all the source files into executable binaries. Finally, **make install** will install the binaries and any supporting files into the appropriate locations. Other specifics commands that you'll see in this book for compilation and installation procedure will be:

```
make depend
strip
chown
```

The **make depend** command will build and make the necessary dependencies for different files. The **strip** command will discard all symbols from the object files. This means that our binary file will be smaller in size. This will improve the performance of the program, since there will be fewer lines to read by the system when it executes the binary. The **chown** command will set the correct file owner and group permissions for the binaries. More commands will be explained in the concerned installation sections.

Editing files with the vi editor tool

The vi program is a text editor that you can use to edit any text and particularly programs. During installation of software, the user will often have to edit text files, like Makefiles or configuration files. The following are some of the more important keystroke commands to get around in vi. I decided to introduce the vi commands now since it is necessary to use vi throughout this book.

Command	Result
`i`	Notifies `vi` to insert text before the cursor
`a`	Notifies `vi` to append text after the cursor
`dd`	Notifies `vi` to delete the current line
`x`	Notifies `vi` to delete the current character
`Esc`	Notifies `vi` to end the insert or append mode
`u`	Notifies `vi` to undo the last command
`Ctrl+f`	Scroll up one page
`Ctrl+b`	Scroll down one page
`/string`	Search forward for string
`:f`	Display filename and current line number
`:q`	Quit editor
`:q!`	Quit editor without saving changes
`:wq`	Save changes and exit editor

Recommended software to include in each type of servers

If you buy binaries, you will not get any equity and ownership of source code. Source code is a very valuable asset and binaries have no value. Buying software may become a thing of the past. You only need to buy good hardware; it is worth spending money on the hardware and get the software from Internet. Important point, is that it is the computer hardware that is doing the bulk of the job. Hardware is the real workhorse and software is just driving it. It is for this reason that we believe in working with and using the Open source software. Much of the software and services that come with Linux are open source and allow the user to use and modify them in an undiscriminating way according to the General Public License.

Linux has quickly become the most practical and friendly used platform for e-business -- and with good reason. Linux offers users stability, functionality and value that rivals any platform in the industry. Millions of users worldwide have chosen Linux for applications, from web and email servers to departmental and enterprise vertical application servers. To respond to your needs and to let you know how you can share services between systems I have developed ten different types of servers, which cover the majority of servers' functions and enterprise demands.

Often companies try to centralize many services into one server to save money, it is well known and often seen that there are conflicts between the technical departments and purchasing agents of companies about investment and expenditure when it comes to buying new equipment. When we consider security and optimization, it is of the utmost importance not to run too many services in one server, it is highly recommended to distribute tasks and services between multiple systems. The table below show you which software and services we recommend to for each type of Linux server.

The following conventions will explain the interpretations of these tables:

- **Optional Components**: components that may be included to improve the features of the server or to fit special requirements.
- **Security Software Required**: what we consider as minimum-security software to have installed on the server to improve security.
- **Security Software Recommended**: what we recommend for the optimal security of the servers.

Mail Server	Web Server	Gateway Server
Sendmail or qmail (SMTP Server) BIND/DNS (Caching) IPTABLES Firewall ---------- IMAP/POP only for Sendmail	Apache (Web Server) qmail (Standalone) BIND/DNS (Caching) IPTABLES Firewall	BIND/DNS (Caching) qmail (Standalone) IPTABLES Firewall ---------- Squid Proxy (Server)
Optional Components	Optional Components	Optional Components
	Mod_PHP4 Capability Mod_SSL Capability Mod-Perl Capability MM Capability Webmail Capability	
Security Software Required	Security Software Required	Security Software Required
Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Server) Tripwire Integrity Tool	Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Server) Tripwire Integrity Tool	Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Client & Server) Tripwire Integrity Tool
Security Software recommended	Security Software recommended	Security Software recommended
GnuPG sXid Logcheck PortSentry Quota	GnuPG sXid Logcheck PortSentry Quota	GnuPG sXid Logcheck PortSentry

FTP Server	Domain Name Server	File Sharing Server
Wu-FTPD (Server) qmail (Standalone) BIND/DNS (Caching) IPTABLES Firewall	Primary BIND/DNS (Server) qmail (Standalone) IPTABLES Firewall ---------- Secondary BIND/DNS (Server)	Samba LAN (Server) qmail (Standalone) BIND/DNS (Caching) IPTABLES Firewall
Optional Components	Optional Components	Optional Components
Anonymous FTP (Server)		
Security Software Required	Security Software Required	Security Software Required
Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Server) Tripwire Integrity Tool	Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Server) Tripwire Integrity Tool	Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Server) Tripwire Integrity Tool
Security Software recommended	Security Software recommended	Security Software recommended
GnuPG sXid Logcheck PortSentry Quota	GnuPG sXid Logcheck PortSentry	GnuPG sXid Logcheck PortSentry

Database server	Backup server	VPN Server
PostgreSQL (Client & Server) qmail (Standalone) BIND/DNS (Caching) IPTABLES Firewall ---------- MySQL (Client & Server) ---------- OpenLDAP (Client & Servers)	Amanda (Server) qmail (Standalone) BIND/DNS (Caching) Dump Utility IPTABLES Firewall	FreeS/WAN VPN (Server) qmail (Standalone) BIND/DNS (Caching) IPTABLES Firewall
Optional Components	Optional Components	Optional Components
Security Software Required	Security Software Required	Security Software Required
Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Server) Tripwire Integrity Tool	Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Client & Server) Tripwire Integrity Tool	Secure Linux Kernel Patches OpenSSL Encryption Software OpenSSH (Server) Tripwire Integrity Tool
Security Software recommended	Security Software recommended	Security Software recommended
GnuPG sXid Logcheck PortSentry	GnuPG sXid Logcheck PortSentry	GnuPG sXid Logcheck PortSentry

Some last comments

Before reading the rest of the book, it should be noted that the text assumes that certain files are placed in certain directories. Where they have been specified, the conventions we adopt here for locating these files are those of the Red Hat Linux distribution. If you are using a different distribution of Linux or some other operating system that chooses to distribute these files in a different way, you should be careful when copying examples directly from the text.

It is important to note that all software-listed from Part IV through Part IX of the book is required if you want to run a fully operational and secure Linux system. Without them, you will have one that it is not as secure as you expect it to be. Therefore I highly recommend you read at least Part IV through Part IX before going into the specific services you may want to install on your server.

2 Installation - Installing a Linux Server

In this Chapter

Linux Installation

Abstract
We have prepared and structured this chapter in a manner that follows the original installation of the Red Hat Linux operating system from CD-ROM. Each section below refers to, and will guide you through, the different screens that appear during the setup of your system after booting from the Red Hat boot diskette. We promise that it will be interesting to have the machine you want to install Linux on ready and near you when you follow the steps described below.

You will see that through the beginning of the installation of Linux, there are many options, parameters, and hacks that you can set before the system logs in for the first time.

Know your Hardware!
Understanding the hardware of your computer is essential for a successful installation of Linux. Therefore, you should take a moment and familiarize yourself with your computer hardware. Be prepared to answer the following questions:

1. How many hard drives do you have?
2. What size is each hard drive (eg, 15GB)?
3. If you have more than one hard drive, which is the primary one?
4. What kind of hard drive do you have (eg, IDE ATA/66, SCSI)?
5. How much RAM do you have (eg, 256MB RAM)?
6. Do you have a SCSI adapter? If so, who made it and what model is it?
7. Do you have a RAID system? If so, who made it and what model is it?
8. What type of mouse do you have (eg, PS/2, Microsoft, Logitech)?
9. How many buttons does your mouse have (2/3)?
10. If you have a serial mouse, what COM port is it connected to (eg, COM1)?
11. What is the make and model of your video card? How much video RAM do you have (eg, 8MB)?
12. What kind of monitor do you have (make and model)?
13. Will you be connected to a network? If so, what will be the following:
    a. Your IP address?
    b. Your netmask?
    c. Your gateway address?
    d. Your domain name server's IP address?
    e. Your domain name?
    f. Your hostname?
    g. Your types of network(s) card(s) (makes and model)?
    h. Your number of card(s) (makes and model)?

Creating the Linux Boot Disk
The first thing to do is to create an installation diskette, also known as a boot disk. If you have purchased the official Red Hat Linux CD-ROM, you will find a floppy disk named "Boot Diskette" in the Red Hat Linux box so you don't need to create it.

Sometimes, you may find that the installation will fail using the standard diskette image that comes with the official Red Hat Linux CD-ROM. If this happens, a revised diskette is required in order for the installation to work properly. In these cases, special images are available via the Red Hat Linux Errata web page to solve the problem (http://www.redhat.com/errata).

Since this, is a relatively rare occurrence, you will save time if you try to use the standard diskette images first, and then review the Errata only if you experience any problems completing the installation. Below, we will show you two methods to create the installation Boot Disk, the first method is to use an existing Microsoft Windows computer and the second using an existing Linux computer.

Making a Diskette Under MS-DOS

Before you make the boot disk, insert the Official Red Hat Linux CD-ROM Disk 1 in your computer that runs the Windows operating system. When the program asks for the filename, enter **`boot.img`** for the boot disk. To make the floppies under MS-DOS, you need to use these commands (assuming your CD-ROM is drive D: and contain the Official Red Hat Linux CD-ROM).

- Open the Command Prompt under Windows: Start | Programs | Command Prompt

```
C:\> d:
D:\> cd \dosutils
D:\dosutils> rawrite
Enter disk image source file name: ..\images\boot.img
Enter target diskette drive: a:
Please insert a formatted diskette into drive A: and press -ENTER- :

D:\dosutils>
```

The `rawrite.exe` program asks for the filename of the disk image: Enter **`boot.img`** and insert a blank floppy into drive A. It will then ask for a disk to write to: Enter `a:`, and when complete, label the disk “Red Hat boot disk”, for example.

Making a Diskette Under a Linux-Like OS

To make a diskette under Linux or any other variant of Linux-Like operating system, you must have permission to write to the device representing the floppy drive (known as `/dev/fd0H1440` under Linux).

This permission is granted when you log in the system as the super-user “root”. Once you have logged as “root”, insert a blank formatted diskette into the diskette drive of your computer without issuing a `mount` command on it. Now it’s time to mount the Red Hat Linux CD-ROM on Linux and change to the directory containing the desired image file to create the boot disk.

- Insert a blank formatted diskette into the diskette drive
  Insert the Red Hat Linux CD Part 1 into the CD-ROM drive

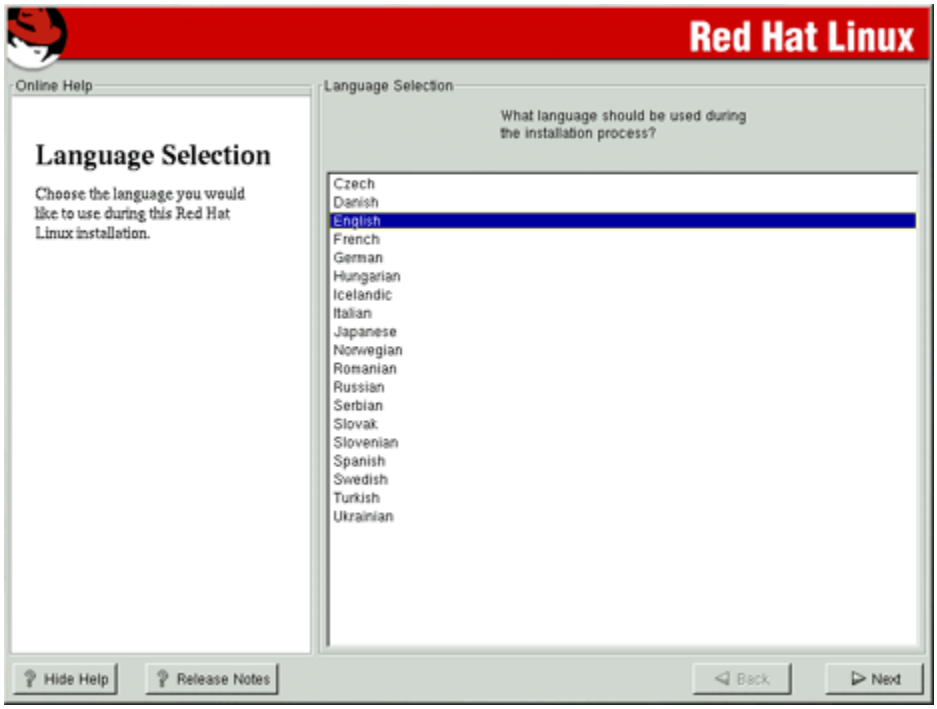
```
[root@deep /]# mount /dev/cdrom /mnt/cdrom
[root@deep /]# cd /mnt/cdrom/images/
[root@deep images]# dd if=boot.img of=/dev/fd0H1440 bs=1440k
1+0 records in
1+0 records out
[root@deep images]# cd /
[root@deep /]# umount /mnt/cdrom
```

Don’t forget to label the diskette “Red Hat boot disk”, for example.

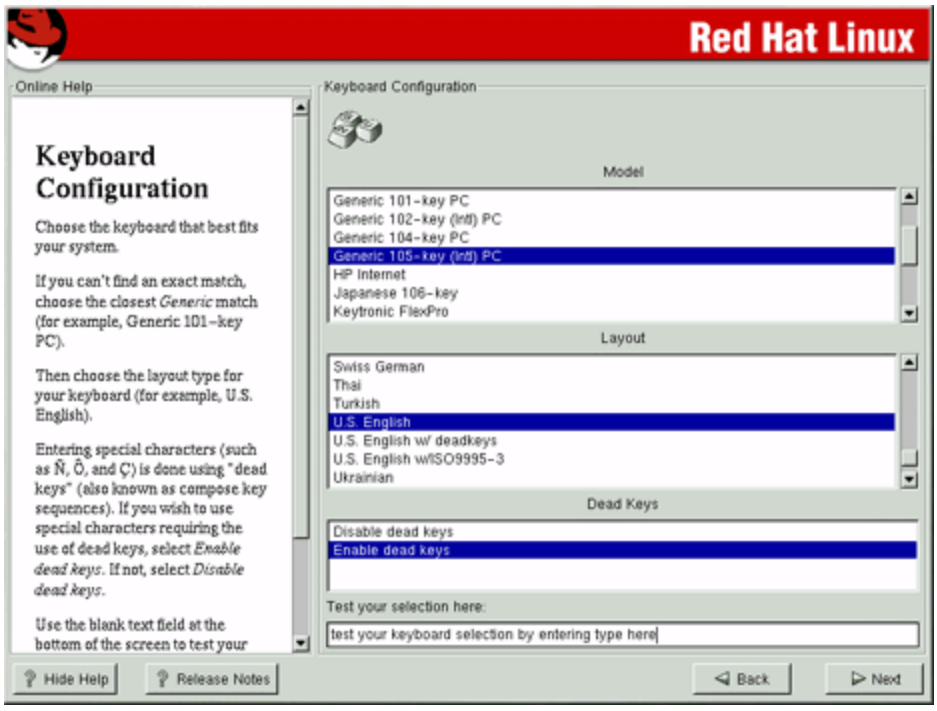
Beginning the installation of Linux

Now that we have made the boot disk, it is time to begin the installation of Linux. Since we'd start the installation directly off the CD-ROM, boot with the boot disk. Insert the boot diskette you create into the drive A: on the computer where you want to install Linux and reboot the computer. At the `boot:` prompt, press **`Enter`** to continue booting and follow the three simple steps below:

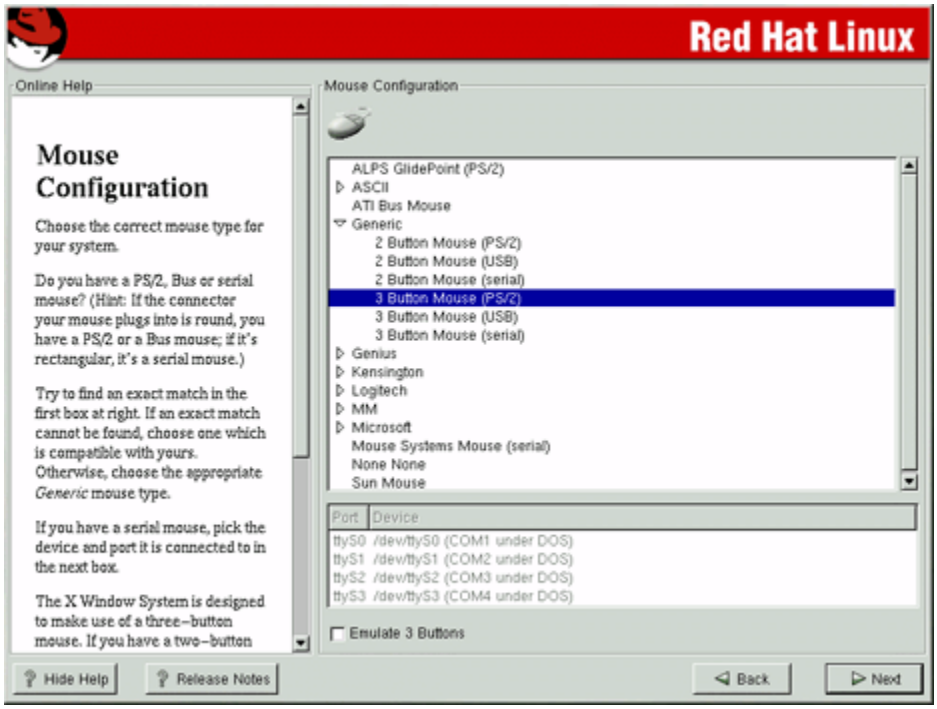
Step 1
The first step is to choose what language should be used during the installation process. In our example we choose the English language.



Step 2
After that, the system allows you to choose your keyboard type, layout type for the keyboard, and the possibility to enable or disable Dead Keys.



Step 3
Finally, we choose the kind of mouse type we use and if this mouse has two or three buttons. If you have a mouse with just two buttons, you can select the option named “Emulate 3 Buttons” and click both mouse buttons at the same time to act as the middle mouse button.



Once we have completed the above three steps, we are ready to begin the installation of Red Hat Linux.

Installation Class and Method (Install Options)

Red Hat Linux 7.1 includes four different classes, or type of installation. They are:

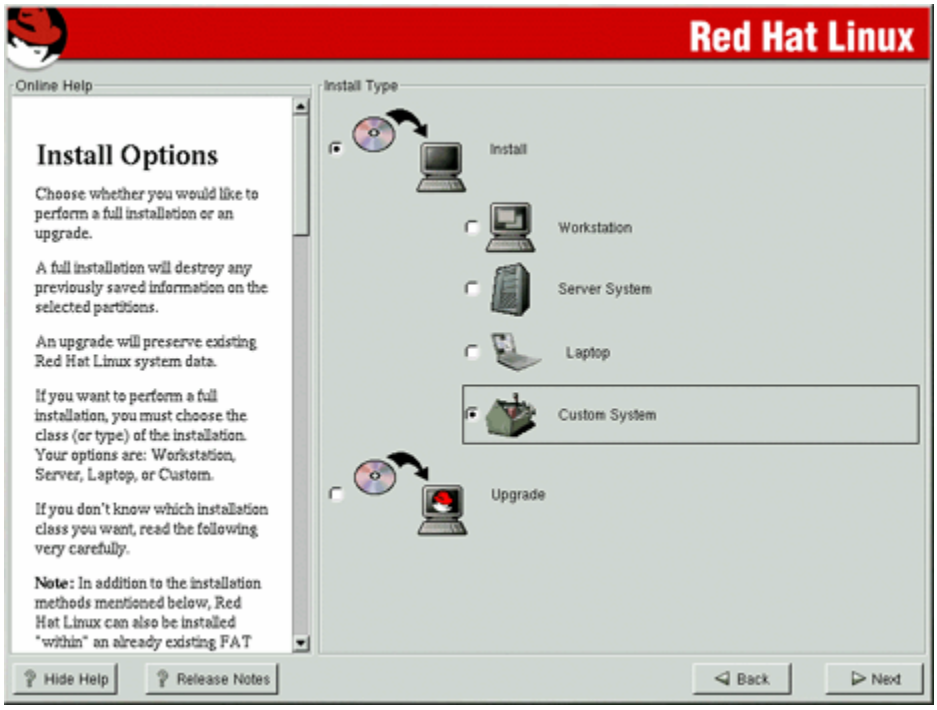
- ✓ Workstation
- ✓ Server System
- ✓ Laptop
- ✓ Custom System

The first two classes (Workstation, and Server System) give you the option of simplifying the installation process with a significant loss of configuration flexibility that we don't want to lose.

For this reason we highly recommend you select the "**Custom System**" installation. Only the custom-class installation gives us complete flexibility. During the custom-class installation, it is up to you how disk space should be partitioned. We also have complete control over the different RPM packages that will be installed on the system.

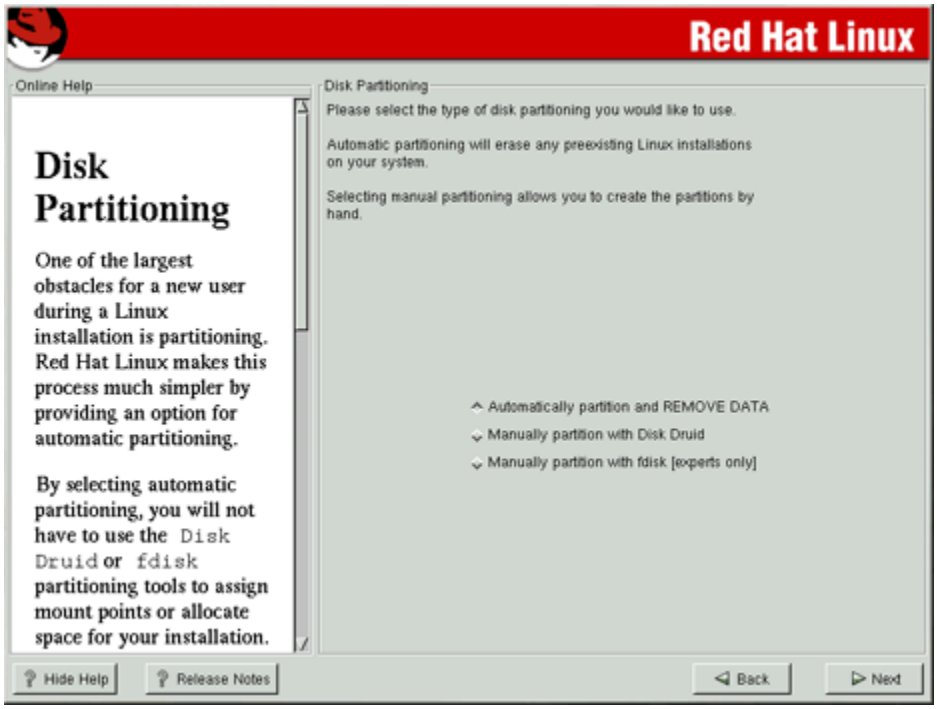
The idea is to load the minimum amount of packages, while maintaining maximum efficiency. The less software that resides on the machine, the fewer potential security exploits or holes may appear.

From the menu that appears on your screen, select the "**Custom System**" installation class and click **Next**.



Partition your system for Linux

The system will show you a new screen from where you can choose the tool you would like to use to partition the disks for Linux.



From here we have two choices, but before we explain each ones, it is important to go and understand partition strategy first.

We assume that you are installing the new Linux server to a new hard drive, with no other existing file system or operating system installed. A good partition strategy is to create a separate partition for each major file system. This enhances security and prevents accidental denial of service or exploit of `SUID` programs.

Creating multiple partitions offers you the following advantages:

- ✓ Protection against denial of service attack.
- ✓ Protection against `SUID` programs.
- ✓ Faster booting.
- ✓ Easy backup and upgrade management.
- ✓ Ability for better control of mounted file system.
- ✓ Limit each file system's ability to grow.
- ✓ Improve performance of some program with special setup.

WARNING: If a previous file system or operating system exists on the hard drive and computer where you want to install your Linux system, we highly recommend, that you make a backup of your current system before proceeding with the disk partitioning.

Partitions Strategy

For performance, stability and security reasons you must create something like the following partitions listed below on your computer. We suppose for this partition configuration the fact that you have a SCSI hard drive of 9.1 GB with 256 MB of physical RAM. Of course you will need to adjust the partition sizes and swap space according to your own needs and disk size.

Minimal recommended partitions that must be created on your system:

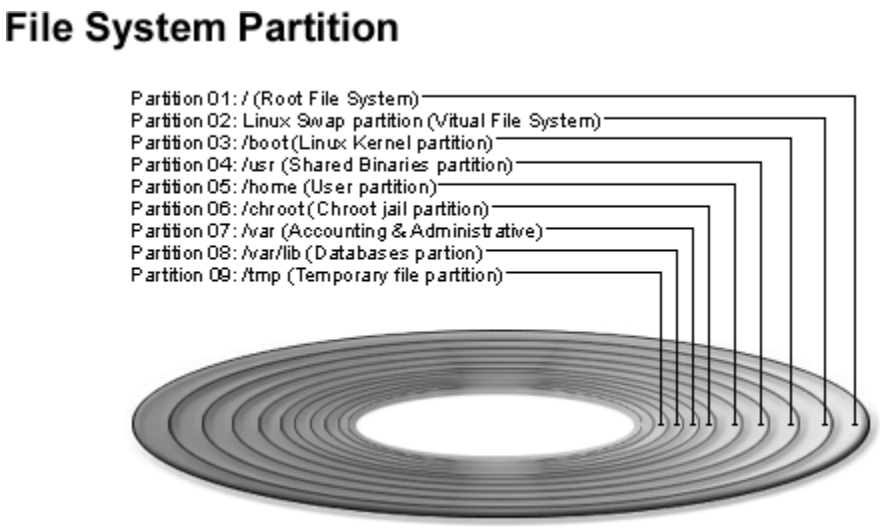
This is the minimum number of partitions we recommend creating whatever you want to setup it for, a Web Server, Mail Server, Gateway or something else.

/boot	5 MB	All Kernel images are kept here.
<Swap>	512 MB	Our swap partition. The virtual memory of the Linux operating system.
/	256 MB	Our root partition.
/usr	512 MB	Must be large, since many Linux binaries programs are installed here.
/home	5700 MB	Proportional to the number of users you intend to host. (i.e. 100 MB per users * by the number of users 57 = 5700 MB)
/var	256 MB	Contains files that change when the system run normally (i.e. Log files).
/tmp	329 MB	Our temporary files partition (must always reside on its own partition).

Additional or optional partitions that can be created on your system:

Depending on what services the Linux system will be assigned to serve or the specific software requirements, there can be some special partitions you can add to the minimum partitions we recommend. You can create as many partitions as you want to fit you needs. What we show you below are partitions related to programs we describe in the book.

/chroot	256 MB	If you want to install programs in chroot jail environment (i.e. DNS, Apache).
/var/lib	1000 MB	Partition to handle SQL or Proxy Database Server files (i.e. MySQL, Squid).



All major file systems are on separate partitions

As you can see, there are two partitions, which are less common than the others. Lets explain each of them in more detail:

The /chroot partition can be used for DNS Server chrooted, Apache Web Server chrooted and other chrooted future programs. The chroot() command is a Unix system call that is often used to provide an additional layer of security when untrusted programs are run. The kernel on Unix variants which support chroot() maintain a note of the root directory each process on the system has. Generally this is /, but the chroot() system call can change this. When chroot() is successfully called, the calling process has its idea of the root directory changed to the directory given as the argument to chroot().

The /var/lib partition can be used to handle SQL or Squid Proxy database files on the Linux Server. This partition can be useful to limit accidental denial of service attack and to improve the performance of the program by tuning the /var/lib file system.

Putting /tmp and /home on separate partitions is pretty much mandatory if users have shell access to the server (protection against SUID programs), splitting these off into separate partitions also prevent users from filling up any critical file system (denial of service attack), putting /var, and /usr on separate partitions is also a very good idea. By isolating the /var partition, you protect your root partition from overfilling (denial of service attack).

In our partition configuration we'll reserve 256 MB of disk space for chrooted programs like Apache, DNS and other software. This is necessary because Apache DocumentRoot files and other binaries, programs related to it will be installed in this partition if you decide to run Apache Web Server in a chrooted jail. Note that the size of the Apache chrooted directory on the chrooted partition is proportional to the size of your DocumentRoot files or number of users.

Swap related issues:

Swap relates to virtual RAM on the system. This special device is needed when you run out of physical RAM because you don't have enough MB of RAM available or your applications required more than what is available on your computer. It is not true that swap space is needed on every system, but to ensure that you do not run out of swap, it is recommended to create a swap partition on the server.

The 2.4 kernel of Linux is more aggressive than the 2.2 kernels in its use of swap space and the optimal sizing of swap space remains dependent on the following:

1. The amount of RAM installed
2. The amount of disk space available for swap
3. The applications being run
4. The mix of applications that are run concurrently

No rule-of-thumb can possibly take all these data points into account. However, we recommend the following swap sizes:

- Single-user systems with less than 128MB physical RAM: 256MB
- Single-user systems and low-end servers with more than 128MB physical RAM: two times physical RAM (2xRAM)
- Dedicated servers with more than 512MB physical RAM: highly dependent on environment and must be determined on a case-by-case basis)

Minimum size of partitions for very old hard disk:

For information purposes only, this is the minimum size in megabytes, which a Linux installation must have to function properly. The sizes of partitions listed below are really small. This configuration can fit into a very old hard disk of 512MB in size that you might find in old i486 computers. We show you this partition just to get an idea of the minimum requirements.

```
/          35MB
/boot      5MB
/chroot    10MB
/home      100MB
/tmp       30MB
/usr       232MB
/var       25MB
```

WARNING: Trying to compile program under a 512 MB of hard drive will fail due to the miss of available space in this kind of hard disk. Instead, install RPM's packages.

Disk Partition (Manual Partitioning)

Now that we know exactly what partitions we need to create for our new Linux server, it is time to choose the partitioning software we will use to make these partitions on the server. With Red Hat Linux two programs exist to assist you during this step. During setup, the installation will give you two choices, which are:

- Manually partition with Disk druid
- Manually partition with fdisk [experts only]

`Disk Druid` is the new software used by default in Red Hat Linux to partition your disk drive, this is an easy to use program, which allows you to work through a graphical interface to create your partitions tables.

`fdisk` was the first partitioning program available on Linux. It is more powerful then `Disk Druid` and allows you to create your partition table in exactly the way you want it (if you want to put your swap partition near the beginning of your drive, then you will need to use `fdisk`). Unfortunately, it is also a little more complicated than `Disk Druid` and many Linux users prefer to use `Disk Druid` for this reason.

Personally, I prefer to create the required partitions with the `fdisk` program and I recommend you use and be familiar with it, because if in future you want to add or change some file systems you will need to use `fdisk`.

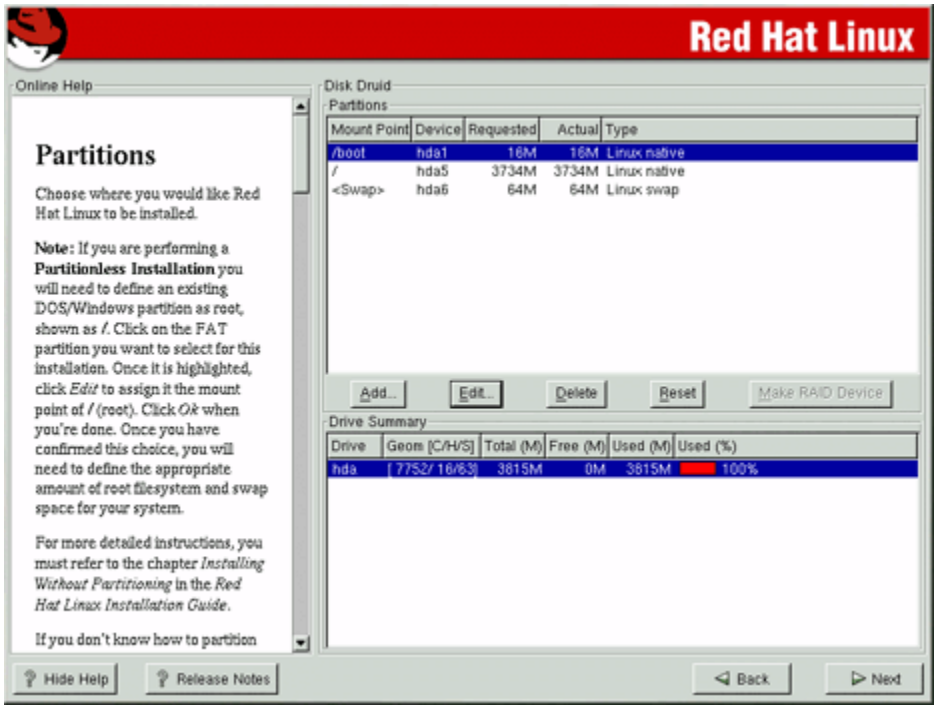
Partitioning with `Disk Druid`

This section applies only if you chose to use **`Disk Druid`** to partition your system.

`Disk Druid` is a program that partitions your hard drive for you. Choose "**`Add`**" to add a new partition, "**`Edit`**" to edit a partition, "**`Delete`**" to delete a partition and "**`Reset`**" to reset the partitions to the original state. When you add a new partition, a new window appears on your screen and gives you parameters to choose.

Different parameters are:

Mount Point: for where you want to mount your new partition in the filesystem.
Size (Megs): for the size of your new partition in megabytes.
Partition Type: Linux native for Linux filesystem and Swap for Linux Swap Partition.



If you have a `SCSI` disk, the device name will be `/dev/sda` and if you have an `IDE` disk it will be `/dev/hda`. If you're looking for high performance and stability, a `SCSI` disk is highly recommended.

Linux refers to disk partitions using a combination of letters and numbers. It uses a naming scheme that is more flexible and conveys more information than the approach used by other operating systems.

Here is a summary:

First Two Letters – The first two letters of the partition name indicate the type of device on which the partition resides. You'll normally see either **`hd`** (for `IDE` disks), or **`sd`** (for `SCSI` disks).

The Next Letter – This letter indicates which device the partition is on. For example: `/dev/hda` (the first `IDE` hard disk) and `/dev/hdb` (the second `IDE` disk), etc.

Keep this information in mind, it will make things easier to understand when you're setting up the partitions Linux requires.

Now, as an example:
To make the partitions listed below on your system (this is the partition we'll need for our server installation example); the commands below are for `Disk Druid`:

Step 1
Execute all of the following commands with `Disk Druid` to create the require partitions.

```
Add
Mount Point: /boot ← our /boot directory (all Kernel images are kept here).
Size (Megs): 5
Partition Type: Linux Native
Ok

Add
Mount Point: ← our /Swap partition (leave the Mount Point Blank).
Size (Megs): 512
Partition Type: Linux Swap
Ok

Add
Mount Point: / ← our / directory (the root partition).
Size (Megs): 256
Partition Type: Linux Native
Ok

Add
Mount Point: /usr ← our /usr directory (many Linux binaries programs are installed here).
Size (Megs): 512
Partition Type: Linux Native
Ok

Add
Mount Point: /home ← our /home directory (where users files & directories reside).
Size (Megs): 5700
Partition Type: Linux Native
Ok

Add
Mount Point: /chroot ← our /chroot directory (for programs installed in chroot jail environment).
Size (Megs): 256
Partition Type: Linux Native
Ok

Add
Mount Point: /var ← our /var directory (files that change when the system run are keep here).
Size (Megs): 256
Partition Type: Linux Native
Ok

Add
Mount Point: /var/lib ← our /var/lib directory (special partition to handle SQL or Proxy Database files).
Size (Megs): 1000
Partition Type: Linux Native
Ok

Add
Mount Point: /tmp ← our /tmp directory (partition for temporary files on the system).
Size (Megs): 227
Partition Type: Linux Native
Ok
```

Step 2
After you have executed the above commands to create and partition your drive with **`Disk Druid`**, press the **Next** button and continue the installation to choose partitions to format.

Partitioning with `fdisk`

This section applies only if you chose to use **`fdisk`** to partition your system.

The first thing you will want to do is using the **`p`** key to check the current partition information. You need to first add your root partition. Use the **`n`** key to create a new partition and then select either **`e`** or **`p`** keys for extended or primary partition.

Most likely you will want to create a primary partition. You are asked what partition number should be assigned to it, at which cylinder the partition should start (you will be given a range – **just choose the lowest number (1)**), and the size of the partition. For example, for a 5MB partition, you would enter +5M for the size when asked.

Next, you need to add your extended partition. Use the **`n`** key to create a new partition and then select the **`e`** key for extended partition. You are asked what partition number should be assigned to it, at which cylinder the partition should start (you will be given a range – **just choose the lowest number (2)**), and the size of the partition. **You would enter the last number for the size when asked (or just press Enter)**.

You will now want to create the swap partition. You need to use the **`n`** key for a new partition. Choose logical; tell it where the first cylinder should be **(2)**. Tell `fdisk` how big you want your swap partition. You then need to change the partition type to `Linux swap`. Enter the **`t`** key to change the type and enter the partition number of your swap partition. Enter the number **`82`** for the hex code for the `Linux swap` partition.

Now that you have created your Linux boot and Linux swap partition, it is time to add any additional partitions you might need. Use the **`n`** key again to create a new partition, and enter all the information just as before. Keep repeating this procedure until all your partitions are created. You can create up to four primary partitions; then you must start putting extended partitions into each primary partition.

> **NOTE**: None of the changes you make take effect until you save then and exit `fdisk` using the **`w`** command. You may quit `fdisk` at any time without saving changes by using the **`q`** command.

An overview of `fdisk`

- The command for help is **`m`**
- To list the current partition table, use **`p`**
- To add a new partition, use **`n`**
- To delete a partiotion, use **`d`**
- To set or changes the partition type, use **`t`**
- To provide a listing of the different partition types and their ID numbers, use **`l`**
- To saves your information and quits `fdisk`, use **`w`**

Now, as an example:
To make the partitions listed below on your system (these are the partitions we'll need for our server installation example); the commands below are for `fdisk`:

Step 1
Execute all of the following commands with `fdisk` to create the require partitions.

```
Command (m for help): n
Command action
    e extended
    p primary partition (1-4)
p
Partition number (1-4): 1
First cylinder (1-1116, default 1): 1
Last cylinder or +size or +sizeM or +sizeK (1-1116, default 1116): +5M
```
← our `/boot` directory.

```
Command (m for help): n
Command action
    e extended
    p primary partition (1-4)
e
Partition number (1-4): 2
First cylinder (2-1116, default 2): 2
Last cylinder or +size or +sizeM or +sizeK (2-1116, default 1116): 1116
```
← our extended partition.

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (2-1116, default 2): 2
Last cylinder or +size or +sizeM or +sizeK (2-1116, default 1116): +512M
```
← our `Swap` partition.

```
Command (m for help): t
Partition number (1-5): 5
```
← this is our `Swap` partition number on this example.
```
Hex code (type L to list codes): 82
Changed system type of partition 5 to 82 )Linux swap)

Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (68-1116, default 68): 68
Last cylinder or +size or +sizeM or +sizeK (68-1116, default 1116): +256M
```
← our `/` directory.

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (101-1116, default 101): 101
Last cylinder or +size or +sizeM or +sizeK (101-1116, default 1116): +512M
```
← our `/usr` directory.

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (167-1116, default 167): 167
Last cylinder or +size or +sizeM or +sizeK (167-1116, default 1116): +5700M
```
← our /home directory.

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (894-1116, default 894): 894
Last cylinder or +size or +sizeM or +sizeK (894-1116, default 1116): +256M
```
← our /chroot directory.

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (927-1116, default 927): 927
Last cylinder or +size or +sizeM or +sizeK (927-1116, default 1116): +256M
```
← our /var directory.

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (960-1116, default 960): 960
Last cylinder or +size or +sizeM or +sizeK (960-1116, default 1116): +1000M
```
← our /var/lib directory.

```
Command (m for help): n
Command action
    l logical (5 or over)
    p primary partition (1-4)
l
First cylinder (1088-1116, default 1088): 1088
```
Last cylinder or +size or +sizeM or +sizeK (1088-1116, default 1116): **1116** ← our /tmp directory.

Step 2

Now, use the **p** command to list the partition that we've created, you must see something like the following information on your screen.

```
Command (m for help): p
```

```
Disk /tmp/sda: 255 heads, 63 sectors, 1116 cylinders
Units = cylinders of 16065 * 512 bytes

Device Boot    Start         End          Blocks        Id           System
/tmp/sda1      1             1            8001          83           Linux
/tmp/sda2      2             1116         8956237+      5            Extended
/tmp/sda5      2             67           530113+       82           Linux swap
/tmp/sda6      68            100          265041        83           Linux
/tmp/sda7      101           166          530113+       83           Linux
/tmp/sda8      167           893          5839596       83           Linux
/tmp/sda9      894           926          265041        83           Linux
/tmp/sda10     927           959          265041        83           Linux
/tmp/sda11     960           1087         1028128+      83           Linux
/tmp/sda12     1088          1116         232911        83           Linux
```

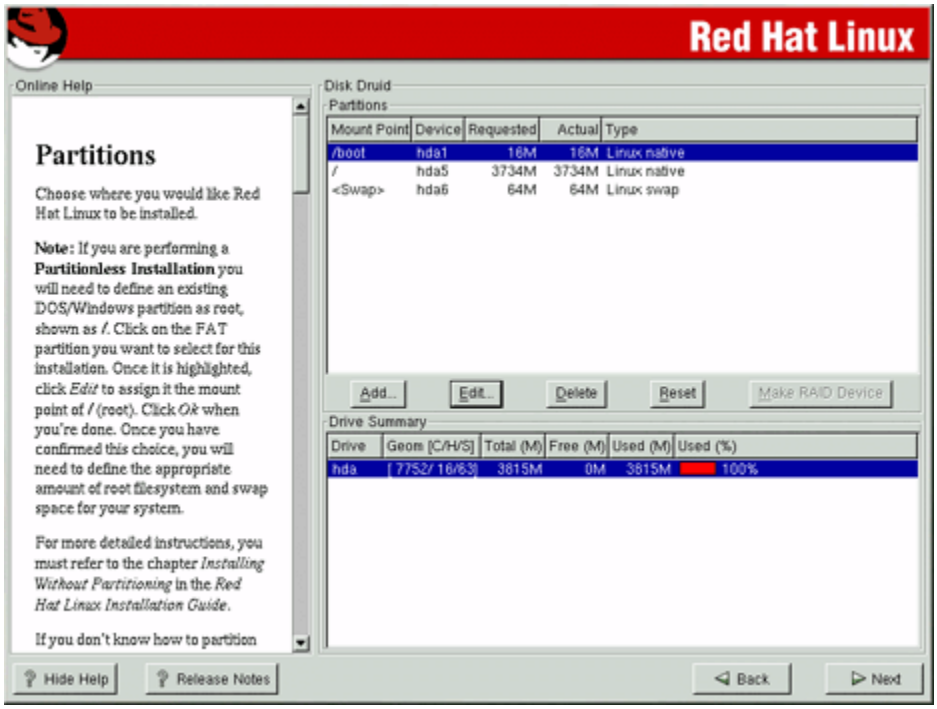
Step 3

If all the partitions look fine and meet your requirements, use the **`w`** command to write the table to disk and exit `fdisk` program:

```
Command (m for help): w
The partition table has been altered
```

Step 4

After you have partitioned your drive with **`fdisk`**, press **Next** and continue the installation with **`Disk Druid`** to choose the mount point of the directories. `Disk Druid` contains a list of all disk partitions with filesystems readable by Linux. This gives you the opportunity to assign these partitions to different parts of your Linux system when it boots. Select the partition you wish to assign and press **`Enter`**; then enter the mount point for that partition, e.g., `/var`.



Step 5
After the mount points for the directories have been completed, you must see something like the following on your screen. Our mount points look like this:

Disk Druid

Partitions

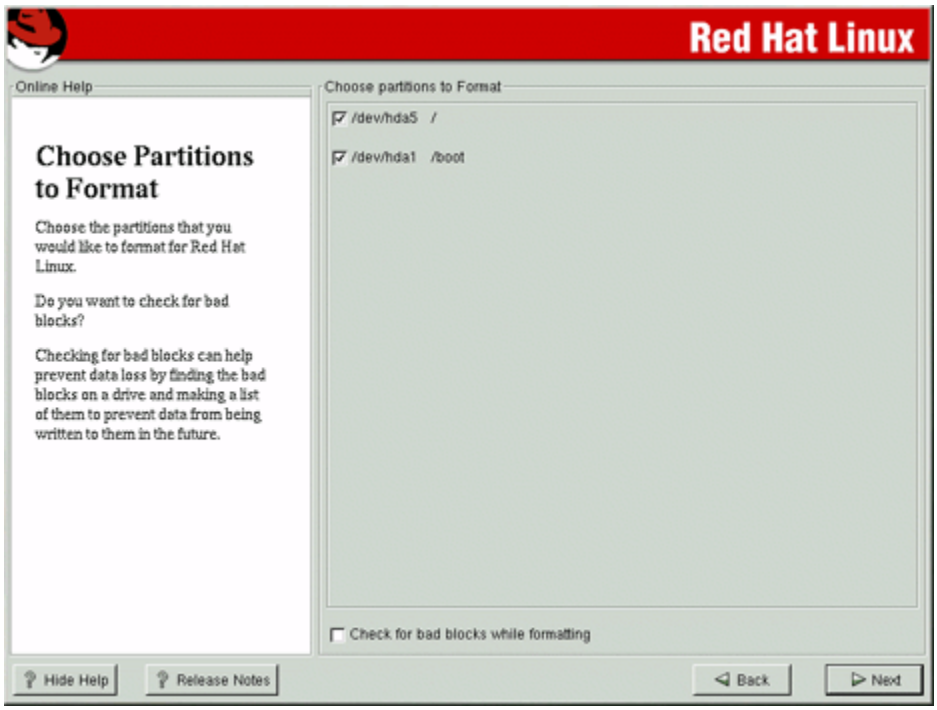
Mount Point	*Device*	*Requested*	*Actual*	*Type*
/boot	sda1	7M	7M	Linux Native
<Swap>	sda5	517M	517M	Linux Swap
/	sda6	258M	258M	Linux Native
/usr	sda7	517M	517M	Linux Native
/home	sda8	5702M	5702M	Linux Native
/chroot	sda9	258M	258M	Linux Native
/var	sda10	258M	258M	Linux Native
/var/lib	sda11	1004M	1004M	Linux Native
/tmp	sda12	227M	227M	Linux Native

Drive Summary

Drive	Geom [C/H/S]	Total (M)	Free (M)	Used (M)	Used (%)
sda	[1116/255/63]	8754M	1M	8753M	99%

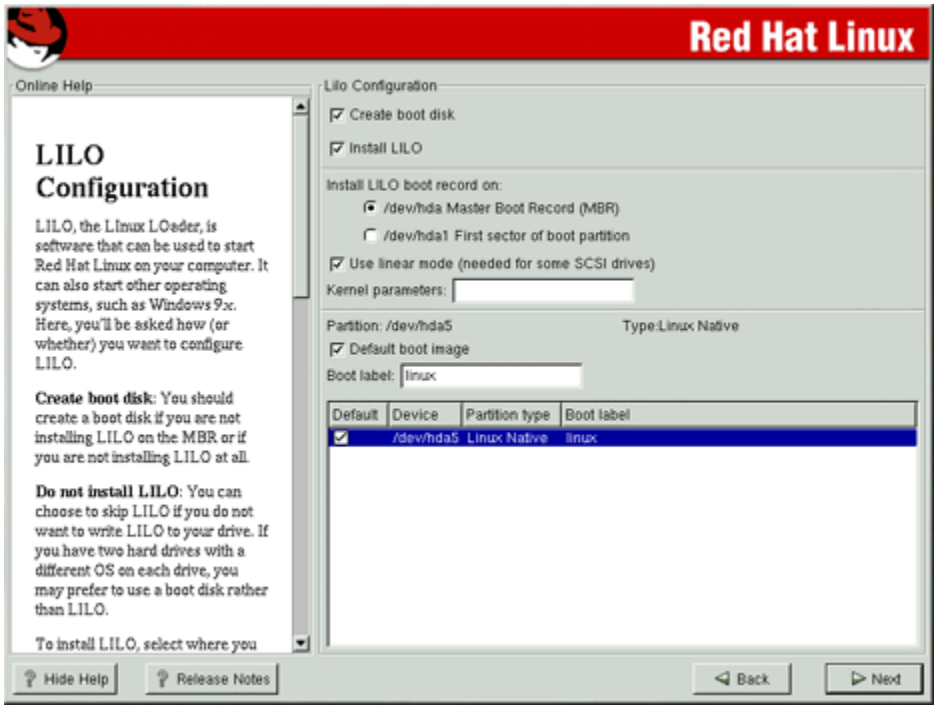
Step 6
Now that you have partitioned and chosen the mount points of your directories, select **Next** to continue. After your partitions are created, the installation program will ask you to choose which partitions to format. Choose the partitions, check the **(Check for bad blocks while formatting)** box, and press **Next** again. This formats the partitions and makes them active so Linux can use them.

NOTE: Checking for bad blocks can help prevent data loss by finding the bad blocks on a drive and making a list of them to prevent data from being written to them in the future.



System Configuration

On the next screen you will see the LILO Configuration screen. LILO, the **LI**nux **LO**ader, is software that can be used to start Linux on your computer. From this screen, you will see different configurable options related to LILO.



The first option is:

- Create boot disk

The **Create boot disk** option is checked by default. If you do not want to create a boot disk, you should deselect this option. Also, this option must be checked if you decide to not install LILO on the MBR (the **M**aster **B**oot **R**ecord) or if you are not installing LILO at all.

The second option is:

- Do not install LILO

This option allows you to skip installing LILO if you use a boot disk rather than LILO to start your system. This can greatly improve security in some case since you need to have a bootable Linux floppy with the kernel on it to start the server. But in other hand, you will not be able to restart the server remotely if something happens.

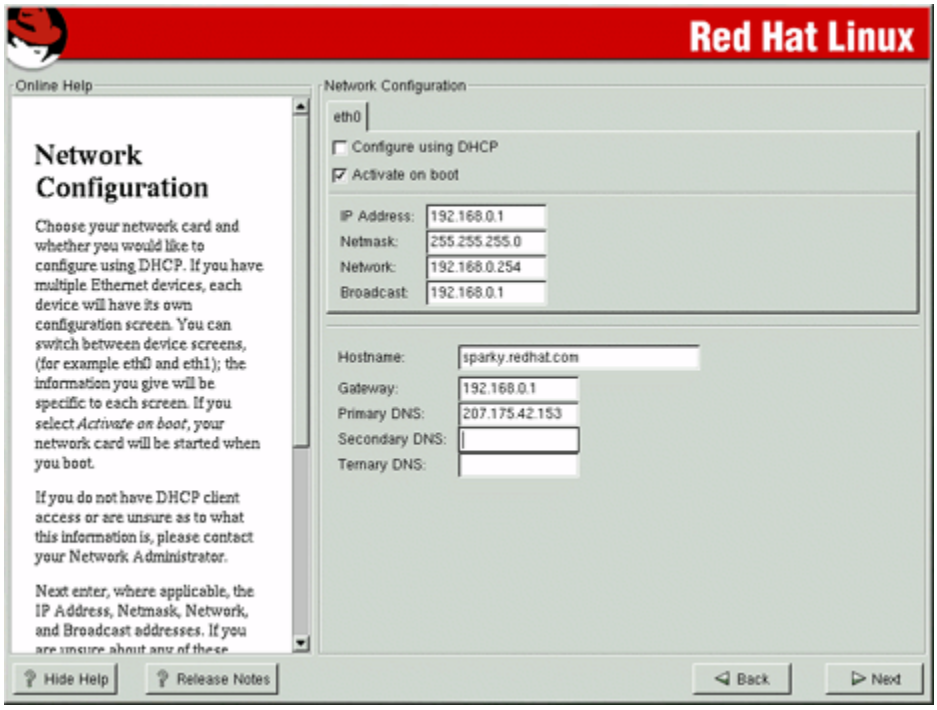
The third option **(the one that we will chose)** installs `LILO` in your Linux system and gives you the choice to install `LILO` boot record on:

- Master Boot Record (MBR)
- First Sector of Boot Partition

Usually, if Linux is the only Operating System on your machine (and this must be the case in a server installation), you should choose the "**Master Boot Record (MBR)**" option.

Network Configuration

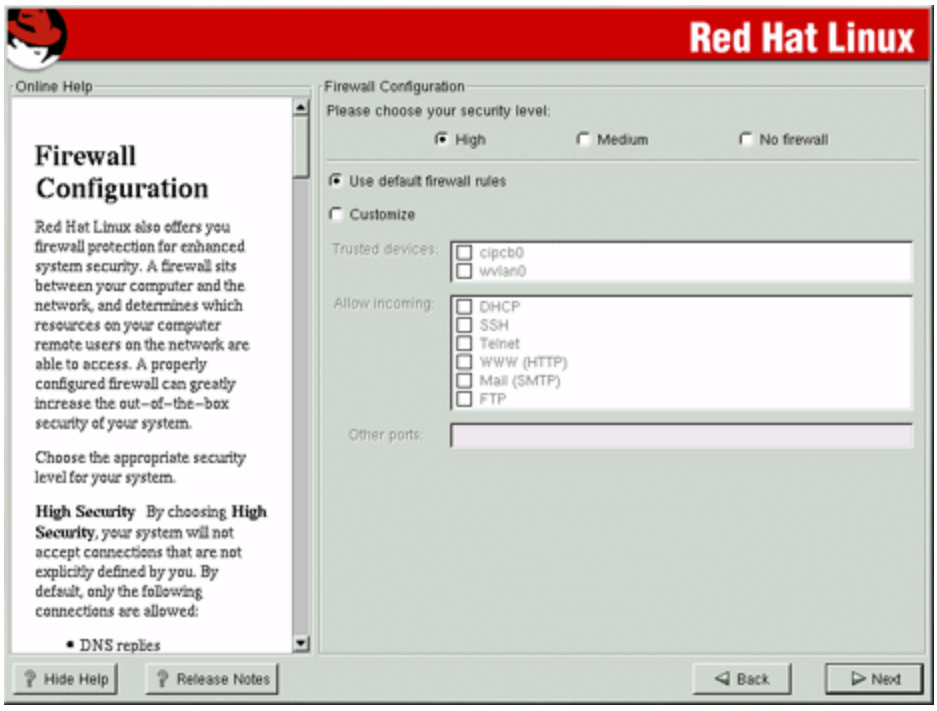
After that, you need to configure your network. If you have multiple Ethernet devices, each device will have its own configuration screen.



Firewall Configuration

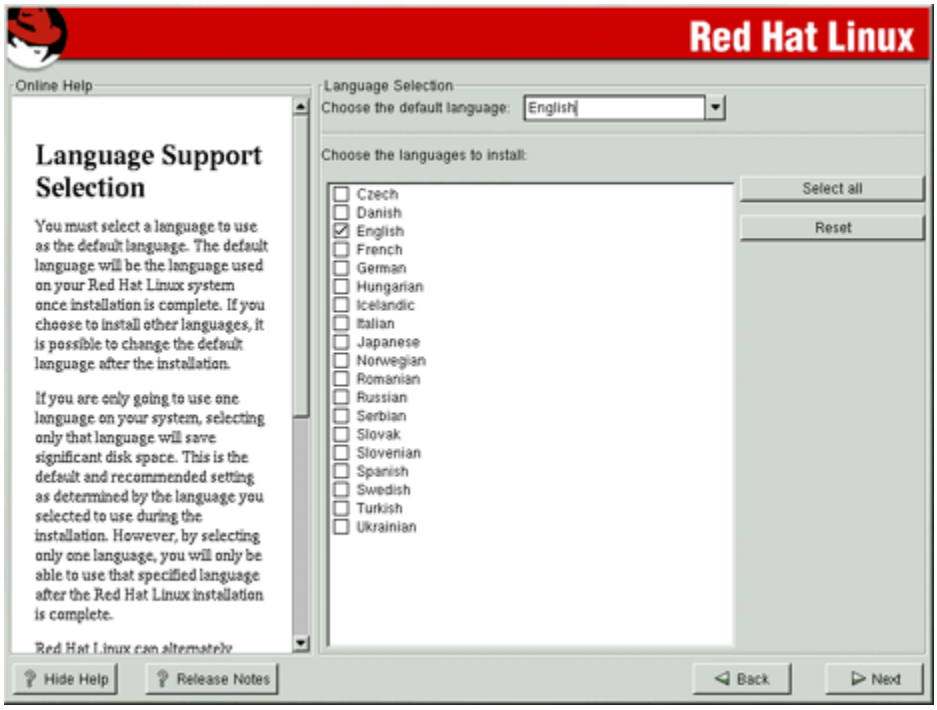
The latest release of Red Hat Linux now offers the possibility to configure a Firewall during installation. This is OK for the average end user but **NOT** for serious Firewall security. This newly added feature uses the old `IPCHAINS` tool of Linux with the help of a small utility named "`lokkit`" to set up your firewall. I highly recommend you to deactivate this feature now and see later chapters on how to install and configure `IPTABLES`, which is the new Firewall tool to use with Linux and kernel 2.4 generation.

From the next screen that appears, you will see three different security levels available, choose the "**No firewall**" option and click **Next**.



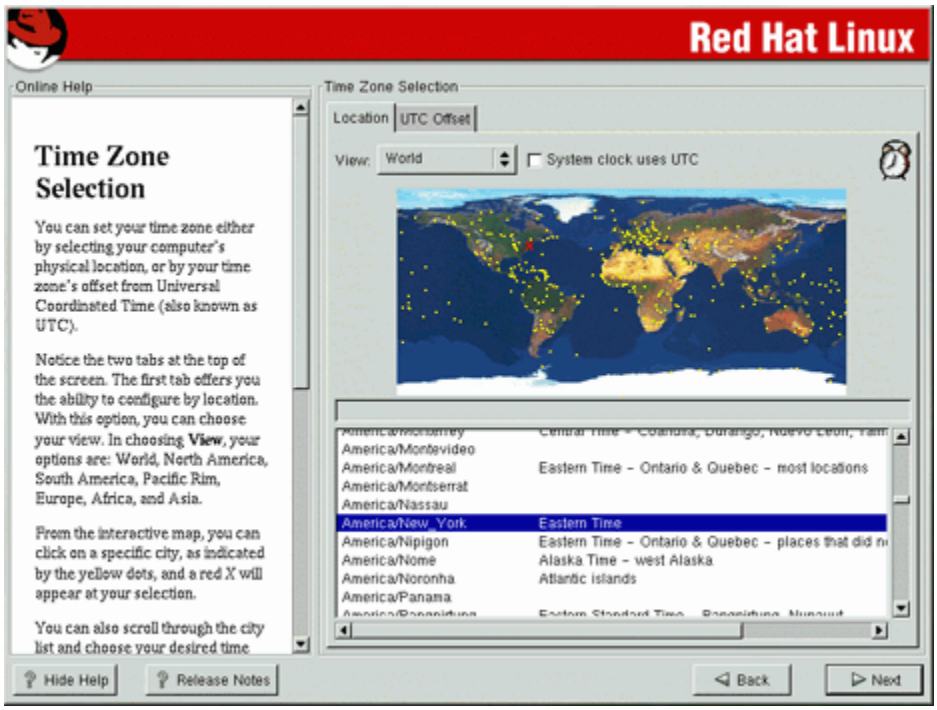
Language Support Selection

Multiple language selection is now possible with this release of Linux. With the internalization, a need for different language support has appeared. From here the installation will ask you to choose the default language that will be used on your Linux system once the installation is complete. If you are only going to use one language on your system, selecting only this language will save significant disk space.



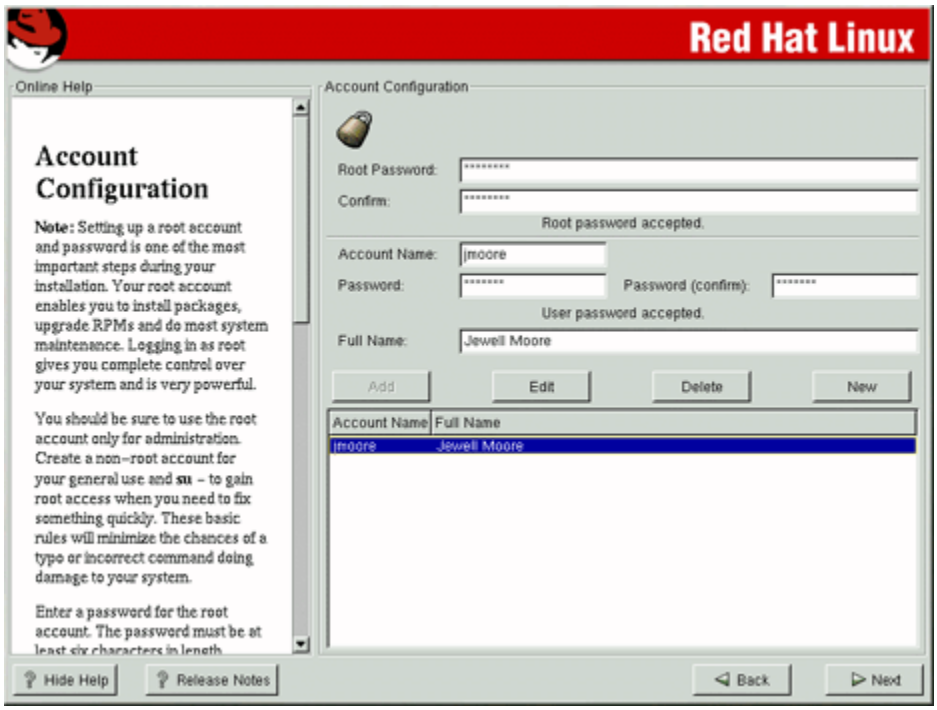
Time Zone Selection

On the next screen, you will have the opportunity to set your time zone. Once selected click **Next**.



Account Configuration

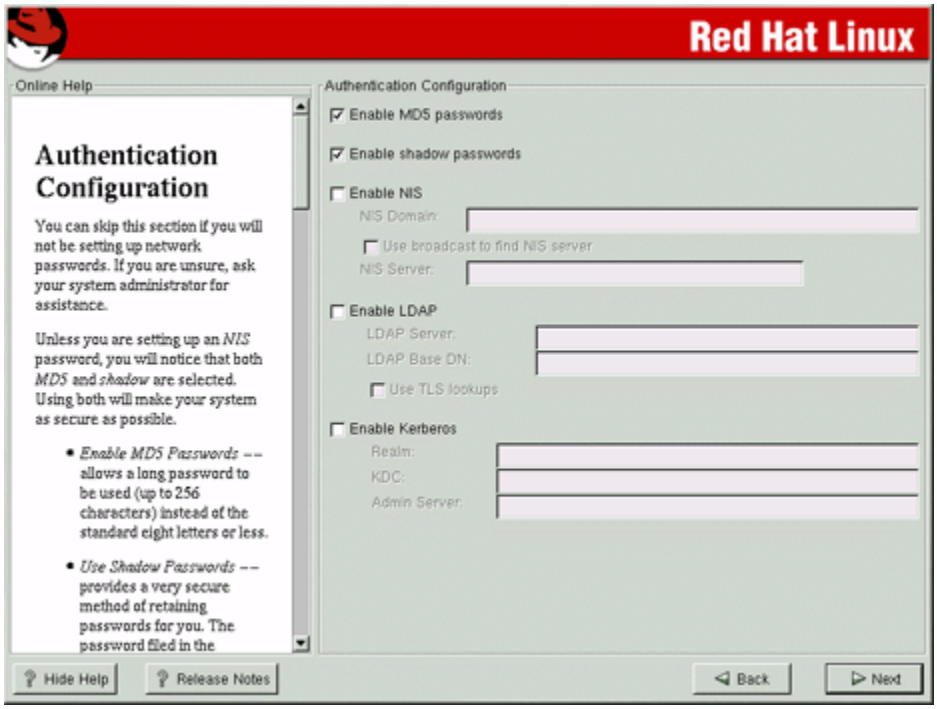
After the clock has been configured, you need to give your system a root password account.



Authentication Configuration

Finally, the last stage is the authentication configuration. For Authentication Configuration don't forget to select:

- ✓ Enable `MD5` passwords
- ✓ Enable Shadow passwords



Enable `MD5` passwords - allows a long password to be used (up to 256 characters), instead of the Unix standard eight letters or less.

Enable shadow passwords - provides a very secure method of retaining passwords for you. All passwords are stored in a file named `shadow`, which is readable only by the super-user root.

Enable `NIS`, `LDAP`, and `Kerberos` doesn't need to be selected since we are not configuring these services on this server right know.

Selecting Package Groups

After your partitions have been configured and selected for formatting and configurations have been set for your specific system, you are ready to select packages for installation. By default, Linux is a powerful operating system that runs many useful services. However, many of these services are unneeded and pose potential security risks.

Ideally, each network service should be on a dedicated, single-purpose host. Many Linux operating systems are configured by default to provide a wider set of services and applications than are required to provide a particular network service, so you may need to configure the server to eliminate unneeded services. Offering only essential services on a particular host can enhance your network security in several ways:

- ✓ Other services cannot be used to attack the host and impair or remove desired network services.

- ✓ The host can be configured to better suit the requirements of the particular service. Different services might require different hardware and software configurations, which could lead to needless vulnerabilities or service restrictions.

- ✓ By reducing services, the number of logs and log entries is reduced so detecting unexpected behavior becomes easier.

- ✓ Different individuals may administer different services. By isolating services so each host and service has a single administrator you will minimize the possibility of conflicts between administrators.

A proper installation of your Linux server is the first step to a stable, secure system. From the screen menu that appears (Selecting Package Groups), you first have to choose which system components you want to install, in our case, we must **DESELECT ALL CHECKED** Package Groups on the list.

Since we are configuring a Linux Server, we don't need to install a graphical interface (`XFree86`) on our system (a graphical interface on a server means less processes, less CPU availability, less memory, security risks, and so on), also computers are subject to the treachery of images as well. The image on your computer screen is not a computer file -- it's only an image on a computer screen. Images of files, processes, and network connections are very distant cousins of the actual bits in memory, in network packets, or on disks.

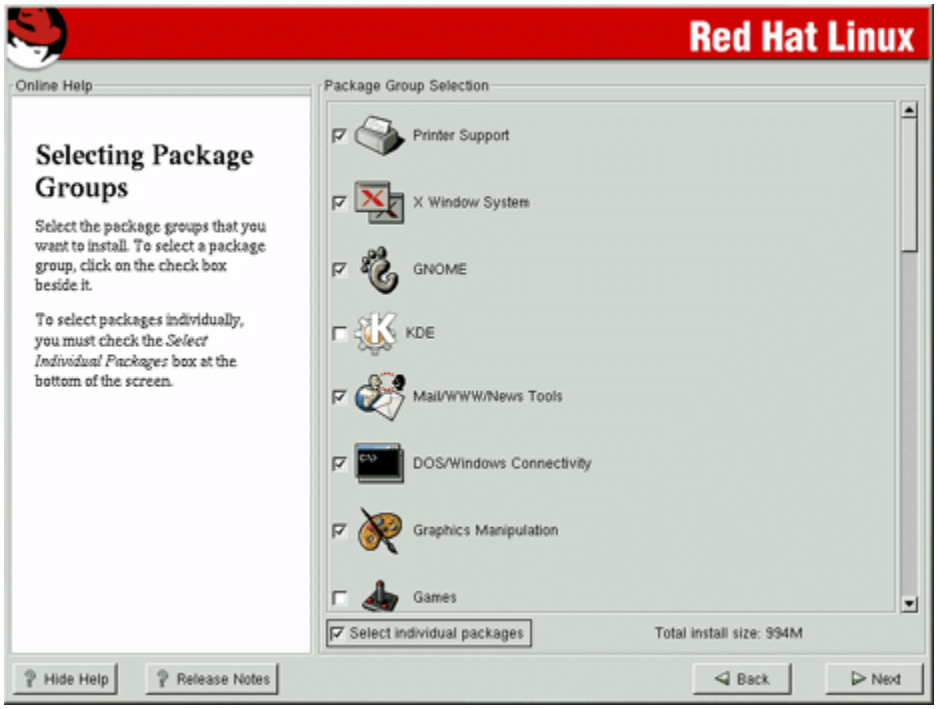
Layer upon layer of hardware and software produces the images that you see. When an intruder "owns" a machine, any of those layers could be tampered with. Application software can lie, OS kernels can lie, boot PROMs can lie, and even hard disk drives can lie. Graphical interfaces are usually used on only workstations.

Step 1
First of all, it is vital to verify and be **SURE** to deselect all of the following Package Group:

- ✓ Printer Support
- ✓ X Window System
- ✓ GNOME
- ✓ KDE
- ✓ Mail/WWW/News Tools
- ✓ DOS/Windows Connectivity
- ✓ Graphics Manipulation
- ✓ Games
- ✓ Multimedia Support
- ✓ Laptop Support
- ✓ Networked Workstation
- ✓ Dialup Workstation
- ✓ News Server
- ✓ NFS Server
- ✓ SMB (Samba) Server
- ✓ IPX/Netware™ Connectivity
- ✓ Anonymous FTP Server
- ✓ SQL Server
- ✓ Web Server
- ✓ DNS Name Server
- ✓ Network Management Workstation
- ✓ Authoring/Publishing
- ✓ Emacs
- ✓ Development
- ✓ Kernel Development
- ✓ Utilities
- ✓ Everything

To resume, it is very important and I say VERY IMPORTANT to deselect (none is selected) every selected Packages Group before clicking on the **Next** button for continuing the installation.

We don't want and don't need to install any additional packages. The default install of this Linux distribution already comes with the most essential programs we need for the functionality of the operating system.



NOTE ABOUT SYSTEM SIZE: At this stage of our installation of Linux, the total install size will be **224MB** if you have deselected all menu packages group as described above.

Step 2
At this point, the installation program will check dependencies in packages selected for installation (in our case no packages are selected) and format every partition you selected for formatting in you system. This can take several minutes depending on the speed of your machine. Once all partitions have been formatted, the installation program starts to install Linux to your hard drive.

How to use RPM Commands

This section contains an overview of using RPM for installing, uninstalling, upgrading, querying, listing, and checking RPM packages on your Linux system. You must be familiar with these RPM commands now because we'll use them often in this book and especially later in this chapter for software that must be uninstalled after installation of the server.

- To install a RPM package, use the command:

```
[root@deep /]# rpm -ivh foo-1.0-2.i386.rpm
foo                  ##################################################
```

Note that RPM packages have a file of names like **`foo-1.0-2.i386.rpm`**, which include the package name (**`foo`**), version (**`1.0`**), release (**`2`**), and architecture (**`i386`**).

- To uninstall a RPM package, use the command:

```
[root@deep /]# rpm -e foo
```

Notice that we used the package name "**`foo`**", not the name of the original package file "**`foo-1.0-2.i386.rpm`**".

- To upgrade a RPM package, use the command:

```
[root@deep /]# rpm -Uvh foo-1.0-2.i386.rpm
foo                  ##################################################
```

With this command, RPM automatically uninstalls the old version of `foo` package and installs the new one. Always use `rpm -Uvh` to install packages, since it works fine even when there are no previous versions of the package installed.

- To query a RPM package, use the command:

```
[root@deep /]# rpm -q foo
foo-2.3-8
```

This command will print the package name, version, and release number of installed package `foo`. Use this command to verify that a package is or is not installed on your system.

- To display package information, use the command:

```
[root@deep /]# rpm -qi foo
Name        : foo                Relocations: none
Version     : 2.3                Vendor: OpenNA.com, Inc.
Release     : 8                   Build Date: Thu 24 Aug 2000 11:16:53 AM EDT
Install date: Mon 12 Feb 2001 01:17:24 AM EST      Build Host: openna.com
Group       : Applications/Archiving        Source RPM: foo-2.3-8.src.rpm
Size        : 271467                           License: distributable
Packager    : OpenNA.com, Inc. <http://www.openna.com/>
Summary     : Here will appears summary of the package.
Description : Here will appears the description of the package.
```

This command displays package information; includes name, version, and description of the installed program. Use this command to get information about the installed package.

- To display package information before installing the program, use the command:

```
[root@deep /]# rpm -qpi foo-2.3-8.i386.rpm
Name        : foo                Relocations: none
Version     : 2.3                Vendor: OpenNA.com, Inc.
Release     : 8                   Build Date: Thu 24 Aug 2000 11:16:53 AM EDT
Install date: Mon 12 Feb 2001 01:17:24 AM EST      Build Host: openna.com
Group       : Applications/Archiving        Source RPM: foo-2.3-8.src.rpm
Size        : 271467                           License: distributable
Packager    : OpenNA.com, Inc. <http://www.openna.com/>
Summary     : Here will appears summary of the package.
Description : Here will appears the description of the package.
```

This command displays package information; includes name, version, and description of the program without the need to install the program first. Use this command to get information about a package before you install it on your system.

- To list files in a installed RPM package, use the command:

```
[root@deep /]# rpm -ql foo
/usr/bin/foo
/usr/bin/foo1
/usr/sbin/foo2
```

This command will list all files in a installed RPM package. It works only when the package is already installed on your system.

- To list files in package that is not already installed, use the command:

```
[root@deep /]# rpm -qpl foo
/usr/lib/foo
/usr/bin/foo1
/usr/sbin/foo2
```

This command will list all files in a RPM package that is not already installed on your system. It is useful when you want to know which components are included in the package before installing it.

- To know which files is part of which package, use the command:
```
[root@deep /]# rpm -qf /etc/passwd
setup-2.3.4-1
```

This command will show you from which RPM package the file comes from. It works only when the package is already installed on your system and it is very useful when you see some files into Linux that you do not know about it and want to get more information about its RPM provenance.

- To check a RPM signature package, use the command:
```
[root@deep /]# rpm --checksig foo
```

This command checks the PGP signature of specified package to ensure its integrity and origin. Always use this command first before installing new RPM package on your system.GnuPG or PGP software must be already installed on your system before you can use this command. See the chapter related to GnuPG installation and configuration for more information.

- To examine only the md5sum of the package, use the command:
```
[root@deep /]# rpm --checksig --nogpg foo
```

The RPM md5sum is useful to verify that a package has not been corrupted or tampered with. You can use it to be sure that the download of your new RPM package was not corrupted during network transfer.

Starting and stopping daemon services

The **init** program of Linux (also known as process control initialization) is in charge of starting all the normal and authorized processes that need to run at boot time on your system. These may include the APACHE daemons, NETWORK daemons, and anything else that must be running when your machine boots. Each of these processes has a script under the /etc/rc.d/init.d directory written to accept an argument, which can be start, stop, restart, etc. You can also execute those scripts by hand:

For example:

- To start the httpd Web Server daemon manually under Linux, you'll type:
```
[root@deep /]# /etc/rc.d/init.d/httpd start
Starting httpd:              [OK]
```

- To stop the httpd Web Server daemon manually under Linux, you'll type:
```
[root@deep /]# /etc/rc.d/init.d/httpd stop
Shutting down http:          [OK]
```

- To restart the httpd Web Server daemon manually under Linux, you'll type:
```
[root@deep /]# /etc/rc.d/init.d/httpd restart
Shutting down http:          [OK]
Starting httpd:              [OK]
```

Check inside your /etc/rc.d/init.d directory for services available and use the commands **start** | **stop** | **restart** to work around.

Software that must be uninstalled after installation of the server

Red Hat Linux installs other programs on your system by default and doesn't give you the choice to uninstall them during the install setup or programs which are going to be compiled from tarballs (source code). For this reason, you must uninstall the following software on your system after the installation of your Linux server.

In the table below, you'll find a partial list of software that must be uninstalled once the installation of your Linux server has been completed.

anacron	hotplug	pciutils
apmd	ipchains	pump
at	ksymoops	raidtools
dhcpcd	kudzu	redhat-logos
dosfstools	lokkit	redhat-release
eject	mailcap	setserial

Use the following RPM command to uninstall them:

- The command to uninstall RPM's software is:
  `[root@deep /]# rpm -e <softwarenames>`

Where `<softwarenames>` is the name of the software you want to uninstall e.g. (`foo`).

Step 1
Programs like **`apmd`**, **`Sendmail`**, **`at`** and **`anacron`** are daemons that run as process. It is better to stop those processes before uninstalling them from the system.

- To stop those processes, use the following commands:

```
[root@deep /]# /etc/rc.d/init.d/apmd stop
Shutting down APM daemon:            [OK]

[root@deep /]# /etc/rc.d/init.d/sendmail stop
Shutting down sendmail:              [OK]

[root@deep /]# /etc/rc.d/init.d/atd stop
Stopping at daemon:                  [OK]

[root@deep /]# /etc/rc.d/init.d/anacron stop
Shutting down anacron:               [OK]
```

Step 2
Once the processes `apmd`, `sendmail`, `at` and `anacron` programs have been stopped, you can safely uninstall them, and all the other packages, as shown below:

- To remove all the unneeded packages together, use the following commands:

```
[root@deep /]# rpm -e --nodeps anacron apmd at dhcpcd dosfstools eject
hotplug ipchains ksymoops kudzu lokkit mailcap pciutils pump raidtools
redhat-logos redhat-release setserial

[root@deep /]# rm -rf /var/spool/anacron/
```

Step 3
The program `hdparm` is needed by `IDE` hard disks but not `SCSI` hard disks. If you have an `IDE` disk on your system you must keep this program (`hdparm`), but if you don't have an `IDE` hard disk you can remove it safely from your system. `hdparm` is used to optimize your `IDE` hard drive. `SCSI` hard drives doesn't need to be optimized since they are capable to run at their full speed (80 Mps to 160 Mps) without modification.

- To remove the **`hdparm`** package from your system, use the following command:
  `[root@deep /]#` **`rpm -e hdparm`**

Step 4
The program `mkinitrd` is needed by `SCSI` or `RAID` hard disk but not `IDE` hard disks. If you have a `SCSI` or `RAID` disk on your system you must keep this program (`mkinitrd`), but if you don't have a `SCSI` or `RAID` hard disk you can safely remove it from your system.

- To remove the **`mkinitrd`** package from your system, use the following command:
  `[root@deep /]#` **`rpm -e --nodeps mkinitrd`**

Step 5
Use the programs `kbdconfig`, `mouseconfig`, `timeconfig`, `authconfig`, `ntsysv`, and `setuptool` in order to set your keyboard language and type, your mouse type, your default time zone, your NIS and shadow passwords, your numerous symbolic links in `/etc/rc.d` directory, and text mode menu utility which allow you to access all of these features. After those configurations have been set during the installation stage of your Linux server it's rare that you would need to change them again. So, you can uninstall them, and if in the future you need to change your keyboard, mouse, default time, etc again via test mode menu, all you have to do is to install the program with the RPM from your original CD-ROM.

- To remove all the above programs from your system, use the following command:
  `[root@deep /]#` **`rpm -e kbdconfig mouseconfig timeconfig authconfig ntsysv setuptool`**

Step 6
The program `quota` is a system administration tools for monitoring and limiting user/group disk usage, per file system. This program must be installed only on servers where the need for monitoring and restricting amount of disk space in users directories is require.

- To remove the **`quota`** package from your system, use the following command:
  `[root@deep /]#` **`rpm -e quota`**

Step 7
Even if you have not intending to install a mail server on your Linux system, the program `Sendmail` (or equivalent program) is always needed on your servers for potential messages sent to the root user by different software services installed on your machine.

Sendmail is a **M**ail **T**ransport **A**gent (MTA) program that sends mail from one machine to another. It can be configured in different manners; it can serve as an internal delivery mail system to a Mail Hub Server, or can be configured to be a Central Mail Hub Server for all Sendmail machines on your network. So depending on what you want to do with Sendmail, you must configure it to respond to your specific needs and speed. For this reason you must uninstall Sendmail and see the part in this book that is related to Mail Transfer Agent configuration and installation.

- To remove the **sendmail** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e sendmail
  ```

Step 8
Procmail is a mail-processing program, which can be used by Sendmail for all local mail delivery. This program is required only if you decide to install and use Sendmail on your server as a Central Mail Hub Server, and only if Sendmail is installed as a Central Hub Server. Since only a mail server with Sendmail as a MTA required procmail, it is better to uninstall procmail and install it only on the machine that will become your mail server with Sendmail.

- To remove the **procmail** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e procmail
  ```

Step 9
The OpenLDAP software is a set of protocols for accessing directory services like phone book style information and other kinds of information over the Internet. This useful program is not suitable for everyone and depends of what you want to do with your system. If you want to give it a try, see later in this book under the chapter related to databases for more information.

- To remove the **OpenLDAP** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e openldap
  ```

Step 10
The Cyrus SASL implementation is the Simple Authentication and Security Layer, a method for adding authentication support to connection-based protocols. It is used in conjunction with Cyrus, which is an electronic messaging program like Sendmail. Since we don't use and don't talk about it in this book, we can safety remove it.

- To remove the **Cyrus SASL** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e cyrus-sasl
  ```

Step 11
OpenSSL is an SSL encryption mechanism which ensures and provides safe and secure transactions of data over networks. This piece of software is one of the most important tools for a Linux server and it is highly recommended that it is installed. Unfortunately, the one that comes with Red Hat Linux is not up to date and not optimized for our specific server. For this reason, we will uninstall it now and see later in this book, under the chapters related to security software, how to install, secure, optimize and use it.

- To remove the **OpenSSL** package from your system, use the following command:
  ```
  [root@deep /]# rpm -e openssl
  [root@deep /]# rm -rf /usr/share/ssl/
  ```

Step 12
The `ash` package is a smaller version of the Bourne shell (`sh`). Since we already use `sh`, we can uninstall this package from our system. If you use this program in your regular administration task, then keep it installed on your server.

- To remove the **`ash`** package from your system, use the following command:
  `[root@deep /]#` **`rpm -e ash`**

Step 13
The `time` package is a utility for monitoring a program's use of system resources and can be used by developer to optimize their programs. This program is useful for developers.

- To remove the **`time`** package from your system, use the following command:
  `[root@deep /]#` **`rpm -e time`**

Step 14
The `krb5-libs` package contains the shared libraries needed by `Kerberos 5`. Because we're not using `Kerberos`, we'll need to uninstall this package. `Kerberos` is not secure as you can think and can be cracked easily with some good knowledge of this program. Anyway it is yours to decide if you really need it.

- To remove the **`krb5-libs`** package from your system, use the following command:
  `[root@deep /]#` **`rpm -e krb5-libs`**
  `[root@deep /]#` **`rm -rf /usr/kerberos/`**

Descriptions of programs that must be uninstalled after installation of the server

Below is the list of programs and a short description of their purpose. We must uninstall them for increased security and to make more space on our server. For more information and an explanation of their capabilities and uses, please see your Red Hat manual or query the package by making an "**`rpm -qi foo`**" command before uninstalling it.

- The `anacron` package is similar to the `cron` package but differ in the way that it does not assume that the system is running continuously and it is a good command scheduler for system which don't run 24 hours a day. **[Unnecessary for a server]**

- The `apmd` package, or advanced Power Management daemon utilities, can watch your notebook's battery and warn all users when the battery is low. **[Unnecessary for a server]**

- The `at` package is a utility that will do time-oriented job control by scheduling a command to run later. Unfortunately, it has had a rich history of problems and we can achieve the same functionality with the more secure `vixie-cron` package. For this reason I recommend you to uninstall it. **[Security Risks]**

- The `dhcpcd` package contains the protocol, which allows systems to get their own network configuration information from a DHCP server. If your are going to use DHCP on your network, it is recommended to install the DHCP client included in the `pump` package, which provides a faster and simpler DHCP client. **[Unnecessary]**

- The dosfstools package contains utilities for making and checking MS-DOS FAT file systems on Linux. Remember that we want to install a Linux server on our system and not a PC with two different operating systems on it. Therefore we must uninstall this program from our computer. **[Unnecessary, we run a server]**

- The eject package contains an eject program that allows the user to eject removable media (typically CD-ROMs, floppy disks, Iomega Jaz or Zip disks) using software. **[Necessary only if you have a tape backup on this server]**

- The hotplug package is a helper application for loading modules for USB devices. On a server environment, USB devices are not used at all and are required only on Linux workstation. **[Unnecessary, we run a server]**

- The ipchains package is the old tool used with Linux kernel 2.2 for managing Linux kernel packet filtering capabilities and to set up firewalling on the network. A new and more powerful tool named "IPTABLES" exists and this is the one that we will use later in the book to set up our firewall on Linux. **[Unnecessary]**

- The ksymoops package is a small tool used to report kernel oops and error message decoder. This package is useful for developers that work on the Linux kernel and want to debug or for users that want to report bugs with the kernel. The same result can be achieved with the dmesg command of Linux. **[Unnecessary]**

- The kudzu package is a hardware-probing tool run at system boot time to determine what hardware has been added or removed from the system. **[Unnecessary, we run a server]**

- The lokkit package is a Firewall configuration application for an average end user and it is not designed to configure arbitrary firewalls since it is solely designed to handle typical dialup user and cable modem set-ups. It is not the answer to a complex firewall configuration, and it is not the equal of an expert firewall designer. **[Unnecessary]**

- Metamail is a program that uses the mailcap file to determine how it should display non-text or multimedia material. **[Unnecessary]**

- The pciutils package contains various utilities for inspecting and setting devices connected to the PCI bus. **[We use other methods]**

- The Pump DHCP package allows individual diskless clients on a network to get their own IP network configuration information from network servers. **[Unnecessary]**

- The raidtools package includes the tools you need to set up and maintain a software RAID device on a Linux system. **[Depending if you use Raid or not]**

- The redhat-logos package contains files of the Red Hat "Shadow Man" logo and the RPM logo. **[Unnecessary on a server]**

- The redhat-release package contains the Red Hat Linux release file. **[Unnecessary]**

- The setserial package is a basic system utility for displaying or setting serial port information. **[Unnecessary]**

NOTE ABOUT SYSTEM SIZE: If all the packages described in this section have been uninstalled from the system, then our install size of Linux is now **132MB**.

Remove unnecessary documentation files

Well, 132MB is very good but we can do more. By default the majority of each RPM's packages installed under Linux comes with documentation files related to the software. This documentation contains original files from the program tar archive like `README`, `FAQ`, `BUG`, `INSTALL`, `NEWS`, `PROJECTS` and more.

Many of them can be easily retrieved from the website where the program has been downloaded and it makes no sense for them to be kept on your system. I know that hard drives costs have come down considerably recently, but why keep this kind of documentation on a secure server if it unlikely they will not be read more than once. Anyway, have a look inside those files and decide for yourself if you want to remove them or not.

- To remove all documentation files from your system, use the following commands:

```
[root@deep /]# cd /usr/share/doc/
[root@deep doc]# rm -rf *
```

NOTE ABOUT SYSTEM SIZE: If all the documentation files have been removed from the system, then our install size of Linux is now **118MB**.

Remove unnecessary/empty files and directories

There are some files and directories we can remove manually from the file system of Linux to make a clean install. These files and directories are not needed but still exist after our secure installation of Linux and can be removed safely. Some are bugs from the Red Hat installation script and others are created by default even if you don't use them.

- To remove all unnecessary/empty files and directories from your system, use the following commands:

```
[root@deep /]# rm -f /etc/exports
[root@deep /]# rm -f /etc/printcap
[root@deep /]# rm -f /etc/ldap.conf
[root@deep /]# rm -f /etc/yp.conf
[root@deep /]# rm -f /etc/hosts.allow
[root@deep /]# rm -f /etc/hosts.deny
[root@deep /]# rm -rf /etc/xinetd.d/
[root@deep /]# rm -rf /etc/hotplug/
[root@deep /]# rm -rf /etc/ppp/
[root@deep /]# rm -rf /etc/opt/
[root@deep /]# rm -rf /etc/X11/
[root@deep /]# rm -rf /opt/
[root@deep /]# rm -rf /var/opt/
[root@deep /]# rm -rf /var/nis/
[root@deep /]# rm -rf /var/spool/lpd/
[root@deep /]# rm -rf /usr/X11R6/
[root@deep /]# rm -rf /usr/etc/
[root@deep /]# rm -rf /usr/local/
[root@deep /]# rm -rf /usr/dict/
[root@deep /]# rm -f /usr/bin/X11
[root@deep /]# rm -f /usr/bin/kbdrate
[root@deep /]# rm -f /usr/lib/X11
```

```
[root@deep /]# rm -f /usr/lib/libcrypto.so.1
[root@deep /]# rm -f /usr/lib/libssl.so.1
[root@deep /]# rm -rf /usr/lib/games/
[root@deep /]# rm -rf /usr/share/empty/
[root@deep /]# rm -rf /usr/share/pixmaps/
```

NOTE: If in the future you want to install a program which needs some of the files/directories we have removed, then the program will automatically recreate the missing files or directories. Good!

Software that must be installed after installation of the server

There are certain programs required to be able to compile programs on your server, for this reason you must install the following RPM packages. This part of the installation is very important and requires that you install all the packages described below.

These are on your Red Hat Part 1 and Part 2 CD-ROMs under RedHat/RPMS directory and represents the necessary base software needed by Linux to compile and install programs. Please note that if you don't want to compile software in your server or if you only use RPM's packages to update programs or if you use a dedicated server to develop, compile or create your own RPM's packages which will be installed later along your network on the servers, then you **DON'T** need to install the packages described here.

Step 1
First, we mount the CD-ROM drive and move to the RPMS subdirectory of the CD-ROM.

- To mount the CD-ROM drive and move to RPM directory, use the following commands:

```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
had: ATAPI 32X CD-ROM drive, 128kB Cache
mount: block device dev/cdrom is write-protected, mounting read-only
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
```

These are the packages that we need to be able to compile and install programs on the Linux system. Remember, this is the minimum number of packages that permits you to compile most of the tarballs available for Linux. Other compiler packages exist on the Linux CD-ROM, so verify with the `README` file that came with the tarballs program you want to install if you receive error messages during compilation of the specific software.

The compiler packages:
Compiler packages contains programs and languages used to build software on the system. Remember to uninstall all of the following compiler packages after succesfull installation of all software required on your Linux server.

binutils-2.10.91.0.2-3.i386.rpm	flex-2.5.4a-13.i386.rpm
bison-1.28-5.i386.rpm	gcc-2.96-81.i386.rpm
byacc-1.9-18.i386.rpm	gcc-c++-2.96-81
cdecl-2.5-17.i386.rpm	kernel-headers-2.4.2-2.i386.rpm
cpp-2.96-81.i386.rpm	m4-1.4.1-4.i386.rpm
cproto-4.6-7.i386.rpm	make-3.79.1-5.i386.rpm
ctags-4.0.3-1.i386.rpm	patch-2.5.4-9.i386.rpm
dev86-0.15.0-5.i386.rpm	perl-5.6.0-12.i386.rpm

The development packages:
Development packages contain header and other files required during compilation of software. In general, development packages are needed when we want to add some specific functionality to the program that we want to compile. For example if I want to add `PAM` support to `IMAP`, I'll need `pam-devel`, which contains the required header files for `IMAP` to compile successfully.

As for compiler packages, all development packages must be uninstalled after successful compilation of all the software that you need on your Linux server. Remember to uninstall them since they are not needed for proper functionality of the server, but just to compile the programs.

aspell-devel-0.32.6-2	libpng-devel-1.0.9-1
db3-devel-3.1.17-7	libstdc++-devel-2.96-81
freetype-devel-2.0.1-4	ncurses-devel-5.2-8
gd-devel-1.8.3-7	pam-devel-0.74-22
gdbm-devel-1.8.0-5	pspell-devel-0.11.2-2
glibc-devel-2.2.2-10	zlib-devel-1.1.3-22
libjpeg-devel-6b-15	

Dependencies packages:
Dependencies packages are other RPM packages needed by the RPM packages that we want to install. This happens because some RPM's are directly linked with others and depend on each one to function properly. The following packages are required by the above RPM packages and we will install them to satisfy dependencies. After proper compilation and installation of all needed software on the Linux server, we can uninstall them (if not needed by special program that we will install) safety.

gd-1.8.3-7	libpng-1.0.9-1
freetype-2.0.1-4	libtool-libs-1.3.5-8
libjpeg-6b-15	pspell-0.11.2-2

Step 2
It is better to install the software described above together if you don't want to receive dependencies error messages during the install. Some of the RPMs reside on CD-ROM Part 1 and other on CD-ROM Part2 of Red Hat. For easy installation, I recommend you to copy all of the required packages (compilers and development) to your hard drive and install them from there.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rpm -Uvh *.rpm
binutils          ##################################################
bison             ##################################################
byacc             ##################################################
cdecl             ##################################################
cpp               ##################################################
cproto            ##################################################
ctags             ##################################################
dev86             ##################################################
flex              ##################################################
gcc               ##################################################
kernel-headers    ##################################################
gcc-c++           ##################################################
m4                ##################################################
make              ##################################################
patch             ##################################################
perl              ##################################################
aspell-devel      ##################################################
db3-devel         ##################################################
freetype-devel    ##################################################
```

```
gd-devel          ##################################################
gdbm-devel        ##################################################
glibc-devel       ##################################################
libjpeg-devel     ##################################################
libpng-devel      ##################################################
libstdc++-devel   ##################################################
ncurses-devel     ##################################################
pam-devel         ##################################################
pspell-devel      ##################################################
zlib-devel        ##################################################
gd                ##################################################
libjpeg           ##################################################
libpng            ##################################################
pspell            ##################################################
freetype          ##################################################
libtool-libs      ##################################################
```

NOTE: Some of the RPM reside on CD-ROM part 1 and other on CD-ROM Part2 of Red Hat. For easy installation, I recommend you to copy all of the required packages (compilers and development) to your hard drive and install them from there.

NOTE ABOUT SYSTEM SIZE: If you have installed all the require packages described above to be able to make compilation in the system, then our install size of Linux is now **222MB**.

Step 3
This step is required only if you also want to use the Linux server to compile programs and services. If you have a dedicated system to compile and build RPM packages, which can be installed on the other servers on your network, you don't need this step.

After installation and compilation of all programs and services, it's a good idea to remove all sharp objects (compilers, etc) described above unless they are required by your system. A few reasons are:

- ✓ If a cracker gains access to your server he or she cannot compile or modify binary programs. Also, this will free a lot of space and will help to improve regular scanning of the files on your server for integrity checking.

- ✓ When you run a server you will give it a special task to accomplish. You will never put all services you want to offer in one machine or you will lose speed (resources available divided by the number of process running on the server)

- ✓ Decrease your security with a lot of services running on the same machine, if a cracker accesses this server, he or she can attack directly all the others available.

- ✓ Having different servers doing different tasks will simplify the administration, and management. You know what task each server is supposed to do, what services should be available, which ports are open to clients access and which one are closed, you know what you are supposed to see in the log files, etc, and give you more control and flexibility on each one (server dedicated for mail, web pages, database, development, backup, etc.

- ✓ For example, one server specialized just for development and testing will mean you will not be compelled to install compiler programs on a server each time you want to compile and install new software on it, and be obliged afterwards to uninstall the compilers, or other sharp objects.

Verifying installed programs on your Server

If you have followed each step exactly as described, this is the list of all installed programs that you should have on your server after the complete installation of Linux.

Step 1
This list must match exactly the `install.log` file located in your `/tmp` directory or you could run into problems.

glibc-common	ed	**mkinitrd**
mailcap	fileutils	lilo
redhat-logos	**at**	**mkbootdisk**
redhat-release	findutils	**mouseconfig**
setup	gawk	**time**
filesystem	gettext	tmpwatch
basesystem	grep	crontabs
glibc	**ash**	utempter
termcap	**dhcpcd**	vim-common
bdflush	gzip	vim-minimal
chkconfig	less	which
cracklib	man	words
db1	net-tools	cracklib-dicts
db2	**openssl**	pam
db3	popt	**authconfig**
dosfstools	logrotate	**cyrus-sasl**
e2fsprogs	**procmail**	gpm
eject	procps	**kudzu**
file	psmisc	passwd
gdbm	pwdb	sh-utils
glib	**raidtools**	**krb5-libs**
hdparm	readline	**openldap**
ksymoops	rootfiles	**sendmail**
libtermcap	sed	SysVinit
losetup	console-tools	zlib
mailx	**setserial**	rpm
mingetty	shadow-utils	util-linux
mktemp	dev	initscripts
bash	slang	apmd
bzip2	newt	devfsd
hotplug	**kbdconfig**	**ipchains**
libstdc++	**ntsysv**	kernel
groff	**setuptool**	**lokkit**
MAKEDEV	slocate	**pciutils**
modutils	sysklogd	**pump**
ncurses	syslinux	**quota**
info	tar	**timeconfig**
cpio	textutils	vixie-cron
diffutils	mount	**anacron**

NOTE: All texts in bold are packages that we have uninstalled from the default install list. Remember that some of these RPM packages will be reinstalled manually later in this book and most are unnecessary for daily work of the system.

Step 2
After we have uninstalled all the software that must be uninstalled and the addition of the necessary RPM packages to be able to compile programs we must verify the list of all installed RPM programs again, but this time with the following command:

- To verify the list of all installed RPM package on your system, use the command:
  `[root@deep /]# rpm -qa > installed_rpm`

The "`-qa`" option will query all installed RPM packages on your system and the special character "`>`" will redirect the output to the file named `installed_rpm`.

The content of the `installed_rpm` file must look exactly like this:

filesystem-2.0.7-1	kernel-2.4.2-2	passwd-0.64.1-4
glibc-2.2.2-10	vixie-cron-3.0.1-62	zlib-1.1.3-22
bdflush-1.5-16	glibc-common-2.2.2-10	util-linux-2.10s-12
cracklib-2.7-8	setup-2.4.7-1	**binutils-2.10.91.0.2-3**
db2-2.4.14-5	basesystem-7.0-2	**byacc-1.9-18**
gdbm-1.8.0-5	termcap-11.0.1-8	**cpp-2.96-81**
libtermcap-2.0.8-26	chkconfig-1.2.22-1	**ctags-4.0.3-1**
mailx-8.1.1-20	db1-1.85-5	**dev86-0.15.0-5**
mktemp-1.5-8	db3-3.1.17-7	**kernel-headers-2.4.2-2**
bzip2-1.0.1-3	e2fsprogs-1.19-4	**gcc-2.96-81**
libstdc++-2.96-81	file-3.33-1	**gcc-c++-2.96-81**
MAKEDEV-3.1.0-14	glib-1.2.9-1	**make-3.79.1-5**
ncurses-5.2-8	losetup-2.10r-5	**perl-5.6.0-12**
cpio-2.4.2-20	mingetty-0.9.4-16	**bison-1.28-5**
ed-0.2-19	bash-2.04-21	**cdecl-2.5-17**
gawk-3.0.6-1	groff-1.16.1-7	**cproto-4.6-7**
grep-2.4.2-5	modutils-2.4.2-5	**flex-2.5.4a-13**
less-358-16	info-4.0-20	**glibc-devel-2.2.2-10**
net-tools-1.57-6	diffutils-2.7-21	**m4-1.4.1-4**
popt-1.6.2-8	fileutils-4.0.36-4	**patch-2.5.4-9**
psmisc-19-4	findutils-4.1.6-2	**aspell-devel-0.32.6-2**
rootfiles-7.0-4	gettext-0.10.35-31	**db3-devel-3.1.17-7**
console-tools-19990829-34	gzip-1.3-12	**freetype-devel-2.0.1-4**
shadow-utils-20000826-4	man-1.5h1-20	**gd-devel-1.8.3-7**
slang-1.4.2-2	logrotate-3.5.4-1	**gdbm-devel**
sysklogd-1.4-7	procps-2.0.7-8	**libjpeg-devel-6b-15**
tar-1.13.19-4	pwdb-0.61.1-1	**libpng-devel-1.0.9-1**
mount-2.10r-5	readline-4.1-9	**libstdc++-devel-2.96-81**
lilo-21.4.4-13	sed-3.02-9	**ncurses-devel-5.2-8**
tmpwatch-2.7.1-1	dev-3.1.0-14	**pam-devel-0.74-22**
utempter-0.5.2-4	newt-0.50.22-2	**pspell-devel-0.11.2-2**
vim-minimal-6.0-0.27	slocate-2.5-5	**zlib-devel-1.1.3-22**
words-2-16	syslinux-1.52-1	**gd-1.8.3-7**
pam-0.74-22	textutils-2.0.11-7	**freetype-2.0.1-4**
sh-utils-2.0-13	crontabs-1.9-2	**libjpeg-6b-15**
SysVinit-2.78-15	vim-common-6.0-0.27	**libpng-1.0.9-1**
rpm-4.0.2-8	which-2.12-1	**libtool-libs-1.3.5-8**
initscripts-5.83-1	cracklib-dicts-2.7-8	**pspell-0.11.2-2**
devfsd-2.4.2-2	gpm-1.19.3-16	

NOTE: All texts in bold are compiler packages that we have added to be able to compile programs on the system. Remember that these packages can be uninstalled after complete compilation of all software safety and without problem.

This second step is required to be sure we have not forgotten to remove some unnecessary RPM or to add some important packages that permit us to compile programs on the system. If the result looks the same as our `installed_rpm` file above, we are now ready to play with our new Linux server.

In the above list, I assume that all sharp objects required for making compilation of programs and services on the system are installed. Of course they must be uninstalled and removed from the list if we don't want to use this server to compile programs and services but prefer to use RPM packages made on another system for all servers on our network.

Update of the latest software

Keep all software (especially network software) up to date with the latest versions. Check the errata pages for the Red Hat Linux distribution, available at http://www.redhat.com/apps/support/updates.html. The errata pages are perhaps the best resource for fixing 90% of the common problems with Red Hat Linux. In addition, security holes for which a solution exists are generally on the errata page 24 hours after Red Hat has been notified. You should always check there first.

Step 1
For all software packages described here and later in this book, I assume that you use another computer on your network to retrieve and download the required software. If this is not the case, I suggest you at least install the `FTP` client program that comes with your OS CD-ROM and install it, to be able to make remote connections and download files.

Of course if for some obscure reasons the networking feature of your server doesn't work at this stage, I recommend you to read the part of the book called "Networking Related Reference" and especially the chapter under it called "Networking - `TCP/IP` Network Management" for troubleshooting and more information on the subject.

This secure Linux server installation requires that the software listed below be installed on your system to be able to download packages from the Internet. if you don't use another computer on your network to retrieve and download programs.

- ✓ `ftp`, which provides the standard UNIX command-line `FTP` (**F**ile **T**ransfer **P**rotocol) client, must already be installed on your system to be able to download software on the Internet.

- ➢ To verify if `ftp` package is installed on your system, use the command:
```
[root@deep /]# rpm -q ftp
package ftp is not installed
```

- To mount your CD-ROM drive before installing the require package, use the command:
```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
mount: block device /dev/cdrom is write-protected, mounting read-only
```

- To install the `ftp` package on your Linux system, use the following command:
```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh ftp-version.i386.rpm
```

```
ftp       ##################################################
```

- To unmount your CD-ROM drive, use the following command:
  `[root@deep RPMS]# cd /; umount /mnt/cdrom/`

The following are based on information listed by Red Hat as of 2001/04/23. Please check regularly at http://www.redhat.com/ for the latest status.

Errata: Bug, Fixes & Advisories are available from:
Red Hat Updates Web Site: http://www.redhat.com/apps/support/updates.html
Red Hat Updates FTP Site: 216.148.218.202, 63.240.14.64, 216.148.218.201, 63.240.14.63, 216.148.218.192, 63.240.14.62

Step 2
Software that must be updated at this time for your Red Hat Linux Secure Server are:

```
mount-2.11b-3.i386.rpm
```

NOTE: You can also retrieve all present and future software RPM packages that will need to be updated directly from our OpenNA.com website at: www.openna.com/downloads/downloads.php

Part II Security and Optimization Related Reference

In this Part

Security and Optimization - General System Security
Security and Optimization - Pluggable Authentication Modules
Security and Optimization - General System Optimization
Security and Optimization - Kernel Security & Optimization

Now that we have installed a base system, the next four chapters will concentrate on how to tighten the security of our configured system, optimize our system to perform at its peak and upgrade our machine for the latest kernel.

Please note that when we talk of tightening the security we are referring to the features available within the base installed system and not to any additional software. We will talk about them later in this book.

3 Security and Optimization - General System Security

In this Chapter

Linux General System Security

Abstract
A secure Linux server depends on how the administrator makes it. Once we have eliminated the potential security risk by removing unneeded services, we can start to secure our existing services and software on our server. Within a few hours of installing and configuring your system, you can prevent many attacks before they occur. In this chapter we will discuss some of the more general, basic techniques used to secure your system. The following is a list of features that can be used to help prevent attacks from external and internal sources.

BIOS
It is recommended to disallow booting from floppy drives and set passwords on BIOS features. You can check your BIOS manual or look at it thoroughly the next time you boot up your system to find out how to do this. Disabling the ability to boot from floppy drives and being able to set a password to access the BIOS features will improve the security of your system.

This will block unauthorized people from trying to boot your Linux system with a special boot disk and will protect you from people trying to change BIOS features like allowing boot from floppy drive or booting the server without prompt password. It is important to note that there is a possibility to bypass this security measure if someone has a physical access to your server since they can open the computer and unplug the BIOS battery. This will reset all features to their initial values.

Unplug your server from the network
It is not wise to apply security changes in your newly installed Linux server if you are online. So it is preferable to deactivate all network interfaces in the system before applying security changes.

- To stop specific network devices manually on your system, use the following command:
```
[root@deep /]# ifdown eth0
```
- To start specific network devices manually on your system, use the following command:
```
[root@deep /]# ifup eth0
```
- To shut down all network interfaces, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/network stop
Shutting down interface eth0          [OK]
Disabling Ipv4 packet forwarding      [OK]
```
- To start all network interfaces, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/network start
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
```

Security as a policy
It is important to point out that you cannot implement security if you have not decided what needs to be protected, and from whom. You need a security policy--a list of what you consider allowable and what you do not consider allowable upon which to base any decisions regarding security. The policy should also determine your response to security violations. What you should consider when compiling a security policy will depend entirely on your definition of security. The following questions should provide some general guidelines:

- ✓ How do you classify confidential or sensitive information?
- ✓ Does the system contain confidential or sensitive information?
- ✓ Exactly whom do you want to guard against?
- ✓ Do remote users really need access to your system?
- ✓ Do passwords or encryption provide enough protection?
- ✓ Do you need access to the Internet?
- ✓ How much access do you want to allow to your system from the Internet?
- ✓ What action will you take if you discover a breach in your security?

This list is short, and your policy will probably encompass a lot more before it is completed. Any security policy must be based on some degree of paranoia; deciding how much you trust people, both inside and outside your organization. The policy must, however, provide a balance between allowing your users reasonable access to the information they require to do their jobs and totally disallowing access to your information. The point where this line is drawn will determine your policy.

Choose a right password

The starting point of our Linux General Security tour is the password. Many people keep their valuable information and files on a computer, and the only thing preventing others from seeing it is the eight-character string called a password. An unbreakable password, contrary to popular belief, does not exist. Given time and resources all passwords can be guessed either by social engineering or by brute force.

Social engineering of server passwords and other access methods are still the easiest and most popular way to gain access to accounts and servers. Often, something as simple as acting as a superior or executive in a company and yelling at the right person at the right time of the day yields terrific results.

Running a password cracker on a weekly basis on your system is a good idea. This helps to find and replace passwords that are easily guessed or weak. Also, a password checking mechanism should be present to reject a weak password when choosing an initial password or changing an old one. Character strings that are plain dictionary words, or are all in the same case, or do not contain numbers or special characters should not be accepted as a new password.

We recommend the following rules to make passwords effective:

- ✓ They should be at least six characters in length, preferably eight characters including at least one numeral or special character.
- ✓ They must not be trivial; a trivial password is one that is easy to guess and is usually based on the user's name, family, occupation or some other personal characteristic.
- ✓ They should have an aging period, requiring a new password to be chosen within a specific time frame.
- ✓ They should be revoked and reset after a limited number of concurrent incorrect retries.

The root account

The "root" account is the most privileged account on a Unix system. The "root" account has no security restrictions imposed upon it. This means the system assumes you know what you are doing, and will do exactly what you request -- no questions asked. Therefore it is easy, with a mistyped command, to wipe out crucial system files. When using this account it is important to be as careful as possible. For security reasons, never log in on your server as "root" unless it is absolutely an instance that necessitates root access. Also, if you are not on your server, never sign in and leave yourself on as "root"--this is VERY, VERY. VERY BAD.

Set login time out for the root account

Despite the notice to never, if they are not on the server, sign in as “root” and leave it unattended, administrators still stay on as “root” or forget to logout after finishing their work and leave their terminals unattended.

The answer to solve this problem is to make the bash shell automatically logout after not being used for a period of time. To do that, you must set the special variable of Linux named “`TMOUT`” to the time in seconds of no input before logout.

- Edit your **`profile`** file (`vi /etc/profile`) and add the following line somewhere after the line that read “`HISTSIZE=`” on this file:

  `TMOUT=7200`

The value we enter for the variable “`TMOUT=`” is in seconds and represents 2 hours (60 * 60 = 3600 * 2 = 7200 seconds). It is important to note that if you decide to put the above line in your `/etc/profile` file, then the automatic logout after two hours of inactivity will apply for all users on the system. So, instead, if your prefer to control which users will be automatically logged out and which ones are not, you can set this variable in their individual `.bashrc` file.

After this parameter has been set on your system, you must logout and login again (as root) for the change to take effect.

The `/etc/exports` file

If you are exporting file systems using the `NFS` service, be sure to configure the **`/etc/exports`** file with the most restrictive access possible. This means not using wildcards, not allowing root write access, and mounting read-only wherever possible.

Step 1

- Edit the **`exports`** file (`vi /etc/exports`) and add:

  As an example:
  `/dir/to/export host1.mydomain.com(ro,root_squash)`
  `/dir/to/export host2.mydomain.com(ro,root_squash)`

Where `/dir/to/export` is the directory you want to export, `host1.mydomain.com` is the machine allowed to log in this directory, the `<ro>` option mean mounting read-only and the `<root_squash>` option for not allowing root write access in this directory.

Step 2

- For this change to take effect you will need to run this command on your terminal:
  `[root@deep]# **/usr/sbin/exportfs -a**`

WARNING: Please be aware that having an NFS service available on your system can be a security risk. Personally, I don't recommend using it. If you are follow our installation, the NFS service is not installed in your system.

The single-user login mode of Linux

Linux has a special command (`linux single`) also known as 'single-user mode', which can be entered at the boot prompt during startup of the system. The single-user mode is generally used for system maintenance. You can boot Linux in single-user mode by typing at the `LILO` boot prompt the following command:

```
LILO:  linux single
```

This will place the system in Run level 1 where you'll be logged in as the super-user 'root', and where you won't even have to type in a password!

Step 1
Requiring no password to boot into root under single-user mode is a bad idea! You can fix this by editing the **`inittab`** file (`vi /etc/inittab`) and change the following line:

```
id:3:initdefault:
```

To read:

```
id:3:initdefault:
~~:S:wait:/sbin/sulogin
```

The addition of the above line will require to enter the root password before continuing to boot into `single-user` mode by making `init (8)` run the program `sulogin (8)` before dropping the machine into a root shell for maintenance.

Step 2

- Now, for the change to take effect type in the following at a prompt:

```
[root@deep /]# /sbin/init q
```

The `LILO` and `/etc/lilo.conf` file

`LILO` is the most commonly used boot loader for Linux. It manages the boot process and can boot Linux kernel images from floppy disks, hard disks or can even act as a "boot manager" for other operating systems.

`LILO` is very important in the Linux system and for this reason, we must protect it the best we can. The most important configuration file of `LILO` is the `lilo.conf` file, and it resides under the `/etc` directory. It is with this file that we can configure and improve the security of our `LILO` program and Linux system. Following are three important options that will improve the security of our valuable `LILO` program.

- Adding: **timeout=00**

This option controls how long (in seconds) LILO waits for user input before booting to the default selection. One of the requirements of C2 security is that this interval be set to 0 unless the system dual boots something else.

- Adding: **restricted**

This option asks for a password only, if parameters are specified on the command line (e.g. linux single). The option "restricted" can only be used together with the "password" option. Make sure you use this one on each additional image you may have.

- Adding: **password=<password>**

This option asks the user for a password when trying to load the image. Actually the effect of using the password parameter in /etc/lilo.conf will protect the Linux image from booting. This means, it doesn't matter if you load Linux in single mode or if you just do a normal boot. It will always ask you for the password.

Now this can have a very bad effect, namely you are not able to reboot Linux remotely any more since it won't come up until you type in the root password at the console. It is for this reason that adding "restricted" with "password" is very important since the option "restricted" relaxes the password protection and a password is required only if parameters are specified at the LILO prompt, (e.g. single).

Passwords are always case-sensitive, also make sure the /etc/lilo.conf file is no longer world readable, or any user will be able to read the password. Here is an example of our protected LILO with the lilo.conf file.

Step 1

- Edit the **lilo.conf** file (vi /etc/lilo.conf) and add or change the three options above as show:

```
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
prompt ← remove this line if you don't want to pass options at the LILO prompt.
timeout=00 ← change this line to 00 to disable the LILO prompt.
linear
message=/boot/message ← remove this line if you don't want the welcome screem.
default=linux
restricted ← add this line to relaxes the password protection.
password=<password> ← add this line and put your password.

image=/boot/vmlinuz-2.4.2-2
    label=linux
    initrd=/boot/initrd-2.4.2-2.img
    read-only
    root=/dev/sda6
```

Step 2

Because the configuration file /etc/lilo.conf now contains unencrypted passwords, it should only be readable for the super-user "root".

- To make the /etc/lilo.conf file readable only by the super-user "root", use the following command:

```
[root@deep /]# chmod 600 /etc/lilo.conf (will be no longer world readable).
```

Step 3
Now we must update our configuration file `/etc/lilo.conf` for the change to take effect.

- To update the `/etc/lilo.conf` file, use the following command:

```
[root@deep /]# /sbin/lilo -v
LILO version 21.4-4, copyright © 1992-1998 Wernerr Almesberger
'lba32' extentions copyright © 1999,2000 John Coffman

Reading boot sector from /dev/sda
had : ATAPI 32X CD-ROM drive, 128kB Cache
Merging with /boot/boot.b
Mapping message file /boot/message
Boot image : /boot/vmlinuz-2.2.16-22
Mapping RAM disk /boot/initrd-2.2.16-22.img
Added linux *
/boot/boot.0800 exists - no backup copy made.
Writing boot sector.
```

Step 4
One more security measure you can take to secure the `lilo.conf` file is to set it immutable, using the **`chattr`** command.

- To set the file immutable simply, use the following command:

```
[root@deep /]# chattr +i /etc/lilo.conf
```

And this will prevent any changes (accidental or otherwise) to the `lilo.conf` file. If you wish to modify the `lilo.conf` file you will need to unset the immutable flag:

- To unset the immutable flag, use the following command:

```
[root@deep /]# chattr -i /etc/lilo.conf
```

WARNING: When you use the `password` option, then `LILO` will always ask you for the password, regardless if you pass options at the `LILO` prompt (e.g. `single`) or not **EXCEPT** when you set the "`restricted`" option in `/etc/lilo.conf`.

The option "`restricted`" relaxes the password protection and a password is required only if parameters are specified at the `LILO` prompt, (e.g. `single`).

If you didn't had this option set "`restricted`", Linux will always ask you for the password and you will not be able to remotely reboot your system, therefore don't forget to add the option "`restricted`" with the option "`password`" into the `/etc/lilo.conf` file.

Disabling `Ctrl-Alt-Delete` keyboard shutdown command

Commenting out the line (with a "**#**") listed below in your `/etc/inittab` file will disable the possibility of using the `Control-Alt-Delete` command to shutdown your computer. This is pretty important if you don't have the best physical security to the machine.

Step 1

- To do this, edit the **inittab** file (`vi /etc/inittab`) and change/comment the line:

  ```
  ca::ctrlaltdel:/sbin/shutdown -t3 -r now
  ```

  To read:

  ```
  #ca::ctrlaltdel:/sbin/shutdown -t3 -r now
  ```

Step 2

- Now, for the change to take effect type in the following at a prompt:

  ```
  [root@deep /]# /sbin/init q
  ```

The `/etc/services` file

The port numbers on which certain "standard" services are offered are defined in the RFC 1700 "Assigned Numbers". The `/etc/services` file enables server and client programs to convert service names to these numbers (ports). The list is kept on each host and it is stored in the file `/etc/services`. Only the "root" user is allowed to make modifications to this file. It is rare to edit the `/etc/services` file. since it already contains the more common service names / port numbers. To improve security, we can set the immutable flag on this file to prevent unauthorized deletion or modification.

- To immunize the `/etc/services` file, use the following command:
  `[root@deep /]# chattr +i /etc/services`

The `/etc/securetty` file

The `/etc/securetty` file allows you to specify which **TTY** and **VC** (virtual console) devices the “root” user is allowed to login on. The `/etc/securetty` file is read by the login program (usually `/bin/login`). Its format is a list of the **tty** and **vc** devices names allowed, and for all others that are commented out or do not appear in this file, root login is disallowed.

Disable any **tty** and **vc** devices that you do not need by commenting them out (**#** at the beginning of the line) or by removing them.

- Edit the **securetty** file (`vi /etc/securetty`) and comment out or remove the following lines:

```
vc/1                                    tty1
#vc/2                                   #tty2
#vc/3                                   #tty3
#vc/4                                   #tty4
#vc/5                                   #tty5
#vc/6                                   #tty6
#vc/7                                   #tty7
#vc/8                                   #tty8
#vc/9                                   #tty9
#vc/10                                  #tty10
#vc/11                                  #tty11
```

Which means root is allowed to login on only **tty1** and **vc/1**. This is my recommendation, allowing “root” to log in on only one `tty` or `vc` device and use the `su` command to switch to “root” if you need more devices to log in as “root”.

Special accounts

It is important to **DISABLE ALL default vendor accounts that** you don't use on your system (some accounts exist by default even if you have not installed the related services on your server). This should be checked after each upgrade or new software installation. Linux provides these accounts for various system activities, which you may not need if the services are not installed on your server. If you do not need the accounts, remove them. The more accounts you have, the easier it is to access your system.

We assume that you are using the Shadow password suite on your Linux system. If you are not, you should consider doing so, as it helps to tighten up security somewhat. This is already set if you've followed our Linux installation procedure and selected, under the "Authentication Configuration", the option to "Enable Shadow Passwords" (see the chapter related to the "Installation of your Linux Server" for more information).

- To delete user on your system, use the following command:
  ```
  [root@deep /]# userdel username
  ```

- To delete group on your system, use the following command:
  ```
  [root@deep /]# groupdel username
  ```

Step 1

First we will remove all default vendor accounts into the `/etc/passwd` file that are unnecessary for the operation of the secure server configuration that we use in this book.

- Type the following commands to delete all default users accounts listed below:
  ```
  [root@deep /]# userdel adm
  [root@deep /]# userdel lp
  [root@deep /]# userdel shutdown
  [root@deep /]# userdel halt
  [root@deep /]# userdel news
  [root@deep /]# userdel mail
  [root@deep /]# userdel uucp
  [root@deep /]# userdel operator
  [root@deep /]# userdel games
  [root@deep /]# userdel gopher
  [root@deep /]# userdel ftp
  ```

WARNING: By default, the `userdel` command will not delete a user's home directory. If you want the home directories of accounts to be deleted too, then add the **`-r`** option to the `userdel` command. Finally, the **`-r`** option must be used only when you have added a new user to the server and want to remove them. It doesn't need to be used for the removal of the above default users accounts. The user account called "`mail`" must be removed from the system only if you don't use `Sendmail` as your default Mail Server with `mailx` package. This user is related to `mailx` and not `Sendmail`.

Once the above list of users has been deleted from your Linux system, the `/etc/passwd` file will look like this:

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:
daemon:x:2:2:daemon:/sbin:
sync:x:5:0:sync:/sbin:/bin/sync
nobody:x:99:99:Nobody:/:
```

Step 2
After that we have removed all the unnecessary default vendor accounts into the `/etc/passwd` file from our system, we will remove all default vendor accounts into the `/etc/group` file.

- Type the following commands to delete all default usersgroups accounts listed below:
```
[root@deep /]# groupdel adm
[root@deep /]# groupdel lp
[root@deep /]# groupdel news
[root@deep /]# groupdel mail
[root@deep /]# groupdel uucp
[root@deep /]# groupdel games
[root@deep /]# groupdel dip
```

NOTE: The group account called "`mail`" must be removed from the system only if you don't use the `mailx` program for "`mail`". This is probably not what you want except if you use `qmail`.

Once the above list of group users has been deleted from your Linux system the `/etc/group` file will like this:

```
root:x:0:root
bin:x:1:root,bin,daemon
daemon:x:2:root,bin,daemon
sys:x:3:root,bin
tty:x:5:
disk:x:6:root
mem:x:8:
kmem:x:9:
wheel:x:10:root
man:x:15:
nobody:x:99:
users:x:100:
floppy:x:19:
slocate:x:21:
utmp:x:22:
```

Step 3
Finally it is time to add the necessary and allowed users into the system:

- To add a new user on your system, use the following command:
```
[root@deep /]# useradd username
```
For example:
```
[root@deep /]# useradd admin
```

- To add or change password for user on your system, use the following command:
```
[root@deep /]# passwd username
```
For example:
```
[root@deep /]# passwd admin
```
The output should look something like this:
```
Changing password for user admin
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

Step 4
The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file, which has been the source of attacks involving the deletion of `/etc/passwd`, `/etc/shadow`, `/etc/group` or `/etc/gshadow` files.

- To set the immutable bit on the passwords and groups files, use the following commands:
```
[root@deep /]# chattr +i /etc/passwd
[root@deep /]# chattr +i /etc/shadow
[root@deep /]# chattr +i /etc/group
[root@deep /]# chattr +i /etc/gshadow
```

WARNING: In the future, if you intend to add or delete users, passwords, usergroups, or group files, you must unset the immutable bit on all those files or you will not be able to make and update your changes. Also if you intend to install an RPM program that will automatically add a new user to the different immunized `passwd` and `group` files, then you will receive an error message during the install if you have not unset the immutable bit from those files.

- To unset the immutable bit on the passwords and groups files, use the commands:
```
[root@deep /]# chattr -i /etc/passwd
[root@deep /]# chattr -i /etc/shadow
[root@deep /]# chattr -i /etc/group
[root@deep /]# chattr -i /etc/gshadow
```

Control mounting a file system

You can have more control on mounting file systems like `/cache/`, `/home/` or `/tmp/` partitions with some nifty options like **`noexec`**, **`nodev`**, and **`nosuid`**. This can be setup in the `/etc/fstab` text file. The `fstab` file contains descriptive information about the various file system mount options; each line addresses one file system.

Information related to security options in the `fstab` text file are:

✓ `defaults`	Allow everything (quota, read-write, and suid) on this partition.
✓ `noquota`	Do not set users quotas on this partition.
✓ `nosuid`	Do not set SUID/SGID access on this partition.
✓ `nodev`	Do not set character or special devices access on this partition.
✓ `noexec`	Do not set execution of any binaries on this partition.
✓ `quota`	Allow users quotas on this partition.
✓ `ro`	Allow read-only on this partition.
✓ `rw`	Allow read-write on this partition.
✓ `suid`	Allow SUID/SGID access on this partition.

NOTE: For more information on options that you can set in this file (`fstab`), see the man pages about `mount (8)`.

Step 1

- Edit the **fstab** file (`vi /etc/fstab`) and change it depending on your needs.

For example change:

```
LABEL=/cache        /cache      ext2    defaults        1 2
LABEL=/home         /home       ext2    defaults        1 2
LABEL=/tmp          /tmp        ext2    defaults        1 2
```

To read:

```
LABEL=/cache        /cache      ext2    defaults,nodev      1 2
LABEL=/home         /home       ext2    defaults,nosuid     1 2
LABEL=/tmp          /tmp        ext2    defaults,nosuid,noexec      1 2
```

Meaning, **<nosuid>**, do not allow set-user-identifier or set-group-identifier bits to take effect, **<nodev>**, do not interpret character or block special devices on this file system partition, and **<noexec>**, do not allow execution of any binaries on the mounted file system.

Step 2

Once you have made the necessary adjustments to the `/etc/fstab` file, it is time to inform the Linux system about the modifications.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /cache -oremount
[root@deep /]# mount /home -oremount
[root@deep /]# mount /tmp -oremount
```

Each file system that has been modified must be remounted with the command show above. In our example we have modified the `/cache`, `/home`, and `/tmp` file system and it is for this reason that we remount these files systems with the above commands.

- You can verify if the modifications have been correctly applied to the Linux system with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root /   ext2    rw 0 0
/proc /proc   proc    rw 0 0
/dev/sda1     /boot       ext2    rw 0 0
/dev/sda10    /cache      ext2    rw,nodev 0 0
/dev/sda9     /chroot     ext2    rw 0 0
/dev/sda8     /home       ext2    rw,nosuid 0 0
/dev/sda13    /tmp        ext2    rw,noexec,nosuid 0 0
/dev/sda7     /usr        ext2    rw 0 0
/dev/sda11    /var        ext2    rw 0 0
/dev/sda12    /var/lib    ext2    rw 0 0
none  /dev/pts    devpts  rw 0 0
```

This command will show you all the files systems on your Linux server with parameters applied to them.

Mounting the /boot directory of Linux as read-only

The /boot directory is where the Linux kernel and some of its related files are kept. On many Linux variants this directory resides in its own partition and the default parameter is to mount it as read-write. We can change this parameter to make it read-only for better security.

Mounting the /boot partition as read-only eliminates possible problems that someone may try to change or modify vital files inside it. To mount the /boot file system of Linux as read-only, follow the simple steps below.

Step 1

- Edit the **fstab** file (vi /etc/fstab) and change the line:

```
LABEL=/boot       /boot       ext2    defaults        1 2
```

To read:

```
LABEL=/boot       /boot       ext2    defaults,ro     1 2
```

We add the "**ro**" option to this line to specify to mount this partition as read-only.

Step 2

Make the Linux system aware about the modification you have made to the /etc/fstab file.

- This can be accomplished with the following command:

```
[root@deep /]# mount /boot -oremount
```

- Then test your results with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root /   ext2    rw 0 0
/proc /proc   proc    rw 0 0
/dev/sda1     /boot       ext2    ro 0 0
/dev/sda10    /cache      ext2    rw,nodev 0 0
/dev/sda9     /chroot     ext2    rw 0 0
/dev/sda8     /home       ext2    rw,nosuid 0 0
/dev/sda13    /tmp        ext2    rw,noexec,nosuid 0 0
/dev/sda7     /usr        ext2    rw 0 0
/dev/sda11    /var        ext2    rw 0 0
/dev/sda12    /var/lib    ext2    rw 0 0
none /dev/pts    devpts  rw 0 0
```

If you see something like: /dev/sda1 /boot ext2 **ro** 0 0, congratulations!

WARNING: If in the future you want to upgrade your Linux kernel, it is important to reset the modification you have made to the /boot directory to its initial state (read-write) or you will not be able to install the new kernel because the /boot partition is set as read-only. All you have to do if you want to put the /boot partition to its original state is to edit the /etc/fstab file again and remove the "ro" option then remount the /boot file system with the "**mount -oremount**" command again.

Conceal binary RPM

Once you have installed all the software that you need on your Linux server with the RPM command, it's a good idea to move it to a safe place like a floppy disk or other safe place of your choice. With this method if someone accesses your server and has the intention to install nasty software with the RPM command, he wouldn't be able to. Of course, if in the future you want to install or upgrade new software via RPM, all you have to do is to replace the RPM binary to its original directory again.

- To move the RPM binary on the floppy disk, use the command:

```
[root@deep /]# mount /dev/fd0H1440 /mnt/floppy/
[root@deep /]# mv /bin/rpm /mnt/floppy/
[root@deep /]# umount /mnt/floppy/
```

WARNING: Never uninstall the RPM program completely from your system or you will be unable to reinstall it again later, since to install RPM or other software you need to have RPM commands available.

One more thing you can do is change the default permission of the "`rpm`" command from 755 to 700. With this modification, non-root users can't use the "`rpm`" program to query, install etc; in case you forget to move it to a safe place after installation of new programs.

- To change the default permission of `/bin/rpm`, use the command:

```
[root@deep /]# chmod 700 /bin/rpm
```

Shell logging

To make it easy for you to repeat long commands, the bash shell stores up to 500 old commands in the `~/.bash_history` file (where "**~/**" is your home directory). Each user that has an account on the system will have this file `.bash_history` in their home directory. Reducing the number of old commands the `.bash_history` files can hold may protect users on the server who enter by mistake their password on the screen in plain text and have their password stored for a long time in the `.bash_history` file.

Step 1

The `HISTSIZE` line in the `/etc/profile` file determine the size of old commands the `.bash_history` file for all users on your system can hold. For all accounts I would highly recommend setting the `HISTSIZE` in `/etc/profile` file to a low value such as **`10`**.

- Edit the **`profile`** file (`vi /etc/profile`) and change the line:

```
HISTSIZE=1000
```

To read:

```
HISTSIZE=10
```

Which means, the `.bash_history` file in each users home directory can store 10 old commands and no more. Now, if a cracker tries to see the `~/.bash_history` file of users on your server to find some password typed by mistake in plain text, he or she has less chance to find one.

Step 2
The administrator should also add into the `/etc/profile` file the "`HISTFILESIZE=0`" line, so that each time a user logs out, its `.bash_history` file will be deleted so crackers will not be able to use `.bash_history` file of users who are not presently logged into the system.

- Edit the **`profile`** file (`vi /etc/profile`) and add the following parameter below the "`HISTSIZE=`" line:

  `HISTFILESIZE=0`

After this parameter has been set on your system, you must logout and login again (as root) for the change to take effect.

Physical hard copies of all-important logs

One of the most important security considerations is the integrity of the different log files under the `/var/log/` directory on your server. If despite each of the security functions put in place on our server, a cracker can gain access to it, our last defense is the log file system, so it is very important to consider a method of being sure of the integrity of our log files.

If you have a printer installed on your server, or on a machine on your network, a good idea would be to have actual physical hard copies of all-important logs. This can be easily accomplished by using a continuous feed printer and having the syslog program sending all logs you seem important out to `/dev/lp0` (the printer device). Cracker can change the files, programs, etc on your server, but can do nothing when you have a printer that prints a real paper copy of all of your important logs.

As an example:
For logging of all `telnet`, `mail`, `boot` messages and `ssh` connections from your server to the printer attached to THIS server, you would want to add the following line to the `/etc/syslog.conf` file:

Step 1

- Edit the **`syslog.conf`** file (`vi /etc/syslog.conf`) and add at the end of this file the following line:

  `authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0`

Step 2

- Now restart your `syslog` daemon for the change to take effect:

```
[root@deep /]# /etc/rc.d/init.d/syslog restart
Shutting down kernel logger:          [OK]
Shutting down system logger:          [OK]
Starting system logger:               [OK]
Starting kernel logger:               [OK]
```

As an example:
For logging of all `telnet`, `mail`, `boot` messages and `ssh` connections from your server to the printer attached to a REMOTE server in your local network, then you would want to add the following line to `/etc/syslog.conf` file on the REMOTE server.

Step 1

- Edit the **syslog.conf** file (`vi /etc/syslog.conf`) on the REMOTE server (for example: `printer.openna.com`) and add at the end of this file the following line:

```
authpriv.*;mail.*;local7.*;auth.*;daemon.info /dev/lp0
```

If you don't have a printer in your network, you can also copy all the log files to another machine; simply omit the above first step of adding `/dev/lp0` to your `syslog.conf` file on remote and go directly to the "**-r**" option second step on remote. Using the feature of copying all the log files to another machine will give you the possibility to control all `syslog` messages on one host and will tear down administration needs.

Step 2

Since the default configuration of the `syslog` daemon is to not receive any messages from the network, we must enable on the REMOTE server the facility to receive messages from the network. To enable the facility to receive messages from the network on the REMOTE server, add the following option "**-r**" to your `syslog` daemon script file (only on the REMOTE host):

- Edit the `syslog` daemon (`vi +24 /etc/rc.d/init.d/syslog`) and change:

```
daemon syslogd -m 0
```

To read:

```
daemon syslogd -r -m 0
```

Step 3

- Restart your syslog daemon on the remote host for the change to take effect:

```
[root@mail /]# /etc/rc.d/init.d/syslog restart
Shutting down kernel logger:          [OK]
Shutting down system logger:          [OK]
Starting system logger:               [OK]
Starting kernel logger:               [OK]
```

Step 4

- If we have a firewall on the REMOTE server (you are supposed to have one), we must add or verify the existence of the following lines:

```
# SYSLOG server (514)
# -----------------

# Provides full remote logging. Using  this feature you're able to
# control all syslog messages on one host.

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp \
         -s $SYSLOG_CLIENT --source-port $UNPRIVPORTS \
         -d $IPADDR --destination-port 514 -j ACCEPT
```

Where `EXTERNAL_INTERFACE="eth0"` # Internet or Internal connected interface
Where `IPADDR="208.164.186.10"` # Your IP address
Where `SYSLOG_CLIENT="208.164.168.0/24"` # Your syslog clients IP ranges

Step 5

- Now restart your firewall on the remote host for the change to take effect:

```
[root@printer /]# /etc/rc.d/init.d/iptables restart
Shutting Firewalling Services:       [OK]
Starting Firewalling Services:       [OK]
```

This firewall rule will allow incoming UDP packets on port 514 (`syslog port`) on the remote server that comes from our internal client to be accepted. For more information on Firewalls see the chapter relating to network firewalls.

Step 6

- Edit the **`syslog.conf`** file (`vi /etc/syslog.conf`) on the LOCAL server, and add at the end of this file the following line:

```
authpriv.*;mail.*;local7.*;auth.*;daemon.info @printer
```

Where "`printer`" is the hostname of the REMOTE server. Now if anyone ever hacks your machine and attempts to erase vital system logs, you still have a hard copy of everything. It should then be fairly simple to trace where they came from and deal with it accordingly.

Step 7

- Restart your `syslog` daemon on the LOCAL server for the change to take effect:

```
[root@deep /]# /etc/rc.d/init.d/syslog restart
Shutting down kernel logger:         [OK]
Shutting down system logger:         [OK]
Starting system logger:              [OK]
Starting kernel logger:              [OK]
```

Step 8

- Same as on the REMOTE host, we must add or verify the existence of the following lines in our firewall script file on the LOCAL host:

```
# SYSLOG client (514)
# -----------------

iptables -A OUTPUT  -o $EXTERNAL_INTERFACE -p udp \
         -s $IPADDR --source-port 514 \
         -d $SYSLOG_SERVER --destination-port $UNPRIVPORTS -j ACCEPT
```

Where `EXTERNAL_INTERFACE="eth0"` # Internet or Internal connected interface
Where `IPADDR="208.164.186.1"` # Your IP address
Where `SYSLOG_SERVER="printer.openna.com"` # Your Printer Server in our example

Step 9

- Finally restart your firewall on the LOCAL host for the change to take effect:

```
[root@deep /]# /etc/rc.d/init.d/iptables restart
Shutting Firewalling Services:       [OK]
Starting Firewalling Services:       [OK]
```

This firewall rule will allow outgoing UDP packets on unprivileged ports on the local server destined to the remote `syslog` server to be accepted. Repeat step 6 through steps 9 for each additional server you may have and want all-important logs to be logged on remote printer server. For more information on Firewalls see the chapter relating to network firewalls.

WARNING: Never use your Gateway Server as a host to control all `syslog` messages; this is a very bad idea. More options and strategies exist with the `sysklogd` program, see the man pages about `sysklogd (8), syslog(2),` and `syslog.conf(5)` for more information.

Tighten scripts under `/etc/rc.d/init.d/`

Fix the permissions of the script files that are responsible for starting and stopping all your normal processes that need to run at boot time.

- To fix the permissions of those files, use the following command:
  `[root@deep /]#` **`chmod -R 700 /etc/init.d/*`**

Which means just the super-user “root” is allowed to Read, Write, and Execute scripts files on this directory. I don’t think regular users need to know what’s inside those script files.

WARNING: If you install a new program or update a program that use the init system V script located under `/etc/rc.d/init.d/` directory, don’t forget to change or verify the permission of this script file again.

The `/etc/rc.local` file

By default, when you login to a Linux machine, it tells you the Linux distribution name, version, kernel version, and the name of the server. This is giving away too much info. We’d rather just prompt users with a "Login:" prompt.

Step 1

- To do this, edit the **`rc.local`** file (vi `/etc/rc.local)` and place "**#**" in front of the following lines as shown:

```
--
# This will overwrite /etc/issue at every boot.  So, make any changes you
# want to make to /etc/issue here or you will lose them when you reboot.
#echo "" > /etc/issue
#echo "$R" >> /etc/issue
#echo "Kernel $(uname -r) on $a $(uname -m)" >> /etc/issue
#
#cp -f /etc/issue /etc/issue.net
#echo >> /etc/issue
--
```

Step 2

- Then, remove the following files: `issue.net` and `issue` under `/etc/` directory:
  `[root@deep /]#` **`rm -f /etc/issue`**
  `[root@deep /]#` **`rm -f /etc/issue.net`**

WARNING: The `/etc/issue.net` file is the login banner that users will see when they make a networked (i.e. `telnet`, `SSH`) connection to your machine. You will find it in the `/etc` directory, along with a similar file called `issue`, which is the login banner that gets displayed to local users.

It is simply a text file and can be customized to your own tastes, but be aware that as noted above, if you do change it or remove it like we do, you'll also need to modify the `/etc/rc.d/rc.local` shell script, which re-creates both the `issue` and `issue.net` files every time the system boots.

Bits from root-owned programs

A regular user will be able to run a program as root if it is set to `SUID` root. All programs and files on your computer with the 's' bits appearing on its mode, have the `SUID` (`-rw`**`s`**`r-xr-x`) or `SGID` (`-r-xr-`**`s`**`r-x`) bit enabled. Because these programs grant special privileges to the user who is executing them, it is important to remove the '`s`' bits from root-owned programs that won't absolutely require such privilege. This can be accomplished by executing the command **`chmod a-s`** with the name(s) of the `SUID/SGID` files as its arguments.

Such programs include, but aren't limited to:

- ✓ Programs you never use.
- ✓ Programs that you don't want any non-root users to run.
- ✓ Programs you use occasionally, and don't mind having to `su` (1) to root to run.

We've placed an asterisk (**`*`**) next to each program we personally might disable and consider to be not absolutely required for the duty work of the server. Remember that your system needs some suid root programs to work properly, so be careful.

Step 1

- To find all files with the 's' bits from root-owned programs, use the command:

```
[root@deep]# find / -type f \( -perm -04000 -o -perm -02000 \) -exec ls -
l {} \;

*-rwsr-xr-x   1 root     root        34220    Jul 18 14:13  /usr/bin/chage
*-rwsr-xr-x   1 root     root        36344    Jul 18 14:13  /usr/bin/gpasswd
-rwxr-sr-x    1 root     man         35196    Jul 12 03:50  /usr/bin/man
-r-s--x--x    1 root     root        13536    Jul 12 07:56  /usr/bin/passwd
-rwxr-sr-x    1 root     mail        10932    Jul 12 10:03  /usr/bin/suidperl
-rwsr-sr-x    1 root     mail        63772    Jul 12 10:03  /usr/bin/sperl5.6.0
-rwxr-sr-x    1 root     slocate     23964    Jul 23 17:48  /usr/bin/slocate
*-r-xr-sr-x   1 root     tty         6524     Jul 12 03:19  /usr/bin/wall
*-rws--x-x    1 root     root        13184    Jul 21 19:15  /usr/bin/chfn
*-rws--x-x    1 root     root        12640    Jul 21 19:15  /usr/bin/chsh
*-rws--x-x    1 root     root        5464     Jul 21 19:15  /usr/bin/newgrp
*-rwxr-sr-x   1 root     tty         8500     Jul 21 19:15  /usr/bin/write
*-rwsr-xr-x   1 root     root        6288     Jul 26 10:22  /usr/sbin/usernetctl
-rwxr-sr-x    1 root     utmp        6584     Jul 13 00:46  /usr/sbin/utempter
*-rwsr-xr-x   1 root     root        20540    Jul 25 07:33  /bin/ping
-rwsr-xr-x    1 root     root        14184    Jul 12 20:47  /bin/su
*-rwsr-xr-x   1 root     root        55356    Jul 12 05:01  /bin/mount
*-rwsr-xr-x   1 root     root        25404    Jul 12 05:01  /bin/umount
*-rwxr-sr-x   1 root     root        4116     Jul 26 10:22  /sbin/netreport
-r-sr-xr-x    1 root     root        14732    Jul 26 14:06  /sbin/pwdb_chkpwd
-r-sr-xr-x    1 root     root        15340    Jul 26 14:06  /sbin/unix_chkpwd
```

Step 2

- To disable the suid bits on selected programs above, type the following commands:

```
[root@deep /]# chmod a-s /usr/bin/chage
[root@deep /]# chmod a-s /usr/bin/gpasswd
[root@deep /]# chmod a-s /usr/bin/wall
[root@deep /]# chmod a-s /usr/bin/chfn
[root@deep /]# chmod a-s /usr/bin/chsh
[root@deep /]# chmod a-s /usr/bin/newgrp
[root@deep /]# chmod a-s /usr/bin/write
[root@deep /]# chmod a-s /usr/sbin/usernetctl
[root@deep /]# chmod a-s /bin/ping
[root@deep /]# chmod a-s /bin/mount
[root@deep /]# chmod a-s /bin/umount
[root@deep /]# chmod a-s /sbin/netreport
```

If you want to know what those programs do, type “man program-name” and read the man page.

As an example:

- To read the `netreport` man page, use the following command:

```
[root@deep /]# man netreport
```

Finding all files with the `SUID/SGID` bit enabled

All `SUID` and `SGID` files that still exist on your system after we have removed those that won't absolutely require such privilege are a potential security risk, and should be monitored closely. Because these programs grant special privileges to the user who is executing them, it is necessary to ensure that insecure programs are not installed.

A favorite trick of crackers is to exploit `SUID` "root" programs, and leave a `SUID` program as a backdoor to get in the next time. Find all `SUID` and `SGID` programs on your system, and keep track of what they are so that you are aware of any changes, which could indicate a potential intruder.

- Use the following command to find all `SUID/SGID` programs on your system:

```
[root@deep /]# find / -type f \( -perm -04000 -o -perm -02000 \) -exec ls
-l {} \;
```

When you have, for example, the home directories of the users accounts mountable on all servers, then this find command will check the same home directory on every server (SUIDs on mounted file systems are not effective). If there are more mounted file systems on the servers, then this can take some time which actually a waste of time.

- In this case, you can avoid this by executing the following command (see '`-fstype`'):

```
[root@deep /]# find / \( ! -fstype nfs -o -prune \) -type f \( -perm -
04000 -o -perm -02000 \) -exec ls -l {} \;
```

NOTE: See later in this book the chapter related to “Securities Software - Monitoring Tools” for more information about the software named “**`sXid`**” that will do the job for you automatically each day and report the results via mail.

Don't let internal machines tell the server what their MAC address is

To avoid the risk that a user could easily change a computers IP address and appear as someone else to the firewall, you can force the ARP cache entries of Linux using the `arp` command utility. A special option can be used with the `arp` utility to avoid letting INTERNAL machines tell the server what their MAC (Media Access Control) address is and the IP address associated with it. ARP is a small utility, which manipulates the kernel's ARP (Address Resolution Protocol) cache. Through all possible options associated with this utility, the primary one is clearing an address mapping entry and manually setting up one. In the hope to more secure our server from the INTERNAL, we will manually set MAC address (sometimes called Hardware addresses) of all know computers in our network statically by using static ARP entries.

Step1

- For each IP address of INTERNAL computers in your network, use the following command to know the MAC address associate with the IP address:

```
[root@deep /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:50:DA:C6:D3:FF
          inet addr:207.35.78.3  Bcast:207.35.78.32  Mask:255.255.255.224
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1887318 errors:0 dropped:0 overruns:1 frame:0
          TX packets:2709329 errors:0 dropped:0 overruns:0 carrier:1
          collisions:18685 txqueuelen:100
          Interrupt:10 Base address:0xb000

eth1      Link encap:Ethernet  HWaddr 00:50:DA:C6:D3:09
          inet addr:192.168.1.11  Bcast:192.168.1.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:182937 errors:0 dropped:0 overruns:0 frame:0
          TX packets:179612 errors:0 dropped:0 overruns:0 carrier:0
          collisions:7434 txqueuelen:100
          Interrupt:11 Base address:0xa800

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          UP LOOPBACK RUNNING  MTU:3924  Metric:1
          RX packets:7465 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7465 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
```

The MAC (Media Access Control) address will be the letters and numbers that come after "**HWaddr**" (the Hardware Address). In the above example our MAC address are: **00:50:DA:C6:D3:FF** for the interface `eth0` and **00:50:DA:C6:D3:09** for the interface `eth1`.

Step 2
Once we know the MAC (Media Access Control) address associated with IP address, we can add them manually to the ARP entries of the Linux server.

- To add manually MAC address to ARP entries, use the following command:

```
[root@deep /]# arp -s 207.35.78.3 00:50:DA:C6:D3:FF
[root@deep /]# arp -s 192.168.1.11 00:50:DA:C6:D3:09
```

The "**-s**" option means to manually create an ARP address mapping entry for host `hostname` with hardware address set to `hw_addr` class. You can add you ARP commands to the `/etc/rc.d/rc.local` file if you want to keep your configuration if the system reboot.

Step 3

- To verify if the modifications have been added to the system, use the following command:

```
[root@deep /]# arp
Address       Hwtype  Hwaddress           Flags Mask  Iface
207.35.78.3   ether   00:20:78:13:86:92   CM          eth1
192.168.1.11  ether   00:E0:18:90:1B:56   CM          eth1
```

WARNING: If you receive error message like: **SIOCSARP: Invalid argument**, it is because the MAC (Media Access Control) address you want to add is the one of your server. You must add only MAC address of INTERNAL computers in your private network. This hack doesn't apply to external node on the Internet.

You can now be reassured that someone will not change the system's IP address of an INTERNAL system and get through. If they do change the IP address, the server simply won't talk to them. With the new `iptables` tool of Linux, which replace the old `ipchains` utility for packet filter administration and firewall setup, MAC addresses can be filtered and configured in the firewall rules too.

Unusual or hidden files

It is important to look everywhere on the system for unusual or hidden files (files that start with a period and are normally not shown by the "**`ls`**" command), as these can be used to hide tools and information (password cracking programs, password files from other systems, etc.). A common technique on UNIX systems is to put a hidden directory or file in a user's account with an unusual name, something like '`...`' or '`..  `' (dot dot space) or '`..^G`' (dot dot control-G). The `find` program can be used to look for hidden files.

- To look for hidden files, use the following commands:

```
[root@deep /]# find / -name ".. " -print -xdev
[root@deep /]# find / -name ".*" -print -xdev | cat -v
```

WARNING: Files with names such as '`.xx`' and '`.mail`' have been used (that is, files that might appear to be normal).

Finding Group and World Writable files and directories

Group and world writable files and directories, particularly system files (partions), can be a security hole if a cracker gains access to your system and modifies them. Additionally, world-writable directories are dangerous, since they allow a cracker to add or delete files as he or she wishes in these directories. In the normal course of operation, several files will be writable, including some from the `/dev/`, `/var/catman/` directories, and all symbolic links on your system.

- To locate all group & world-writable files on your system, use the command:

```
[root@deep /]# find / -type f \( -perm -2 -o -perm -20 \) -exec ls -lg {} \;
```

- To locate all group & world-writable directories on your system, use the command:

```
[root@deep /]# find / -type d \( -perm -2 -o -perm -20 \) -exec ls -ldg {} \;
```

WARNING: A file and directory integrity checker like "**Tripwire**" software can be used regularly to scan, manage and find modified group or world writable files and directories easily. See later in this book the chapter related to "Securities Software - System Integrity" for more information about `Tripwire`.

Unowned files

Don't permit any unowned file. Unowned files may also be an indication that an intruder has accessed your system. If you find unowned file or directory on your system, verify its integrity, and if all looks fine, give it an owner name. Some time you may uninstall a program and get an unowned file or directory related to this software; in this case you can remove the file or directory safely.

- To locate files on your system that do not have an owner, use the following command:
  `[root@deep /]#` **`find / -nouser -o -nogroup`**

WARNING: It is important to note that files reported under `/dev/` directory don't count.

Finding `.rhosts` files

Finding all existing `.rhosts` files that could exist on your server should be a part of your regular system administration duties, as these files should not be permitted on your system. Remember that a cracker only needs one insecure account to potentially gain access to your entire network.

Step 1

- You can locate all existing `.rhosts` files on your system with the following command:
  `[root@deep /]#` **`find /home -name .rhosts`**

If the result returns nothing, then you are safe and your system contain no `.rhosts` files in the `/home/` directory at this time. If you are doing a new install of Linux (like we did), you should not have any `.rhosts` files on your system.

Step 2

You can also use a `cron` job to periodically check for, report the contents of, and delete `$HOME/.rhosts` files. Also, users should be made aware that you regularly perform this type of audit, as directed by your security policy.

- To use a `cron` job to periodically check and report via mail all `.rhosts` files, create as "root" the **`find_rhosts_files`** script file under `/etc/cron.daily/` directory (`touch /etc/cron.daily/find_rhosts_files`) and add the following lines in this script:

```
#!/bin/sh
/usr/bin/find /home -name .rhosts | (cat <<EOF
This is an automated report of possible existent ".rhosts" files on the server
deep.openna.com, generated by the find utility command.

New detected ".rhosts" files under the "/home/" directory include:
EOF
cat
) | /bin/mail -s "Content of .rhosts file audit report" root
```

- Now make this script executable, verify the owner, and change the group to “root”.
```
[root@deep /]# chmod 755 /etc/cron.daily/find_rhosts_files
[root@deep /]# chown 0.0 /etc/cron.daily/find_rhosts_files
```

Each day mail will be sent to “root” with a subject:” Content of .rhosts file audit report” containing potential new `.rhosts` files.

System is compromised!

If you believe that your system has been compromised, contact CERT ® Coordination Center or your representative in FIRST (Forum of Incident Response and Security Teams).

Internet Email: cert@cert.org
CERT Hotline: (+1) 412-268-7090
Facsimile: (+1) 412-268-6989
CERT/CC personnel answer 8:00 a.m. – 8:00 p.m. EST (GMT –5)/EDT (GMT –4)) on working days; they are on call for emergencies during other hours and on weekends and holidays.

4 Security and Optimization - Pluggable Authentication Modules

In this Chapter

The password length
Disabling console program access
Disabling all console access
The Login access control table
Tighten console permissions for privileged users
Putting limits on resource
Controlling access time to services
Blocking; `su` to root, by one and sundry

Linux Pluggable Authentication Modules

Abstract
The **P**luggable **A**uthentication **M**odules (PAM) consists of shared libraries, which enable administrators to choose how applications authenticate users.

Basically, PAM enables the separation of authentication schemes from the applications. This is accomplished by providing a library of functions that applications can use for requesting user authentications. ssh, pop, imap, etc. are PAM-aware applications, hence these applications can be changed from providing a password to providing a voice sample or fingerprint by simply changing the PAM modules without having to rewrite any code in these applications.

The configuration files of the PAM modules are located in the directory /etc/pam.d and the modules (shared libraries) themselves are located in the directory /lib/security. The /etc/pam.d directory has a collection of named files of its own, e.g. ssh, pop, imap, etc. PAM-aware applications that do not have a configuration file will automatically be pointed to the default configuration file 'other'.

In the next section we will set up some recommended minimum-security restrictions using PAM.

The password length
The minimum acceptable password length by default when you install your Linux system is 5. This means that when a new user is given access to the server, his/her password length will be at minimum 5 mixes of character strings, letter, number, special character etc. This is not enough and must be 8 or more. The password length under Linux by the use of its PAM feature is controlled by 5 arguments minlen, dcredit, ucredit, lcredit, and ocredit.

Step 1
To prevent non-security-minded people or administrators from being able to enter just 5 characters for the valuable password, edit the rather important /etc/pam.d/passwd file and enforce the minimum password length.

- Edit the **passwd** file (vi /etc/pam.d/passwd) and remove the following line:

```
password    required      /lib/security/pam_stack.so service=system-auth
```

Step 2
Once the above line has been removed from the passwd file, we must remove the following three lines as shown below from the system-auth file. This is a bug in the PAM RPM package of Red Hat that we must correct here to be able to use this feature with Linux.

- Edit the **system-auth** file (vi /etc/pam.d/system-auth) and remove the lines:

```
password    required      /lib/security/pam_cracklib.so retry=3
password    sufficient    /lib/security/pam_unix.so nullok use_authtok md5 shadow
password    required      /lib/security/pam_deny.so
```

Step 3
Now add the following lines to `/etc/pam.d/passwd`. We use the `PAM` "`pam_cracklib`" module here with the argument "`minlen`" to enforce the password length.

```
password    required     /lib/security/pam_cracklib.so retry=3 minlen=12
password    sufficient   /lib/security/pam_unix.so nullok use_authtok md5
shadow
password    required     /lib/security/pam_deny.so
```

After adding the above lines, the `/etc/pam.d/passwd` file should look like this:

```
#%PAM-1.0
auth        required     /lib/security/pam_stack.so service=system-auth
account     required     /lib/security/pam_stack.so service=system-auth
password    required     /lib/security/pam_cracklib.so retry=3 minlen=12
password    sufficient   /lib/security/pam_unix.so nullok use_authtok md5
shadow
password    required     /lib/security/pam_deny.so
```

And the `/etc/pam.d/system-auth` file should look like this:

```
#%PAM-1.0
# This file is auto-generated.
# User changes will be destroyed the next time authconfig is run.
auth        required      /lib/security/pam_env.so
auth        sufficient    /lib/security/pam_unix.so likeauth nullok
auth        required      /lib/security/pam_deny.so
account     required      /lib/security/pam_unix.so
session     required      /lib/security/pam_limits.so
session     required      /lib/security/pam_unix.so
```

WARNING: It is important to note that when you set the password for a user under 'root', then these restrictions don't apply!! This is the case on all Unix OS. The user 'root' can override pretty much everything. Instead, log as the user account from which you apply this restriction and try to change the password. You will see that it works.

You need to keep in mind that this module includes a credit mechanism. E.g. if you define `minlen=12`, then you will get 1 credit for e.g. including a single digit number in your password, or for including a non-alphanumeric character. Getting 1 credit means that the module will accept a password of the length of minlen-credit. When you check the parameters of the cracklib module, you will see that it has some parameters that let you define what a credit is (http://www.us.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html).

For example:

minlen	The following password was accepted
14	gjtodgsdf1$

You can see that I got 1 credit for a alphanumeric character and a credit for each non-alphanumeric character. "gjtodgsdf1$" has a length of 11, 1 credit for alpha-numeric, 2 credits for non-alphanumeric character (1 and $) which gives me a credit of 3, hence the password length of 11 was accepted.

At any rate, the minimum length is adjusted by the mixture of types of characters used in the password. Using digits (up to the number specified with the "dcredit=" parameter, which defaults to 1) or uppercase letters "ucredit" or lowercase letters "lcredit" or other types of letters "ocredit" will decrease the minimum length by up to four since the default parameter for these arguments is 1 and there is four different arguments that you can add.

A password with 9 lowercase letters in it will pass a minimum length set to 10 unless "lcredit=0" is used, because a credit is granted for the use of a lowercase letter. If the mixture includes an uppercase letter, a lowercase letter, and a digit, then a minlength of 8 effectively becomes 5.

NOTE: With the new MD5 passwords capability, which is installed by default in all modern Linux operating system, a long password can be used now (up to 256 characters), instead of the Unix standard eight letters or less. If you want to change the password length of 8 characters to example 16 characters, all you have to do is to replace the number 12 by 20 in the "minlen=12" line of the /etc/pam.d/passwd file.

Disabling console program access

In a safe environment, where we are sure that console is secured because passwords for BIOS and LILO are set and all physical power and reset switches on the system are disabled, it may be advantageous to entirely disable all console-equivalent access to programs like shutdown, reboot, and halt for regular users on your server.

- To do this, run the following command:
  ```
  [root@deep /]# rm -f /etc/security/console.apps/<servicename>
  ```

Where <servicename> is the name of the program to which you wish to disable console-equivalent access. Unless you use xdm, however, be careful to not remove the xserver file or no one but only 'root' will be able to start the X server. (If you always use xdm to start the X server, 'root' is the only user that needs to start X, in which case you might actually want to remove the xserver file).

- To disable console program access, use the following commands:
  ```
  [root@deep /]# rm -f /etc/security/console.apps/halt
  [root@deep /]# rm -f /etc/security/console.apps/poweroff
  [root@deep /]# rm -f /etc/security/console.apps/reboot
  [root@deep /]# rm -f /etc/security/console.apps/shutdown
  [root@deep /]# rm -f /etc/security/console.apps/xserver (if removed, root
  will be the only user able to start X).
  ```

This will disable console-equivalent access to programs halt, poweroff, reboot, and shutdown. Once again, the program xserver applies only if you installed the Xwindow interface on your system.

WARNING: If you are following our setup installation, the Xwindow interface is not installed on your server and all the files described above will not appear in the `/etc/security/console.apps` directory, so don't pay attention to the above step.

Disabling all console access

The Linux-`PAM` library installed by default on your system allows the system administrator to choose how applications authenticate users, such as for console access, program and file access. In order to disable all these accesses for the users, you must comment out all lines that refer to **`pam_console.so`** in the `/etc/pam.d` directory. This step is a continuation of the hack "Disabling console program access". The following script will do the trick automatically for you.

Step 1

- As 'root' creates the **`disabling.sh`** script file (`touch disabling.sh`) and add the following lines inside:

```
# !/bin/sh
cd /etc/pam.d
for i in * ; do
sed '/[^#].*pam_console.so/s/^/#/' < $i > foo && mv foo $i
done
```

Step 2

- Make this script executable with the following command and execute it:

```
[root@deep /]# chmod 700 disabling.sh
[root@deep /]# ./disabling.sh
```

This will comment out all lines that refer to **`pam_console.so`** for all files located under `/etc/pam.d` directory. Once the script has been executed, you can remove it from your system.

The Login access control table

On a server environment where authorized and legitimate logins can come from everywhere, it is important to have the possibility to use a security file which allow us to have more control over users who can connect to the server. What we are looking here is to have more control on not allowing some legitimated accounts to login from anywhere. Fortunately, this file exists and is called "`access.conf`", you can find it under your `/etc/security` directory.

The `access.conf` file which comes already installed with your native Linux system allow us to control which authorized users can/cannot log in to the server or to the console and from where. Don't forget that users access can come everywhere from remote host or directly from the console of the system. Configuration of the `access.conf` file of Linux is not complicated to understand. Below I show you how to configure it to be very restrictive and secure.

Step 1

By default denying access to every one, is the first step of a reliable security policy. In this way we eliminate the possibility of forgetting someone or to making a mistake.

- Edit the **`access.conf`** file (`vi /etc/security/access.conf`) and add the following line at the end of the file.

```
-:ALL EXCEPT root gmourani:ALL
```

This access policy means to disallow console logins as well as remote accounts login to all from anywhere except for user '`root`' and '`gmourani`'. With this choice of policy, we deny non-networked and remote logins to every user with a shell account on the system from everywhere and allow only the selected users.

Take a note that many possibilities exist as for example allowing the same users '`root`' and '`gmourani`' to log only to the system from remote host with IP address `207.35.78.2`. To enable this policy, all we need to do is to change the above policy to this one:

- Edit the **`access.conf`** file (`vi /etc/security/access.conf`) and add the following lines at the end of the file.

```
-:ALL EXCEPT root gmourani:207.35.78.2
-:ALL:LOCAL
```

Here the second policy line means to disallow all local access to the console for every users even for the super-user '`root`', therefore if you want to log as '`root`' you need first to log as user '`gmourani`' from remote host with IP address `207.35.78.2` and `su` to '`root`' (this is why I added '`root`' to the users allowed to connect from remote host `207.35.78.2`).

Step 2
To be able to use the `access.conf` feature of Linux, make sure to add the following line to `/etc/pam.d/login` and `sshd` if you use this service or it will not work.

- Edit the **`login`** file (`vi /etc/pam.d/login`) and add the following line.

```
account    required     /lib/security/pam_access.so
```

After adding the above line, the `/etc/pam.d/login` file should look like this:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_access.so
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
```

NOTE: Please read information about possible configurations of this file inside the `access.conf` file since your policies will certainly differ from the example that I show you above.

Tighten console permissions for privileged users

The `console.perms` security file of Linux, which use the `pam_console.so` module to operate, is designed to give to privileged users at the physical console (virtual terminals and local xdm-managed X sessions) capabilities that they would not otherwise have, and to take those capabilities away when they are no longer logged in at the console.

It provides two main kinds of capabilities: file permissions and authentication. When a user logs in at the console and **no other user is currently logged in at the console**, the `pam_console.so` module will change permissions and ownership of files as described in the file `/etc/security/console.perms`.

Please note that privileged users are nothing in common with regular users you may add to the server, they are special users like `floppy`, `cdrom`, `scanner`, etc which in an networking server environment are also considered and treated as users.

Step 1
The default `console.perms` configuration file of Linux is secure enough for regular use of the system where an Xwindow interface is considered to be installed but in a highly secure environment where the **G**raphical **U**ser **I**nterface (`GUI`) is not installed or where some special devices like `sound`, `jaz`, etc have no reason to exist, we can tighten the `console.perms` security file of Linux to be more secure by eliminating non-existent or unneeded privileged users to have capabilities that they would not otherwise have.

- Edit the **`console.perms`** file (`vi /etc/security/console.perms`), and change the default lines inside this file:

```
# file classes -- these are regular expressions
<console>=tty[0-9][0-9]* :[0-9]\.[0-9] :[0-9]
<xconsole>=:[0-9]\.[0-9] :[0-9]

# device classes -- these are shell-style globs
<floppy>=/dev/fd[0-1]*
<sound>=/dev/dsp* /dev/audio* /dev/midi* \
        /dev/mixer* /dev/sequencer
<cdrom>=/dev/cdrom* /dev/cdwriter*
<pilot>=/dev/pilot
<jaz>=/dev/jaz
<zip>=/dev/zip
<scanner>=/dev/scanner
<fb>=/dev/fb /dev/fb[0-9]*
<kbd>=/dev/kbd
<joystick>=/dev/js*
<v4l>=/dev/video* /dev/radio* /dev/winradio* /dev/vtx* /dev/vbi*
<gpm>=/dev/gpmctl
<dri>=/dev/dri/* /dev/nvidia*

# permission definitions
<console>  0660 <floppy>    0660 root.floppy
<console>  0600 <sound>     0640 root.sys
<console>  0600 <cdrom>     0600 root.disk
<console>  0600 <pilot>     0660 root.tty
<console>  0600 <jaz>       0660 root.disk
<console>  0600 <zip>       0660 root.disk
<console>  0600 <scanner>   0600 root
<console>  0600 <fb>        0600 root
<console>  0600 <kbd>       0600 root
<console>  0600 <joystick>  0600 root
<console>  0600 <v4l>       0600 root
<console>  0700 <gpm>       0700 root
```

```
<xconsole> 0600 /dev/console 0600 root.root
<xconsole> 0600 <dri>       0600 root
```

To read :

```
# file classes -- these are regular expressions
<console>=tty[0-9][0-9]* :[0-9]\.[0-9] :[0-9]

# device classes -- these are shell-style globs
<floppy>=/dev/fd[0-1]*
<cdrom>=/dev/cdrom* /dev/cdwriter*
<pilot>=/dev/pilot
<fb>=/dev/fb /dev/fb[0-9]*
<kbd>=/dev/kbd
<gpm>=/dev/gpmctl
<dri>=/dev/dri/* /dev/nvidia*

# permission definitions
<console>  0660 <floppy>    0660 root.floppy
<console>  0600 <cdrom>     0600 root.disk
<console>  0600 <pilot>     0660 root.tty
<console>  0600 <fb>        0600 root
<console>  0600 <kbd>       0600 root
<console>  0700 <gpm>       0700 root
```

Here we removed every privileged user related to the **G**raphical **U**ser **I**nterface and others related to `sound`, `zip` drive, `jaz` drive, `scanner`, `joystick` and `video` media at the physical console on the server.

Putting limits on resource

The `limits.conf` file located under the `/etc/security` directory can be used to control and limit resources for the users on your system. It is important to set resource limits on all your users so they can't perform denial of service attacks (number of processes, amount of memory, etc) on the server. These limits will have to be set up for the user when he or she logs in.

For example, limits for all users on your system might look like this.

Step 1

- Edit the **`limits.conf`** file (`vi /etc/security/limits.conf`) and add or change the lines to read:

```
*     hard    core    0
*     hard    rss     5000
*     hard    nproc   35
```

This says to prohibit the creation of core files "**`core 0`**", restrict the number of processes to `20` "**`nproc 20`**", and restrict memory usage to `5M` "**`rss 5000`**" for everyone except the super user "root". All of the above only concerns users who have entered through the login prompt on your system. With this kind of quota, you have more control on the processes, core files, and memory usage that users may have on your system. The asterisk "**`*`**" mean: all users that logs in on the server.

Putting an asterisk "**`*`**" to cover all users can pose problem with daemon users account like "`www`" for a Web Server, "`mysql`" for a `SQL` Database Server, etc. If we put an asterisk, then, these users will be affected by the restriction and limitation of processes or memory usage.

To solve the problem, we can choose an existing group name in our system and add every regular user to this group. In this manner, the restrictions and limitations will apply to all users who are members of this group name only. A special group account named "users" can be used for this purpose.

- Edit the **limits.conf** file (vi /etc/security/limits.conf) and add or change the lines to read:

```
@users    hard    core    0
@users    hard    rss     5000
@users    hard    nproc   35
```

If you decide to use a group name like "**@users**" to control and limit resources for the users on your system, then it is important to not forget to change the GUI (**G**roup **U**ser **I**D) of these users to be "100". "100" is the numeric value of the user's ID "users".

- The command to create a new user with group name which is set by default to users is:
  `[root@deep /]# useradd -g100 admin`

The "-g100" option represents the number of the user's initial login group and in our case "100" is the group account name "users". The "admin" parameter is the user name we want to add to the group name "users".

WARNING: Use the same command above for all users on your system you want to be member of the "users" group account. It is also preferable to set this parameter first before adding users to the system.

Step 2

- You must also edit the /etc/pam.d/login file and add the following line to the bottom of the file:

```
session       required    /lib/security/pam_limits.so
```

After adding the line above, the /etc/pam.d/login file should look like this:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_stack.so services=system-auth
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_stack.so services=system-auth
account    required     /lib/security/pam_access.so
password   required     /lib/security/pam_stack.so services=system-auth
session    required     /lib/security/pam_stack.so services=system-auth
session    required     /lib/security/pam_limits.so
```

Controlling access time to services

As the Linux-PAM system said, running a well-regulated system occasionally involves restricting access to certain services in a selective manner. The time.conf security file, which is provided by the pam_time.so module of Linux, offers some time control for access to services offered by a system. Its actions are determined through the configuration file called time.conf and located under /etc/security directory.

Step 1
The time.conf file can be configured to deny access to (individual) users based on their name, the time of day, the day of week, the service they are applying for and their terminal from which they are making their request.

- Edit the **time.conf** file (vi /etc/security/time.conf), and add the following line:

  login ; tty* & !ttyp* ; !root !gmourani ; !Al0000-2400

The above time control access line means to deny all user access to console-login at all times except for the super-user 'root' and the user 'gmourani'.

Take a note that many combinations exist as described in the time.conf file, we can, for example, allow user 'admin' to access the console-login any time except at the weekend and on Tuesday from 8AM to 6PM with the following statement.

- Edit the time.conf file (vi /etc/security/time.conf), and add the following line:

  login ; * ; !admin ; !Wd0000-2400 !Tu0800-1800

Step 2
To be able to use the time.conf feature of Linux, make sure to add the following line to /etc/pam.d/login and sshd if you use this service or nothing will work.

- Edit the **login** file (vi /etc/pam.d/login) and add the following line.

  ```
  account    required     /lib/security/pam_time.so
  ```

After adding the line above, the /etc/pam.d/login file should look like this:

```
#%PAM-1.0
auth       required     /lib/security/pam_securetty.so
auth       required     /lib/security/pam_stack.so services=system-auth
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_stack.so services=system-auth
account    required     /lib/security/pam_access.so
account    required     /lib/security/pam_time.so
password   required     /lib/security/pam_stack.so services=system-auth
session    required     /lib/security/pam_stack.so services=system-auth
session    required     /lib/security/pam_limits.so
```

NOTE: Please read information about possible configurations of this file inside the time.conf file since your policies will certainly differ from the examples that I show you above.

Blocking; `su` to root, by one and sundry

The `su` (**S**ubstitute **U**ser) command allows you to become other (existing) users on the system. For example you can temporarily become 'root' and execute commands as the super-user 'root'. If you don't want anyone to `su` to root or want to restrict the `su` command to certain users then uncomment the following line of your `su` configuration file in the `/etc/pam.d` directory. We highly recommend that you limit the persons allowed to `su` to the root account.

Step 1

- Edit the **`su`** file (`vi /etc/pam.d/su`) and uncomment the following line in the file:

```
auth      required        /lib/security/pam_wheel.so use_uid
```

After this line has been uncommented, the `/etc/pam.d/su` file should look like this:

```
#%PAM-1.0
auth        sufficient      /lib/security/pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
#auth       sufficient      /lib/security/pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
auth        required        /lib/security/pam_wheel.so use_uid
auth        required        /lib/security/pam_stack.so service=system-auth
account     required        /lib/security/pam_stack.so service=system-auth
password    required        /lib/security/pam_stack.so service=system-auth
session     required        /lib/security/pam_stack.so service=system-auth
session     optional        /lib/security/pam_xauth.so
```

Which means only those who are members of the "**`wheel`**" group can `su` to root; it also includes logging. Note that the "`wheel`" group is a special account on your system that can be used for this purpose. You cannot use any group name you want to make this hack. This hack combined with specifying which **`TTY`** and **`VC`** devices root is allowed to login on will improve your security a lot on the system.

Step 2

Now that we have defined the "**`wheel`**" group in our `/etc/pam.d/su` file configuration, it is time to add some users who will be allowed to `su` to "root" account.

- If you want to make, for example, the user "`admin`" a member of the "`wheel`" group, and thus be able to `su` to root, use the following command:

```
[root@deep /]# usermod -G10 admin
```

Which means "`G`" is a list of supplementary groups, where the user is also a member of. "`10`" is the numeric value of the user's ID "`wheel`", and "`admin`" is the user we want to add to the "`wheel`" group. Use the same command above for all users on your system you want to be able to `su` to "root" account.

> **NOTE**: For Linux users, who use the Xwindow interface, it is important to note that if you can't `su` in a `GNOME` terminal, it's because you've used the wrong terminal. (So don't think that this advice doesn't work simply because of a `GNOME` terminal problem!)

Facultative:
With the latest Linux operating system, a special line exists in the `su` file `/etc/pam.d/su` which allows you to implicitly trust users in the "**`wheel`**" group (for security reasons, I don't recommend using this option). This mean that all users who are members of the "**`wheel`**" group can `su` to root without the need to enter the "root" password.

- To allow users who are members of the "**`wheel`**" group to `su` to root account without the need to enter the "root" password, edit the **`su`** file (`vi /etc/pam.d/su`) and uncomment the following line in the file:

  `auth sufficient /lib/security/pam_wheel.so trust use_uid`

After this line has been uncommented, the `/etc/pam.d/su` file should look like this:

```
#%PAM-1.0
auth        sufficient      /lib/security/pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
auth        sufficient      /lib/security/pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
auth        required        /lib/security/pam_wheel.so use_uid
auth        required        /lib/security/pam_stack.so service=system-auth
account     required        /lib/security/pam_stack.so service=system-auth
password    required        /lib/security/pam_stack.so service=system-auth
session     required        /lib/security/pam_stack.so service=system-auth
session     optional        /lib/security/pam_xauth.so
```

5 Security and Optimization - General System Optimization

In this Chapter

Linux General System Optimization

Abstract
At this stage of your configuration, you should now have a Linux server optimally configured and secured. Our server contains the most essential package and programs installed to be able to work properly and the most essential general system security configuration. Before we continue and begin to install the services we want to share with our customers, it is important to tune our Linux server.

The tuning we will perform in the following part will be applied to the whole system. It also applies to present as well as future programs, such as services that we will later install. Generally, if you don't use a `x386` Intel processor, Red Hat Linux out of the box is not optimized for your specific CPU architecture (most people now run Linux on a Pentium processor). The sections below will guide you through different steps to optimize your Linux server for your specific processor, memory, and network.

Static vs. shared libraries
During compilation and build time of a program, the last stage (where all the parts of the program are joined together) is to link the software through the Linux libraries if needed. These libraries, which come in both shared and static formats, contain common system code which are kept in one place and shared between programs. Obviously there are some tasks that many programs will want to do, like opening files, and the codes that perform these functions are provided by the Linux libraries. On many Linux system these libraries files can be found into the `/lib`, `/usr/lib`, and `/usr/share` directories. The default behavior of Linux is to link shared and if it cannot find the shared libraries, then is to link statically.

One of the differences between using static or shared libraries are: When using a static library, the linker finds the bits that the program modules need, and directly copies them into the executable output file that it generates. For shared libraries, it leaves a note in the output saying, "when this program is run, it will first have to load this library".

As Gregory A Lundberg from the WU-FTPD Development Group said:
Performance-wise, for most systems, worrying about static vs. dynamic is a moot point. There simply isn't enough difference to measure.

Security-wise there are valid arguments both ways. Static linking is less secure because it locks in the library bugs; unless you rebuild all such programs, your system won't be properly secured. Static linking is more secure because it avoids library attacks. The choice is yours: run a daemon which will remain vulnerable to library attacks, or run one which remains vulnerable to library bugs.

Portability-wise, the only difference is the size of the file you'll be transferring between systems.

To make setup easier, a statically linked daemon is only needed when the libraries are completely unavailable. That is rarely the case. Finally, on a busy system (when performance becomes a true issue), by statically linking you'll be DEGRADING performance. Being bigger, as more and more statically linked daemons are running, your system begins to swap sooner and since none of the code is shared, swapping will have a larger effect on performance. So, when looking to improve performance, you'll want to use shared libraries as much as possible.

If you decide to compile program statically, you will generally need to add the "**-static**" and/or "**--disable-shared**" options flag to your compile line during compilation of your software. Be aware that it is not always possible to use and compile statically all programs, this highly depends on how developers are coding and developed the software.

To resume:

1. If you want to compile program with shared libraries, you will use something like the following:
```
CFLAGS='-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer'
./Configure \
```

2. If you want to compile program with static libraries, you will use something like the following:
```
CFLAGS='-O3 -static -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer'
./Configure \
--disable-shared \
```

WARNING: On Linux, static libraries have names like `libc.a`, while shared libraries are called `libc.so.x.y.z` where `x.y.z` is some form of version number since it would be quite a pain to recompile programs each time the version number changed so instead programs reference libraries by these shorter names and depend on the dynamic linker to make these shorter names symlinks to the current version. Shared libraries often have links pointing to them.

The `Glibc 2.2` library of Linux

The `Glibc 2.2`, which replaces the `libc4` and `libc5` that came before it, is the latest version of the `GNU C` Library for Linux and it contains standard libraries used by multiple programs on the system as described in the previous section. This particular package contains the most important sets of shared and static libraries, which provides the core functionality for `C` programs to run and without it, a Linux system would not function.

Under Red Hat Linux this package comes configured to run under i386 processor for portability reasons and this will pose problems for us if we want to compile programs under Linux because even if we have put in all the optimization flags we need to improve the speed of our server, when the compiler includes static or shared libraries files to our program, these library files will run optimized for an i386 processor.

In this case, our program will have some parts of its binaries optimized for an i686 processor (the program itself) and another parts optimized for an i386 processor (the `GLIBC` libraries). To solve the problem, we have made new RPM's packages at your disposal at the following Internet address:

- Go to this URL and download the following RPM's packages for an i686 CPU:
  URL: *No longer available (Use GLIBC for i686 from the Red Hat Linnux CD-ROM)*

  glibc-2.2.2-1.i686.rpm
  glibc-common-2.2.2-1.i686.rpm
  glibc-devel-2.2.2-1.i686.rpm

For each RPM for your particular architecture, run:

```
[root@deep /]# rpm -Uvh [filename]
```

Why Linux programs are distributed as source

Linux has been ported to run on a large number of different machines and rather than provide a copy for each machine Linux can run on, it's much simpler just to distribute the source and let the end user compile it. The creators of the distribution have no idea if you're going to be running it on a 386 or on a Pentium III and above so they have to write programs that work on all processors and this is where the problem comes, because all the programs that were installed with your distribution are going to be compiled so they work on the 386 for portability, meaning that they don't use any new feature like MMX which can only be found on newer generation of processors.

Fortunately, various compiler options exist to optimize program you want to install under Linux for your specific CPU architecture. This is great for those of us that want to tweak every ounce of performance out of the program, now we get to decide how the program is compiled. If you want some speed out of your programs you've got to know a fair amount about the various option flags you can use to compile.

The first thing you want to set is your CPU type, that's done with the "**`-march=cpu_type`**" (processor machine architecture) flag, an example would be "**`-march=i686`**" or "**`-march=k6`**", this will allow the compiler to select the appropriate optimizations for the processor, but this is only the beginning of what can be done.

You can set the "**`-O`**" flag anywhere from 1 to 3 to tell the compiler how aggressive to be with the optimizations, "**`-O3`**" will produce the fastest programs assuming the compiler didn't optimize an important part of a subroutine out. The next thing you might want to do is check out the "**`-f`**" options of the compiler, these are things like "**`-funroll-loops`**", and "**`-fomit-frame-pointer`**".

> **WARNING:** Compiling with the "**`-fomit-frame-pointer`**" switch option will use the stack for accessing variables. Unfortunately, debugging is almost impossible with this option. Also take special attention to the above optimization number "`-O3`"; "`O`" is a capital `o` and not a `0 (zero)`.

Some misunderstanding in the compiler flags options

At lot of discussions exist in the Linux community about the "**`-O`**" option and its level numbers. Some Linux users try to convince that level number up to "**`-O3`**" like "**`-O9`**" will produce faster program. The "**`-O9`**" flag doesn't do anything over "**`-O3`**", if you don't believe me make a small file, call it `testO3.c` and see:

Step 1

- Create the `testO3.c` file with the following command:
  `[root@deep tmp]#` **`touch testO3.c`**

Step 2

- Run the `GCC` compiler with "**`-O3`**" flag through the `testO3.c` file with the command:
  `[root@deep tmp]#` **`gcc -O3 -S -fverbose-asm testO3.c`**

Step 3
Look at `testO3.s` that it made, then run again with "**-O9**" and compare the output.

- Create the `testO9.c` file with the following command:
  `[root@deep tmp]#` **`touch testO9.c`**

Step 4

- Run the `GCC` compiler again with "**-O9**" flag through the `testO9.c` file with the command:
  `[root@deep tmp]#` **`gcc -O9 -S -fverbose-asm testO9.c`**

Step 5
Now if you compare the output you will see no difference between the both files.

- To compare the output, use the following command:
  `[root@deep tmp]#` **`diff testO3.s testO9.s > difference`**

WARNING: The "**-O3**" flag level number is the best and highest optimization flag you can use during optimization of programs under Linux.

The `gcc 2.96 specs` file

The `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file of Red Hat Linux is a set of defines that the `gcc` compiler uses internally to set various aspects of the compile environment. All customizations that you put in this file will apply for the entire variable environment on your system, so putting optimization flags in this file is a good choice.

To squeeze the maximum performance from your x86 programs, you can use full optimization when compiling with the "**-O3**" flag. Many programs contain "**-O2**" in the Makefile. The "**-O3**" level number is the highest level of optimization. It will increase the size of what it produces, but it runs faster. You can also use the "**-march=cpu_type**" switch to optimize the program for the CPU listed to the best of GCC's ability. However, the resulting code will only be run able on the indicated CPU or higher.

Below are the optimization flags that **we recommend** you to put in your `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file depending on your CPU architecture. The optimization options apply only when we compile and install a new program in our server. These optimizations don't play any role in our Linux base system; it just tells our compiler to optimize the new programs that we will install with the optimization flags we have specified in the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file. Adding options listed below depending of your CPU architecture to the `gcc 2.96` specs file will save you having to change every CFLAGS in future Makefiles.

Step 1
The first thing to do is to verify the compiler version installed on your Linux server.

- To verify the compiler version installed on your system, use the command:

```
[root@deep /]#  gcc -v
Reading specs from /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs
gcc version 2.96 20000731 (Red Hat Linux 7.1 2.96-81)
```

Step 2

For CPU i686 or PentiumPro, Pentium II, Pentium III, and Athlon

Edit the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__    %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_i686__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__    %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: -O3 -march=i686 -funroll-loops -fomit-frame-pointer %{m386:-
mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium} %{mpentiumpro:-
mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O3` and not `-03` (dash zero three).

For CPU i586 or Pentium

Edit the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_i586__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: -O3 -march=i586 -funroll-loops -fomit-frame-pointer %{m386:-
mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium} %{mpentiumpro:-
mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O3` and not `-03` (dash zero three).

For CPU i486

Edit the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_i486__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: -O3 -march=i486 -funroll-loops -fomit-frame-pointer %{m386:-
mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium} %{mpentiumpro:-
mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O3` and not `-03` (dash zero three).

For CPU AMD K6 or K6-2

Edit the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file, scroll down a ways... You'll see a section like the following:

```
*cpp_cpu_default:
-D__tune_i386__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: %{m386:-mcpu=i386} %{m486:-mcpu=i486} %{mpentium:-mcpu=pentium}
%{mpentiumpro:-mcpu=pentiumpro}}
```

Change it for the following:

```
*cpp_cpu_default:
-D__tune_k6__

*cpp_cpu:
-Acpu(i386) -Amachine(i386) %{!ansi:-Di386} -D__i386 -D__i386__
%{march=i386:%{!mcpu*:-D__tune_i386__ }}%{march=i486:-D__i486 -D__i486__
%{!mcpu*:-D__tune_i486__ }}%{march=pentium|march=i586:-D__pentium -D__pentium__
%{!mcpu*:-D__tune_pentium__ }}%{march=pentiumpro|march=i686:-D__pentiumpro -
D__pentiumpro__   %{!mcpu*:-D__tune_pentiumpro__ }}%{march=k6:-D__k6 -D__k6__
%{!mcpu*:-D__tune_k6__ }}%{march=athlon:-D__athlon -D__athlon__ %{!mcpu*:-
D__tune_athlon__ }}%{m386|mcpu=i386:-D__tune_i386__ }%{m486|mcpu=i486:-
D__tune_i486__ }%{mpentium|mcpu=pentium|mcpu=i586:-D__tune_pentium__
}%{mpentiumpro|mcpu=pentiumpro|mcpu=i686:-D__tune_pentiumpro__ }%{mcpu=k6:-
D__tune_k6__ }%{mcpu=athlon:-D__tune_athlon__
}%{!march*:%{!mcpu*:%{!m386:%{!m486:%{!mpentium*:%(cpp_cpu_default)}}}}}

*cc1_cpu:
%{!mcpu*: -O3 -march=k6 -funroll-loops -fomit-frame-pointer %{m386:-mcpu=i386}
%{m486:-mcpu=i486} %{mpentium:-mcpu=pentium} %{mpentiumpro:-mcpu=pentiumpro}}
```

WARNING: Make sure that you're putting `-O3` and not `-03` (dash zero three).

Step3
Once our optimization flags have been applied to the `gcc 2.96` specs file, it time to verify if the modification work.

- To verify if the optimization work, use the following commands:
```
[root@deep tmp]# touch cpu.c
[root@deep tmp]# gcc cpu.c -S -fverbose-asm
[root@deep tmp]# less cpu.s
```

What you'll get is a file that contains depending of options you have chose, something like:

```
        .file   "cpu.c"
        .version        "01.01"
# GNU C version 2.96 20000731 (Red Hat Linux 7.1) (i386-redhat-linux) compiled
by GNU C version 2.96 20000731 (Red Hat Linux 7.1).
# options passed:  -O3 -march=i686 -funroll-loops -fomit-frame-pointer
# -fverbose-asm
# options enabled:  -fdefer-pop -fomit-frame-pointer
# -foptimize-sibling-calls -fcse-follow-jumps -fcse-skip-blocks
# -fexpensive-optimizations -fthread-jumps -fstrength-reduce -funroll-loops
# -fpeephole -fforce-mem -ffunction-cse -finline-functions -finline
# -fkeep-static-consts -fcaller-saves -fpcc-struct-return -fgcse
# -frerun-cse-after-loop -frerun-loop-opt -fdelete-null-pointer-checks
# -fschedule-insns2 -fsched-interblock -fsched-spec -fbranch-count-reg
# -fnew-exceptions -fcommon -fverbose-asm -fgnu-linker -fregmove
# -foptimize-register-move -fargument-alias -fstrict-aliasing -fident
# -fpeephole2 -fmath-errno -m80387 -mhard-float -mno-soft-float -mieee-fp
# -mfp-ret-in-387 -march=i686

gcc2_compiled.:
        .ident  "GCC: (GNU) 2.96 20000731 (Red Hat Linux 7.1 2.96-81)"
```

WARNING: In our example we are optimized the specs file for a i686 CPU processor. It is important to note that most of the "**`-f`**" options are automatically included when you use "**-O3**" and don't need to be specified again. The changes that were shown were made so that a command like "**`gcc`**" would really be the command "**`gcc -march=i686`**" without having to change every single Makefile which can really be a pain.

Below is the explanation of the different optimization options we use:

- **The "`-march=cpu_type`" optimization flag**
  The "`-march=cpu_type`" optimization option will set the default CPU to use for the machine type when scheduling instructions.

- **The "`-funroll-loops`" optimization flag**
  The "`-funroll-loops`" optimization option will perform the optimization of loop unrolling and will do it only for loops whose number of iterations can be determined at compile time or run time.

- **The "`-fomit-frame-pointer`" optimization flag**
  The "`-fomit-frame-pointer`" optimization option, one of the most interesting, will allow the program to not keep the frame pointer in a register for functions that don't need one. This avoids the instructions to save, set up and restores frame pointers; it also makes an extra register available in many functions and makes debugging impossible on most machines.

WARNING: All future optimizations that we will describe in this book refer by default to a Pentium PRO/II/III and higher i686 CPU family. So you must adjust the compilation flags for your specific CPU processor type in the `/usr/lib/gcc-lib/i386-redhat-linux/2.96/specs` file and during your compilation time.

Tuning IDE Hard Disk Performance

The `hdparm` is a tool, which can be used to tune and improve the performance of your `IDE` hard disk. By default, any `IDE` drives you have in your Linux system are not optimized. Even if you have an `ULTRA DMA` system you will not be able to take full advantage of its speed if you are not using the `hdparm` tool to enable its features. This is because there is many different hard drive makes and models and Linux cannot know every feature of each one.

Performance increases have been reported on massive disk I/O operations by setting the `IDE` drivers to use `DMA`, 32-bit transfers and multiple sector modes. The kernel seems to use more conservative settings unless told otherwise. The magic command to change the setting of your drive is `hdparm`.

Before going into the optimization of your hard drive, it is important to verify that the `hdparm` package is installed in your system. If you have followed every step during the installation of Linux on your computer, then this package is not installed.

- To verify if `hdparm` package is installed on your system, use the command:
```
[root@deep /]# rpm -q hdparm
package hdparm is not installed
```

If the `hdparm` package seems not to be installed, you'll need to mount your CD-ROM drive containing the Linux CD-ROM Part 1 and install it.

- To mount the CD-ROM drive, use the following commands:
```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
had: ATAPI 32X CD-ROM drive, 128kB Cache
mount: block device dev/cdrom is write-protected, mounting read-only
```

- To install the `hdparm` package on your Linux system, use the following command:
```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh hdparm-version.i386.rpm
hdparm       ##################################################
```

- To unmount your CD-ROM drive, use the following command:
```
[root@deep RPMS]# cd /; umount /mnt/cdrom/
```

Once `hdparm` package is installed on the system, it is time to go into the optimization of your hard drive. It is important to note that depending on your model and make, there will be some parameters that will apply and other that don't. It is to your responsibility to know and understand your disk drive before applying any optimization parameters as described below.

Finally, and especially for `UltraDMA` systems, it is vital to verify under your `BIOS` settings if the parameters related to `DMA` support on your computer are enabled or you will inevitably break your hard disk. You have been warned.

Step 1
The first parameter applies to the majority of all modern drives and models in the market and enables 32-bit I/O over PCI buses. This option is one of the most important and will usually double the speed of your drive.

- To enable 32-bit I/O over the PCI buses, use the following command:
  `[root@deep /]#  /sbin/hdparm -c3 /dev/hda (or hdb, hdc etc).`

This will usually, depending on your IDE Disk Drive model, cut the timing buffered disk reads time by two. The hdparm (8) manpage says that you may need to use "**-c3**" for many chipsets since it works with nearly all 32-bit IDE chipsets. All (E)IDE drives still have only a 16-bit connection over the ribbon cable from the interface card.

Step 2
The second parameter applies only on standard DMA disk and will activate the simple DMA feature of the disk. This feature is for old disk drives with DMA capabilities.

- To enable DMA, use the following command:
  `[root@deep /]#  /sbin/hdparm -d1 /dev/hda (or hdb, hdc etc).`

This may depend on support for your motherboard chipset being compiled into your kernel. Also, this command will enable DMA support for your hard drive only for interfaces which support DMA, it will cut the timing buffered disk reads time and will improve the performance by two.

Step 3
Multiword DMA mode 2, also kown as ATA2 disk drive is the successor of the simple DMA drive. If you have this kind of hard drive, then you must enable the parameter in your Linux system.

- To enable multiword DMA mode 2 transfers, use the following command:
  `[root@deep /]#  /sbin/hdparm -d1 -X34 /dev/hda (or hdb, hdc etc).`

This sets the IDE transfer mode for newer (E)IDE/ATA2 drives. (Check your hardware manual to see if you have it).

Step 4
As for DMA mode 2, the UltraDMA mode 2 is an improvement of the DMA technology. If you have this kind of drive in your system, then choose this mode.

- To enable UltraDMA mode 2 transfers, use the following command:
  `[root@deep /]#  /sbin/hdparm -d1 -X66 /dev/hda (or hdb, hdc etc)`

See your manual page about hdparm for more information. USE THIS OPTION WITH EXTREME CAUTION!

Step 5
The UltraDMA mode 4 is one of the latest entries and one of the most popular at this time; it is also known and referred as ATA/66. I guess that most of you have this kind of drive installed and if it is the case then it is the one that you must choose for sure.

- To enable UltraDMA mode4 transfers, use the following command:
  `[root@deep /]#  /sbin/hdparm -d1 -X12 -X68 /dev/hda (or hdb, hdc etc)`

This will enable UltraDMA ATA/66 mode on your drive. See your manual page about hdparm for more information. USE THIS OPTION WITH EXTREME CAUTION!

Step 6
Multiple sector mode (aka IDE Block Mode), is a feature of most modern IDE hard drives, permitting the transfer of multiple sectors per I/O interrupt, rather than the usual one sector per interrupt. When this feature is enabled, it typically reduces operating system overhead for disk I/O by 30-50%. On many systems it also provides increased data throughput of anywhere from 5% to 50%.

- To set multiple sector mode I/O, use the following command:
  `[root@deep /]#  /sbin/hdparm -mXX /dev/hda (or hdb, hdc etc)`

Where "XX" is the maximum setting supported by your drive. The "**-i**" flag can be used to find the maximum setting supported by an installed drive: look for **MaxMultSect** in the output.

- To find the maximum setting of your drive, use the following command:
  `[root@deep /]#  /sbin/hdparm -i /dev/hda (or hdb, hdc etc)`

```
/dev/hda:

 Model=QUANTUM FIREBALLP LM15, FwRev=A35.0700, SerialNo=883012661990
 Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs }
 RawCHS=16383/16/63, TrkSize=32256, SectSize=21298, ECCbytes=4
 BuffType=3(DualPortCache), BuffSize=1900kB, MaxMultSect=16, MultSect=16
 DblWordIO=no, OldPIO=2, DMA=yes, OldDMA=2
 CurCHS=16383/16/63, CurSects=-66060037, LBA=yes, LBAsects=29336832
 tDMA={min:120,rec:120}, DMA modes: mword0 mword1 mword2
 IORDY=on/off, tPIO={min:120,w/IORDY:120}, PIO modes: mode3 mode4
 UDMA modes: mode0 mode1 mode2 mode3 *mode4
```

Step 7
The get/set sector count is used to improve performance in sequential reads of large files! The default setting is 8 sectors (4KB) and we will double and change it for 16. USE THIS OPTION WITH EXTREME CAUTION!

- To improve the get/set sector count for file system read-ahead, use the command:
  `[root@deep /]#  /sbin/hdparm -a16 /dev/hda (or hdb, hdc etc)`

Step 8
The get/set interrupt-unmask flag will greatly improve Linux's responsiveness and eliminates "serial port overrun" errors. USE THIS OPTION WITH EXTREME CAUTION!

- To improve and get/set interrupt-unmask flag for the drive, use the command:
  `[root@deep /]#  /sbin/hdparm -u1 /dev/hda (or hdb, hdc etc)`

Step 9
The IDE drive's write-caching feature will improve the performance of the hard disk. USE THIS OPTION WITH EXTREME CAUTION!

- To enable the IDE drive's write-caching feature, use the following command:
  `[root@deep /]#  /sbin/hdparm -W1 /dev/hda (or hdb, hdc etc)`

Step 10
These options will allow the drive to retain your settings over a soft reset (as done during the error recovery sequence). It is important to note that not all drives support this feature.

- To enables the drive to retain your settings, use the command:
  `[root@deep /]#` **`/sbin/hdparm -K1 -k1 /dev/hda`** `(or hdb, hdc etc)`

Step 11
Once every tuning related to your specific drive have been set, you can test the results and see if you want to keep them or not.

- You can test the results of your changes by running `hdparm` in performance test mode:
  `[root@deep /]#` **`/sbin/hdparm -vtT /dev/hda`** `(or hdb, hdc etc).`

```
/dev/hda:
 multcount       = 16 (on)
 I/O support     =  3 (32-bit w/sync)
 unmaskirq       =  1 (on)
 using_dma       =  1 (on)
 keepsettings    =  1 (on)
 nowerr          =  0 (off)
 readonly        =  0 (off)
 readahead       = 16 (on)
 geometry        = 1826/255/63, sectors = 29336832, start = 0
 Timing buffer-cache reads:  128 MB in  0.85 seconds = 150.59 MB/sec
 Timing buffered disk reads:  64 MB in  2.54 seconds = 25.20 MB/sec
```

Once you have a set of `hdparm` options, you can put the commands in your `/etc/rc.d/rc.local` file to run it every time you reboot the machine. When running from `/etc/rc.d/rc.local`, you can add the "**-q**" option for reducing screen clutter. In my case, I will put the following configuration in the end of my `rc.local` file:

```
/sbin/hdparm -q -c3 -d1 -X12 -X68 -m16 -a16 -u1 -W1 -k1 -K1 /dev/had
```

NOTE: The latest realese of Red Hat Linux (7.1) now by default automatically optimizes your `IDE` hard drive. Therefore, you don't have to configure it as shown above but I prefer to tell you this now to let you read this section and understand how hard disk optimization works with the `hdparm` tool of Linux.

6 Security and Optimization – Kernel Security & Optimization

In this Chapter

Making an emergency boot floppy
Checking the `/boot` partition of Linux
Tuning the `Kernel`
Applying the Openwall `kernel` patch
Cleaning up the `Kernel`
Configuring the `Kernel`
Compiling the `Kernel`
Installing the `Kernel`
Reconfiguring `/etc/modules.conf` file
Delete programs, edit files pertaining to modules
Remounting the `/boot` partition of Linux as read-only
Rebooting your system to load the new `kernel`
Making a new rescue floppy for Modularized `Kernel`
Making a emergency boot floppy disk for Monolithic `Kernel`
Optimizing `Kernel`

Linux Kernel

Abstract

Well, our Linux server seems to be getting in shape now! But wait, what is the most important part of our server? Yes, it's the kernel. The Linux kernel is the core of our operating system, and without it there is no Linux at all. So we must take care of our kernel and configure it to fit our needs and compile just features we really need.

The new generation of Linux Kernel 2.4 was seemingly written with the server in mind. Many of the old limits, which prevented Linux adoption in the "enterprise" market, have been lifted. The first thing to do next is to build a kernel that best suits your system. It's very simple to do but, in any case, refer to the `README` file in the `/usr/src/linux` source directory after uncompressing the archive on your system. When configuring your kernel only compile in code that you need and use. Few main reasons that come to mind are:

- ✓ The Kernel will be faster (less code to run),
- ✓ You will have more memory (Kernel parts are NEVER swapped to the virtual memory),
- ✓ More stable (Ever probed for a non-existent card?),
- ✓ Unnecessary parts can be used by an attacker to gain access to the machine or other machines on the network.
- ✓ Modules are also slower than support compiled directly in the kernel.

In our configuration and compilation we will firstly show you how to build a `monolithic kernel`, which is the recommended method for better performance and a `modularized kernel` for easily portability between different Linux systems. `Monolithic kernel` means to only answer **yes** or **no** to the questions (don't make anything modular) and omit the steps: **`make modules`** and **`make modules_install`**.

Unfortunately with Linux kernel 2.4 generation, patching our new kernel with the buffer overflow protection from Openwall kernel patches will not work since the Openwall project announced that Linux 2.4 is NOT going to be supported until 2.4.10 or so. Patches for the Linux kernel exist, like Solar Designer's non-executable stack patch, which disallows the execution of code on the stack, making a number of buffer overflow attacks harder - and defeating completely a number of current exploits used by "script kiddies" worldwide.

These installation instructions assume

Commands are Unix-compatible.
The source path is `/usr/src`.
Installations were tested on Red Hat Linux 7.1.
All steps in the installation will happen using the super-user account "root".
Latest Kernel version number is `2.4.5`
Latest Secure Linux Kernel Patches version number is not available with this kernel.

Packages

The following are based on information as listed by The Linux Kernel Archives as of 2001/05/26 and by the Openwall project as of 2001/05/26. Please regularly check at www.kernel.org and www.openwall.com/linux/ for the latest status.

Pristine source code is available from:
Kernel Homepage: http://www.kernel.org/
Kernel FTP Site: 209.10.41.242
You must be sure to download: linux-2.4.5.tar.gz

Secure Linux Kernel Patches Homepage: http://www.openwall.com/linux/
Secure Linux Kernel Patches FTP Site: 195.42.162.180
You must be sure to download: Not available at this time.

Prerequisites

Depending on whether you want a firewall or users quota support with your system, the Linux Kernel requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install them from your Linux CD-ROM or source archive files. Please make sure you have all of these programs installed on your system before proceeding with this chapter.

- ✓ iptables package, is the new secure and more powerful program used by Linux to set up firewalls as well as IP masquerading in your system. Install this package if you want to support Firewalls in your server.

- ✓ quota package, is a system administration tool for monitoring and limiting users' and/or groups' disk usage, per file system. Install this package if you want a tool to control users directories sizes in your server.

➢ To verify if iptables package is installed on your system, use the command:

```
[root@deep /]# rpm -q iptables
package iptables is not installed
```

➢ To verify if quota package is installed on your system, use the command:

```
[root@deep /]# rpm -q quota
package quota is not installed
```

- To mount your CD-ROM drive before installing the required packages, use the command:

```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
had: ATAPI 32X CD-ROM drive, 128kB Cache
mount: block device dev/cdrom is write-protected, mounting read-only
```

- To install the iptables package on your Linux system, use the following command:

```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh iptables-version.i386.rpm
iptables      ##################################################
```

- To install the quota package on your Linux system, use the following command:

```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh quota-version.i386.rpm
quota     ##################################################
```

NOTE: For more information on `Iptables` Netfilter Firewall configuration or `quota` software, see further down there related chapter in this book.

Making an emergency boot floppy

The first pre-install step is to make an emergency boot floppy. Linux has a small utility named **`mkbootdisk`** to do this. The first step is to find out what kernel version you are currently using. Check out your `/etc/lilo.conf` file and see which image was booted from and from this image we can find the kernel version we need to make our emergency boot floppy. In my example, I have the following in the `lilo.conf` file.

```
[root@deep /]# cat /etc/lilo.conf
boot=/dev/sda
map=/boot/map
install=/boot/boot.b
timeout=00
default=linux
restricted
password=mypasswd

image=/boot/vmlinuz-2.4.2-2      ← the kernel version
    label=linux                  ← the image we booted from
    initrd=/boot/initrd-2.4.2-2.img
    read-only
    root=/dev/sda6
```

Now you'll need to find the image that you booted from. On a standard new first install, it will be the one-labeled **`linux`**. In the above example we show that the machine booted using the `/boot/vmlinuz-2.4.2-2` original kernel version of the system. Now we simply need to put a formatted 1.44 floppy in our system and execute the following command as root:

```
[root@deep /]# mkbootdisk --device /dev/fd0H1440 2.4.2-2
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:
```

Following these guidelines, you will now have a boot floppy with a known working kernel in case of problems with the upgrade. I recommend rebooting the system with the floppy to make sure that the floppy works correctly.

Checking the `/boot` partition of Linux

It is important before going into the compilation and installation of a new kernel to check if the `/boot` file system of Linux is mounted as read-write. If you have follow the steps described in chapter related to "General System Security" under the section named "Mounting the `/boot` directory of Linux as read-only", then your `/boot` file system is mounted as read-only. In this case we must remount it as read-write or you will not be able to install the new kernel on the system. To remount the `/boot` partition as read-write, follow the simple steps below.

Step1

- Edit the **fstab** file (`vi /etc/fstab`) and change the line:

```
LABEL=/boot       /boot       ext2    defaults,ro     1 2
```

To read:

```
LABEL=/boot       /boot       ext2    defaults        1 2
```

We remove the "**ro**" option (read-only) from this line to specify to mount this partition as read-write.

Step 2
Make the Linux system aware about the modification you have made to the `/etc/fstab` file.

- This can be accomplished with the following command:
```
[root@deep /]# mount /boot -oremount
```

- Then test your results or check the state of your `/boot` partition with the command:
```
[root@deep /]# cat /proc/mounts
/dev/root /   ext2        rw      0 0
/proc /proc   proc        rw      0 0
/dev/sda1     /boot       ext2    rw 0 0
/dev/sda10    /cache      ext2    rw,nodev 0 0
/dev/sda9     /chroot     ext2    rw 0 0
/dev/sda8     /home       ext2    rw,nosuid 0 0
/dev/sda13    /tmp        ext2    rw,noexec,nosuid 0 0
/dev/sda7     /usr        ext2    rw 0 0
/dev/sda11    /var        ext2    rw 0 0
/dev/sda12    /var/lib    ext2    rw 0 0
none /dev/pts    devpts  rw 0 0
```

If you see something like: `/dev/sda1 /boot ext2 rw 0 0`, congratulations!

Tuning the Kernel

Ok first of all, it is important to copy the new kernel tar archive in the appropriate location on your server `/usr/src` and then remove the old kernel from your system before installing a new one. Removing the old kernel will not freeze your computer until you try to reboot it before installing the new one because the Linux kernel resides in memory.

Step 1
We must copy the archive file of the kernel to the `/usr/src` directory and move to this directory.

- To copy the tar archive of the Linux kernel to the `/usr/src` directory, use the command:
```
[root@deep /]# cp linux-version.tar.gz /usr/src/
```

- To move to the `/usr/src` directory, use the following command:
```
[root@deep /]# cd /usr/src/
```

Step 2
Depending on how the Linux Kernel has been previously installed on your system, there are two possibilities too uninstall it as shown below.

If you already have installed a Linux kernel with a tar archive before
These steps are required only if you already have installed a Linux kernel with a tar archive before. If it is a first, fresh install of Linux kernel, then instead uninstall the `kernel-headers-version.i386.rpm`, `kernel-version.i386.rpm` packages that are on your system.

- Move to the `/usr/src` directory if you are not already in it with the following command:
  `[root@deep /]#` **`cd /usr/src/`**
- Remove the Linux symbolic link with the following command:
  `[root@deep src]#` **`rm -f linux`**
- Remove the Linux kernel headers directory with the following command:
  `[root@deep src]#` **`rm -rf linux-2.4.x/`**
- Remove the Linux kernel with the following command:
  `[root@deep src]#` **`rm -f /boot/vmlinuz-2.4.x`**
- Remove the Linux `System.map` file with the following command:
  `[root@deep src]#` **`rm -f /boot/System.map-2.4.x`**
- Remove the Linux kernel modules directory (if available) with the following command:
  `[root@deep src]#` **`rm -rf /lib/modules/2.4.x/`**

NOTE: Removing the old kernel modules is required only if you have installed a `modularized kernel` version before. If the modules directory doesn't exist under the `/lib/modules` directory, it's because your old kernel version is not a `modularized kernel`.

If the original kernel's RPM packages are installed on your system
If the original kernel RPM packages are installed on your system instead of the Linux kernel tar archive, because you have just finished installing your new Linux system, or have used an RPM package before to upgrade your Linux system, then use the following command to uninstall the Linux kernel:

- You can verify which kernel RPM packages are installed on your system with the following command:
```
[root@deep src]# rpm -qa | grep kernel
kernel-2.4.2-2
kernel-headers-2.4.2-2
```

The above command shows us that **`kernel`** and **`kernel-headers`** are the only kernel RPM packages installed on our system. We uninstall them as show below.

- To uninstall the linux kernel RPM, use the following command:
  `[root@deep src]#` **`rpm -e --nodeps kernel kernel-headers`**

NOTE: If you receive an error message like: `cannot remove /lib/modules/2.4.x directory, directory not empty`, then remove the directory manually with command like: `rm –rf /lib/modules/2.4.x/` form your system. This directory is related to the old kernel and it is not required for the new kernel we want to install.

Step 3
Once we have uninstalled the old kernel and after our new kernel tar archive has been copied to the `/usr/src` directory, we must uncompress it and remove the tar archive (`linux-version.tar.gz`) from the system if we wish to conserve disk space.

- To uncompress the kernel, use the following command:
  `[root@deep src]# **tar xzpf linux-version.tar.gz**`

- To remove the kernel tar archive from the system, use the following command:
  `[root@deep src]# **rm -f linux-version.tar.gz**`

WARNING: If kernel compilation is something new for you, then it is recommended to keep the kernel tar archive (`linux-version.tar.gz`) until the end of the installation. In this way, if you make some mistake during compilation, you always have the source available to try again.

Step 4
Ok, the old kernel has been uninstalled from our system; we have copied the new one to its appropriate location and uncompressed it. Now, we must tune our new Linux kernel to the maximum of its capabilities. All optimizations shown below are just an increase of the default kernel parameters.

- Edit the **`sem.h`** file (`vi +66 /usr/src/linux/include/linux/sem.h`) and change the following parameter:

  ```
  #define SEMMNI   128     /* <= IPCMNI  max # of semaphore identifiers */
  ```

  To read:

  ```
  #define SEMMNI   512     /* <= IPCMNI  max # of semaphore identifiers */
  ```

- Edit the **`printk.c`** file (`vi +26 /usr/src/linux/kernel/printk.c`) and change the following parameter:

  ```
  #define LOG_BUF_LEN  (16384)
  ```

  To read:

  ```
  #define LOG_BUF_LEN  (65536)
  ```

Step 5
Finally, we must instruct the kernel to fit our specific CPU architecture and optimization flags. Depending of your CPU architecture and optimization flags, this step will improve the performance of the kernel. As an example with a PII 400MHz the BogoMIPS will become **799.54** instead of the default number of **400.00**. Also take a note that it is not because BogoMIPS show you a number of **799.54** for a 400MHz CPU that your processor runs at this speed now. The BogoMIPS result can just be considered as a benchmark since it was a meaningless benchmark measurement.

- Edit the **`Makefile`** file (`vi +19 /usr/src/linux/Makefile`) and change the line:

  ```
  HOSTCFLAGS      = -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
  ```

  To read:

  ```
  HOSTCFLAGS      = -Wall -Wstrict-prototypes -O3 -funroll-loops -fomit-
  frame-pointer
  ```

- Edit the **`Makefile`** file (`vi +90 /usr/src/linux/Makefile`) and change the line:

  ```
  CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes -O2 -fomit-frame-pointer
  -fno-strict-aliasing
  ```

  To read:

  ```
  CFLAGS := $(CPPFLAGS) -Wall -Wstrict-prototypes -O3 -funroll-loops -
  fomit-frame-pointer -fno-strict-aliasing
  ```

WARNING: These changes turn on aggressive optimization tricks that may or may not work with all kernels. Please, if the optimization flags above do not work for you, don't try to force it to work. I wouldn't want to make your system unstable like Microsoft Windows. Also take a note that we are not specifying the "`-march=i686`" option in the above lines since the kernel and related to what processor you will choose during kernel configuration will add automatically this option for you during compilation.

Applying the Openwall kernel patch

The Secure Linux Kernel patches from the Openwall Project are a great way to prevent attacks like Stack Buffer Overflows, and others. The Openwall patch is a collection of security-related features for the Linux kernel, all configurable via the new '"**Security options**" configuration section that will be added to your new kernel.

This patch may change from version to version, and some may contain various other security fixes. Unfortunately Openwall announced that Linux 2.4 is NOT going to be supported until 2.4.10 or so. Below, I'm continuing to show you how to apply this security patch to the kernel in the eventuality that Openwall release a patch for kernel 2.4 generation. As you can see, I use a fictitious version for my example.

New features of patch version linux-2.4.5-ow1.tar.gz are:
Non-executable user stack area
Restricted links in /tmp
Restricted FIFOs in /tmp
Restricted /proc
Special handling of fd 0, 1, and 2
Enforce RLIMIT_NPROC on execve(2)
Destroy shared memory segments not in use

WARNING: When applying the linux-2.4.5-ow1 patch, a new "**Security options**" section will be added at the end of your kernel configuration. For more information and description of the different features available with this patch, see the `README` file that come with the source code of the patch.

- To apply the Openwall Secure Kernel Patch to the Linux kernel, use the commands:

```
[root@deep /]# cp linux-2.4.5-ow1.tar.gz /usr/src/
[root@deep /]# cd /usr/src/
[root@deep src]# tar xzpf linux-2.4.5-ow1.tar.gz
[root@deep src]# cd linux-2.4.5-ow1/
[root@deep linux-2.4.5-ow1]# mv linux-2.4.5-ow1.diff /usr/src/
[root@deep linux-2.4.5-ow1]# cd ..
[root@deep src]# patch -p0 < linux-2.4.5-ow1.diff
[root@deep src]# rm -rf linux-2.4.5-ow1
[root@deep src]# rm -f linux-2.4.5-ow1.diff
[root@deep src]# rm -f linux-2.4.5-ow1.tar.gz
```

First we copy the program archive to the `/usr/src` directory, then we move to this directory and uncompress the linux-2.4.5-ow1.tar.gz archive. We then move to the new uncompressed Linux patch, move the file linux-2.4.5-ow1.diff file containing the patch to the `/usr/src`, return to `/usr/src` and patch our kernel with the file linux-2.4.5-ow1.diff. Afterwards, we remove all files related to the patch.

WARNING: All security messages related to the linux-2.4.5-ow1 patch, like the non-executable stack part, should be logged to the log file `/var/log/messages`. The "**Restricted links in /tmp**" feature of this patch will make Mailing List like `Mailman` to not work properly on the system. The "**Destroy shared memory segments not in use**" feature of this patch will make `SQL` database like `PostgreSQL` to not work properly on the system but this seem to be ok with `MySQL` database now. So if you use or are intended to use one of these services, don't enable the related feature during compilation of the Kernel.

The step of patching your new kernel is completed. Now follow the rest of this installation to build the Linux kernel and reboot your system.

Cleaning up the Kernel

It is important to be sure that your `/usr/include/asm`, and `/usr/include/linux` subdirectories are just symlinks to the kernel sources.

Step 1
The `asm`, and `linux` subdirectories are soft links to the real include kernel source header directories needed for our Linux architecture, for example `/usr/src/linux/include/asm-i386` for `asm`.

- To symlink the `asm`, and `linux` subdirectories to the kernel sources, type the following commands on your terminal:

```
[root@deep src]# cd /usr/include/
[root@deep include]# rm -f asm linux
[root@deep include]# ln -s /usr/src/linux/include/asm-i386 asm
[root@deep include]# ln -s /usr/src/linux/include/linux linux
```

This is a very important part of the configuration: we remove the `asm`, and `linux` directories under `/usr/include` then rebuild a new links that point to the same name directories under the new Linux kernel source version directory. The `/usr/include` directory contains important header files needed by your Linux kernel and programs to be able to compile on your system.

WARNING: If the previously installed kernel in your system was made by RPM packages, then the `asm` and `linux` soft links will not exist since the uninstall of kernel-headers RPM package removes them automatically for you. Don't forget to create them.

Step 2
Make sure you have no stale `.o` files and dependencies lying around.

- To be sure that we have no stale `.o` files and dependencies lying around, type the following commands on your terminal:

```
[root@deep include]# cd /usr/src/linux/
[root@deep linux]# make mrproper
```

NOTE: These two steps above simply clean up anything that might have accidentally been left in the source tree by the development team.

You should now have the sources correctly installed. You can configure the Linux kernel in one of three ways. The first method is to use the **`make config`** command. It provides you with a text-based interface for answering all the configuration options. You are prompted for all the options you need to set up your kernel.

The second method is to use the **`make menuconfig`** command, which provides all the kernel options in an easy-to-use menu. The third is to use the **`make xconfig`** command (only available if the graphical interface of Linux is installed on the system), which provides a full graphical interface to all the kernel options.

Step 3
For configuration in this chapter, you will use the **make config** command because we have not installed the XFree86 Window Interface on our Linux server or the necessary packages to use **make menuconfig** command.

- Type the following commands on your terminal to load the kernel configuration:
```
[root@deep /]# cd /usr/src/linux/ (if you are not already in this directory).
[root@deep linux]# make config
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
```

Configuring the Kernel

As soon as you enter **make config** at the prompt as described in the previous step, a list of kernel configurable options will be displayed for you to choose to configure the kernel, you must indicate what features and devices drivers you want to include in your Linux system and select how to include support for specific devices. Typically, for each configuration option, you have to respond with one of the following choices:

[y] To compile into the kernel and always be loaded.
[m] To use a module for that feature and load that segment of code on demand.
[n] To skip and excludes the support for that specific device from the kernel.

WARNING: It is important to note that an **n** or **y** means the default choice. If a device does not have a modular device driver, you will not see the **[m]** option. Some time an **[?]** option will appear in the choices. This mean that you can get more information about the feature when you type the **? + ENTER key**. Choosing the **[?]** help option will opens another terminal describing the option.

Monolithic kernel configuration

As we know now, they are two possible different configurations for the kernel. The first is called a monolithic kernel the second is called a modularized kernel. Below we begin by showing you the configuration of a monolithic kernel which is to compile the required code and drivers directly into the kernel by answering the different kernel questions only by **yes** or **no**. Don't forget to only compile code that you need and use.

A new kernel is very specific to your computer hardware, in the monolithic kernel configuration part below; we assume the following hardware for our example. Of course you must change them to fit your system components.

1 Pentium-III 667 MHz (i686) processor
1 Motherboard Asus P3V4X Pro 133Mhz EIDE
1 Hard Disk Ultra ATA/66 EIDE
1 Chipset Apollo Pro133A
1 CD-ROM ATAPI IDE
1 Floppy Disk
2 Ethernet Cards 3COM 3c597 PCI 10/100
1 Mouse PS/2

If you don't want some options listed in the `monolithic kernel` configuration that I enable by default, answer **`n`** (for no) instead of **`y`** (for yes) to the related questions. If you want some other options that I disable, then answer **`y`** instead of **`n`**.

In the configuration below, we tune our kernel for a Pentium III family i686 CPU processor, enable generic firewall support, to be able to implement `IPTABLE` Netfilter firewall feature on the system, as well as `DMA` support for `IDE` disk drive and disable `SCSI` disk support. We configure the kernel to work with a `3COM` Ethernet card, disable insecure `NFS` services, `USB` technology and sound features for our server. This kind of kernel configuration can be used for all kind of Linux server except for a system, which is supposed to run as a Gateway/Proxy Server by forwarding packets.

```
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [N/y/?]
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?] n
*
* Processor type and features
*
Processor family (386, 486, 586/K5/5x86/6x86/6x86MX, Pentium-Classic, Pentium-MMX, Pentium-
Pro/Celeron/Pentium-II, Pentium-III, Pentium-4, K6/K6-II/K6-III, Athlon/K7, Crusoe, Winchip-C6, Winchip-2,
Winchip-2A/Winchip-3) [Pentium-III]
  defined CONFIG_M686FXSR
Toshiba Laptop support (CONFIG_TOSHIBA) [N/y/?]
/dev/cpu/microcode - Intel IA32 CPU microcode support (CONFIG_MICROCODE) [N/y/?]
/dev/cpu/*/msr - Model-specific register support (CONFIG_X86_MSR) [N/y/?]
/dev/cpu/*/cpuid - CPU information support (CONFIG_X86_CPUID) [N/y/?]
High Memory Support (off, 4GB, 64GB) [off]
  defined CONFIG_NOHIGHMEM
MTRR (Memory Type Range Register) support (CONFIG_MTRR) [N/y/?]
Symmetric multi-processing support (CONFIG_SMP) [Y/n/?] n
APIC and IO-APIC support on uniprocessors (CONFIG_X86_UP_IOAPIC) [N/y/?] (NEW) y
*
* General setup
*
Networking support (CONFIG_NET) [Y/n/?]
SGI Visual Workstation support (CONFIG_VISWS) [N/y/?]
PCI support (CONFIG_PCI) [Y/n/?]
  PCI access mode (BIOS, Direct, Any) [Any]
  defined CONFIG_PCI_GOANY
PCI device name database (CONFIG_PCI_NAMES) [Y/n/?] n
EISA support (CONFIG_EISA) [N/y/?]
MCA support (CONFIG_MCA) [N/y/?]
Support for hot-pluggable devices (CONFIG_HOTPLUG) [Y/n/?] n
System V IPC (CONFIG_SYSVIPC) [Y/n/?]
BSD Process Accounting (CONFIG_BSD_PROCESS_ACCT) [N/y/?]
Sysctl support (CONFIG_SYSCTL) [Y/n/?]
Kernel core (/proc/kcore) format (ELF, A.OUT) [ELF]
  defined CONFIG_KCORE_ELF
Kernel support for a.out binaries (CONFIG_BINFMT_AOUT) [Y/n/?]
Kernel support for ELF binaries (CONFIG_BINFMT_ELF) [Y/n/?]
```

```
Kernel support for MISC binaries (CONFIG_BINFMT_MISC) [Y/n/?]
Power Management support (CONFIG_PM) [Y/n/?] n
*
* Memory Technology Devices (MTD)
*
Memory Technology Device (MTD) support (CONFIG_MTD) [N/y/?]
*
* Parallel port support
*
Parallel port support (CONFIG_PARPORT) [N/y/?]
*
* Plug and Play configuration
*
Plug and Play support (CONFIG_PNP) [Y/n/?] n
*
* Block devices
*
Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y/m/n/?]
XT hard disk support (CONFIG_BLK_DEV_XD) [N/y/m/?]
Compaq SMART2 support (CONFIG_BLK_CPQ_DA) [N/y/m/?]
Compaq CISS Array support (CONFIG_BLK_CPQ_CISS_DA) [N/y/m/?]
Mylex DAC960/DAC1100 PCI RAID Controller support (CONFIG_BLK_DEV_DAC960) [N/y/m/?]
Loopback device support (CONFIG_BLK_DEV_LOOP) [N/y/m/?]
Network block device support (CONFIG_BLK_DEV_NBD) [N/y/m/?]
RAM disk support (CONFIG_BLK_DEV_RAM) [N/y/m/?]
*
* Multi-device support (RAID and LVM)
*
Multiple devices driver support (RAID and LVM) (CONFIG_MD) [N/y/?]
*
* Networking options
*
Packet socket (CONFIG_PACKET) [Y/m/n/?]
  Packet socket: mmapped IO (CONFIG_PACKET_MMAP) [N/y/?] y
Kernel/User netlink socket (CONFIG_NETLINK) [N/y/?] y
  Routing messages (CONFIG_RTNETLINK) [N/y/?] (NEW) y
  Netlink device emulation (CONFIG_NETLINK_DEV) [N/y/m/?] (NEW) y
Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) [N/y/?] y
  Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) [N/y/?] (NEW) y
Socket Filtering (CONFIG_FILTER) [N/y/?]
Unix domain sockets (CONFIG_UNIX) [Y/m/n/?]
TCP/IP networking (CONFIG_INET) [Y/n/?]
  IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] n
  IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) [N/y/?]
  IP: kernel level autoconfiguration (CONFIG_IP_PNP) [N/y/?]
  IP: tunneling (CONFIG_NET_IPIP) [N/y/?]
  IP: GRE tunnels over IP (CONFIG_NET_IPGRE) [N/y/?]
  IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) [N/y/?]
  IP: TCP syncookie support (disabled per default) (CONFIG_SYN_COOKIES) [N/y/?] y
*
*   IP: Netfilter Configuration
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) [N/y/?] (NEW)
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) [N/y/?] (NEW) y
  limit match support (CONFIG_IP_NF_MATCH_LIMIT) [N/y/m/?] (NEW) y
  MAC address match support (CONFIG_IP_NF_MATCH_MAC) [N/y/m/?] (NEW) y
  netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) [N/y/m/?] (NEW) y
  Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) [N/y/m/?] (NEW) y
  TOS match support (CONFIG_IP_NF_MATCH_TOS) [N/y/m/?] (NEW) y
    tcpmss match support (CONFIG_IP_NF_MATCH_TCPMSS) [N/y/?] (NEW) y
  Packet filtering (CONFIG_IP_NF_FILTER) [N/y/m/?] (NEW) y
    REJECT target support (CONFIG_IP_NF_TARGET_REJECT) [N/y/m/?] (NEW) y
```

```
  Packet mangling (CONFIG_IP_NF_MANGLE) [N/y/m/?] (NEW) y
    TOS target support (CONFIG_IP_NF_TARGET_TOS) [N/y/m/?] (NEW) y
    MARK target support (CONFIG_IP_NF_TARGET_MARK) [N/y/m/?] (NEW) y
  LOG target support (CONFIG_IP_NF_TARGET_LOG) [N/y/m/?] (NEW) y
  TCPMSS target support (CONFIG_IP_NF_TARGET_TCPMSS) [N/y/m/?] (NEW) y
*
*
*
The IPX protocol (CONFIG_IPX) [N/y/?]
Appletalk protocol support (CONFIG_ATALK) [N/y/?]
DECnet Support (CONFIG_DECNET) [N/y/?]
802.1d Ethernet Bridging (CONFIG_BRIDGE) [N/y/?]
*
* QoS and/or fair queuering
*
QoS and/or fair queuring (EXPERIMENTAL) (CONFIG_NET_SCHED) [N/y/?]
*
* Telephony Support
*
Linux telephony support (CONFIG_PHONE) [N/y/?]
*
* ATA/IDE/MFM/RLL support
*
ATA/IDE/MFM/RLL support (CONFIG_IDE) [Y/n/?]
*
* IDE, ATA and ATAPI Block devices
*
Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support (CONFIG_BLK_DEV_IDE) [Y/n/?]
*
* Please see Documentation/ide.txt for help/info on IDE drives
*
  Use old disk-only driver on primary interface (CONFIG_BLK_DEV_HD_IDE) [N/y/?]
  Include IDE/ATA-2 DISK support (CONFIG_BLK_DEV_IDEDISK) [Y/n/?]
    Use multi-mode by default (CONFIG_IDEDISK_MULTI_MODE) [N/y/?]
  Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [Y/n/?]
  Include IDE/ATAPI TAPE support (CONFIG_BLK_DEV_IDETAPE) [N/y/?]
  Include IDE/ATAPI FLOPPY support (CONFIG_BLK_DEV_IDEFLOPPY) [N/y/?]
  SCSI emulation support (CONFIG_BLK_DEV_IDESCSI) [N/y/?]
*
* IDE chipset support/bugfixes
*
  CMD640 chipset bugfix/support (CONFIG_BLK_DEV_CMD640) [Y/n/?] n
  RZ1000 chipset bugfix/support (CONFIG_BLK_DEV_RZ1000) [Y/n/?] n
  Generic PCI IDE chipset support (CONFIG_BLK_DEV_IDEPCI) [Y/n/?]
    Sharing PCI IDE interrupts support (CONFIG_IDEPCI_SHARE_IRQ) [Y/n/?]
    Generic PCI bus-master DMA support (CONFIG_BLK_DEV_IDEDMA_PCI) [N/y/?] y
    Boot off-board chipsets first support (CONFIG_BLK_DEV_OFFBOARD) [N/y/?]
      Use PCI DMA by default when available (CONFIG_IDEDMA_PCI_AUTO) [N/y/?] y
    AEC62XX chipset support (CONFIG_BLK_DEV_AEC62XX) [N/y/?]
    ALI M15x3 chipset support (CONFIG_BLK_DEV_ALI15X3) [N/y/?]
    AMD Viper support (CONFIG_BLK_DEV_AMD7409) [N/y/?]
    CMD64X chipset support (CONFIG_BLK_DEV_CMD64X) [N/y/?]
    CY82C693 chipset support (CONFIG_BLK_DEV_CY82C693) [N/y/?]
    Cyrix CS5530 MediaGX chipset support (CONFIG_BLK_DEV_CS5530) [N/y/?]
    HPT34X chipset support (CONFIG_BLK_DEV_HPT34X) [N/y/?]
    HPT366 chipset support (CONFIG_BLK_DEV_HPT366) [N/y/?]
    Intel PIIXn chipsets support (CONFIG_BLK_DEV_PIIX) [N/y/?]
    NS87415 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_NS87415) [N/y/?]
    PROMISE PDC20246/PDC20262/PDC20267 support (CONFIG_BLK_DEV_PDC202XX) [N/y/?]
    ServerWorks OSB4 chipset support (CONFIG_BLK_DEV_OSB4) [N/y/?]
    SiS5513 chipset support (CONFIG_BLK_DEV_SIS5513) [N/y/?]
    SLC90E66 chipset support (CONFIG_BLK_DEV_SLC90E66) [N/y/?]
```

```
   Tekram TRM290 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_TRM290) [N/y/?]
   VIA82CXXX chipset support (CONFIG_BLK_DEV_VIA82CXXX) [N/y/?] y
  Other IDE chipset support (CONFIG_IDE_CHIPSETS) [N/y/?]
  IGNORE word93 Validation BITS (CONFIG_IDEDMA_IVB) [N/y/?] (NEW)
*
* SCSI support
*
SCSI support (CONFIG_SCSI) [Y/n/?] n
*
* I2O device support
*
I2O support (CONFIG_I2O) [N/y/?]
*
* Network device support
*
Network device support (CONFIG_NETDEVICES) [Y/n/?]
*
* ARCnet devices
*
ARCnet support (CONFIG_ARCNET) [N/y/?]
Dummy net driver support (CONFIG_DUMMY) [Y/n/?]
Bonding driver support (CONFIG_BONDING) [N/y/?]
EQL (serial line load balancing) support (CONFIG_EQUALIZER) [N/y/?]
Universal TUN/TAP device driver support (CONFIG_TUN) [N/y/?]
General Instruments Surfboard 1000 (CONFIG_NET_SB1000) [N/y/?]
*
* Ethernet (10 or 100Mbit)
*
Ethernet (10 or 100Mbit) (CONFIG_NET_ETHERNET) [Y/n/?]
  3COM cards (CONFIG_NET_VENDOR_3COM) [N/y/?] y
    3c501 "EtherLink" support (CONFIG_EL1) [N/y/?] (NEW)
    3c503 "EtherLink II" support (CONFIG_EL2) [N/y/?] (NEW)
    3c505 "EtherLink Plus" support (CONFIG_ELPLUS) [N/y/?] (NEW)
    3c509/3c529 (MCA)/3c579 "EtherLink III" support (CONFIG_EL3) [N/y/?] (NEW)
    3c515 ISA "Fast EtherLink" (CONFIG_3C515) [N/y/?] (NEW)
    3c590/3c900 series (592/595/597) "Vortex/Boomerang" support (CONFIG_VORTEX) [N/y/?] (NEW) y
  AMD LANCE and PCnet (AT1500 and NE2100) support (CONFIG_LANCE) [N/y/?]
  Western Digital/SMC cards (CONFIG_NET_VENDOR_SMC) [N/y/?]
  Racal-Interlan (Micom) NI cards (CONFIG_NET_VENDOR_RACAL) [N/y/?]
  DEPCA, DE10x, DE200, DE201, DE202, DE422 support (CONFIG_DEPCA) [N/y/?]
  HP 10/100VG PCLAN (ISA, EISA, PCI) support (CONFIG_HP100) [N/y/?]
  Other ISA cards (CONFIG_NET_ISA) [N/y/?]
  EISA, VLB, PCI and on board controllers (CONFIG_NET_PCI) [Y/n/?] n
  Pocket and portable adapters (CONFIG_NET_POCKET) [N/y/?]
*
* Ethernet (1000 Mbit)
*
Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support (CONFIG_ACENIC) [N/y/?]
Packet Engines Hamachi GNIC-II support (CONFIG_HAMACHI) [N/y/?]
SysKonnect SK-98xx support (CONFIG_SK98LIN) [N/y/?]
FDDI driver support (CONFIG_FDDI) [N/y/?]
PPP (point-to-point protocol) support (CONFIG_PPP) [N/y/?]
SLIP (serial line) support (CONFIG_SLIP) [N/y/?]
*
* Wireless LAN (non-hamradio)
*
Wireless LAN (non-hamradio) (CONFIG_NET_RADIO) [N/y/?]
*
* Token Ring devices
*
Token Ring driver support (CONFIG_TR) [N/y/?]
Fibre Channel driver support (CONFIG_NET_FC) [N/y/?]
```

```
*
* Wan interfaces
*
Wan interfaces support (CONFIG_WAN) [N/y/?]
*
* Amateur Radio support
*
Amateur Radio support (CONFIG_HAMRADIO) [N/y/?]
*
* IrDA (infrared) support
*
IrDA subsystem support (CONFIG_IRDA) [N/y/?]
*
* ISDN subsystem
*
ISDN support (CONFIG_ISDN) [N/y/?]
*
* Old CD-ROM drivers (not SCSI, not IDE)
*
Support non-SCSI/IDE/ATAPI CDROM drives (CONFIG_CD_NO_IDESCSI) [N/y/?]
*
* Input core support
*
Input core support (CONFIG_INPUT) [N/y/?]
*
* Character devices
*
Virtual terminal (CONFIG_VT) [Y/n/?]
  Support for console on virtual terminal (CONFIG_VT_CONSOLE) [Y/n/?]
Standard/generic (8250/16550 and compatible UARTs) serial support (CONFIG_SERIAL) [Y/n/?]
  Support for console on serial port (CONFIG_SERIAL_CONSOLE) [N/y/?]
Extended dumb serial driver options (CONFIG_SERIAL_EXTENDED) [N/y/?]
Non-standard serial port support (CONFIG_SERIAL_NONSTANDARD) [N/y/?]
Unix98 PTY support (CONFIG_UNIX98_PTYS) [Y/n/?]
Maximum number of Unix98 PTYs in use (0-2048) (CONFIG_UNIX98_PTY_COUNT) [256] 128
*
* I2C support
*
I2C support (CONFIG_I2C) [N/y/?]
*
* Mice
*
Bus Mouse Support (CONFIG_BUSMOUSE) [N/y/?]
Mouse Support (not serial and bus mice) (CONFIG_MOUSE) [Y/n/?]
  PS/2 mouse (aka "auxiliary device") support (CONFIG_PSMOUSE) [Y/n/?]
  C&T 82C710 mouse port support (as on TI Travelmate) (CONFIG_82C710_MOUSE) [N/y/?]
  PC110 digitizer pad support (CONFIG_PC110_PAD) [N/y/?]
*
* Joysticks
*
QIC-02 tape support (CONFIG_QIC02_TAPE) [N/y/?]
*
* Watchdog Cards
*
Watchdog Timer Support (CONFIG_WATCHDOG) [N/y/?]
Intel i8x0 Random Number Generator support (CONFIG_INTEL_RNG) [N/y/?]
/dev/nvram support (CONFIG_NVRAM) [N/y/?]
Enhanced Real Time Clock Support (CONFIG_RTC) [N/y/?]
Double Talk PC internal speech card support (CONFIG_DTLK) [N/y/?]
Siemens R3964 line discipline (CONFIG_R3964) [N/y/?]
Applicom intelligent fieldbus card support (CONFIG_APPLICOM) [N/y/?]
*
```

*** Ftape, the floppy tape device driver**
*
Ftape (QIC-80/Travan) support (CONFIG_FTAPE) [N/y/?]
/dev/agpgart (AGP Support) (CONFIG_AGP) [Y/m/n/?] **n**
Direct Rendering Manager (XFree86 DRI support) (CONFIG_DRM) [Y/n/?] **n**
*
*** Multimedia devices**
*
Video For Linux (CONFIG_VIDEO_DEV) [N/y/?]
*
*** File systems**
*
Quota support (CONFIG_QUOTA) [N/y/?]
Kernel automounter support (CONFIG_AUTOFS_FS) [N/y/?]
Kernel automounter version 4 support (also supports v3) (CONFIG_AUTOFS4_FS) [Y/n/?] **n**
DOS FAT fs support (CONFIG_FAT_FS) [N/y/?]
Compressed ROM file system support (CONFIG_CRAMFS) [N/y/?]
Simple RAM-based file system support (CONFIG_RAMFS) [N/y/?]
ISO 9660 CDROM file system support (CONFIG_ISO9660_FS) [Y/n/?]
  Microsoft Joliet CDROM extensions (CONFIG_JOLIET) [N/y/?]
Minix fs support (CONFIG_MINIX_FS) [N/y/?]
NTFS file system support (read only) (CONFIG_NTFS_FS) [N/y/?]
OS/2 HPFS file system support (CONFIG_HPFS_FS) [N/y/?]
/proc file system support (CONFIG_PROC_FS) [Y/n/?]
/dev/pts file system for Unix98 PTYs (CONFIG_DEVPTS_FS) [Y/n/?]
ROM file system support (CONFIG_ROMFS_FS) [N/y/?]
Second extended fs support (CONFIG_EXT2_FS) [Y/n/?]
System V and Coherent file system support (read only) (CONFIG_SYSV_FS) [N/y/?]
UDF file system support (read only) (CONFIG_UDF_FS) [N/y/?]
UFS file system support (read only) (CONFIG_UFS_FS) [N/y/?]
*
*** Network File Systems**
*
Coda file system support (advanced network fs) (CONFIG_CODA_FS) [N/y/?]
NFS file system support (CONFIG_NFS_FS) [Y/n/?] **n**
NFS server support (CONFIG_NFSD) [Y/n/?] **n**
SMB file system support (to mount Windows shares etc.) (CONFIG_SMB_FS) [N/y/?]
NCP file system support (to mount NetWare volumes) (CONFIG_NCP_FS) [N/y/?]
*
*** Partition Types**
*
Advanced partition selection (CONFIG_PARTITION_ADVANCED) [N/y/?]
*
*** Console drivers**
*
VGA text console (CONFIG_VGA_CONSOLE) [Y/n/?]
Video mode selection support (CONFIG_VIDEO_SELECT) [N/y/?]
*
*** Sound**
*
Sound card support (CONFIG_SOUND) [Y/n/?] **n**
*
(Security options will appear only if you are patched your kernel with the Openwall Project patch).
*** Security options**
*
Non-executable user stack area (CONFIG_SECURE_STACK) [Y]
Autodetect and emulate GCC trampolines (CONFIG_SECURE_STACK_SMART) [Y]
Restricted links in /tmp (CONFIG_SECURE_LINK) [Y] **n**
Restricted FIFOs in /tmp (CONFIG_SECURE_FIFO) [Y]
Restricted /proc (CONFIG_SECURE_PROC) [N] **y**
Special handling of fd 0, 1, and 2 (CONFIG_SECURE_FD_0_1_2) [Y]
Enforce RLIMIT_NPROC on execve(2) (CONFIG_SECURE_RLIMIT_NPROC) [Y]

```
Destroy shared memory segments not in use (CONFIG_SECURE_SHM) [N]
*
* USB support
*
Support for USB (CONFIG_USB) [Y/n/?] n
*
* Kernel hacking
*
Magic SysRq key (CONFIG_MAGIC_SYSRQ) [N/y/?]

*** End of Linux kernel configuration.
*** Check the top-level Makefile for additional configuration.
*** Next, you must run 'make dep'.
```

WARNING: If you want to enable `IPTABLES` support into the kernel, the `iptables` program must be installed first or you will receive error messages during kernel compilation. This is because when `iptables` support is enabled, the kernelwill associate some part of the `iptables` program with it configuration. Therefore don't forget to install `IPTABLES` before configuring kernel with `IPTABLES` support. Finally the same warning is true for `quota` support into the kernel.

Modularized kernel configuration

Building kernel with modules (`modularized kernel`) has some advantages. It allow easy portability between different Linux systems, since you can choose and build different parts of the kernel as a module and load that segment of code on demand. Below we show you the configuration of `modularized kernel`, which is to compile some needed codes and drivers as a module into the kernel by answering to the different questions by **`y`**, **`n`** or **`m`**. As for the previous `monolithic kernel` configuration, don't forget to only compile code that you need and use.

A new kernel is very specific to your computer hardware, in the `modularized kernel` configuration part below; we assume the following hardware for our example. Of course you must change them to fit your system components.

1 Pentium II 400 MHz (i686) processor
1 SCSI Motherboard
1 SCSI Hard Disk
1 SCSI Controler Adaptec AIC 7xxx
1 CD-ROM ATAPI IDE
1 Floppy Disk
2 Ethernet Cards Intel EtherExpressPro 10/100
1 Mouse PS/2

If you don't want some options listed in the `modularized kernel` configuration that I enable by default, answer **`n`** (for no) instead of **`y`** (for yes) or **`m`** (for modularized if possible) to the related questions. If you want some other options that I disable, then answer **`y`** or **`m`** instead of **`n`**.

In the configuration below, we have enable loadable module support in the kernel, tune our kernel for a Pentium II family i686 CPU processor, enable full Firewall Netfilter with masquerading and forwarding support. This is a perfect configuration if you want to run your system as a Gateway/Proxy Server since it will be capable to forward and redistribute network packet. After that, we enable `DMA` support for `IDE` disk drives since our CD-ROM in this example is an `IDE` model (if your system is pure `SCSI` we can disable support for `IDE` and `DMA`) and enable `SCSI` disk support for Adaptec `AIC7xxx` model. We configure the kernel to work with Intel EtherExpressPro/100 network cards, disable insecure `NFS` services, `USB` technology and sound features for our Linux server.

```
rm -f include/asm
( cd include ; ln -sf asm-i386 asm)
/bin/sh scripts/Configure arch/i386/config.in
#
# Using defaults found in arch/i386/defconfig
#
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (CONFIG_EXPERIMENTAL) [N/y/?]
*
* Loadable module support
*
Enable loadable module support (CONFIG_MODULES) [Y/n/?]
  Set version information on all module symbols (CONFIG_MODVERSIONS) [Y/n/?] n
  Kernel module loader (CONFIG_KMOD) [Y/n/?]
*
* Processor type and features
*
Processor family (386, 486, 586/K5/5x86/6x86/6x86MX, Pentium-Classic, Pentium-MMX, Pentium-
Pro/Celeron/Pentium-II, Pentium-III, Pentium-4, K6/K6-II/K6-III, Athlon/K7, Crusoe, Winchip-C6, Winchip-2,
Winchip-2A/Winchip-3) [Pentium-III] Pentium-Pro/Celeron/Pentium-II
  defined CONFIG_M686
Toshiba Laptop support (CONFIG_TOSHIBA) [N/y/m/?]
/dev/cpu/microcode - Intel IA32 CPU microcode support (CONFIG_MICROCODE) [N/y/m/?]
/dev/cpu/*/msr - Model-specific register support (CONFIG_X86_MSR) [N/y/m/?]
/dev/cpu/*/cpuid - CPU information support (CONFIG_X86_CPUID) [N/y/m/?]
High Memory Support (off, 4GB, 64GB) [off]
  defined CONFIG_NOHIGHMEM
Math emulation (CONFIG_MATH_EMULATION) [N/y/?] (NEW)
MTRR (Memory Type Range Register) support (CONFIG_MTRR) [N/y/?]
Symmetric multi-processing support (CONFIG_SMP) [Y/n/?] n
APIC and IO-APIC support on uniprocessors (CONFIG_X86_UP_IOAPIC) [N/y/?] (NEW) y
*
* General setup
*
Networking support (CONFIG_NET) [Y/n/?]
SGI Visual Workstation support (CONFIG_VISWS) [N/y/?]
PCI support (CONFIG_PCI) [Y/n/?]
  PCI access mode (BIOS, Direct, Any) [Any]
  defined CONFIG_PCI_GOANY
PCI device name database (CONFIG_PCI_NAMES) [Y/n/?] n
EISA support (CONFIG_EISA) [N/y/?]
MCA support (CONFIG_MCA) [N/y/?]
Support for hot-pluggable devices (CONFIG_HOTPLUG) [Y/n/?] n
System V IPC (CONFIG_SYSVIPC) [Y/n/?]
BSD Process Accounting (CONFIG_BSD_PROCESS_ACCT) [N/y/?]
Sysctl support (CONFIG_SYSCTL) [Y/n/?]
Kernel core (/proc/kcore) format (ELF, A.OUT) [ELF]
  defined CONFIG_KCORE_ELF
Kernel support for a.out binaries (CONFIG_BINFMT_AOUT) [Y/m/n/?]
```

```
Kernel support for ELF binaries (CONFIG_BINFMT_ELF) [Y/m/n/?]
Kernel support for MISC binaries (CONFIG_BINFMT_MISC) [Y/m/n/?]
Power Management support (CONFIG_PM) [Y/n/?] n
*
* Memory Technology Devices (MTD)
*
Memory Technology Device (MTD) support (CONFIG_MTD) [N/y/m/?]
*
* Parallel port support
*
Parallel port support (CONFIG_PARPORT) [N/y/m/?]
*
* Plug and Play configuration
*
Plug and Play support (CONFIG_PNP) [Y/m/n/?] n
*
* Block devices
*
Normal PC floppy disk support (CONFIG_BLK_DEV_FD) [Y/m/n/?]
XT hard disk support (CONFIG_BLK_DEV_XD) [N/y/m/?]
Compaq SMART2 support (CONFIG_BLK_CPQ_DA) [N/y/m/?]
Compaq CISS Array support (CONFIG_BLK_CPQ_CISS_DA) [N/y/m/?]
Mylex DAC960/DAC1100 PCI RAID Controller support (CONFIG_BLK_DEV_DAC960) [N/y/m/?]
Loopback device support (CONFIG_BLK_DEV_LOOP) [N/y/m/?]
Network block device support (CONFIG_BLK_DEV_NBD) [N/y/m/?]
RAM disk support (CONFIG_BLK_DEV_RAM) [N/y/m/?]
*
* Multi-device support (RAID and LVM)
*
Multiple devices driver support (RAID and LVM) (CONFIG_MD) [N/y/?]
*
* Networking options
*
Packet socket (CONFIG_PACKET) [Y/m/n/?]
  Packet socket: mmapped IO (CONFIG_PACKET_MMAP) [N/y/?] y
Kernel/User netlink socket (CONFIG_NETLINK) [N/y/?] y
  Routing messages (CONFIG_RTNETLINK) [N/y/?] (NEW) y
  Netlink device emulation (CONFIG_NETLINK_DEV) [N/y/m/?] (NEW) y
Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) [N/y/?] y
  Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) [N/y/?] (NEW) y
Socket Filtering (CONFIG_FILTER) [N/y/?]
Unix domain sockets (CONFIG_UNIX) [Y/m/n/?]
TCP/IP networking (CONFIG_INET) [Y/n/?]
  IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] n
  IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) [N/y/?] y
    IP: policy routing (CONFIG_IP_MULTIPLE_TABLES) [N/y/?] (NEW) y
      IP: use netfilter MARK value as routing key (CONFIG_IP_ROUTE_FWMARK) [N/y/?] (NEW) y
      IP: fast network address translation (CONFIG_IP_ROUTE_NAT) [N/y/?] (NEW) y
    IP: equal cost multipath (CONFIG_IP_ROUTE_MULTIPATH) [N/y/?] (NEW) y
    IP: use TOS value as routing key (CONFIG_IP_ROUTE_TOS) [N/y/?] (NEW) y
    IP: verbose route monitoring (CONFIG_IP_ROUTE_VERBOSE) [N/y/?] (NEW) y
    IP: large routing tables (CONFIG_IP_ROUTE_LARGE_TABLES) [N/y/?] (NEW) y
  IP: kernel level autoconfiguration (CONFIG_IP_PNP) [N/y/?]
  IP: tunneling (CONFIG_NET_IPIP) [N/y/m/?]
  IP: GRE tunnels over IP (CONFIG_NET_IPGRE) [N/y/m/?]
  IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) [N/y/?]
  IP: TCP syncookie support (disabled per default) (CONFIG_SYN_COOKIES) [N/y/?] y
*
*   IP: Netfilter Configuration
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) [N/y/m/?] (NEW) m
  FTP protocol support (CONFIG_IP_NF_FTP) [N/m/?] (NEW) m
```

IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) [N/y/m/?] (NEW) **m**
  limit match support (CONFIG_IP_NF_MATCH_LIMIT) [N/m/?] (NEW) **m**
  MAC address match support (CONFIG_IP_NF_MATCH_MAC) [N/m/?] (NEW) **m**
  netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) [N/m/?] (NEW) **m**
  Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) [N/m/?] (NEW) **m**
  TOS match support (CONFIG_IP_NF_MATCH_TOS) [N/m/?] (NEW) **m**
    tcpmss match support (CONFIG_IP_NF_MATCH_TCPMSS) [N/m/?] (NEW) **m**
  Connection state match support (CONFIG_IP_NF_MATCH_STATE) [N/m/?] (NEW) **m**
  Packet filtering (CONFIG_IP_NF_FILTER) [N/m/?] (NEW) **m**
    REJECT target support (CONFIG_IP_NF_TARGET_REJECT) [N/m/?] (NEW) **m**
  Full NAT (CONFIG_IP_NF_NAT) [N/m/?] (NEW) **m**
    MASQUERADE target support (CONFIG_IP_NF_TARGET_MASQUERADE) [N/m/?] (NEW) **m**
    REDIRECT target support (CONFIG_IP_NF_TARGET_REDIRECT) [N/m/?] (NEW) **m**
  Packet mangling (CONFIG_IP_NF_MANGLE) [N/m/?] (NEW) **m**
    TOS target support (CONFIG_IP_NF_TARGET_TOS) [N/m/?] (NEW) **m**
    MARK target support (CONFIG_IP_NF_TARGET_MARK) [N/m/?] (NEW) **m**
  LOG target support (CONFIG_IP_NF_TARGET_LOG) [N/m/?] (NEW) **m**
TCPMSS target support (CONFIG_IP_NF_TARGET_TCPMSS) [N/m/?] (NEW) **m**
ipchains (2.2-style) support (CONFIG_IP_NF_COMPAT_IPCHAINS) [N/y/m/?] (NEW)
ipfwadm (2.0-style) support (CONFIG_IP_NF_COMPAT_IPFWADM) [N/y/m/?] (NEW)
*
*
*
The IPX protocol (CONFIG_IPX) [N/y/m/?]
Appletalk protocol support (CONFIG_ATALK) [N/y/m/?]
DECnet Support (CONFIG_DECNET) [N/y/m/?]
802.1d Ethernet Bridging (CONFIG_BRIDGE) [N/y/m/?]
*
*** QoS and/or fair queuering**
*
QoS and/or fair queuring (EXPERIMENTAL) (CONFIG_NET_SCHED) [N/y/?]
*
*** Telephony Support**
*
Linux telephony support (CONFIG_PHONE) [N/y/m/?]
*
*** ATA/IDE/MFM/RLL support**
*
ATA/IDE/MFM/RLL support (CONFIG_IDE) [Y/m/n/?] **m**
*
*** IDE, ATA and ATAPI Block devices**
*
Enhanced IDE/MFM/RLL disk/cdrom/tape/floppy support (CONFIG_BLK_DEV_IDE) [M/n/?]
*
* Please see Documentation/ide.txt for help/info on IDE drives
*
  Use old disk-only driver on primary interface (CONFIG_BLK_DEV_HD_IDE) [N/y/?]
  Include IDE/ATA-2 DISK support (CONFIG_BLK_DEV_IDEDISK) [M/n/?]
    Use multi-mode by default (CONFIG_IDEDISK_MULTI_MODE) [N/y/?]
  Include IDE/ATAPI CDROM support (CONFIG_BLK_DEV_IDECD) [M/n/?]
  Include IDE/ATAPI TAPE support (CONFIG_BLK_DEV_IDETAPE) [N/y/m/?]
  Include IDE/ATAPI FLOPPY support (CONFIG_BLK_DEV_IDEFLOPPY) [N/y/m/?]
  SCSI emulation support (CONFIG_BLK_DEV_IDESCSI) [N/y/m/?]
*
*** IDE chipset support/bugfixes**
*
  CMD640 chipset bugfix/support (CONFIG_BLK_DEV_CMD640) [Y/n/?] **n**
  RZ1000 chipset bugfix/support (CONFIG_BLK_DEV_RZ1000) [Y/n/?] **n**
  Generic PCI IDE chipset support (CONFIG_BLK_DEV_IDEPCI) [Y/n/?]
    Sharing PCI IDE interrupts support (CONFIG_IDEPCI_SHARE_IRQ) [Y/n/?]
    Generic PCI bus-master DMA support (CONFIG_BLK_DEV_IDEDMA_PCI) [N/y/?] **y**
    Boot off-board chipsets first support (CONFIG_BLK_DEV_OFFBOARD) [N/y/?]

```
    Use PCI DMA by default when available (CONFIG_IDEDMA_PCI_AUTO) [N/y/?] y
  AEC62XX chipset support (CONFIG_BLK_DEV_AEC62XX) [N/y/?]
  ALI M15x3 chipset support (CONFIG_BLK_DEV_ALI15X3) [N/y/?]
  AMD Viper support (CONFIG_BLK_DEV_AMD7409) [N/y/?]
  CMD64X chipset support (CONFIG_BLK_DEV_CMD64X) [N/y/?]
  CY82C693 chipset support (CONFIG_BLK_DEV_CY82C693) [N/y/?]
  Cyrix CS5530 MediaGX chipset support (CONFIG_BLK_DEV_CS5530) [N/y/?]
  HPT34X chipset support (CONFIG_BLK_DEV_HPT34X) [N/y/?]
  HPT366 chipset support (CONFIG_BLK_DEV_HPT366) [N/y/?]
  Intel PIIXn chipsets support (CONFIG_BLK_DEV_PIIX) [N/y/?]
  NS87415 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_NS87415) [N/y/?]
  PROMISE PDC20246/PDC20262/PDC20267 support (CONFIG_BLK_DEV_PDC202XX) [N/y/?]
  ServerWorks OSB4 chipset support (CONFIG_BLK_DEV_OSB4) [N/y/?]
  SiS5513 chipset support (CONFIG_BLK_DEV_SIS5513) [N/y/?]
  SLC90E66 chipset support (CONFIG_BLK_DEV_SLC90E66) [N/y/?]
  Tekram TRM290 chipset support (EXPERIMENTAL) (CONFIG_BLK_DEV_TRM290) [N/y/?]
  VIA82CXXX chipset support (CONFIG_BLK_DEV_VIA82CXXX) [N/y/?]
Other IDE chipset support (CONFIG_IDE_CHIPSETS) [N/y/?]
IGNORE word93 Validation BITS (CONFIG_IDEDMA_IVB) [N/y/?] (NEW)
*
* SCSI support
*
SCSI support (CONFIG_SCSI) [Y/m/n/?]
*
* SCSI support type (disk, tape, CD-ROM)
*
SCSI disk support (CONFIG_BLK_DEV_SD) [Y/m/n/?]
Maximum number of SCSI disks that can be loaded as modules (CONFIG_SD_EXTRA_DEVS) [40]
SCSI tape support (CONFIG_CHR_DEV_ST) [N/y/m/?]
SCSI OnStream SC-x0 tape support (CONFIG_CHR_DEV_OSST) [N/y/m/?]
SCSI CD-ROM support (CONFIG_BLK_DEV_SR) [N/y/m/?]
SCSI generic support (CONFIG_CHR_DEV_SG) [N/y/m/?]
*
* Some SCSI devices (e.g. CD jukebox) support multiple LUNs
*
Enable extra checks in new queueing code (CONFIG_SCSI_DEBUG_QUEUES) [Y/n/?] n
Probe all LUNs on each SCSI device (CONFIG_SCSI_MULTI_LUN) [Y/n/?] n
Verbose SCSI error reporting (kernel size +=12K) (CONFIG_SCSI_CONSTANTS) [Y/n/?] n
SCSI logging facility (CONFIG_SCSI_LOGGING) [N/y/?]
*
* SCSI low-level drivers
*
3ware Hardware ATA-RAID support (CONFIG_BLK_DEV_3W_XXXX_RAID) [N/y/m/?]
7000FASST SCSI support (CONFIG_SCSI_7000FASST) [N/y/m/?]
ACARD SCSI support (CONFIG_SCSI_ACARD) [N/y/m/?]
Adaptec AHA152X/2825 support (CONFIG_SCSI_AHA152X) [N/y/m/?]
Adaptec AHA1542 support (CONFIG_SCSI_AHA1542) [N/y/m/?]
Adaptec AHA1740 support (CONFIG_SCSI_AHA1740) [N/y/m/?]
Adaptec AIC7xxx support (CONFIG_SCSI_AIC7XXX) [N/y/m/?] y
  Enable Tagged Command Queueing (TCQ) by default (CONFIG_AIC7XXX_TCQ_ON_BY_DEFAULT)
[N/y/?] (NEW) y
  Maximum number of TCQ commands per device (CONFIG_AIC7XXX_CMDS_PER_DEVICE) [8] (NEW)
  Collect statistics to report in /proc (CONFIG_AIC7XXX_PROC_STATS) [N/y/?] (NEW)
  Delay in seconds after SCSI bus reset (CONFIG_AIC7XXX_RESET_DELAY) [5] (NEW)
AdvanSys SCSI support (CONFIG_SCSI_ADVANSYS) [N/y/m/?]
Always IN2000 SCSI support (CONFIG_SCSI_IN2000) [N/y/m/?]
AM53/79C974 PCI SCSI support (CONFIG_SCSI_AM53C974) [N/y/m/?]
AMI MegaRAID support (CONFIG_SCSI_MEGARAID) [N/y/m/?]
BusLogic SCSI support (CONFIG_SCSI_BUSLOGIC) [N/y/m/?]
Compaq Fibre Channel 64-bit/66Mhz HBA support (CONFIG_SCSI_CPQFCTS) [N/y/m/?]
DMX3191D SCSI support (CONFIG_SCSI_DMX3191D) [N/y/m/?]
DTC3180/3280 SCSI support (CONFIG_SCSI_DTC3280) [N/y/m/?]
```

EATA ISA/EISA/PCI (DPT and generic EATA/DMA-compliant boards) support (CONFIG_SCSI_EATA) [N/y/m/?]
EATA-DMA [Obsolete] (DPT, NEC, AT&T, SNI, AST, Olivetti, Alphatronix) support (CONFIG_SCSI_EATA_DMA) [N/y/m/?]
EATA-PIO (old DPT PM2001, PM2012A) support (CONFIG_SCSI_EATA_PIO) [N/y/m/?]
Future Domain 16xx SCSI/AHA-2920A support (CONFIG_SCSI_FUTURE_DOMAIN) [N/y/m/?]
GDT SCSI Disk Array Controller support (CONFIG_SCSI_GDTH) [N/y/m/?]
Generic NCR5380/53c400 SCSI support (CONFIG_SCSI_GENERIC_NCR5380) [N/y/m/?]
IBM ServeRAID support (CONFIG_SCSI_IPS) [N/y/m/?]
Initio 9100U(W) support (CONFIG_SCSI_INITIO) [N/y/m/?]
Initio INI-A100U2W support (CONFIG_SCSI_INIA100) [N/y/m/?]
NCR53c406a SCSI support (CONFIG_SCSI_NCR53C406A) [N/y/m/?]
NCR53c7,8xx SCSI support (CONFIG_SCSI_NCR53C7xx) [N/y/m/?]
NCR53C8XX SCSI support (CONFIG_SCSI_NCR53C8XX) [N/y/m/?]
SYM53C8XX SCSI support (CONFIG_SCSI_SYM53C8XX) [Y/m/n/?] **n**
PAS16 SCSI support (CONFIG_SCSI_PAS16) [N/y/m/?]
PCI2000 support (CONFIG_SCSI_PCI2000) [N/y/m/?]
PCI2220i support (CONFIG_SCSI_PCI2220I) [N/y/m/?]
PSI240i support (CONFIG_SCSI_PSI240I) [N/y/m/?]
Qlogic FAS SCSI support (CONFIG_SCSI_QLOGIC_FAS) [N/y/m/?]
Qlogic ISP SCSI support (CONFIG_SCSI_QLOGIC_ISP) [N/y/m/?]
Qlogic ISP FC SCSI support (CONFIG_SCSI_QLOGIC_FC) [N/y/m/?]
Qlogic QLA 1280 SCSI support (CONFIG_SCSI_QLOGIC_1280) [N/y/m/?]
Seagate ST-02 and Future Domain TMC-8xx SCSI support (CONFIG_SCSI_SEAGATE) [N/y/m/?]
Simple 53c710 SCSI support (Compaq, NCR machines) (CONFIG_SCSI_SIM710) [N/y/m/?]
Symbios 53c416 SCSI support (CONFIG_SCSI_SYM53C416) [N/y/m/?]
Tekram DC390(T) and Am53/79C974 SCSI support (CONFIG_SCSI_DC390T) [N/y/m/?]
Trantor T128/T128F/T228 SCSI support (CONFIG_SCSI_T128) [N/y/m/?]
UltraStor 14F/34F support (CONFIG_SCSI_U14_34F) [N/y/m/?]
UltraStor SCSI support (CONFIG_SCSI_ULTRASTOR) [N/y/m/?]
*
*** I2O device support**
*
I2O support (CONFIG_I2O) [N/y/m/?]
*
*** Network device support**
*
Network device support (CONFIG_NETDEVICES) [Y/n/?]
*
*** ARCnet devices**
*
ARCnet support (CONFIG_ARCNET) [N/y/m/?]
Dummy net driver support (CONFIG_DUMMY) [M/n/y/?]
Bonding driver support (CONFIG_BONDING) [N/y/m/?]
EQL (serial line load balancing) support (CONFIG_EQUALIZER) [N/y/m/?]
Universal TUN/TAP device driver support (CONFIG_TUN) [N/y/m/?]
General Instruments Surfboard 1000 (CONFIG_NET_SB1000) [N/y/m/?]
*
*** Ethernet (10 or 100Mbit)**
*
Ethernet (10 or 100Mbit) (CONFIG_NET_ETHERNET) [Y/n/?]
  3COM cards (CONFIG_NET_VENDOR_3COM) [N/y/?]
  AMD LANCE and PCnet (AT1500 and NE2100) support (CONFIG_LANCE) [N/y/m/?]
  Western Digital/SMC cards (CONFIG_NET_VENDOR_SMC) [N/y/?]
  Racal-Interlan (Micom) NI cards (CONFIG_NET_VENDOR_RACAL) [N/y/?]
  DEPCA, DE10x, DE200, DE201, DE202, DE422 support (CONFIG_DEPCA) [N/y/m/?]
  HP 10/100VG PCLAN (ISA, EISA, PCI) support (CONFIG_HP100) [N/y/m/?]
  Other ISA cards (CONFIG_NET_ISA) [N/y/?]
  EISA, VLB, PCI and on board controllers (CONFIG_NET_PCI) [Y/n/?]
    AMD PCnet32 PCI support (CONFIG_PCNET32) [N/y/m/?]
    Apricot Xen-II on board Ethernet (CONFIG_APRICOT) [N/y/m/?]
    CS89x0 support (CONFIG_CS89x0) [N/y/m/?]

```
    DECchip Tulip (dc21x4x) PCI support (CONFIG_TULIP) [N/y/m/?]
    Generic DECchip & DIGITAL EtherWORKS PCI/EISA (CONFIG_DE4X5) [N/y/m/?]
    Digi Intl. RightSwitch SE-X support (CONFIG_DGRS) [N/y/m/?]
    EtherExpressPro/100 support (CONFIG_EEPRO100) [Y/m/n/?]
    National Semiconductor DP83810 series PCI Ethernet support (CONFIG_NATSEMI) [N/y/m/?]
    PCI NE2000 and clones support (see help) (CONFIG_NE2K_PCI) [N/y/m/?]
    RealTek RTL-8139 PCI Fast Ethernet Adapter support (CONFIG_8139TOO) [N/y/m/?]
    SiS 900/7016 PCI Fast Ethernet Adapter support (CONFIG_SIS900) [N/y/m/?]
    SMC EtherPower II (CONFIG_EPIC100) [N/y/m/?]
    Sundance Alta support (CONFIG_SUNDANCE) [N/y/m/?]
    TI ThunderLAN support (CONFIG_TLAN) [N/y/m/?]
    VIA Rhine support (CONFIG_VIA_RHINE) [N/y/m/?]
    Winbond W89c840 Ethernet support (CONFIG_WINBOND_840) [N/y/m/?]
    Sun Happy Meal 10/100baseT PCI support (CONFIG_HAPPYMEAL) [N/y/m/?]
  Pocket and portable adapters (CONFIG_NET_POCKET) [N/y/?]
*
* Ethernet (1000 Mbit)
*
Alteon AceNIC/3Com 3C985/NetGear GA620 Gigabit support (CONFIG_ACENIC) [N/y/m/?]
Packet Engines Hamachi GNIC-II support (CONFIG_HAMACHI) [N/y/m/?]
SysKonnect SK-98xx support (CONFIG_SK98LIN) [N/y/m/?]
FDDI driver support (CONFIG_FDDI) [N/y/?]
PPP (point-to-point protocol) support (CONFIG_PPP) [N/y/m/?]
SLIP (serial line) support (CONFIG_SLIP) [N/y/m/?]
*
* Wireless LAN (non-hamradio)
*
Wireless LAN (non-hamradio) (CONFIG_NET_RADIO) [N/y/?]
*
* Token Ring devices
*
Token Ring driver support (CONFIG_TR) [N/y/?]
Fibre Channel driver support (CONFIG_NET_FC) [N/y/?]
*
* Wan interfaces
*
Wan interfaces support (CONFIG_WAN) [N/y/?]
*
* Amateur Radio support
*
Amateur Radio support (CONFIG_HAMRADIO) [N/y/?]
*
* IrDA (infrared) support
*
IrDA subsystem support (CONFIG_IRDA) [N/y/m/?]
*
* ISDN subsystem
*
ISDN support (CONFIG_ISDN) [N/y/m/?]
*
* Old CD-ROM drivers (not SCSI, not IDE)
*
Support non-SCSI/IDE/ATAPI CDROM drives (CONFIG_CD_NO_IDESCSI) [N/y/?]
*
* Input core support
*
Input core support (CONFIG_INPUT) [N/y/m/?]
*
* Character devices
*
Virtual terminal (CONFIG_VT) [Y/n/?]
  Support for console on virtual terminal (CONFIG_VT_CONSOLE) [Y/n/?]
```

Standard/generic (8250/16550 and compatible UARTs) serial support (CONFIG_SERIAL) [Y/m/n/?]
  Support for console on serial port (CONFIG_SERIAL_CONSOLE) [N/y/?]
Extended dumb serial driver options (CONFIG_SERIAL_EXTENDED) [N/y/?]
Non-standard serial port support (CONFIG_SERIAL_NONSTANDARD) [N/y/?]
Unix98 PTY support (CONFIG_UNIX98_PTYS) [Y/n/?]
Maximum number of Unix98 PTYs in use (0-2048) (CONFIG_UNIX98_PTY_COUNT) [256] **128**
*
*** I2C support**
*
I2C support (CONFIG_I2C) [N/y/m/?]
*
*** Mice**
*
Bus Mouse Support (CONFIG_BUSMOUSE) [N/y/m/?]
Mouse Support (not serial and bus mice) (CONFIG_MOUSE) [Y/m/n/?]
PS/2 mouse (aka "auxiliary device") support (CONFIG_PSMOUSE) [Y/n/?]
C&T 82C710 mouse port support (as on TI Travelmate) (CONFIG_82C710_MOUSE) [N/y/m/?]
PC110 digitizer pad support (CONFIG_PC110_PAD) [N/y/m/?]
*
*** Joysticks**
*
*
* Input core support is needed for joysticks
*
QIC-02 tape support (CONFIG_QIC02_TAPE) [N/y/m/?]
*
*** Watchdog Cards**
*
Watchdog Timer Support (CONFIG_WATCHDOG) [N/y/?]
Intel i8x0 Random Number Generator support (CONFIG_INTEL_RNG) [N/y/m/?]
/dev/nvram support (CONFIG_NVRAM) [N/y/m/?]
Enhanced Real Time Clock Support (CONFIG_RTC) [N/y/m/?]
Double Talk PC internal speech card support (CONFIG_DTLK) [N/y/m/?]
Siemens R3964 line discipline (CONFIG_R3964) [N/y/m/?]
Applicom intelligent fieldbus card support (CONFIG_APPLICOM) [N/y/m/?]
*
*** Ftape, the floppy tape device driver**
*
Ftape (QIC-80/Travan) support (CONFIG_FTAPE) [N/y/m/?]
/dev/agpgart (AGP Support) (CONFIG_AGP) [Y/m/n/?] **n**
Direct Rendering Manager (XFree86 DRI support) (CONFIG_DRM) [Y/n/?] **n**
*
*** Multimedia devices**
*
Video For Linux (CONFIG_VIDEO_DEV) [N/y/m/?]
*
*** File systems**
*
Quota support (CONFIG_QUOTA) [N/y/?]
Kernel automounter support (CONFIG_AUTOFS_FS) [N/y/m/?]
Kernel automounter version 4 support (also supports v3) (CONFIG_AUTOFS4_FS) [Y/m/n/?] **n**
DOS FAT fs support (CONFIG_FAT_FS) [N/y/m/?]
Compressed ROM file system support (CONFIG_CRAMFS) [N/y/m/?]
Simple RAM-based file system support (CONFIG_RAMFS) [N/y/m/?]
ISO 9660 CDROM file system support (CONFIG_ISO9660_FS) [Y/m/n/?] **m**
  Microsoft Joliet CDROM extensions (CONFIG_JOLIET) [N/y/?]
Minix fs support (CONFIG_MINIX_FS) [N/y/m/?]
NTFS file system support (read only) (CONFIG_NTFS_FS) [N/y/m/?]
OS/2 HPFS file system support (CONFIG_HPFS_FS) [N/y/m/?]
/proc file system support (CONFIG_PROC_FS) [Y/n/?]
/dev/pts file system for Unix98 PTYs (CONFIG_DEVPTS_FS) [Y/n/?]
ROM file system support (CONFIG_ROMFS_FS) [N/y/m/?]

Second extended fs support (CONFIG_EXT2_FS) [Y/m/n/?]
System V and Coherent file system support (read only) (CONFIG_SYSV_FS) [N/y/m/?]
UDF file system support (read only) (CONFIG_UDF_FS) [N/y/m/?]
UFS file system support (read only) (CONFIG_UFS_FS) [N/y/m/?]
*
*** Network File Systems**
*
Coda file system support (advanced network fs) (CONFIG_CODA_FS) [N/y/m/?]
NFS file system support (CONFIG_NFS_FS) [Y/m/n/?] **n**
NFS server support (CONFIG_NFSD) [Y/m/n/?] **n**
SMB file system support (to mount Windows shares etc.) (CONFIG_SMB_FS) [N/y/m/?]
NCP file system support (to mount NetWare volumes) (CONFIG_NCP_FS) [N/y/m/?]
*
*** Partition Types**
*
Advanced partition selection (CONFIG_PARTITION_ADVANCED) [N/y/?]
*
*** Console drivers**
*
VGA text console (CONFIG_VGA_CONSOLE) [Y/n/?]
Video mode selection support (CONFIG_VIDEO_SELECT) [N/y/?]
*
*** Sound**
*
Sound card support (CONFIG_SOUND) [Y/m/n/?] **n**
*
(Security options will appear only if you are patched your kernel with the Openwall Project patch).
*** Security options**
*
Non-executable user stack area (CONFIG_SECURE_STACK) [Y]
Autodetect and emulate GCC trampolines (CONFIG_SECURE_STACK_SMART) [Y]
Restricted links in /tmp (CONFIG_SECURE_LINK) [Y] **n**
Restricted FIFOs in /tmp (CONFIG_SECURE_FIFO) [Y]
Restricted /proc (CONFIG_SECURE_PROC) [N] **y**
Special handling of fd 0, 1, and 2 (CONFIG_SECURE_FD_0_1_2) [Y]
Enforce RLIMIT_NPROC on execve(2) (CONFIG_SECURE_RLIMIT_NPROC) [Y]
Destroy shared memory segments not in use (CONFIG_SECURE_SHM) [N]
*****
*** USB support**
*
Support for USB (CONFIG_USB) [Y/m/n/?] **n**
*
*** Kernel hacking**
*
Magic SysRq key (CONFIG_MAGIC_SYSRQ) [N/y/?]

*** End of Linux kernel configuration.
*** Check the top-level Makefile for additional configuration.
*** Next, you must run 'make dep'.

WARNING: With the new kernel 2.4 and SCSI system you don't have the choice to configure a modularized kernel because of the option "Maximum number of SCSI disks that can be loaded as modules (CONFIG_SD_EXTRA_DEVS) [40]" which doesn't let us to compile it directly into the kernel.

If you want to enable IPTABLES support into the kernel, the iptables program must be installed first or you will receive error messages during kernel compilation. This is because when iptables support is enabled, the kernel will associate some part of the iptables program with it configuration. Therefore don't forget to install IPTABLES before configuring kernel with IPTABLES support. Finally the same warning is true for quota support into the kernel.

Finally, it is important to note that the kernel configuration part related to "IP: Netfilter Configuration" has been configured as loadable module in this example. This is because I want to show you a different kernel configuration than the first for monolithic kernel that you may have. With kernel 2.4.x generation, we have now the possibility to compile all "IP: Netfilter Configuration" options related to Masquerading and Forwarding support directly into the kernel. Therefore it is for you to decide how you want to configure this part of the kernel for your system, you can configure it as modules or compiled and included directly into the kernel.

Compiling the Kernel

This section applies to monolithic kernel and modularized kernel. Now, return to the /usr/src/linux directory (if you are not already in it). You need to compile the new kernel. You do so by using the following command:

- To compile the Kernel, use the following command:
  ```
  [root@deep linux]# make dep; make clean; make bzImage
  ```

This line contains three commands in one. The first one, **make dep**, actually takes your configuration and builds the corresponding dependency tree. This process determines what gets compiled and what doesn't. The next step, **make clean**, erases all previous traces of a compilation so as to avoid any mistakes in which the wrong version of a feature gets tied into the kernel. Finally, **make bzImage** does the full compilation of the kernel.

After the process is complete, the kernel is compressed and ready to be installed on your system. Before we can install the new kernel, we must know if we need to compile the corresponding modules. This is required ONLY if you said **yes** to "**Enable loadable module support (CONFIG_MODULES)**" and have compiled some options in the kernel configuration above as a module (See Modularized kernel configuration). In this case, you must execute the following commands:

- To compile the corresponding modules for your kernel, use the following commands:
  ```
  [root@deep linux]# make modules
  [root@deep linux]# make modules_install
  ```

WARNING: The **make modules** and **make modules_install** commands are required ONLY if you say **yes** to "**Enable loadable module support (CONFIG_MODULES)**" in your kernel configurations (See Modularized kernel configuration) because you want to build a **modularized kernel**.

Installing the Kernel

This section applies to monolithic kernel and modularized kernel. Ok, kernel has been configured, compiled and is now ready to be installed in your system. Below are the required steps to install all the necessary kernel components into your server.

Step 1
Copy the file /usr/src/linux/arch/i386/boot/bzImage from the kernel source tree to the /boot directory, and give it an appropriate new name.

- To copy the **bzImage** file to the /boot directory, use the following commands:
```
[root@deep /]# cd /usr/src/linux/ (if you are not already in it)
[root@deep linux]# cp arch/i386/boot/bzImage /boot/vmlinuz-2.4.5
```

NOTE: An appropriate or recommended new name is something like **vmlinuz-2.4.5**, this is important if you want a new rescue floppy or emergency boot floppy using the **mkbootdisk** tool that require some specific needs like for example: vmlinuz-2.4.5 instead of vmlinuz-2.4.5.a

Step 2
A new System.map file is generated when you compile a kernel, and is a list of all the addresses in that kernel and their corresponding symbols. Every time that you create a new kernel, such a file System.map is created and saved in /usr/src/linux. In it you will find information about offsets within kernel that are required by the modules if you have compiled the kernel as modularized. It's a text file, which is read by a few programs (like ps) to do address <-> symbol translation, and which you need if you ever get an Oops.

Certain commands, like klog, ps, and lsof, use the System.map file to get the name of kernel symbols. Without it some commands like lsof will complain that they can't find a System.map file to match the currently booted kernel.

Copy the file /usr/src/linux/System.map from the kernel source tree to the /boot directory, and give it an appropriate new name.

- To copy the **System.map** file to the /boot directory, use the following commands:
```
[root@deep /]# cd /usr/src/linux/ (if you are not already in it)
[root@deep linux]# cp System.map /boot/System.map-2.4.5
```

Step 3
Move into the /boot directory and rebuild the links **vmlinuz** and **System.map**.

- To rebuild the **vmlinuz** and **System.map** files, use the following commands:
```
[root@deep linux]# cd /boot/
[root@deep /boot]# ln -fs vmlinuz-2.4.5 vmlinuz
[root@deep /boot]# ln -fs System.map-2.4.5 System.map
```

We must rebuild the links of **vmlinuz** and **System.map** to point them to the new installed kernel version. Without the new links LILO program will look, by default, for the old version of your Linux kernel.

Step 4
Remove obsolete and unnecessary files under the /boot directory to increase disk space:

- To remove obsolete and unnecessary files under the /boot directory, use commands:
```
[root@deep /]# cd /boot/ (if you are not already in it)
[root@deep /boot]# rm -f module-info
[root@deep /boot]# rm -f initrd-2.4.x.img
```

The **module-info** is a link, which points to the old modules directory of your original kernel. Since we have installed a brand new kernel, we don't need to keep this broken link.

The **initrd-2.4.x.img** is a file that contains an initial RAM disk image that serves as a system before the disk is available. This file is only available and is installed from the Linux initial setup installation if your system has a SCSI adapter present and only if your system has a SCSI adapter. If we use and have a SCSI system, the required driver now will be incorporated into our new Linux kernel since we have build it by answering **Yes** to the question related to our SCSI model during the configuration of the kernel, so we can remove this file (**initrd-2.4.x.img**) safely.

Step 5
Create a new Linux kernel directory that will handle all header files related to Linux kernel for future compilation of other programs on your system.

Recall, we had created two symlinks under the /usr/include directory that point to the Linux kernel header files to be able to compile it without receiving error and also be able to compile future programs. The /usr/include directory is where all the header files for your Linux system are kept for reference and dependencies when you compile and install new programs.

The **asm**, and **linux** links are used when programs need to know some functions which are compile-time specific to the kernel installed on your system. Programs call other headers as well in the /usr/include directory when they must know specific information, dependencies, etc of your system.

- To create a new Linux kernel directory to handle all header files, use the commands:
```
[root@deep /]# mkdir -p /usr/src/linux-2.4.5/include
[root@deep /]# cd /usr/src/linux/
[root@deep linux]# cp -r include/asm-generic ../linux-2.4.5/include/
[root@deep linux]# cp -r include/asm-i386 ../linux-2.4.5/include/
[root@deep linux]# cp -r include/linux ../linux-2.4.5/include/
[root@deep linux]# cd ../
[root@deep src]# rm -rf /usr/src/linux
[root@deep src]# cd /usr/src/ (to be sure that we are into the src directory)
[root@deep src]# ln -s /usr/src/linux-2.4.5 linux
```

First we create a new directory named "**linux-2.4.5**" based on the version of the kernel we have installed for easy interpretation, then we copy directories asm-generic, asm-i386, and linux from /usr/src/linux/include to our new location /usr/src/linux-2.4.5/include.

After we remove the entire source directory where we had compiled the new kernel, we create a new symbolic link named "**linux**" under /usr/src that points to our new /usr/src/linux-2.4.5 directory. With these steps, future compiled programs will know where to look for headers related to the kernel on your server.

NOTE: This step will allow us to gain space on our hard drive and will reduce the risk of security. The Linux kernel source directory handles a lot files and is about **94M** in size when uncompressed. With the procedure described above, our Linux kernel directory began approximately **4M** in size so we save **90MB** for the same functionalities.

Step 6
Finally, you need to edit the `/etc/lilo.conf` file to make your new kernel one of the boot time options:

Edit the **`lilo.conf`** file (`vi /etc/lilo.conf`) and make the appropriate change on the line that read "`image=/boot/vmlinuz-x.x.x`".

```
[root@deep /]# vi /etc/lilo.conf

boot=/dev/sda
map=/boot/map
install=/boot/boot.b
timeout=00
default=linux
restricted
password=somepasswd

image=/boot/vmlinuz
 label=linux
 read-only
 root=/dev/sda6
```

WARNING: I recommend you to put on the line "**`image=/boot/vmlinuz-x.x.x`**" only the word "**`vmlinuz`**"; this allow us to not have to edit the `lilo.conf` file each time we upgrade our kernel. The word "**`vmlinuz`**" always point to your latest kernel image.

Also, for `SCSI` system only, don't forget to remove the line that read "**`initrd=/boot/initrd-x.x.x.img`**" in the `lilo.conf` file, since this line is not necessary now since we have built our `SCSI` system directly into the kernel by answering **Yes** to the question related to our `SCSI` model during configuration of the kernel.

Once the necessary modifications has been made into the `/etc/lilo.conf` file as shown above, we update our `lilo.conf` file for the change to take effect with the following command:

```
[root@deep /]# /sbin/lilo -v
LILO version 21.4-4, copyright © 1992-1998 Wernerr Almesberger
'lba32' extentions copyright © 1999,2000 John Coffman

Reading boot sector from /dev/sda
had : ATAPI 32X CD-ROM drive, 128kB Cache
Merging with /boot/boot.b
Mapping message file /boot/message
Boot image : /boot/vmlinuz
Added linux *
/boot/boot.0800 exists - no backup copy made.
Writing boot sector.
```

Reconfiguring `/etc/modules.conf` file

This section applies only if you chose to install a `modularized kernel` in your system. The `/etc/modules.conf` file represents the (optional) configuration file for loading some kernel modules in your system. It is used to modify the behavior of `modprobe` and `depmod` programs. This file consists of a set of lines with different parameters. It is important after each upgrade of a `modularized kernel` to verify if all information and parameters contained inside it, are valid and correct.

All the contents of the `/etc/modules.conf` file apply only for systems where the kernel has been configured with modules (`modularized kernel`). So if you have recompiled your new kernel with some new options as modules or if you have removed some modules from it, it is important to update or remove the **`modules.conf`** file to reflect the changes and eliminate possible error message during booting.

As an example, the following is the content of the **`modules.conf`** file on my system. Linux has added these parameters automatically, depending of the system hardware during the primary install stage of the operating system.

```
alias scsi_hostadapter aic7xxx
alias eth0 eepro100
alias eth1 eepro100
alias parport_lowlevel parport_pc
alias usb-controller uhci
```

One important use of the `modules.conf` file is the possibility of using the "`alias`" directive to give alias names to modules and link object files to a module.

After recompilation of the kernel, and depending of how we have answered the different kernel questions during kernel configuration, it may be possible that we need to make some adjustments to the default parameters, especially if we have answered **`yes`** during kernel configuration to some devices available in our system, like network cards and `SCSI` adapters.

If the configuration file `/etc/modules.conf` is missing, or if any directive is not overridden, the default will be to look under `/lib/modules` directory containing modules compiled for the current release of the kernel. Therefore, we can remove the `/etc/modules.conf` file from the system and let the `modprobe` and `depmod` programs manage all existing modules for us.

To summarize, you can:

1) Keep the **`modules.conf`** file; only kernel options which you have answered **m** during kernel configuration time (of course only if these modules did exist into `modules.conf`). Any kernel options where you have answered **yes** or **no** will not appears into the **`modules.conf`** file.

2) Or remove the `/etc/modules.conf` file from your system and let `modprobe` and `depmod` programs manage all existing modules for you. On a server environment, I prefer to use this choice.

Delete programs, edit files pertaining to modules

This section applies only if you chose to install a `monolithic kernel` in your system. By default when you install Linux for the first time (like we did), the kernel is built as a `modularized kernel`. This means that each device or function we need exists as a module and is controlled by the Kernel Daemon program named **`kmod`**. `kmod` automatically loads some modules and functions into memory as they are needed, and unloads them when they're no longer being used.

Step 1

`kmod` and other module management programs included in the **`modutils`** RPM package use the `modules.conf` file located in the `/etc` directory to know for example which Ethernet card you have, if your Ethernet card requires special configuration and so on. If we don't use any modules in our new compiled kernel because we have compiled the kernel as `monolithic kernel` and ONLY in this case, we can remove the `modules.conf` file and uninstall completely the **`modutils`** RPM package.

- To remove the **`modules.conf`** file, use the following command:
  `[root@deep /]#` **`rm -f /etc/modules.conf`**

- To uninstall the **`modutils`** package, use the following command:
  `[root@deep /]#` **`rpm -e --nodeps modutils`**

Step 2

One last thing to do is to edit the file **`devfsd.conf`** and comment out the line related to module autoloading by inserting a "**`#`**" at the beginning of the line.

- Edit the **`devfsd.conf`** file (`vi /etc/devfsd.conf`), and change the line:

  ```
  LOOKUP    .*  MODLOAD
  ```

  To read:

  ```
  #LOOKUP   .*  MODLOAD
  ```

Step 3

Finaly, it is important to remove the file named "`modules.devfs`" under `/etc` since it is no longer needed for a monolithic kernel.

- To remove the **`modules.devfs`** file, use the following command:
  `[root@deep /]# rm -f /etc/modules.devfs`

> **WARNING:** Once again, the above ("Delete program, file and lines related to modules") is required only if you said **no** to "`Enable loadable module support (CONFIG_MODULES)`" in your kernel configuration because you have decided to build a `monolithic kernel`.

Remounting the /boot partition of Linux as read-only

This section applies to monolithic kernel and modularized kernel. Once our new kernel has been installed in the system, we can now remount the /boot partition of Linux as read-only to eliminate possible problems that someone might try to change or modify vital files inside it. To remount the /boot directory as read-only, follow the simple steps below.

Step1

- Edit the **fstab** file (vi /etc/fstab) and change the line:

```
LABEL=/boot       /boot       ext2    defaults        1 2
```

To read:

```
LABEL=/boot       /boot       ext2    defaults,ro     1 2
```

Step 2
Make the Linux system aware of the modification you have made to the /etc/fstab file.

- This can be accomplished with the following command:

```
[root@deep /]# mount /boot -oremount
```

- Then test your results with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root /   ext2    rw 0 0
/proc /proc   proc    rw 0 0
/dev/sda1     /boot       ext2    ro 0 0
/dev/sda10    /cache      ext2    rw,nodev 0 0
/dev/sda9     /chroot     ext2    rw 0 0
/dev/sda8     /home       ext2    rw,nosuid 0 0
/dev/sda13    /tmp        ext2    rw,noexec,nosuid 0 0
/dev/sda7     /usr        ext2    rw 0 0
/dev/sda11    /var        ext2    rw 0 0
/dev/sda12    /var/lib    ext2    rw 0 0
none /dev/pts    devpts  rw 0 0
```

If you see something like: /dev/sda1 /boot ext2 **ro** 0 0, congratulations!

Rebooting your system to load the new kernel

Whether you have installed a new monolithic kernel where codes and drivers are compiled into the kernel and are always loaded or a modularized kernel where some segment of codes are compiled into the kernel as a module and loaded on demand, it is time to **Reboot** your system and test your results.

- To reboot your Linux system, use the following command:

```
[root@deep /]# reboot
```

When the system is rebooted and you are logged in, verify the new version of your kernel with the following command:

- To verify the version of your new kernel, use the following command:

```
[root@deep /]# uname -a
Linux deep 2.4.5 #1 Sat Mar 24 09:38:35 EDT 2001 i686 unknown
```

Congratulations!

NOTE ABOUT SYSTEM SIZE: After recompilation of the kernel and installation of all packages necessary to make compilation on the system plus the update of required RPM packages, our install size of Linux is now **162MB**. Note that it can be smaller than 162 MB if we don't install compilers packages and use another computer to develop and compile tarballs.

Making a new rescue floppy for Modularized Kernel

This section applies only if you chose to install a `modularized kernel` in your system. After the reboot, you should have now a system with an upgraded kernel. Therefore, it's time is to make a new rescue floppy with the new kernel in case of emergencies. To do this, follow the simple step below:

- Login as root, and insert a new floppy, then execute the following command:

```
[root@deep /]# mkbootdisk --device /dev/fd0H1440 2.4.5
Insert a disk in /dev/fd0. Any information on the disk will be lost.
Press <Enter> to continue or ^C to abort:
```

WARNING: The **`mkbootdisk`** program runs only on `modularized Kernel`. So you can't use it on a `monolithic Kernel`; instead create an emergency boot floppy as shown below.

Making a emergency boot floppy disk for Monolithic Kernel

This section applies only if you chose to install a `monolithic kernel` in your system. Because it is possible to create a rescue floppy only on `modularized kernel`, we must find another way to boot our Linux system for a `monolithic kernel` if the Linux kernel on the hard disk is damaged. This is possible with a Linux emergency boot floppy disk. You should create it immediately after you successfully start your system and log in as root.

- To create the emergency boot floppy disk, follow these steps:

1. Insert a floppy disk and format it with the following command:

```
[root@deep /]# fdformat /dev/fd0H1440
Double-sided, 80 tracks, 18 sec/track. Total capacity 1440 kB.
Formatting ... done
Verifying ... done
```

2. Copy the file "**`vmlinuz`**" from the `/boot` directory to the floppy disk:

```
[root@deep /]# cp /boot/vmlinuz /dev/fd0H1440
cp: overwrite `/dev/fd0H1440'? y
```

The **`vmlinuz`** file is a symbolic link that points to the real Linux kernel.

3. Determine the kernel's root device with the following command:

```
[root@deep /]# rdev
/dev/sda6 /
```

The kernel's root device is the disk partition where the root file system is located. In this example, the root device is `/dev/sda6`; the device name may be different on your system.

4. Set the kernel's root device with the following command:
   `[root@deep /]# rdev /dev/fd0H1440 /dev/sda6`

   To set the kernel's root device, use the device reported by the "**rdev**" command utility in the previous step.

5. Mark the root device as read-only with the following command:
   `[root@deep /]# rdev -R /dev/fd0H1440 1`

   This causes Linux to initially mount the root file system as read-only. By setting the root device as read-only, you avoid several warnings and error messages.

6. Now put the boot floppy in the drive A: and reboot your system with the following command:
   `[root@deep /]# reboot`

Because the **mkbootdisk** program is required only when you have a `modularized kernel` installed in your Linux system, we can remove the unneeded **mkbootdisk** package from the system.

- To uninstall the **mkbootdisk** utility, use the following command:
  `[root@deep /]# rpm –e mkbootdisk`

Optimizing `Kernel`

This section deals with actions we can make to improve and tighten performance of the Linux `Kernel`. Note that we refer to the features available within the base installed Linux system.

`/proc/sys/vm`: The virtual memory subsystem of Linux

All parameters described later in this chaper reside under the `/proc/sys/vm` directory of the server and can be used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel. Be very careful when attempting this. You can optimize your system, but you can also cause it to crash. Since every system is different, you'll probably want some control over these pieces of the system.

Finally, these are advanced setting and if you don't understand them, then don't try to play in this area or try to use all examples below directly in your systems. Remember that all systems are different and require different setting and customization. The majority of the following hacks will work fine on a server with >= at 512MB of RAM or at minimum of 256MB of RAM. Below this amount of memory, nothing is guaranteed and the default setting will just be fine for you.

Below I show you parameters that can be optimized for the system. All suggestions I make in this section are valid for every kind of server. The only difference depends on the amount of RAM your machines have and this is where settings will change.

```
                                    | - bdflush
                                    | - buffermem
                                    | - freepages
/proc/sys/vm ---------------------- | - kswapd
                                    | - avercommit_memory
                                    | - page-cluster
                                    | - pagecache
                                    | - pagetable_cache
```

The above figure shows a snapshot of `/proc/sys/vm` directory on a Red Hat Linux system running kernel version 2.4. Please note that this picture may look different on your system.

The `bdflush` parameters

The `bdflush` file is closely related to the operation of the virtual memory (VM) subsystem of the Linux kernel and has a little influence on disk usage. This file `/proc/sys/vm/bdflush` controls the operation of the `bdflush` kernel daemon. We generally tune this file to improve file system performance. By changing some values from the defaults shown below, the system seems more responsive; e.g. it waits a little more to write to disk and thus avoids some disk access contention.

The `bdflush` parameters currently contains 9 integer values, of which 6 are actually used by the kernel 2.4 generation. The default setup for the `bdflush` parameters under Red Hat Linux is:
`"30 64 64 256 500 3000 60 0 0"`

Step 1
To change the values of **`bdflush`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (vi `/etc/sysctl.conf)` and add the following line:

```
# Improve file system performance
vm.bdflush = 100 1200 128 512 500 6000 500 0 0
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters             [OK]
Bringing up interface lo               [OK]
Bringing up interface eth0             [OK]
Bringing up interface eth1             [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w vm.bdflush="100 1200 128 512 500 6000 500 0 0"
```

In our example above, according to the `/usr/src/linux/Documentation/sysctl/vm.txt` file, the first parameter (**100 %**) governs the maximum number of dirty buffers in the buffer cache. Dirty means that the contents of the buffer still have to be written to disk (as opposed to a clean buffer, which can just be forgotten about). Setting this to a high value means that Linux can delay disk writes for a long time, but it also means that it will have to do a lot of I/O at once when memory becomes short. A low value will spread out disk I/O more evenly.

The second parameter (**1200**) (`ndirty`) gives the maximum number of dirty buffers that `bdflush` can write to the disk in one time. A high value will mean delayed, bursty I/O, while a small value can lead to memory shortage when `bdflush` isn't woken up often enough.

The third parameter (**128**) (`nrefill`) is the number of buffers that `bdflush` will add to the list of free buffers when refill_freelist() is called. It is necessary to allocate free buffers beforehand, since the buffers often are of a different size than memory pages and some bookkeeping needs to be done beforehand. The higher the number, the more memory will be wasted and the less often refill_freelist() will need to run.

When refill_freelist() (**512**) comes across more than `nref_dirt` dirty buffers, it will wake up `bdflush`.

Finally, the `age_buffer` (**50*HZ**) and `age_super` parameters (**5*HZ**) govern the maximum time Linux waits before writing out a dirty buffer to disk. The value is expressed in jiffies (clockticks); the number of jiffies per second is 100. `Age_buffer` is the maximum age for data blocks, while `age_super` is for file system metadata.

The fifth (**500**) and last two parameters (**0** and **0**) are unused by the system so we don't need to change the default ones.

> **NOTE:** Look at `/usr/src/linux/Documentation/sysctl/vm.txt` for more information on how to improve kernel parameters related to virtual memory. Also note that `bdflush` features parameters may vary from kernel version to another.

The `buffermem` parameters

The `buffermem` file is also closely related to the operation of the virtual memory (VM) subsystem of the Linux kernel. The value in this file `/proc/sys/vm/buffermem` controls how much memory should be used for buffer memory (in percentage). It is important to note that the percentage is calculated as a percentage of total system memory.

The `buffermem` parameters currently contains 3 integer values, of which 1 is actually used by the kernel. The default setup for the `buffermem` parameters under Red Hat Linux is:

```
"2 10 60"
```

Step 1
To change the values of **`buffermem`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Improve virtual memory performance
vm.buffermem = 80 10 60
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all networks devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w vm.buffermem="80 10 60"
```

In our example above, according to the `/usr/src/linux/Documentation/sysctl/vm.txt` file, the first parameter (**80 %**) means to use a minimum of 80 percent of memory for the buffer cache; the minimum percentage of memory that should be spent on buffer memory.

The last two parameters (**10** and **60**) are unused by the system so we don't need to change the defaults.

Depending of the amount of RAM you have in the server the value of **80%** may vary. When your server is highly loaded and when all applications are used, you know in detail how much memory is required and used by the system. **80 %** for the `buffermem` parameters seems to be too much for systems under 256 MB of RAM. Doing a "`# free -m`" command on the prompt your system will display amount of free and used memory in the system.

Once you have executed this command "`# free -m`", check for `-/+ buffers/cache:` values and get the one related to the minimal (`-`) to set your value for `buffermem`.

As an example for 128 MB of RAM:

```
128 * 80% = 102.4 MB
128 - 102.4 = 25.6 MB

[root@deep /]# free -m
                total       used       free     shared    buffers     cached
Mem:            124         121        3        30        43          48
-/+ buffers/cache:          29         95
Swap:           128         2          126
```

The result shows us that the `-/+ buffers/cache:` need **29 MB** at minimum to run the system properly and with 128 MB of RAM set at 80% we have only **25.6 MB** available. Hmmm! problem, I guess. so we go back to the calculator again and do this:

To solve the problem:

```
128 * 70% = 89.6
128 - 89.6 = 38.4 MB
```

Well solved!

NOTE: Look at `/usr/src/linux/Documentation/sysctl/vm.txt` for more information on how to improve kernel parameters related to virtual memory. Also note that `buffermem` features parameters may vary from kernel version to another.

The `freepages` parameter

The `freepages` file `/proc/sys/vm/freepages` defines the values in the struct freepages. According to kernel documentation, that struct contains three members: `min`, `low` and `high`, which can be configured to tune the operation of the virtual memory (VM) subsystem of the Linux kernel.

Usually we increase the first member (`min`) to something reasonable like 47.875 for every 32MB of RAM we have and multiply by 2 the result to get the value of member (`low`) and by 3 for the member (`high`) related again to the value of the first member (`mim`): i.e. for a machine with 256 MB of RAM, set it to 383 766 1149 (256/32=8 8*47.875=383 383*2=766 383*3=1149).

One important note here: If the `buffermem` parameters have been changed as shown above, then you don't need to do anything here since the values of `freepages` will be automatically adjusted related to the `buffermem` parameters values.

The default setup for the `freepages` parameter under Red Hat Linux is:
`"2 15 75"`

Step 1
To change the values of **`freepages`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Improve and better control swapping into the system
vm.freepages = 383 766 1149
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
`[root@deep /]# sysctl -w vm.freepages="383 766 1149"`

In our example above, according to the `/usr/src/linux/Documentation/sysctl/vm.txt`, the meaning of the numbers is:

`freepages.min`
When the number of free pages in the system reaches this number, only the kernel can allocate more memory.

`freepages.low`
If the number of free pages gets below this point, the kernel starts swapping aggressively.

`freepages.high`
The kernel tries to keep up to this amount of memory free; if memory comes below this point, the kernel gently starts swapping in the hopes that it never has to do real aggressive swapping.

NOTE: Look at `/usr/src/linux/Documentation/sysctl/vm.txt` for more information on how to improve kernel parameters related to virtual memory. Also take a note that `freepages` features parameters may vary from kernel version to another.

The `kswapd` parameter

The `kswapd` file `/proc/sys/vm/kswapd` is related to the kernel swapout daemon that frees memory when it gets fragmented or full. There are three parameters to tune in this file and two of them (`tries_base` and `swap_cluster`) have the largest influence on system performance. As for the above files, `kswapd` can be used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel.

The default setup for the `kswapd` parameter under Red Hat Linux is:
`"512 32 8"`

Step 1
To change the values of **`kswapd`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Increase swap bandwidth system performance
vm.kswapd = 1024 32 16
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
`[root@deep /]# sysctl -w vm.kswapd=”1024 32 16”`

In our example above, according to the `/usr/src/linux/Documentation/sysctl/vm.txt`, the meaning of the parameters are:

`tries_base`
The maximum number of pages `kswapd` tries to free in one round is calculated from this number. Usually this number will be divided by 4 or 8 (see `mm/vmscan.c`), so it isn't as big as it looks. When you need to increase the bandwidth to/from swap, you'll want to increase this number.

`tries_min`
This is the minimum number of times `kswapd` tries to free a page each time it is called. Basically it's just there to make sure that `kswapd` frees some pages even when it's being called with minimum priority.

`swap_cluster`
This is the number of pages `kswapd` writes in one turn. You want this large so that `kswapd` does it's I/O in large chunks and the disk doesn't have to seek often, but you don't want it to be too large since that would flood the request queue.

NOTE: Look at `/usr/src/linux/Documentation/sysctl/vm.txt` for more information on how to improve kernel parameters related to virtual memory. Also note that `kswapd` features parameters may vary from kernel version to another.

The `page-cluster` parameter

The Linux virtual memory subsystem avoids excessive disk seeks by reading multiple pages on a page fault. The number of pages it reads is highly dependent on the amount of memory in your machine. The number of pages the kernel reads in at once is equal to 2 ^ page-cluster. Values above 2 ^ 5 don't make much sense for swap because we only cluster swap data in 32-page groups. As for the above files, page-cluster is used to tune the operation of the virtual memory (VM) subsystem of the Linux kernel.

The default setup for the `page-cluster` parameter under Red Hat Linux is:
"4"

Step 1
To change the values of **`page-cluster`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Increase number of pages kernel reads in at once
vm.page-cluster = 16
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
`[root@deep /]# sysctl -w vm.page-cluster=16`

The `pagecache` parameter

This file does exactly the same job as `buffermem` parameter, but only controls the struct page_cache, and thus controls the amount of memory used for the page cache. To resume, it controls the amount of memory allowed for memory mapping and generic caching of files.

The page cache is used for 3 main purposes:

- ✓ caching read() data from files
- ✓ caching mmap()ed data and executable files
- ✓ swap cache

When your system is both deep in swap and high on cache, it probably means that a lot of the swapped data is being cached, making for more efficient swapping than possible. You don't want the minimum level to be too low, otherwise your system might thrash when memory is tight or fragmentation is high.

The default setup for the `pagecache` parameter under Red Hat Linux is:
`"2 15 75"`

Step 1
To change the values of **`pagecache`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Improve files memory mapping and generic caching
vm.pagecache = 8 25 85
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
`[root@deep /]# sysctl -w vm.pagecache="8 25 85"`

The `pagetable_cache` parameter

The kernel keeps a number of page tables in a per-processor cache (this helps a lot on `SMP` systems). The cache size for each processor will be between the low and the high value. On `SMP` systems it is used so that the system can do fast pagetable allocations without having to acquire the kernel memory lock.

For large systems, the settings are probably OK. For normal systems they won't hurt a bit. For small systems (<16MB RAM) and on a low-memory, single CPU system it might be advantageous to set both values to 0 so you don't waste the memory.

The default setup for the `pagetable_cache` parameter under Red Hat Linux is:
`"25 50"`

Step 1
To change the values of **`pagetable_cache`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf)` and add the following line:

```
# Improve number of page tables keeps in a per-processor cache
vm.pagetable_cache = 35 60
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: Only change these values on systems with multiple processors (`SMP`) or on small systems (single processor) with less than 16MB of RAM. Recall that on small systems the both values must be set to 0 (`vm.pagetable_cache = 0 0`).

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w vm.pagetable_cache=”35 60”
```

`/proc/sys/fs`: The file system data of Linux

All parameters described later in this chapter reside under the `/proc/sys/fs` directory of the server and can be used to tune and monitor miscellaneous things in the operation of the Linux kernel. Be very careful when attempting this. You can optimize your system, but you can also cause it to crash. Since every system is different, you'll probably want some control over these pieces of the system.

Finally, these are advanced settings and if you don’t understand them, then don’t try to play in this area or try to use all examples below directly in your systems. Remember that all systems are different and required different setting and customization.

Below I show you only parameters that can be optimized for the system. All suggestions I enumerate in this section are valid for every kind of servers. The only difference depends of the amount of MB of RAM your machines have and this is where settings will change.

```
                                    | - binfmt_misc -- | - register
                                    |                  | - status
                                    | - dentry-state
                                    | - dir-notify-enable
                                    | - dquot-max
                                    | - dquot-nr
                                    | - file-max
/proc/sys/fs ---------------------- | - file-nr
                                    | - inode-nr
                                    | - inode-state
                                    | - lease-break-time
                                    | - lease-enable
                                    | - overflowgid
                                    | - overflowuid
                                    | - super-max
                                    | - super-nr
```

The above figure shows a snapshot of `/proc/sys/fs` directory on a Red Hat Linux system running kernel version 2.4. Please note that this picture may look different on your system.

The `file-max` parameter

The `file-max` file `/proc/sys/fs/file-max` sets the maximum number of file-handles that the Linux kernel will allocate. We generally tune this file to improve the number of open files by increasing the value of `/proc/sys/fs/file-max` to something reasonable like 256 for every 4M of RAM we have: i.e. for a machine with 256 MB of RAM, set it to 16384 (256/4=64 64*256=16384).

The default setup for the `file-max` parameter under Red Hat Linux is:
`"8192"`

Step 1
To adjust the value of **`file-max`** to 256 MB of RAM, type the following on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Improve the number of open files
fs.file-max = 16384
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: When you regularly receive from your server a lot of messages with errors about running out of open files, you might want to raise this limit. The default value is 8192. A file server or web server needs a lot of open files.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w fs.file-max=16384
```

Other possible optimization of the system

All information described below relates to other possible tuning we can make on the system. Be very careful when attempting this. You can optimize your system, but you can also cause it to crash. Since every system is different, you'll probably want some control over this part of the system.

The `ulimit` parameter

Linux itself has a “Max Processes” per user limit. This feature allows us to control the number of processes an existing user on the server may be authorized to have. To improve performance, we can safely set the limit of processes for the super-user “root” to be unlimited.

Step 1

- Edit the **`.bashrc`** file (`vi /root/.bashrc`) and add the following line:

```
ulimit -u unlimited
```

The `ulimit` parameter provides control over the resources available to the shell and to processes started by it.

NOTE: You must exit and re-login from your terminal for the change to take effect.

Step 2

To verify that you are ready to go, make sure that when you type as root the command **`ulimit -a`** on your terminal, it shows "`unlimited`" next to **`max user processes`**.

```
[root@deep /]# ulimit -a

core file size (blocks)     1000000
data seg size (kbytes)      unlimited
file size (blocks)          unlimited
max locked memory (Kbytes)  unlimited
max memory size (kbytes)    unlimited
open files                  1024
pipe size (512 bytes)       8
stack size (kbytes)         8192
cpu time (seconds)          unlimited
max user processes          unlimited ← this line
virtual memory (kbytes)     unlimited
```

NOTE: You may also do `ulimit -u unlimited` at the command prompt instead of adding it to the `/root/.bashrc` file but the value will not survive to a reboot.

The `atime` attribute

Linux records information about when files were created and last modified as well as when it was last accessed. There is a cost associated with recording the last access time. The `ext2` file system of Linux has an attribute that allows the super-user to mark individual files such that their last access time is not recorded. This may lead to significant performance improvements on often accessed, frequently changing files such as the contents of News Server, Web Server, Proxy Server, Database Server among other directories.

- To set the attribute to a file, use:
  `[root@deep /]#` **`chattr +A filename`** **← For a specific file**

For a whole directory tree, do something like:

```
[root@deep /root]# chattr -R +A /var/spool          ← For a News and Mail Server directory
[root@deep /root]# chattr -R +A /cache              ← For a Proxy Caches directory
[root@deep /root]# chattr -R +A /home/httpd/openna  ← For a Web Server directory
[root@deep /root]# chattr -R +A /var/lib/mysql      ← For a SQL Database directory
```

The `noatime` attribute

Linux has a special mount option for file systems called **`noatime`** that can be added to each line that addresses one file system in the `/etc/fstab` file. If a file system has been mounted with this option, reading accesses to the file system will no longer result in an update to the **`atime`** information associated with the file like we have explained previously. The importance of the **`noatime`** setting is that it eliminates the need by the system to make writes to the file system for files, which are simply being read. Since writes can be somewhat expensive, this can result in measurable performance gains. Note that the write time information to a file will continue to be updated anytime the file is written to. In our example below, we will set the `noatime` option to our `/chroot` file system.

Step 1

- Edit the **`fstab`** file (`vi /etc/fstab`) and add in the line that refers to the `/chroot` file system, the `noatime` option after the defaults option as show below:

```
LABEL=/chroot    /chroot    ext2        defaults,noatime    1 2
```

Step 2

Once you have made the necessary adjustments to the `/etc/fstab` file, it is time to inform the Linux system about the modification.

- This can be accomplished with the following commands:
  `[root@deep /]#` **`mount /chroot -oremount`**

Each file system that has been modified must be remounted with the command show above. In our example we have modified the `/chroot` file system and it is for this reason that we remount this file system with the above command.

Step 3

- You can verify if the modifications have been correctly applied to the Linux system with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root /   ext2        rw 0 0
/proc /proc   proc        rw 0 0
/dev/sda1     /boot       ext2    rw 0 0
/dev/sda10    /cache      ext2    rw,nodev 0 0
/dev/sda9     /chroot     ext2    rw,noatime 0 0
/dev/sda8     /home       ext2    rw,nosuid 0 0
/dev/sda13    /tmp        ext2    rw,noexec,nosuid 0 0
/dev/sda7     /usr        ext2    rw 0 0
/dev/sda11    /var        ext2    rw 0 0
/dev/sda12    /var/lib    ext2    rw 0 0
none  /dev/pts    devpts  rw 0 0
```

This command will show you all file system in your Linux server with parameters applied to them. If you see something like:

```
/dev/sda11 /chroot    ext2    rw,noatime 0 0
```

Congratulations!

Part III Networking Related Reference

In this Part

Networking - TCP/IP Network Management
Networking - Firewall IPTABLES Packet Filter
Networking - Firewall Masquerading & Forwarding

The last line before going into program security, optimization and installation is the networking part of the Linux system. The next three chapters bring us where we will check, secure and test our network before implementing the iptables firewall packet filter of Linux, which will build a fortress around our secure server.

7 Networking - TCP/IP Network Management

In this Chapter

Linux TCP/IP Network Management

Abstract

This chapter has been inserted here because it is preferable not to be connected to the network if the parts "Installation-Related Reference" and "Security and Optimization-Related Reference" of the book have not been completed. It is not wise to apply new security configurations to your system if you are online. Also, don't forget that the firewall, which represents 50% of networking security, is still not configured on the Linux server. Finally it is very important and I say VERY IMPORTANT that you check all configuration files related to Linux networking to be sure that everything is configured correctly. Please follow all recommendations and steps in this chapter before continuing reading this book. This will allow us to be sure that if something goes wrong in the other chapters, it will be not related to your networking configurations.

- To stop specific network device manually on your system, use the following command:
```
[root@deep /]# ifdown eth0
```

- To start specific network device manually on your system, use the following command:
```
[root@deep /]# ifup eth0
```

- To stop all network devices manually on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/network stop
Shutting down interface eth0          [OK]
Disabling IPv4 packet forwarding      [OK]
```

- To start all network devices manually on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/network start
Enabling IPv4 packet forwarding       [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
```

Until now, we have not played with the networking capabilities of Linux. Linux is one of the best operating systems in the world for networking features. Most Internet sites around the world already know this, and have used it for some time. Understanding your network hardware and all the files related to it is very important if you want to have a full control of what happens on your server. Good knowledge of primary networking commands is vital. Network management covers a wide variety of topics. In general, it includes gathering statistical data and monitoring the status of parts of your network, and taking action as necessary to deal with failures and other changes.

The most primitive technique for network monitoring is periodic "pinging" of critical hosts. More sophisticated network monitoring requires the ability to get specific status and statistical information from a range of devices on the network. These should include various sorts of datagram counts, as well as counts of errors of different kinds. For these reasons, in this chapter we will try to answer fundamental questions about networking devices, files related to network functionality, and essential networking commands.

TCP/IP security problem overview

It is assumed that the reader is familiar with the basic operation of the TCP/IP protocol suite, which includes IP and TCP header field functions and initial connection negotiation. For the uninitiated, a brief description of TCP/IP connection negotiation is given below. The user is strongly encouraged however to research other published literature on the subject.

The IP Packets

The term packet refers to an **I**nternet **P**rotocol (IP) network message. It's the name given to a single, discrete message or piece of information that is sent across an Ethernet network. Structurally, a packet contains an information header and a message body containing the data being transferred. The body of the IP packet- it's data- is all or a piece (a fragment) of a higher-level protocol message.

The IP mechanism

Linux supports three IP message types: ICMP, UDP, and TCP. An ICMP (**I**nternet **C**ontrol **M**essage **P**rotocol) packet is a network-level, IP control and status message.

ICMP messages contains information about the communication between the two end-point computers.

A UDP (**U**ser **D**atagram **P**rotocol) IP packet carries data between two network-based programs, without any guarantees regarding successful delivery or packet delivery ordering. Sending a UDP packet is akin to sending a postcard to another program.

A TCP (**T**ransmission **C**ontrol **P**rotocol) IP packet carries data between two network-based programs, as well, but the packet header contains additional state information for maintaining an ongoing, reliable connection. Sending a TCP packet is akin to carrying on a phone conversation with another process. Most Internet network services use the TCP communication protocol rather than the UDP communication protocol. In other words, most Internet services are based on the idea of an ongoing connection with two-way communication between a client program and a server program.

The IP packet headers

All IP packet headers contain the source and destination IP addresses and the type of IP protocol message (ICMP, UDP or TCP) this packet contains. Beyond this, a packet header contains slightly different fields depending on the protocol type. ICMP packets contain a type field identifying the control or status message, along with a second code field for defining the message more specifically. UDP and TCP packets contain source and destination service port numbers. TCP packets contain additional information about the state of the connection and unique identifiers for each packet.

The TCP/IP Security Problem

The TCP/IP protocol suite has a number of weaknesses that allows an attacker to leverage techniques in the form of covert channels to surreptitiously pass data in otherwise benign packets. This section attempts to illustrate these weaknesses in theoretical examples.

Application

A covert channel is described as: "any communication channel that can be exploited by a process to transfer information in a manner that violates the systems security policy. Essentially, it is a method of communication that is not part of an actual computer system design, but can be used to transfer information to users or system processes that normally would not be allowed access to the information.

In the case of TCP/IP, there are a number of methods available whereby covert channels can be established and data can be surreptitiously passed between hosts. These methods can be used in a variety of areas such as the following:

- ✓ Bypassing packet filters, network sniffers, and "dirty word" search engines.
- ✓ Encapsulating encrypted or non-encrypted information within otherwise normal packets of information for secret transmission through networks that prohibit such activity "TCP/IP Steganography".
- ✓ Concealing locations of transmitted data by "bouncing" forged packets with encapsulated information off innocuous Internet sites.

It is important to realize that TCP is a "connection oriented" or "reliable" protocol. Simply put, TCP has certain features that ensure data arrives at the remote host in a usually intact manner. The basic operation of this relies in the initial TCP "three way handshake" which is described in the three steps below.

Step 1
Send a synchronize (**SYN**) packet and Initial Sequence Number (**ISN**)

Host A wishes to establish a connection to Host B. Host A sends a solitary packet to Host B with the synchronize bit (**SYN**) set announcing the new connection and an Initial Sequence Number (**ISN**) which will allow tracking of packets sent between hosts:

```
Host A  ------  SYN(ISN) ------>     Host B
```

Step 2
Allow remote host to respond with an acknowledgment (**ACK**)

Host B responds to the request by sending a packet with the synchronize bit set (**SYN**) and **ACK** (acknowledgment) bit set in the packet back to the calling host. This packet contains not only the responding clients' own sequence number, but the Initial Sequence Number plus one (**ISN+1**) to indicate the remote packet was correctly received as part of the acknowledgment and is awaiting the next transmission:

```
Host A  <------  SYN(ISN+1)/ACK ------     Host B
```

Step 3
Complete the negotiation by sending a final acknowledgment to the remote host.

At this point Host A sends back a final **ACK** packet and sequence number to indicate successful reception and the connection is complete and data can now flow:

```
Host A  ------ ACK ------>     Host B
```

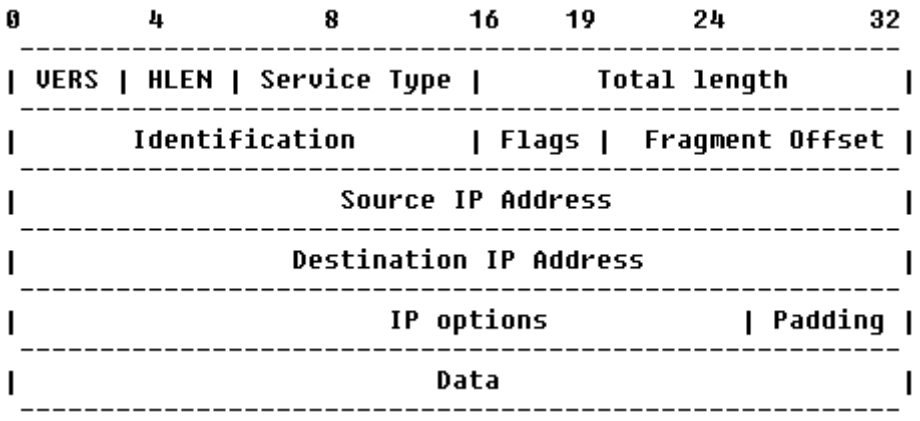
The entire connection process happens in a matter of milliseconds and both sides independently acknowledge each packet from this point. This handshake method ensures a "reliable" connection between hosts and is why TCP is considered a "connection oriented" protocol.

It should be noted that only TCP packets exhibit this negotiation process. This is not so with UDP packets which are considered "unreliable" and do not attempt to correct errors nor negotiate a connection before sending to a remote host.

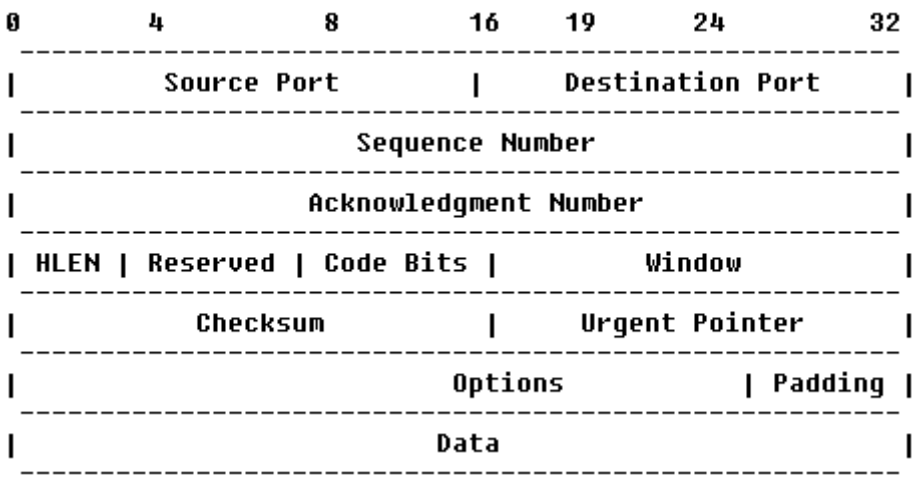
Encoding Information in a TCP/IP Header

The TCP/IP header contains a number of areas where information can be stored and sent to a remote host in a covert manner. Take the following diagrams, which are textual representations of the IP and TCP headers respectively:

IP Header (Numbers represent bits of data from 0 to 32 and the relative position of the fields in the datagram)



TCP Header (Numbers represent bits of data from 0 to 32 and the relative position of the fields in the datagram)



Within each header there are multitudes of areas that are not used for normal transmission or are "optional" fields to be set as needed by the sender of the datagrams. An analysis of the areas of a typical IP header that are either unused or optional reveals many possibilities where data can be stored and transmitted.

The basis of the exploitation relies in encoding ASCII values of the range 0-255. Using this method it is possible to pass data between hosts in packets that appear to be initial connection requests, established data streams, or other intermediate steps. These packets can contain no actual data, or can contain data designed to look innocent. These packets can also contain forged source and destination IP addresses as well as forged source and destination ports.

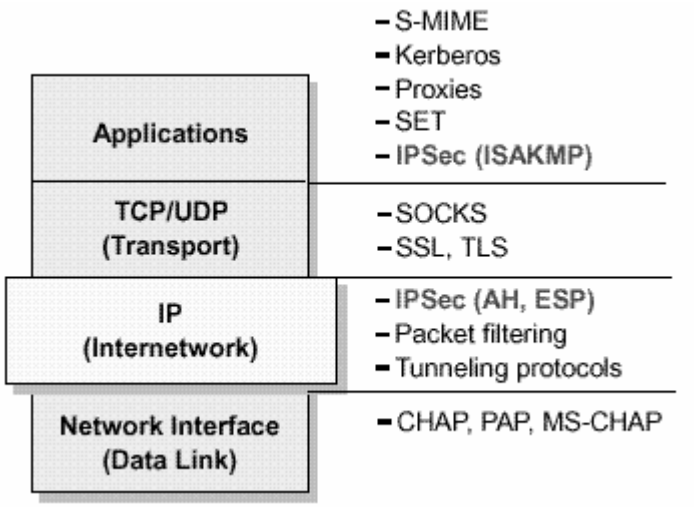
This can be useful for tunneling information past some types of packet filters. Additionally, forged packets can be used to initiate an anonymous TCP/IP "bounced packet network" whereby packets between systems can be relayed off legitimate sites to thwart tracking by sniffers and other network monitoring devices.

Implementations of Security Solutions

The following protocols and systems are commonly used to solve and provide various degrees of security services in a computer network.

- IP filtering
- Network Address Translation (NAT)
- IP Security Architecture (IPSec)
- SOCKS
- Secure Sockets Layer (SSL)
- Application proxies
- Firewalls
- Kerberos and other authentication systems (AAA servers)
- Secure Electronic Transactions (SET)

This graph illustrates where those security solutions fit within the TCP/IP layers:



Security Solutions in the TCP/IP Layers

Installing more than one Ethernet Card per Machine

You might use Linux as a gateway between two Ethernet networks. In that case, you might have two Ethernet cards on your server. To eliminate problems at boot time, the Linux kernel doesn't detect multiple cards automatically. If you happen to have two or more cards, you should specify the parameters of the cards in the `lilo.conf` file for a Monolithic kernel or in the `modules.conf` file for a Modularized kernel. The following are problems you may encounter with your network cards.

Problem 1
If the driver(s) of the card(s) is/are being used as a loadable module (Modularized kernel), in the case of PCI drivers, the module will typically detect all of the installed cards automatically. For ISA cards, you need to supply the I/O base address of the card so the module knows where to look. This information is stored in the file `/etc/modules.conf`.

As an example, consider we have two ISA 3c509 cards, one at I/O 0x300 and one at I/O 0x320.

- For ISA cards, edit the **`modules.conf`** file (`vi /etc/modules.conf`) and add:

```
alias eth0 3c509
alias eth1 3c509
options 3c509 io=0x300,0x320
```

This says that the 3c509 driver should be loaded for either eth0 or eth1 (alias eth0, eth1) and it should be loaded with the options io=0x300,0x320 so that the drivers knows where to look for the cards. Note that 0x is important – things like 300h as commonly used in the DOS world won't work.

For PCI cards, you typically only need the alias lines to correlate the ethN interfaces with the appropriate driver name, since the I/O base of a PCI card can be safely detected.

- For PCI cards, edit the **`modules.conf`** file (`vi /etc/modules.conf`) and add:

```
alias eth0 3c509
alias eth1 3c509
```

Problem 2
If the drivers(s) of the card(s) is/are compiled into the kernel (Monolithic kernel), the PCI probes will find all related cards automatically. ISA cards will also find all related cards automatically, but in some circumstance ISA cards still need to do the following. This information is stored in the file `/etc/lilo.conf`. The method is to pass boot-time arguments to the kernel, which is usually done by LILO.

- For ISA cards, edit the **`lilo.conf`** file (`vi /etc/lilo.conf`) and add:

```
append="ether=0,0,eth1"
```

In this case eth0 and eth1 will be assigned in the order that the cards are found at boot. Remember that this is required only in some circumstance for ISA cards, PCI cards will be found automatically.

NOTE: First test your ISA cards without the boot-time arguments in the `lilo.conf` file, and if this fails, use the boot-time arguments.

Files-Networking Functionality

In Linux, the TCP/IP network is configured through several text files. You may have to edit them to make the network work. It's very important to know the configuration files related to TCP/IP networking, so that you can edit and configure the files if necessary. Remember that our server doesn't have an Xwindow interface (GUI) to configure files via a graphical interface. Even if you use a GUI in your daily activities it is important to know how to configure the network configuration files in text mode. The following sections describe all the basic TCP/IP configuration files under Linux.

The `/etc/sysconfig/network-scripts/ifcfg-ethN` files

The configuration files for each network device you may have or want to add on your system are located in the `/etc/sysconfig/network-scripts` directory with Red Hat Linux, and are named **`ifcfg-eth0`** for the first interface and **`ifcfg-eth1`** for the second, etc. It is recommended to verify if all the parameters in this file are correct.

Following is a sample `/etc/sysconfig/network-scripts/ifcfg-eth0` file:

```
DEVICE=eth0
BOOTPROTO=static
BROADCAST=208.164.186.255
IPADDR=208.164.186.1
NETMASK=255.255.255.0
NETWORK=208.164.186.0
ONBOOT=yes
USERCTL=no
```

If you want to modify your network address manually, or add a new one on a new interface, edit this file (`ifcfg-ethN`), or create a new one and make the appropriate changes.

`DEVICE=`***devicename***, where ***devicename*** is the name of the physical network device.

`BOOTPROTO=`***proto***, where ***proto*** is one of the following:

- static - The default option of Linux (static IP address) sould be used.
- none - No boot-time protocol should be used.
- bootp - The bootp (now pump) protocol should be used.
- dhcp - The dhcp protocol should be used.

`BROADCAST=`***broadcast***, where ***broadcast*** is the broadcast IP address.

`IPADDR=`***ipaddr***, where ***ipaddr*** is the IP address.

`NETMASK=`***netmask***, where ***netmask*** is the netmask IP value.

`NETWORK=`***network***, where ***network*** is the network IP address.

`ONBOOT=`***answer***, where ***answer*** is yes or no (Does the interface will be active or inactive at boot time).

`USERCTL=`***answer***, where ***answer*** is one of the following:

- yes (Non-root users are allowed to control this device).
- no (Only the super-user root is allowed to control this device).

The `/etc/resolv.conf` file

This file `/etc/resolv.conf` is another text file, used by the resolver—a library that determines the IP address for a host name. It is recommended to verify if all parameters included in this file are corrects.

Following is a sample `/etc/resolv.conf` file:

```
domain openna.com
search ns1.openna.com ns2.openna.com openna.com
nameserver 208.164.186.1
nameserver 208.164.186.2
nameserver 127.0.0.1
```

NOTE: Name servers are queried in the order they appear in the file (primary, secondary).

The `/etc/host.conf` file

This file `/etc/host.conf` specifies how names are resolved. Linux uses a resolver library to obtain the IP address corresponding to a host name. It is recommended to verify that all parameters included in this file are correct.

Following is a sample `/etc/host.conf` file:

```
# Lookup names via /etc/hosts first then fall back to DNS resolver.
order hosts,bind
# We have machines with multiple addresses.
multi on
```

The **`order`** option indicates the order of services. The sample entry specifies that the resolver library should first consult the `/etc/hosts` file of Linux to resolve a name and then check the name server (DNS).

The **`multi`** option determines whether a host in the `/etc/hosts` file can have multiple IP addresses (multiple interface ethN). Hosts that have more than one IP address are said to be *multihomed*, because the presence of multiple IP addresses implies that the host has several network interfaces.

The `/etc/sysconfig/network` file

The `/etc/sysconfig/network` file is used to specify information about the desired network configuration on your server. It is recommended that you verify all the parameters included in this file are correct.

Following is a sample /etc/sysconfig/network file:

```
NETWORKING=yes
HOSTNAME=deep
GATEWAY=207.35.78.1
GATEWAYDEV=eth0
```

The following values may be used:

NETWORKING=***answer***, where ***answer*** is yes or no (Configure networking or not configure networking).

HOSTNAME=***hostname***, where ***hostname*** is the hostname of your server.

GATEWAY=***gwip***, where ***gwip*** is the IP address of the remote network gateway (if available).

GATEWAYDEV=***gwdev***, where ***gwdev*** is the device name (eth#) you use to access the remote gateway.

The `/etc/sysctl.conf` file

With the new version of Red Hat Linux, all kernel parameters available under the `/proc/sys/` subdirectory of Linux can be configured at runtime. You can use the new `/etc/sysctl.conf` file to modify and set kernel parameters at runtime. The `sysctl.conf` file is read and loaded each time the system reboots or when you restart your network. All settings are now stored in the `/etc/sysctl.conf` file. All modifications to `/proc/sys` should be made through `/etc/sysctl.conf`, because they are better for control, and are executed before `rc.local` or any other "users" scripts.

Below, we'll focus only on the kernel option for `IPv4` forwarding support. See later in this chapter the `TCP/IP` security parameters related to the `sysctl.conf` file.

To enable `IPv4` forwarding on your Linux system, use the following command:

Step 1

- Edit the `sysctl.conf` file (`vi /etc/sysctl.conf`) and add the following line:

```
# Enable packet forwarding (required only for Gateway, VPN, Proxy, PPP)
net.ipv4.ip_forward = 1
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: You must enable packet forwarding only on a machine that serves as a Gateway Server, `VPN` Server, Proxy Server or with `PPP` connection. Forwarding allows packets that are destined for another network interface (if you have another one) to pass through the network.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.ip_forward=1
```

The /etc/hosts file

As your machine gets started, it will need to know the mapping of some hostnames to IP addresses before DNS can be referenced. This mapping is kept in the /etc/hosts file. In the absence of a name server, any network program on your system consults this file to determine the IP address that corresponds to a host name.

Following is a sample /etc/hosts file:

```
IP Address          Hostname                    Alias
127.0.0.1           localhost.localdomain       localhost
208.164.186.1       deep.openna.com             deep
208.164.186.2       mail.openna.com             mail
208.164.186.3       web.openna.com              web
```

The leftmost column is the IP address to be resolved. The next column is that host's name. Any subsequent columns are the aliases for that host. In the second line, for example, the IP address 208.164.186.1 if for the host deep.openna.com. Another name for deep.openna.com is deep.

WARNING: Some people have reporting that a badly formed line in the /etc/hosts file may result to a "Segmentation fault (core dumped)" with the syslogd daemon, therefore I recommend you to double check your entry under this file and be sure that its respond to the example as shown above. The "Alias" part of the line is important if you want to be able to use the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the system reported by the hostname -f command.

After you are finished adding and configuring your networking files, don't forget to restart your network for the changes to take effect.

- To restart your network, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: Time out problems for telnet or ftp connection are often caused by the server trying to resolve the client IP address to a DNS name. Either DNS isn't configured properly on your server or the client machines aren't known to the DNS server. If you intend to run telnet or ftp services on your server, and aren't using DNS, don't forget to add the client machine name and IP in your /etc/hosts file on the server or you can expect to wait several minutes for the DNS lookup to time out, before you get a login prompt.

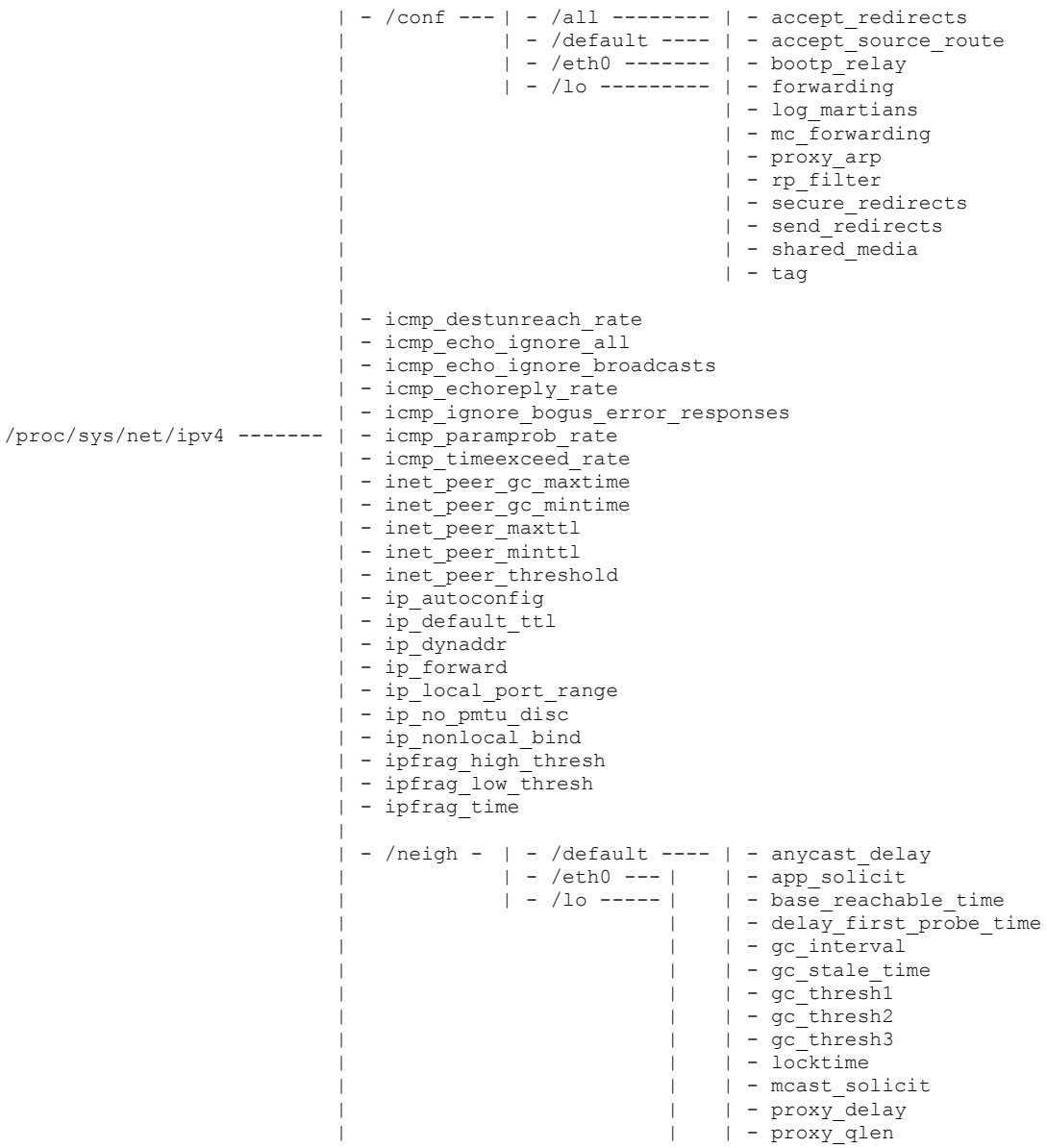
Securing TCP/IP Networking

In Red Hat Linux, many kernel options related to networking security such as dropping packets that come in over interfaces they shouldn't or ignoring ping/broadcasts request, etc can be set in the new `/etc/sysctl.conf` file instead of the `/etc/rc.d/rc.local` file. The `sysctl` settings are stored in `/etc/sysctl.conf`, and are loaded at each boot or networking restart before the `/etc/rc.d/rc.local` file is loaded.

Below, we show you the networking security options that you must definitely configure on your server. To display all `sysctl` values currently available use the "`sysctl -a`" command.

/proc/sys/net/ipv4: IPV4 settings of Linux

All parameters described below reside under the `/proc/sys/net/ipv4` directory of the server and can be used to control the behavior of the IPv4 subsystem of the Linux kernel. Below I show you only the parameters, which can be used for the network security of the system.



```
|                          |    | - retrans_time
|                          |    | - ucast_solicit
|                          |    | - unres_qlen
|                          |
|                          | - anycast_delay
|                          | - app_solicit
|                          | - base_reachable_time
|                          | - delay_first_probe_time
|                          | - gc_stale_time
|                          | - locktime
|                          | - mcast_solicit
|                          | - proxy_delay
|                          | - proxy_qlen
|                          | - retrans_time
|                          | - ucast_solicit
|                          | - unres_qlen
|
| - /route -- | - error_burst
|             | - error_cost
|             | - flush
|             | - gc_elasticity
|             | - gc_interval
|             | - gc_min_interval
|             | - gc_thresh
|             | - gc_timeout
|             | - max_delay
|             | - max_size
|             | - min_adv_mss
|             | - min_delay
|             | - min_pmtu
|             | - mtu_expires
|             | - redirect_load
|             | - redirect_number
|             | - redirect_silence
|
| - tcp_abort_on_overflow
| - tcp_adv_win_scale
| - tcp_app_win
| - tcp_dsack
| - tcp_fack
| - tcp_fin_timeout
| - tcp_keepalive_intvl
| - tcp_keepalive_probes
| - tcp_keepalive_time
| - tcp_max_orphans
| - tcp_max_syn_backlog
| - tcp_max_tw_buckets
| - tcp_mem
| - tcp_orphan_retries
| - tcp_reordering
| - tcp_retrans_collapse
| - tcp_retries1
| - tcp_retries2
| - tcp_rfc1337
| - tcp_rmem
| - tcp_sack
| - tcp_stdurg
| - tcp_syn_retries
| - tcp_synack_retries
| - tcp_syncookies
| - tcp_timestamps
| - tcp_tw_recycle
| - tcp_window_scaling
| - tcp_wmem
```

The above figure shows a snapshot of `/proc/sys/net/ipv4` directory on a Red Hat Linux system running kernel version 2.4. Please note that this picture may look different on your system.

Prevent your system responding to `ping` request

Preventing your system for responding to `ping` request can make a big improvement in your network security since no one can `ping` your server and receive an answer. The `TCP/IP` protocol suite has a number of weaknesses that allows an attacker to leverage techniques in the form of covert channels to surreptitiously pass data in otherwise benign packets. Preventing your server from responding to `ping` requests can help to minimize this problem. Not responding to pings would at least keep most "crackers" out because they would never even know it's there.

Step 1

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Enable ignoring ping request
net.ipv4.icmp_echo_ignore_all = 1
```

Step 2

Once the configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.icmp_echo_ignore_all=1
```

Refuse responding to `broadcasts` request

As for the `ping` request, it's also important to disable `broadcast` requests. When a packet is sent to an IP `broadcast` address (i.e. `192.168.1.255`) from a machine on the local network, that packet is delivered to all machines on that network. Then all the machines on a network respond to this `ICMP` echo request and the result can be severe network congestion or outages (Denial-of-Service attacks). See the RFC 2644 for more information.

Step 1

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Enable ignoring broadcasts request
net.ipv4.icmp_echo_ignore_broadcasts = 1
```

Step 2
Once the configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all networks devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters             [OK]
Bringing up interface lo               [OK]
Bringing up interface eth0             [OK]
Bringing up interface eth1             [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.icmp_echo_ignore_broadcasts=1
```

Routing Protocols

Routing and routing protocols can create several problems. IP source routing, where an IP packet contains details of the path to its intended destination, is dangerous because according to RFC 1122 the destination host must respond along the same path. If an attacker was able to send a source routed packet into your network, then he would be able to intercept the replies and fool your host into thinking it is communicating with a trusted host.

I strongly recommend that you disable IP source routing on all network interfaces on the system to protect your server from this hole. In the configuration below, we disable `IP` source routing on all interfaces on the system even for the interface `eth1`, which represents your second network card if you have one. If `eth1` doesn't exist on your system, then omit the line related to `eth1` in your `sysctl.conf` file.

Step 1
To disable IP source routing on your server, type the following command in your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Disables IP source routing
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.lo.accept_source_route = 0
net.ipv4.conf.eth0.accept_source_route = 0
net.ipv4.conf.eth1.accept_source_route = 0
net.ipv4.conf.default.accept_source_route = 0
```

Step 2
Once configurations have been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters             [OK]
Bringing up interface lo               [OK]
Bringing up interface eth0             [OK]
Bringing up interface eth1             [OK]
```

NOTE: This parameter is dependent on the kernel configuration. If the kernel is configured for a regular host the default setting 'yes' for this parameter can be acceptable and 'no' must be set for a router configuration. 1 means yes, 0 means no.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.accept_source_route=0
[root@deep /]# sysctl -w net.ipv4.conf.lo.accept_source_route=0
[root@deep /]# sysctl -w net.ipv4.conf.eth0.accept_source_route=0
[root@deep /]# sysctl -w net.ipv4.conf.eth1.accept_source_route=0
[root@deep /]# sysctl -w net.ipv4.conf.default.accept_source_route=0
```

Enable TCP SYN Cookie Protection

A "SYN Attack" is a **D**enial **o**f **S**ervice (DoS) attack that consumes all the resources on your machine, forcing you to reboot. Denials of Service attacks (attacks which incapacitate a server due to high traffic volume or ones that tie-up system resources enough that the server cannot respond to a legitimate connection request from a remote system) are easily achievable from internal resources or external connections via extranets and Internet. Enabling `TCP SYN Cookie Protection` will help to eliminate the problem.

Step 1

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Enable TCP SYN Cookie Protection
net.ipv4.tcp_syncookies = 1
```

Step 2

Once the configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: If you receive an error message during execution of the above command, check that you have enable the `TCP` syncookies option in your kernel configuration: `IP: TCP syncookie support (not enabled per default) (CONFIG_SYN_COOKIES) [N/y/?].`

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.tcp_syncookies=1
```

Disable ICMP Redirect Acceptance

When hosts use a non-optimal or defunct route to a particular destination, an ICMP redirect packet is used by routers to inform the hosts what the correct route should be. If an attacker is able to forge ICMP redirect packets, he or she can alter the routing tables on the host and possibly subvert the security of the host by causing traffic to flow via a path you didn't intend. It's strongly recommended to disable ICMP Redirect Acceptance into all available interfaces on the server to protect it from this hole. In the configuration below, we disable the ICMP redirect acceptance for all possible interfaces on the system, even for the interface eth1, which represents your second network card if you have one. If eth1 doesn't exist on your system, then omit the line related to eth1 in your sysctl.conf file.

Step 1

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following lines:

```
# Disable ICMP Redirect Acceptance
net.ipv4.conf.all.accept_redirects = 0
net.ipv4.conf.lo.accept_redirects = 0
net.ipv4.conf.eth0.accept_redirects = 0
net.ipv4.conf.eth1.accept_redirects = 0
net.ipv4.conf.default.accept_redirects = 0
```

Step 2

Once configurations have been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all networks devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: If the kernel is configured for a regular host the default setting 'yes' for this parameter can be acceptable and 'no' must be made for a router configuration. 1 means yes, 0 means no.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.accept_redirects=0
[root@deep /]# sysctl -w net.ipv4.conf.lo.accept_redirects=0
[root@deep /]# sysctl -w net.ipv4.conf.eth0.accept_redirects=0
[root@deep /]# sysctl -w net.ipv4.conf.eth1.accept_redirects=0
[root@deep /]# sysctl -w net.ipv4.conf.default.accept_redirects=0
```

Enable bad error message Protection

This option will alert you about all bad error messages in your network.

Step 1

- Edit the **sysctl.conf** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Enable bad error message Protection
net.ipv4.icmp_ignore_bogus_error_responses = 1
```

Step 2

Once configuration has been set, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters             [OK]
Bringing up interface lo               [OK]
Bringing up interface eth0             [OK]
Bringing up interface eth1             [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.icmp_ignore_bogus_error_responses=1
```

Enable IP spoofing protection

The spoofing protection prevents your network from being the source of spoofed (i.e. forged) communications that are often used in DoS Attacks. In the configuration below, we enable source route verification for all possible interfaces on the system even for the interface `eth1`, which represents your second network card if you have one. If `eth1` doesn't exist on your system, then omit the line related to `eth1` in your `sysctl.conf` file.

Step 1

- Edit the **sysctl.conf** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable IP spoofing protection, turn on source route verification
net.ipv4.conf.all.rp_filter = 1
net.ipv4.conf.lo.rp_filter = 1
net.ipv4.conf.eth0.rp_filter = 1
net.ipv4.conf.eth1.rp_filter = 1
net.ipv4.conf.default.rp_filter = 1
```

Step 2

Once the configurations have been made, you must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters             [OK]
Bringing up interface lo               [OK]
Bringing up interface eth0             [OK]
Bringing up interface eth1             [OK]
```

NOTE: This parameter will prevent spoofing attacks against your internal networks but your external addresses can still be spoofed.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.rp_filter=1
[root@deep /]# sysctl -w net.ipv4.conf.lo.rp_filter=1
[root@deep /]# sysctl -w net.ipv4.conf.eth0.rp_filter=1
[root@deep /]# sysctl -w net.ipv4.conf.eth1.rp_filter=1
[root@deep /]# sysctl -w net.ipv4.conf.default.rp_filter=1
```

Enable Log Spoofed, Source Routed and Redirect Packets

This change will log all Spoofed Packets, Source Routed Packets, and Redirect Packets to your log files. In the configuration below, we enable "Log Spoofed, Source Routed and Redirect Packets" for all possible interfaces on the system even for the interface `eth1`, which represents your second network card if you have one. If `eth1` doesn't exist on your system, then omit the line related to `eth1` in your `sysctl.conf` file.

Step 1

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable Log Spoofed Packets, Source Routed Packets, Redirect Packets
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.lo.log_martians = 1
net.ipv4.conf.eth0.log_martians = 1
net.ipv4.conf.eth1.log_martians = 1
net.ipv4.conf.default.log_martians = 1
```

Step 2

You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters    [OK]
Bringing up interface lo      [OK]
Bringing up interface eth0    [OK]
Bringing up interface eth1    [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.conf.all.log_martians=1
[root@deep /]# sysctl -w net.ipv4.conf.lo.log_martians=1
[root@deep /]# sysctl -w net.ipv4.conf.eth0.log_martians=1
[root@deep /]# sysctl -w net.ipv4.conf.eth1.log_martians=1
[root@deep /]# sysctl -w net.ipv4.conf.default.log_martians=1
```

Optimizing TCP/IP Networking

This section deals with actions we can make to improve and tighten performance of the Linux TCP/IP networking. Note that we refer to the features available within the base installed Linux system. Below I show you only the parameters, which can be used to optimize the TCP/IP networking of your system. All the suggestions I make in this section are valid for all kinds of servers. The only difference depends of the amount of MB of RAM your machines have and this is where some settings will change. The majority of the following hacks will work very fine with servers >= 512MB of RAM or at a minimumof 256MB of RAM. Below this amount of memory, nothing is guaranteed and the default settings will just be fine for you.

Better manage your TCP/IP resources

This hack just make the time default values for TCP/IP connection lower so that more connections can be handled by at a time by your TCP/IP protocol. The following will decrease the amount of time your Linux machine will try take to finish closing a connection and the amount of time before it will kill a stale connection. This will also turn off some IP extensions that aren't needed.

The default setup for the TCP/IP parameters we'll change under Red Hat Linux are:
For the `tcp_fin_timeout "60"`
For the `tcp_keepalive_time "7200"`
For the `tcp_window_scaling "1"`
For the `tcp_sack "1"`
For the `tcp_timestamps "1"`

Step 1
To adjust the new **TCP/IP** values, type the following commands on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Decrease the time default value for tcp_fin_timeout connection
net.ipv4.tcp_fin_timeout = 30

# Decrease the time default value for tcp_keepalive_time connection
net.ipv4.tcp_keepalive_time = 1800

# Turn off the tcp_window_scaling support
net.ipv4.tcp_window_scaling = 0

# Turn off the tcp_sack support
net.ipv4.tcp_sack = 0

# Turn off the tcp_timestamps support
net.ipv4.tcp_timestamps = 0
```

Step 2
Once the parameters have been changed, you must restart your network for the changes to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters       [OK]
Bringing up interface lo         [OK]
Bringing up interface eth0       [OK]
Bringing up interface eth1       [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.tcp_fin_timeout=30
[root@deep /]# sysctl -w net.ipv4.tcp_keepalive_time=1800
[root@deep /]# sysctl -w net.ipv4.tcp_window_scaling=0
[root@deep /]# sysctl -w net.ipv4.tcp_sack=0
[root@deep /]# sysctl -w net.ipv4.tcp_timestamps=0
```

Better manage your `buffer-space` resources

The three parameters below are related to 'total', 'read', and 'write' `TCP buffer-space` that the kernel will allocate on your `TCP/IP` protocol. We generally tune these files to improve the maximum `TCP buffer-space` on the system by increasing the default values to something reasonable like 1 time for every 64M of RAM we have: i.e. for a machine with 256 MB of RAM, set it to 28672 and 16384 for `tcp_mem` and `tcp_wmem` parameters (256/64=4 4*7168=28672 and 256/64=4 4*4096=16384) and three time the values of `tcp_wmem` for `tcp_rmem` (3*16384=49152).

The default setup for the `buffer-space` resources we'll change under Red Hat Linux are:
For the `tcp_mem` "7168 7680 8192"
For the `tcp_wmem` "4096 16384 131072"
For the `tcp_rmem` "4096 87380 174760"

Step 1
To adjust the new **`buffer-space`** values, type the following commands on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Increase the maximum total TCP buffer-space allocatable
net.ipv4.tcp_mem = 28672 28672 32768

# Increase the maximum TCP write-buffer-space allocatable
net.ipv4.tcp_wmem = 16384 65536 524288

# Increase the maximum TCP read-buffer space allocatable
net.ipv4.tcp_rmem = 49152 196608 1572864
```

NOTE: For super computers with a lot of RAM (> 2GB), we can set the new values to:

```
net.ipv4.tcp_mem = 100000000 100000000 100000000
net.ipv4.tcp_wmem = 100000000 100000000 100000000
net.ipv4.tcp_rmem = 30000000 30000000 30000000
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.tcp_mem="28672 28672 32768"
[root@deep /]# sysctl -w net.ipv4.tcp_wmem="16384 65536 524288"
[root@deep /]# sysctl -w net.ipv4.tcp_rmem="49152 196608 1572864"
```

Better manage your `buffer-size` resources

The four parameters below are related to the maximum and default setting of the socket receives and send buffer/buffer-size in bytes. We generally tune these files to improve the maximum and default `socket buffer-size` of the network core option by increasing the default values to something reasonable like 65535 for every 64M of RAM we have: i.e. for a machine with 256 MB of RAM, the new values will be 262140 (256/64=4 4*65535=262140).

The default setup for the `buffer-size` resources we'll change under Red Hat Linux are:
For the `rmem_max "65535"`
For the `rmem_default "65535"`
For the `wmem_max "65535"`
For the `wmem_default "65535"`

Step 1
To adjust the new **`buffer-size`** values, type the following commands on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Increase the maximum and default receive socket buffer size
net.core.rmem_max = 262140
net.core.rmem_default = 262140

# Increase the maximum and default send socket buffer size
net.core.wmem_max = 262140
net.core.wmem_default = 262140
```

NOTE: For super computers with a lot of RAM (> 2GB), we can set the new values to:

```
net.core.rmem_max = 10485760
net.core.rmem_default = 10485760
net.core.wmem_max = 10485760
net.core.wmem_default = 10485760
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.core.rmem_max=262140
[root@deep /]# sysctl -w net.core.rmem_default=262140
[root@deep /]# sysctl -w net.core.wmem_max=262140
[root@deep /]# sysctl -w net.core.wmem_default=262140
```

The `tcp_max_tw_buckets` parameters

The `tcp_max_tw_buckets` `/proc/sys/net/ipv4/tcp_max_tw_buckets` set the `TCP` time-wait buckets pool size for the system. For high-usage systems you may change its default parameter to something reasonable like 180000 for every 64M of RAM we have: i.e. for a machine with 256 MB of RAM, the new values will be 720000 (256/64=4 4*180000=720000) or for super computers with a lot of RAM (> 2GB) you can set this value to 2000000.

The default setup for the `tcp_max_tw_buckets` parameter under Red Hat Linux is:
`"180000"`

Step 1
To change the values of **`tcp_max_tw_buckets`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Increase the tcp-time-wait buckets pool size
net.ipv4.tcp_max_tw_buckets = 720000
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.tcp_max_tw_buckets=720000
```

The `ip_local_port_range` parameters

The `ip_local_port_range` `/proc/sys/net/ipv4/ip_local_port_range` defines the local port range that is used by `TCP` and `UDP` traffic to choose the local port. You will see in the parameters of this file two numbers: The first number is the first local port allowed for `TCP` and `UDP` traffic on the server, the second is the last local port number.

For high-usage systems you may change its default parameters to 16384-65536 (first-last) but only for high-usage systems or you will surely receive error message like: "resources temporally unavailable".

The default setup for the `ip_local_port_range` parameter under Red Hat Linux is:
`"32768 61000"`

Step 1
To change the values of **`ip_local_port_range`**, type the following command on your terminal:

- Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following line:

```
# Allowed local port range
net.ipv4.ip_local_port_range = 16384 65536
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
`[root@deep /]# sysctl -w net.ipv4.ip_local_port_range="16384 65536"`

The `ipfrag_high_thresh` and `ipfrag_low_thresh` parameters

The two parameters below relate to the maximum memory used to reassemble IP fragments. When `ipfrag_high_thresh` bytes of memory are allocated for this purpose, the fragment handler will toss packets until `ipfrag_low_thresh` is reached.

The default setup for the `ipfrag_high_thresh` and `ipfrag_low_thresh` parameters under Red Hat Linux are:
For the `ipfrag_high_thresh` `"262144"`
For the `ipfrag_low_thresh` `"196608"`

Step 1
To change the values of **ipfrag_thresh**, type the following command on your terminal:

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following line:

```
# Increase the maximum memory used to reassemble IP fragments
net.ipv4.ipfrag_high_thresh = 512000
net.ipv4.ipfrag_low_thresh = 446464
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.ipfrag_high_thresh=512000
[root@deep /]# sysctl -w net.ipv4.ipfrag_low_thresh=446464
```

The optmem_max and hot_list_length parameters

Finally the last two parameters we will show here are related to the maximum ancillary buffer size allowed per socket (Ancillary data is a sequence of struct cmsghdr structures with appended data) and the maximum number of skb-heads that can be cached by the TCP/IP networking feature of Linux.

For high-usage systems you may change its default parameter to something reasonable like 7168 for every 64M of RAM we have: i.e. for a machine with 256 MB of RAM, the new values will be 28672 for optmem_max parameter (256/64=4 4*7168=28672) or for super computers with a lot of RAM (> 2GB) you can set this value to 10000000 for optmem_max and 102400 for hot_list_length parameters.

The default setup for the optmem_max and hot_list_length parameters are:
For the optmem_max "10240"
For the hot_list_length "128"

Step 1
To change the values of **optmem_max** and **hot_list_length**, type the following command on your terminal:

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following line:

```
# Increase the maximum amount of option memory buffers
net.core.optmem_max = 28672

# Increase the maximum number of skb-heads to be cached
net.core.hot_list_length = 512
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
```
[root@deep /]# sysctl -w net.core.optmem_max=28672
[root@deep /]# sysctl -w net.core.hot_list_length=512
```

Testing TCP/IP Networking

Once we have applied TCP/IP security and optimization parameters to our server and checked or configured all files related to network functionality, we can run some tests to verify that everything works as expected.

Step 1
Before running these tests, it is important to verify that the iputils package is installed in your system. If you have carefully followed every step during our installation of Linux on your computer, then this package is not installed.

- To verify if iputils package is installed on your system, use the following command:
```
[root@deep /]# rpm -q iputils
package iputils is not installed
```

Step 2
If the iputils package seems to not be installed, you need to mount your CD-ROM drive containing the Red Hat CD-ROM Part 1 and install it.

- To mount the CD-ROM drive, use the following command:
```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
had: ATAPI 32X CD-ROM drive, 128kB Cache
mount: block device dev/cdrom is write-protected, mounting read-only
```

Step 3

- To install the iputils package on your Linux system, use the following command:
```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh iputils-version.i386.rpm
iputils      ##################################################
```

Step 4

- To unmount your CD-ROM drive, use the following command:
```
[root@deep RPMS]# cd /; umount /mnt/cdrom/
```

Once the `iputils` package is installed on your system, it is time to run the tests to see if the network works as expected. It is important to note that at this stage every test must be successful and not have any errors. It is to your responsibility to know and understand networking architecture and basic `TCP/IP` protocols before testing any parts of your networking configuration and topology.

Step 1
To begin, we can use the **`ifconfig`** utility to display all the network interfaces on the server.

- To display all the interfaces you have on your server, use the command:
  `[root@deep /]#` **`ifconfig`**

  The output should look something like this:

```
eth0  Link encap:Ethernet  HWaddr 00:E0:18:90:1B:56
      inet addr:208.164.186.2  Bcast:208.164.186.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xa800

 lo   Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
      UP LOOPBACK RUNNING  MTU:3924  Metric:1
      RX packets:139 errors:0 dropped:0 overruns:0 frame:0
      TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:0
```

NOTE: If the **`ifconfig`** tool is invoked without any parameters, it displays all interfaces you configured. An option of "**`-a`**" shows the inactive one as well.

- To display all interfaces as well as inactive interfaces you may have, use the command:
  `[root@deep /]#` **`ifconfig -a`**

  The output should look something like this:

```
eth0  Link encap:Ethernet  HWaddr 00:E0:18:90:1B:56
      inet addr:208.164.186.2  Bcast:208.164.186.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:11 Base address:0xa800

eth1  Link encap:Ethernet  HWaddr 00:E0:18:90:1B:56
      inet addr:192.168.1.1  Bcast:192.168.1.255  Mask:255.255.255.0
      UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
      RX packets:1295 errors:0 dropped:0 overruns:0 frame:0
      TX packets:1163 errors:0 dropped:0 overruns:0 carrier:0
      collisions:0 txqueuelen:100
      Interrupt:5 Base address:0xa320

 lo   Link encap:Local Loopback
      inet addr:127.0.0.1  Mask:255.0.0.0
```

```
UP LOOPBACK RUNNING  MTU:3924  Metric:1
RX packets:139 errors:0 dropped:0 overruns:0 frame:0
TX packets:139 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:0
```

Step 2
If all network interfaces on the server look as you expect, then it is time to verify that you can reach your hosts. Choose a host from your internal network, for instance `192.168.1.1`

- To verify that you can reach your internal hosts, use the command:
  `[root@deep /]# ping 192.168.1.1`

  The output should look something like this:

```
PING 192.168.1.1 (192.168.1.1) from 192.168.1.1 : 56 data bytes
64 bytes from 192.168.1.1: icmp_seq=0 ttl=128 time=1.0 ms
64 bytes from 192.168.1.1: icmp_seq=1 ttl=128 time=1.0 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=128 time=1.0 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=128 time=1.0 ms

--- 192.168.1.1 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.0/1.0 ms
```

WARNING: Do not try to `ping` a host in which you have applied the previous `TCP/IP` security settings to prevent your system to respond to `ping` request. Instead try to `ping` another host without this feature enable. Also if you don't receive an answer from the internal host you try to `ping`, verify if your hubs, routers, network cards, and network topology are correct.

If you are able to **`ping`** your internal host, congratulations! Now we must **`ping`** an external network, for instance `216.148.218.195`

- To verify that you can reach the external network, use the command:
  `[root@deep /]# ping 216.148.218.195`

  The output should look something like this:

```
PING 216.148.218.195 (216.148.218.195) from 216.148.218.195 :56 data byte
64 bytes from 216.148.218.195: icmp_seq=0 ttl=128 time=1.0 ms
64 bytes from 216.148.218.195: icmp_seq=1 ttl=128 time=1.0 ms
64 bytes from 216.148.218.195: icmp_seq=2 ttl=128 time=1.0 ms
64 bytes from 216.148.218.195: icmp_seq=3 ttl=128 time=1.0 ms

--- 216.148.218.195 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 1.0/1.0/1.0 ms
```

Step 3
You should now display the routing information with the command **`route`** to see if the hosts have the correct routing entries.

- To display the routing information, use the command:
  `[root@deep /]# route -n`

The output should look something like this:

```
Kernel IP routing table
Destination   Gateway      Genmask        Flags  Metric  Ref    Use   Iface
208.164.186.2 0.0.0.0      255.255.255.255UH     0       0      0     eth0
208.164.186.0 208.164.186.2 255.255.255.0  UG     0       0      0     eth0
208.164.186.0 0.0.0.0      255.255.255.0  U      0       0      0     eth0
127.0.0.0     0.0.0.0      255.0.0.0      U      0       0      0     lo
```

Step 4
Another useful option is "**`netstat -vat`**", which shows all active and listen `TCP` connections.

- To shows all active and listen `TCP` connections, use the command:
  `[root@deep /]#` **`netstat -vat`**

The output may look something similar to this example depending if the related services are running. Be aware that your results will almost certainly vary from the ones shown below:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address     State
tcp        0      0 deep.openna.co:domain   *:*                 LISTEN
tcp        0      0 localhost:domain        *:*                 LISTEN
tcp        0      0 deep.openna.com:ssh     gate.openna.com:1682ESTABLISHED
tcp        0      0 *:webcache              *:*                 LISTEN
tcp        0      0 deep.openar:netbios-ssn *:*                 LISTEN
tcp        0      0 localhost:netbios-ssn   *:*                 LISTEN
tcp        0      0 localhost:1032          localhost:1033      ESTABLISHED
tcp        0      0 localhost:1033          localhost:1032      ESTABLISHED
tcp        0      0 localhost:1030          localhost:1031      ESTABLISHED
tcp        0      0 localhost:1031          localhost:1030      ESTABLISHED
tcp        0      0 localhost:1028          localhost:1029      ESTABLISHED
tcp        0      0 localhost:1029          localhost:1028      ESTABLISHED
tcp        0      0 localhost:1026          localhost:1027      ESTABLISHED
tcp        0      0 localhost:1027          localhost:1026      ESTABLISHED
tcp        0      0 localhost:1024          localhost:1025      ESTABLISHED
tcp        0      0 localhost:1025          localhost:1024      ESTABLISHED
tcp        0      0 deep.openna.com:www     *:*                 LISTEN
tcp        0      0 deep.openna.com:https   *:*                 LISTEN
tcp        0      0 *:389                   *:*                 LISTEN
tcp        0      0 *:ssh                   *:*                 LISTEN
```

Step 5
Sometimes machines on your network will discard your IP packets and finding the offending Gateway responsilbe can be difficult. Fortunately the **`tracepath`** utility attempts to trace the route an IP packet would follow to some Internet host. Choose an Internet host, for instance `64.81.28.146`

- To print the route packets take to network host, use the command:
  `[root@deep /]#` **`tracepath 64.81.28.146`**

The output should look something like this:

```
1?: [LOCALHOST]      pmtu 1500
 1?: 207.35.78.1
 2?: 10.70.1.1
 3?: 206.47.228.178
 4?: 206.108.97.149
```

```
 5?: 206.108.103.214
 6?: 206.108.103.228
 7?: 208.51.134.9
 8?: 208.48.234.189
 9?: 206.132.41.78    asymm 10
10?: 204.246.213.226  asymm 13
11?: 206.253.192.217  asymm 13
12?: 206.253.195.218  asymm 14
13:  64.81.28.146     asymm 15 139ms reached
     Resume: pmtu 1500 hops 13 back 15
```

Step 6
Finally, we will use the `hostname` command of Linux to show if our systems host name is correct.

- To display and print the current host name of your server, use the command:
  ```
  [root@deep /]# hostname
  deep
  ```

The `hostname` command without any options will print the current host name of our system, in this example "`deep`".

Now, it's important to verify if the **F**ully **Q**ualified **D**omain **N**ame (`FQDN`) of our server is reported correctly.

- To display and print the `FQDN` of your server, use the command:
  ```
  [root@deep /]# hostname -f
  deep.openna.com
  ```

The last checkup

If you can answer, "Yes" to each of the questions below, then your network is working and you can continue .

- ✓ Parameters inside `ifcfg-ethN` files are corrects
- ✓ The `/etc/resolv.conf` file contain your primary and secondary Domain Name Server
- ✓ All parameters included in the `/etc/host.conf` file are corrects
- ✓ All parameters included in the `/etc/sysconfig/network` file are corrects
- ✓ The `/etc/hosts` file contain the mapping of your hostnames to IP addresses
- ✓ All network interfaces on the server have the right parameter
- ✓ You can reach the internal and external hosts
- ✓ Your hosts have the correct routing entry
- ✓ The status of the interfaces has been checked and looks fine
- ✓ You are able to print the route packets take to network host

8 Networking - Firewall IPTABLES Packet Filter

In this Chapter

Linux IPTABLES Packet Filter

Abstract

The new Linux kernel, like the two previous kernels, supports a new mechanism for building firewalls, network packet filtering (netfilter). The new mechanism, which is controlled by a tool named iptables, is more sophisticated than the previous ones (ipchains) and more secure. This easy to configure new mechanism is also the first stateful firewall on a Linux operating system. Stateful firewalling represents a major technological jump in the intelligence of a firewall and allows, for example, to block/detect many stealth scans that were undetected on previous generations of Linux firewalls, it also blocks most of the DoS attacks by rating limiting user-defined packet types, since it keeps in memory each connection passing through it.

This new technology implies that if foreign packet tries to enter the network by claiming to be part of an existing connection, IPTABLES can consult its list of connections which it keeps in memory and if it finds that the packet doesn't match any of these, it will drop that packet which will defeat the scan in many cases! I will say that 50% of security on a network depends on a good firewall, and everyone should now run IPTABLES on a Linux server to reach this level of security.

Can someone tell me why I might want something like a commercial firewall product rather than simply using the new iptables tool of Linux and restricting certain packets? What am I losing by using iptables? Now, there is undoubtedly room for a lot of debate on this, iptables is as good, and most of the time better, than commercial firewall packages from a functionality and support standpoint. You will probably have more insight into what's going on in your network using iptables than a commercial solution.

That being said, a lot of corporate types want to tell their shareholders, CEO/CTO/etc. that they have the backing of reputable security Software Company. The firewall could be doing nothing more than passing through all traffic, and still the corporate type would be more comfortable than having to rely on the geeky guy in the corner cube who gets grumpy if you turn the light on before noon.

In the end, a lot of companies want to be able to turn around and demand some sort of restitution from a vendor if the network is breached, whether or not they'd actually get anything or even try. All they can typically do with an open source solution is fire the guy that implemented it. At least some of the commercial firewalls are based on Linux or something similar. It’s quite probable that iptables is secure enough for you but not those engaging in serious amounts of high stakes bond trading.

Doing a cost/benefit analysis and asking a lot of pertinent questions is recommended before spending serious money on a commercial firewall---otherwise you may end up with something inferior to your iptables tool. Quite a few of the NT firewalls are likely to be no better than iptables and the general consensus on bugtraq and NT bugtraq are that NT is *far too insecure* to run a serious firewall.

Prerequisites

Linux IPTABLES requires that the listed software below be already installed on your system to be able to run and work successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have all of these programs installed on your machine before you proceed with this chapter.

- ✓ Kernel 2.4 is required to set up firewalls as well as IP masquerading in your system.
- ✓ iptables package, is the new secure and more powerful program used by Linux to set up firewalls as well as IP masquerading in your system.

➢ To verify if iptables package is installed on your system, use the command:

```
[root@deep /]# rpm -q iptables
package iptables is not installed
```

- To mount your CD-ROM drive before installing the require package, use the command:

```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
had: ATAPI 32X CD-ROM drive, 128kB Cache
mount: block device dev/cdrom is write-protected, mounting read-only
```

- To install the iptables package on your Linux system, use the following command:

```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh iptables-version.i386.rpm
iptables     ##################################################
```

- To unmount your CD-ROM drive, use the following command:

```
[root@deep RPMS]# cd /; umount /mnt/cdrom/
```

What is a Network Firewall Security Policy?

Network firewall security policy defines those services that will be explicitly allowed or denied, how these services will be used and the exceptions to these rules. An organization's overall security policy must be determined according to security and business-need analysis. Since a firewall relates to network security alone, a firewall has little value unless the overall security policy is properly defined. Every rule in the network firewall security policy should be implemented on a firewall. Generally, a firewall uses one of the following methods.

Everything not specifically permitted is denied

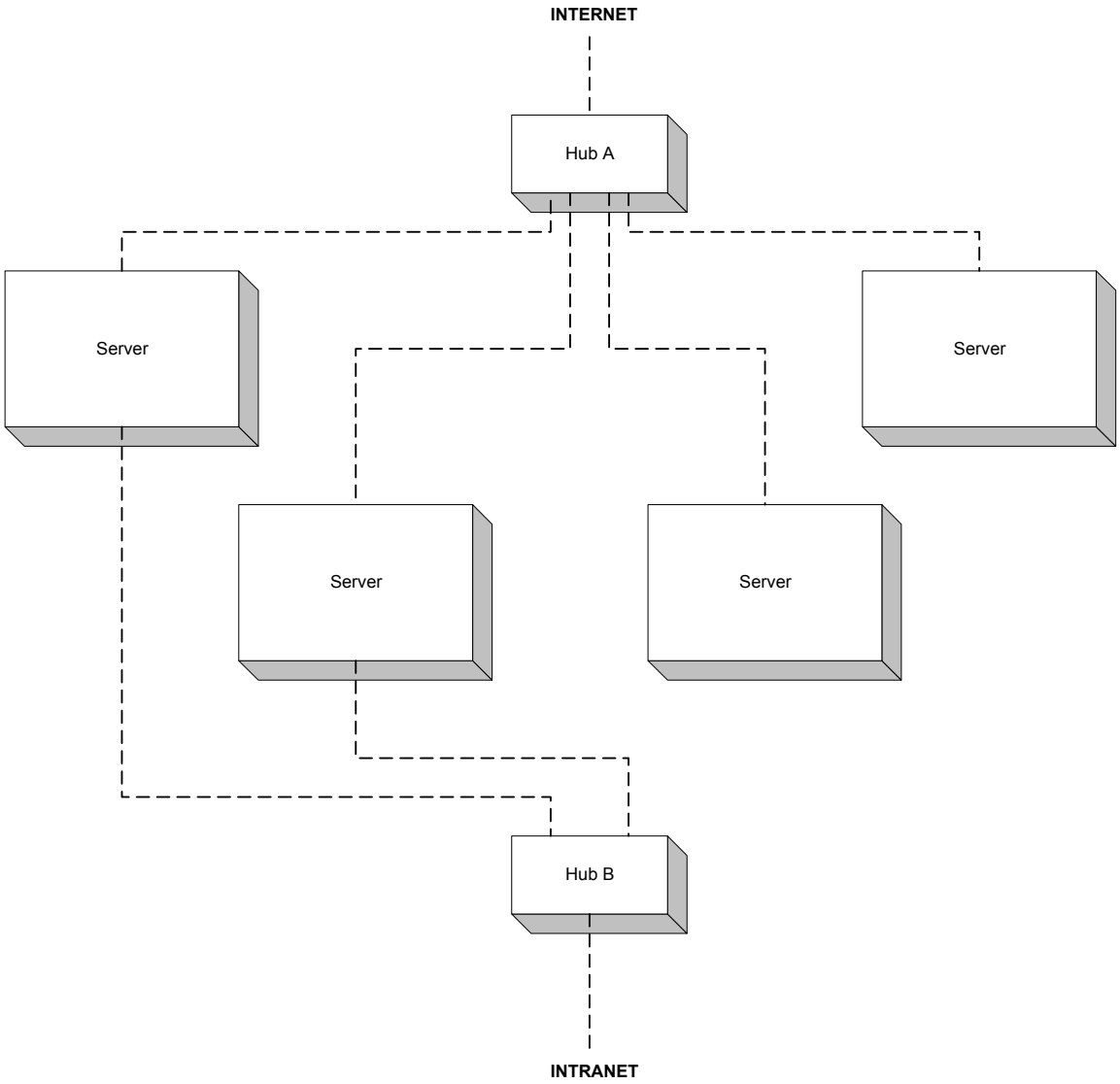
This approach blocks all traffic between two networks except for those services and applications that are permitted. Therefore, each desired service and application should be implemented one by one. No service or application that might be a potential hole on the firewall should be permitted. This is the most secure method, denying services and applications unless explicitly allowed by the administrator. On the other hand, from the point of users, it might be more restrictive and less convenient. This is the method we will use in our Firewall configuration files in this book.

Everything not specifically denied is permitted

This approach allows all traffic between two networks except for those services and applications that are denied. Therefore, each untrusted or potentially harmful service or application should be denied individually. Although this is a flexible and convenient method for the users, it could potentially cause some serious security problems.

The Demilitarized Zone

A demilitarized zone (DMZ) refers to a part of the network that is neither part of the internal network nor directly part of the Internet. Typically, this is the area between your Internet access router and your bastion host (internal network), though it can be between any two policy-enforcing components of your architecture. A DMZ minimizes the exposure of hosts on your external LAN by allowing only recognized and managed services on those hosts to be accessible by hosts on the Internet. This kind of firewall architecture will be the one we will use along this book for all networking services and firewall implementation we want to install on different servers. A demilitarized zone (DMZ) is the most used method in firewall security and most of us use this technique.



The boxes between Hub A and B are in the 'DMZ'. Hub A only routes traffic between the Internet and the DMZ. Hub B only routes traffic between the DMZ and the Intranet. The theory is that all traffic between the Intranet and the Internet has to pass through a machine in the DMZ. The machine in the DMZ can be used to authenticate, record, and control all traffic.

What is Packet Filtering?

Packet Filtering (netfilter) is the type of firewall built into the Linux kernel (as a kernel module, or built right in). A filtering firewall works at the network level. Data is only allowed to leave the system if the firewall rules allow it. As packets arrive they are filtered by their type, source address, destination address, and port information contained in each packet.

Most of the time, packet filtering is accomplished by using a router that can forward packets according to filtering rules. When a packet arrives at the packet-filtering router, the router extracts certain information from the packet header and makes decisions according to the filter rules as to whether the packet will pass through or be discarded.

The following information can be extracted from the packet header:

- ✓ Source IP address
- ✓ Destination IP address
- ✓ TCP/UDP source port
- ✓ TCP/UDP destination port
- ✓ ICMP message type
- ✓ Encapsulated protocol information (TCP, UDP, ICMP or IP tunnel)

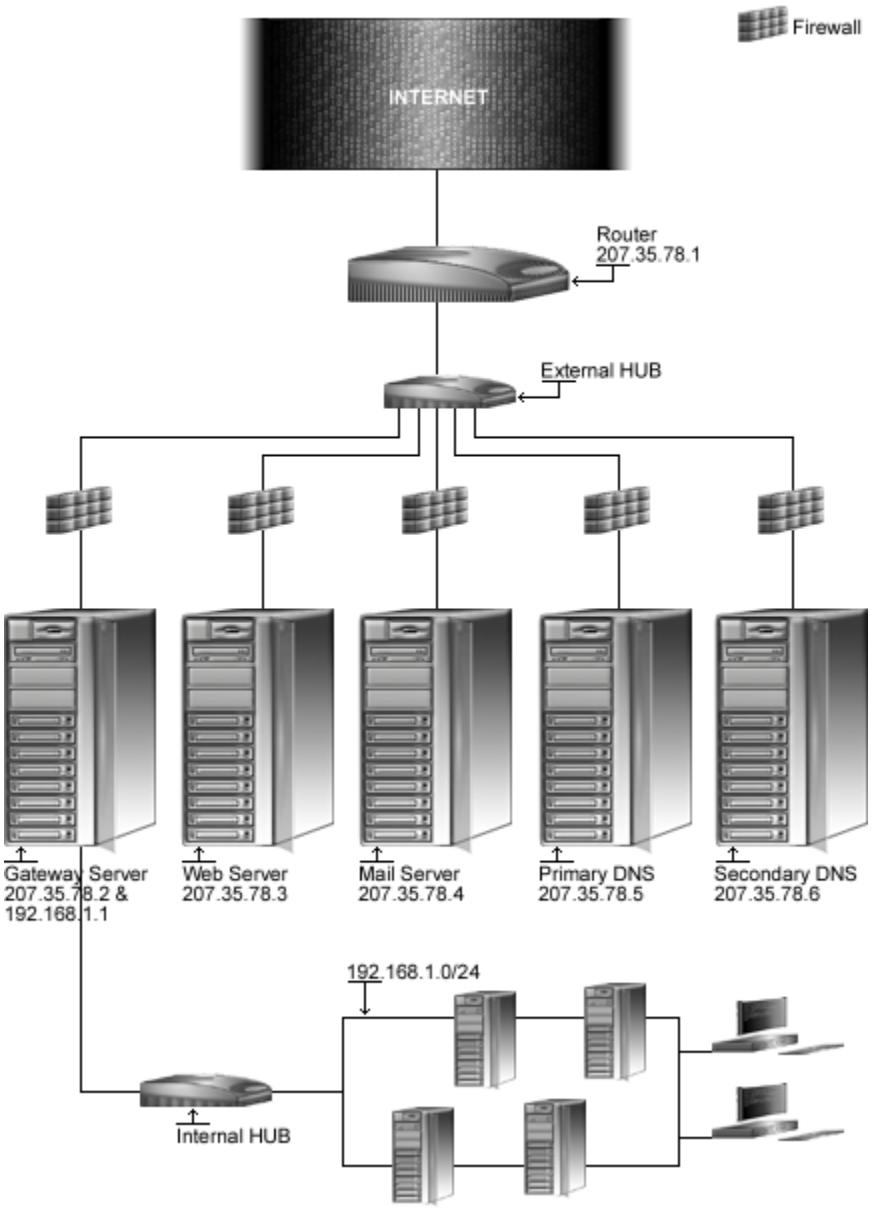
Because very little data is analyzed and logged, filtering firewalls take less CPU power and create less latency in your network. There are lots of ways to structure your network to protect your systems using a firewall.

The topology

All servers should be configured to block at least the unused ports, **even if they are not a firewall server**. This is required for increased security. Imagine that someone gains access to your firewall gateway server: if your servers are not configured to block unused ports, this is a serious network security risk. The same is true for local connections; unauthorized employees can gain access from the inside to your other servers.

In our configuration we will give you five different examples that can help you to configure your firewall rules depending on the type of the server you want to protect and the placement of these servers on your network architecture. It is important to note that the below examples are only a starting point since everyones needs are different, and it is impossible to cover all firewall technique in one chapter, so I recommend you read some good articles or books about firewalls if you need more help to go in deeper with your firewall implementation.

The first example firewall rules file will be for a Web Server, the second for a Mail Server, the third for a Primary Domain Name Server, the fourth for a Secondary Domain Name Server and the last for a Gateway Server that acts as proxy for the inside Workstations and Servers machines. As you can imagine, many possibilities exist for the configuration of your firewall, depending on the tasks you want to assign to the servers in your network. The five examples we show you are the most comon and contain different rules that you can apply or change to fit your own needs. See the diagram below to get an idea.


IPTABLE Firewall
Firewall
INTERNET
Router
207.35.78.1
External HUB
Gateway Server
207.35.78.2 &
192.168.1.1
Web Server
207.35.78.3
Mail Server
207.35.78.4
Primary DNS
207.35.78.5
Secondary DNS
207.35.78.6
192.168.1.0/24
Internal HUB

Building a kernel with `IPTABLES` Firewall support

The first thing you need to do is ensure that your kernel has been built with the netfilter infrastructure compiled in it: netfilter is a general framework inside the Linux kernel, which other things (such as the `iptables` module) can plug into. This means you need kernel 2.4.0 or greater, and answer "**`y`**" or "**`m`**" to the following questions depending of the kernel type you have configured:

*** Networking options**
*
Packet socket (CONFIG_PACKET) [Y/m/n/?]
Packet socket: mmapped IO (CONFIG_PACKET_MMAP) [N/y/?] **y**
Kernel/User netlink socket (CONFIG_NETLINK) [N/y/?] **y**
Routing messages (CONFIG_RTNETLINK) [N/y/?] (NEW) **y**
Netlink device emulation (CONFIG_NETLINK_DEV) [N/y/m/?] (NEW) **y**
Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) [N/y/?] **y**
Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) [N/y/?] (NEW) **y**
Socket Filtering (CONFIG_FILTER) [N/y/?]
Unix domain sockets (CONFIG_UNIX) [Y/m/n/?]
TCP/IP networking (CONFIG_INET) [Y/n/?]
IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] **n**
IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) [N/y/?]
IP: kernel level autoconfiguration (CONFIG_IP_PNP) [N/y/?]
IP: tunneling (CONFIG_NET_IPIP) [N/y/?]
IP: GRE tunnels over IP (CONFIG_NET_IPGRE) [N/y/?]
IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) [N/y/?]
IP: TCP syncookie support (disabled per default) (CONFIG_SYN_COOKIES) [N/y/?] **y**
*
*** IP: Netfilter Configuration**
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) [N/y/?] (NEW)
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) [N/y/?] (NEW) **y**
limit match support (CONFIG_IP_NF_MATCH_LIMIT) [N/y/m/?] (NEW) **y**
MAC address match support (CONFIG_IP_NF_MATCH_MAC) [N/y/m/?] (NEW) **y**
netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) [N/y/m/?] (NEW) **y**
Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) [N/y/m/?] (NEW) **y**
TOS match support (CONFIG_IP_NF_MATCH_TOS) [N/y/m/?] (NEW) **y**
Packet filtering (CONFIG_IP_NF_FILTER) [N/y/m/?] (NEW) **y**
REJECT target support (CONFIG_IP_NF_TARGET_REJECT) [N/y/m/?] (NEW) **y**
Packet mangling (CONFIG_IP_NF_MANGLE) [N/y/m/?] (NEW) **y**
TOS target support (CONFIG_IP_NF_TARGET_TOS) [N/y/m/?] (NEW) **y**
MARK target support (CONFIG_IP_NF_TARGET_MARK) [N/y/m/?] (NEW) **y**
LOG target support (CONFIG_IP_NF_TARGET_LOG) [N/y/m/?] (NEW) **y**

> **WARNING:** If you have followed the Linux Kernel chapter and have recompiled your kernel, all the required options for firewall support, as shown above, are already set. Remember, all servers should be configured to block unused ports, even if they are not a firewall server.

Rules used in the firewall script files

The following is an explanation of a few of the rules that will be used in the Firewalling examples below. This is shown just as a reference, the firewall scripts are well commented and very easy to modify.

Constants used in the firewall scripts files examples

Constants are used for most values. The most basic constants are:

EXTERNAL_INTERFACE
This is the name of the external network interface to the Internet. It's defined as **eth0** in the examples.

LOCAL_INTERFACE_1
This is the name of the internal network interface to the LAN, if any. It's defined as **eth1** in the examples.

LOOPBACK_INTERFACE
This is the name of the loopback interface. It's defined as **lo** in the examples.

IPADDR
This is the IP address of your external interface. It's either a static IP address registered with InterNIC, or else a dynamically assigned address from your ISP (usually via DHCP). For static IP addresses, a script line will automatically find the required IP address on your interface and report it to the firewall program.

INTRANET
This is your LAN network address, if any - the entire range of IP addresses used by the machines on your LAN. These may be statically assigned, or you might run a local DHCP server to assign them. In these examples, the range is 192.168.1.0/24, part of the Class C private address range.

PRIMARY_NAMESERVER
This is the IP address of your Primary DNS Server from your network or your ISP.

SECONDARY_NAMESERVER
This is the IP address of your Secondary DNS Server from your network or your ISP.

NOTE: People with dynamically assigned IPs from an ISP may include the following lines in their declarations for the firewall. The lines will determine the ppp0 IP address, external interface device, and the network of the remote ppp server.

```
EXTERNAL_INTERFACE="ppp0"
IPADDR=`/sbin/ifconfig | grep -A 4 ppp0 | awk '/inet/ { print $2 } ' | sed -e
s/addr://`
```

For DHCP client connection I recommend you to install pump and not dhcpd. Pump is small fast and easy to use than dhcpd. For DHCP connection the value for the IPADDR parameter will be the following line.

```
IPADDR=`/sbin/ifconfig | grep -A 4 eth0 | awk '/inet/ { print $2 } ' | sed -e
s/addr://`
```

Enabling Local Traffic

A firewall has a default policy and a collection of actions to take in response to specific message types. This means that if a given packet has not been selected by any other rule, then the default policy rule will be applied. Since the default policies for all example firewall rule script files in this book are to deny everything, some of these rules must be unset. Local network services do not go through the external network interface. They go through a special, private interface called the loopback interface. None of your local network programs will work until loopback traffic is allowed.

```
# Unlimited traffic on the loopback interface.

iptables -A INPUT  -i $LOOPBACK_INTERFACE  -j ACCEPT
iptables -A OUTPUT -o $LOOPBACK_INTERFACE  -j ACCEPT
```

Source Address Filtering

All IP packet headers contain the source and destination IP addresses and the type of IP protocol message (ICMP, UDP or TCP) the packet contains. The only means of identification under the Internet Protocol (IP) is the source address in the IP packet header. This is a problem that opens the door to source address spoofing, where the sender may replaces its address with either a nonexistent address, or the address of some other site.

```
# Refuse incoming packets pretending to be from the external address.
iptables -A INPUT   -s $IPADDR -j DROP
```

Also, there are at least seven sets of source addresses you should refuse on your external interface in all cases.

These are incoming packets claiming to be from:

- ✓ Your external IP address
- ✓ Class A private IP addresses
- ✓ Class B private IP addresses
- ✓ Class C private IP addresses
- ✓ Class D multicast addresses
- ✓ Class E reserved addresses
- ✓ The loopback interface

With the exception of your own IP address, blocking outgoing packets containing these source addresses protects you from possible configuration errors on your part.

WARNING: Don't forget to exclude your own IP address from outgoing packets blocked. By default I choose to exclude the Class C private IP addresses on the Gateway Server Firewall script file since it's the most used by the majority of people at this time. If you used another class instead of the class C, then you must comment out the lines that refer to your class under the "**SPOOFING & BAD ADDRESSES**" section of the firewall script file. About **SPOOFING & BAD ADDRESSES** in the firewall rules, usually only the Gateway Server must have the rule: `iptables -A INPUT -s $CLASS_C -j DROP` for Class C commented since internal machine on the Class C use the Gateway to have external access. Try to uncomment it and you will see that you could not have access to the Internet from your internal network with IP Class C. Other servers like `Web`, `Mail`, `DNS`, `FTP`, etc must have this line uncommented.

The rest of the rules

Other rules used in the firewall scripts files are:

- ✓ Accessing a Service from the Outside World
- ✓ Offering a Service to the Outside World
- ✓ Masquerading the Internal Machines

The Linux `IPTABLES` firewall scripts files

The tool `iptables` allows you to set up firewalls, IP masquerading, etc. `iptables` talks to the kernel and tells it what packets to filter. Therefore all your firewall setups are stored in the kernel, and thus will be lost on reboot. To avoid this, we recommend using the System V init scripts to make your rules permanent.

To do this, create a firewall script file like shown below in your `/etc/rc.d/init.d` directory for each servers you have. Of course, each server may have a different service to offer and would therefore needs different firewall setup. For this reason, we provide you with five different firewall settings, which you can play with, and examine to fit your needs. Also I assume that you have a minimum amount of knowledge on how filtering firewalls and firewall rules work, since it would take an entire book to cover and talk about Firewalls.

`/etc/rc.d/init.d/iptables:` The Web Server File

This is the configuration script file for our Web Server. This secure configuration allows unlimited traffic on the Loopback interface, `ICMP`, `DNS` forward-only nameserver (53), `SSH` Server and Client (22), `HTTP` Server and Client (80), `HTTPS` Server and Client (443), `SMTP` Client (25), `FTP` Server (20, 21), and Outgoing Traceroute requests by default.

If you don't want some services listed in the firewall rules files for the Web Server that I make ON by default, comment them out with a "**#**" at the beginning of the line. If you want some other services that I commented out with a "**#**", then remove the "**#**" at the beginning of those lines. The text in bold are the parts of the configuration that must be customized and adjusted to satisfy your needs.

Step 1
Create the **`iptables`** script file (`touch /etc/rc.d/init.d/iptables`) on your Web Server and add the following lines:

```
#!/bin/sh
#
# ------------------------------------------------------------------------
# Copyright (C) 1999, 2001 OpenNA.com
# Last modified by Gerhard Mourani:  04-01-2001 <http://www.openna.com/>
# This firewall configuration is suitable for HTTP, HTTPS and FTP Server.
# ------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/iptables.
# chkconfig: - 60 95
# description: Starts and stops the IPTABLES packet filter \
#              used to provide firewall network services.

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network
```

```
# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi

if [ ! -x /sbin/iptables ]; then
    exit 0
fi

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Firewalling: "

# ---------------------------------------------------------------------------
#  Some definitions for easy maintenance.
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

IPADDR=`ifconfig eth0 | fgrep -i inet | cut -d : -f 2 | cut -d \  -f 1`
EXTERNAL_INTERFACE="eth0"                       # Internet connected interface
LOOPBACK_INTERFACE="lo"                         # Your local naming convention
PRIMARY_NAMESERVER="***.**.**.*"                # Your Primary Name Server
SECONDARY_NAMESERVER="***.**.**.*"              # Your Secondary Name Server
#SYSLOG_SERVER="***.**.**.*"                    # Your Syslog Internal Server
SMTP_SERVER="***.**.**.*"                       # Your Central Mail Hub Server

LOOPBACK="127.0.0.0/8"                          # Reserved loopback addr range
CLASS_A="10.0.0.0/8"                            # Class A private networks
CLASS_B="172.16.0.0/12"                         # Class B private networks
CLASS_C="192.168.0.0/16"                        # Class C private networks
CLASS_D_MULTICAST="224.0.0.0/4"                 # Class D multicast addr
CLASS_E_RESERVED_NET="240.0.0.0/5"              # Class E reserved addr
BROADCAST_SRC="0.0.0.0"                         # Broadcast source addr
BROADCAST_DEST="255.255.255.255"                # Broadcast destination addr
PRIVPORTS="0:1023"                              # Privileged port range
UNPRIVPORTS="1024:"                             # Unprivileged port range

# ---------------------------------------------------------------------------

# The SSH client starts at 1023 and works down to 513 for each
# additional simultaneous connection originating from a privileged port.
# Clients can optionally be configured to use only unprivileged ports.
SSH_LOCAL_PORTS="1022:65535"                    # Port range for local clients
SSH_REMOTE_PORTS="513:65535"                    # Port range for remote clients

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ---------------------------------------------------------------------------

# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

    # Remove all existing rules belonging to this filter
    iptables -F

    # Remove any existing user-defined chains.
    iptables -X

    # Set the default policy of the filter to deny.
    iptables -P INPUT   DROP
```

```
    iptables -P OUTPUT  DROP
    iptables -P FORWARD DROP

# ------------------------------------------------------------------------

# LOOPBACK
# --------

    # Unlimited traffic on the loopback interface.

    iptables -A INPUT  -i $LOOPBACK_INTERFACE  -j ACCEPT
    iptables -A OUTPUT -o $LOOPBACK_INTERFACE  -j ACCEPT

# ------------------------------------------------------------------------

# Network Ghouls

    # Deny access to jerks
    # --------------------
    # /etc/rc.d/rc.firewall.blocked contains a list of
    # iptables -A INPUT -i $EXTERNAL_INTERFACE -s address -j DROP
    # rules to block from any access.

    # Refuse any connection from problem sites
    if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    deny_file="/etc/rc.d/rc.firewall.blocked"
    temp_file="/tmp/temp.ip.addresses"
    cat $deny_file | sed -n -e "s/^[ ]*\([0-9.]*\).*$/\1/p" \
    | awk ' $1 ' > $temp_file
    while read ip_addy
    do
      case $ip_addy in
        *) iptables -A INPUT -i $EXTERNAL_INTERFACE -s $ip_addy -j DROP
           iptables -A INPUT -i $EXTERNAL_INTERFACE -d $ip_addy -j DROP
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -s $ip_addy -j REJECT
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -d $ip_addy -j REJECT
        ;;
     esac
    done < $temp_file
    rm -f $temp_file > /dev/null 2>&1
    unset temp_file
    unset deny_file
    fi

# ------------------------------------------------------------------------

# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

    # Refuse incoming packets pretending to be from the external address.
    iptables -A INPUT   -s $IPADDR -j DROP

    # Refuse incoming packets claiming to be from a Class A, B or C private
network
    iptables -A INPUT   -s $CLASS_A -j DROP
    iptables -A INPUT   -s $CLASS_B -j DROP
    iptables -A INPUT   -s $CLASS_C -j DROP

    # Refuse broadcast address SOURCE packets
    iptables -A INPUT   -s $BROADCAST_DEST -j DROP
    iptables -A INPUT   -d $BROADCAST_SRC -j DROP
```

```
    # Refuse Class D multicast addresses
    # Multicast is illegal as a source address.
    # Multicast uses UDP.
    iptables -A INPUT   -s $CLASS_D_MULTICAST -j DROP

    # Refuse Class E reserved IP  addresses
    iptables -A INPUT   -s $CLASS_E_RESERVED_NET -j DROP

    # Refuse special addresses defined as reserved by the IANA.
    # Note:  The remaining reserved addresses are not included
    # filtering them causes problems as reserved blocks are
    # being allocated more often now. The following are based on
    # reservations as listed by IANA as of 2001/01/04. Please regularly
    # check at http://www.iana.org/ for the latest status.

    # Note:  this list includes the loopback, multicast, & reserved addresses.

    # 0.*.*.*           - Can't be blocked for DHCP users.
    # 127.*.*.*                 - LoopBack
    # 169.254.*.*               - Link Local Networks
    # 192.0.2.*                 - TEST-NET
    # 224-255.*.*.*             - Classes D & E, plus unallocated.

    iptables -A INPUT   -s 0.0.0.0/8 -j DROP
    iptables -A INPUT   -s 127.0.0.0/8 -j DROP
    iptables -A INPUT   -s 169.254.0.0/16 -j DROP
    iptables -A INPUT   -s 192.0.2.0/24 -j DROP
    iptables -A INPUT   -s 224.0.0.0/3 -j DROP

# ------------------------------------------------------------------------------

    # UDP TRACEROUTE
    # --------------

    # Traceroute usually uses -S 32769:65535 -D 33434:33523

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $TRACEROUTE_SRC_PORTS \
             -d $IPADDR --destination-port $TRACEROUTE_DEST_PORTS -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $TRACEROUTE_SRC_PORTS \
             --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------------------

    # DNS forward-only nameserver (53)
    # --------------------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             -s $PRIMARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $PRIMARY_NAMESERVER --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $PRIMARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
```

```
        -s $IPADDR --source-port $UNPRIVPORTS \
        -d $PRIMARY_NAMESERVER --destination-port 53 -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
        -s $SECONDARY_NAMESERVER --source-port 53 \
        -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
        -s $IPADDR --source-port $UNPRIVPORTS \
        -d $SECONDARY_NAMESERVER --destination-port 53 -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
        -s $SECONDARY_NAMESERVER --source-port 53 \
        -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
        -s $IPADDR --source-port $UNPRIVPORTS \
        -d $SECONDARY_NAMESERVER --destination-port 53 -j ACCEPT

# ------------------------------------------------------------------

# HTTP server (80)
# ----------------

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
        --source-port $UNPRIVPORTS \
        -d $IPADDR --destination-port 80 -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
        -s $IPADDR --source-port 80 \
        --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# HTTPS server (443)
# ------------------

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
        --source-port $UNPRIVPORTS \
        -d $IPADDR --destination-port 443 -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
        -s $IPADDR --source-port 443 \
        --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# MySQL server (3306)
# -------------------

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
        --source-port $UNPRIVPORTS \
        -d $IPADDR --destination-port 3306 -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
        -s $IPADDR --source-port 3306 \
        --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------

# SSH server (22)
# ---------------
```

```
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $SSH_REMOTE_PORTS \
             -d $IPADDR --destination-port 22 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 22 \
             --destination-port $SSH_REMOTE_PORTS -j ACCEPT


    # SSH client (22)
    # ---------------

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $SSH_LOCAL_PORTS \
             --destination-port 22 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 22 \
             -d $IPADDR --destination-port $SSH_LOCAL_PORTS -j ACCEPT

    # ------------------------------------------------------------------

    # IMAP server (143)
    # -----------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
#             --source-port $UNPRIVPORTS \
#             -d $IPADDR --destination-port 143 -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
#             -s $IPADDR --source-port 143 \
#             --destination-port $UNPRIVPORTS -j ACCEPT

    # IMAP client (143)
    # -----------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
#             --source-port 143 \
#             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
#             -s $IPADDR --source-port $UNPRIVPORTS \
#             --destination-port 143 -j ACCEPT

    # ------------------------------------------------------------------

    # SMTP client (25)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 25 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 25 -j ACCEPT

    # ------------------------------------------------------------------

    # FTP server (21)
    # ---------------

    # incoming request
```

```
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 21 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 21 \
             --destination-port $UNPRIVPORTS -j ACCEPT


    # PORT MODE data channel responses
    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port 20 \
             --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 20 -j ACCEPT


    # PASSIVE MODE data channel responses
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port $UNPRIVPORTS -j ACCEPT

    # ----------------------------------------------------------------

    # SYSLOG client (514)
    # -------------------

#    iptables -A OUTPUT  -o $EXTERNAL_INTERFACE -p udp \
#             -s $IPADDR --source-port 514 \
#             -d $SYSLOG_SERVER --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------------------

# ICMP
# ----

    #    To prevent denial of service attacks based on ICMP bombs, filter
    #    incoming Redirect (5) and outgoing Destination Unreachable (3).
    #    Note, however, disabling Destination Unreachable (3) is not
    #    advisable, as it is used to negotiate packet fragment size.

    # For bi-directional ping.
    #     Message Types:  Echo_Reply (0),  Echo_Request (8)
    #     To prevent attacks, limit the src addresses to your ISP range.
    #
    # For outgoing traceroute.
    #     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
    #     default UDP base: 33434 to base+nhops-1
    #
    # For incoming traceroute.
    #     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
    #     To block this, deny OUTGOING 3 and 11

    #  0: echo-reply (pong)
    #  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
    #  4: source-quench
    #  5: redirect
```

```
    #  8: echo-request (ping)
    # 11: time-exceeded
    # 12: parameter-problem

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type echo-reply \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type destination-unreachable \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type source-quench \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type time-exceeded \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type parameter-problem \
             -d $IPADDR -j ACCEPT


    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type fragmentation-needed -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type source-quench -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type echo-request -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type parameter-problem -j ACCEPT

# --------------------------------------------------------------------------

# Enable logging for selected denied packets

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --destination-port $PRIVPORTS -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --destination-port $UNPRIVPORTS -j DROP


    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type 5 -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type 13/255 -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE  -j REJECT

# --------------------------------------------------------------------------

        ;;
  stop)
        echo -n "Shutting Firewalling: "
```

```
    # Remove all existing rules belonging to this filter
    iptables -F

    # Delete all user-defined chain to this filter
    iptables -X

    # Reset the default policy of the filter to accept.
    iptables -P INPUT   ACCEPT
    iptables -P OUTPUT  ACCEPT
    iptables -P FORWARD ACCEPT

        ;;
  status)
        status iptables
        ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: iptables {start|stop|status|restart|reload}"
        exit 1
esac
echo "done"

exit 0
```

Step 2
Once the script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the script automatically for you at each boot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/iptables
[root@deep /]# chown 0.0 /etc/rc.d/init.d/iptables
```

- To create the symbolic `rc.d` links for your firewall, use the following command:
```
[root@deep /]# chkconfig --add iptables
[root@deep /]# chkconfig --level 2345 iptables on
```

- To manually stop the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables stop
Shutting Firewalling Services:           [OK]
```

- To manually start the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables start
Starting Firewalling Services:           [OK]
```

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time your server boots.

WARNING: Don't try to edit the above script with `MS Wordpad` or some similar program or strange characters will appear in the firewall script file under Linux. Instead use the `vi` editor of Linux to edit the file and everything will work fine for you. You have been warned.

`/etc/rc.d/init.d/iptables`: The Mail Server File

This is the configuration script file for our Mail Server. This secure configuration allows unlimited traffic on the Loopback interface, `ICMP`, `DNS` forward-only nameserver (53), `SSH` Server (22), `SMTP` Server and Client (25), `POP` Server and Client (110), `IMAPS` Server and Client (993), and Outgoing Traceroute requests by default.

If you don't want some services listed in the firewall rules files for the Mail Server that I make ON by default, comment them out with a "**#**" at the beginning of the line. If you want some other services that I commented out with a "**#**", then remove the "**#**" at the beginning of their lines. The text in bold are the parts of the configuration that must be customized and adjusted to satisfy your needs.

Step 1
Create the **`iptables`** script file (`touch /etc/rc.d/init.d/iptables`) on your Mail Server and add the following lines:

```
#!/bin/sh
#
# ------------------------------------------------------------------------
# Copyright (C) 1999, 2001 OpenNA.com
# Last modified by Gerhard Mourani:  04-01-2001 <http://www.openna.com/>
# This firewall configuration is suitable for Central Mail Hub, IMAP/POP
Server.
# ------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/iptables.
# chkconfig: - 60 95
# description: Starts and stops the IPTABLES packet filter \
#              used to provide firewall network services.

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi

if [ ! -x /sbin/iptables ]; then
    exit 0
fi

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Firewalling: "

# ------------------------------------------------------------------------
```

```
#  Some definitions for easy maintenance.
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

IPADDR=`ifconfig eth0 | fgrep -i inet | cut -d : -f 2 | cut -d \  -f 1`
EXTERNAL_INTERFACE="eth0"                       # Internet connected interface
LOOPBACK_INTERFACE="lo"                         # Your local naming convention
PRIMARY_NAMESERVER="***.**.**.*"                # Your Primary Name Server
SECONDARY_NAMESERVER="***.**.**.*"              # Your Secondary Name Server
#SYSLOG_CLIENT="***.**.**.*"                    # Your Syslog Clients IP ranges

LOOPBACK="127.0.0.0/8"                          # Reserved loopback addr range
CLASS_A="10.0.0.0/8"                            # Class A private networks
CLASS_B="172.16.0.0/12"                         # Class B private networks
CLASS_C="192.168.0.0/16"                        # Class C private networks
CLASS_D_MULTICAST="224.0.0.0/4"                 # Class D multicast addr
CLASS_E_RESERVED_NET="240.0.0.0/5"              # Class E reserved addr
BROADCAST_SRC="0.0.0.0"                         # Broadcast source addr
BROADCAST_DEST="255.255.255.255"                # Broadcast destination addr
PRIVPORTS="0:1023"                              # Privileged port range
UNPRIVPORTS="1024:"                             # Unprivileged port range

# ------------------------------------------------------------------------

# The SSH client starts at 1023 and works down to 513 for each
# additional simultaneous connection originating from a privileged port.
# Clients can optionally be configured to use only unprivileged ports.
SSH_LOCAL_PORTS="1022:65535"                    # Port range for local clients
SSH_REMOTE_PORTS="513:65535"                    # Port range for remote clients

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ------------------------------------------------------------------------

# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

    # Remove all existing rules belonging to this filter
    iptables -F

    # Remove any existing user-defined chains.
    iptables -X

    # Set the default policy of the filter to deny.
    iptables -P INPUT   DROP
    iptables -P OUTPUT  DROP
    iptables -P FORWARD DROP

# ------------------------------------------------------------------------

# LOOPBACK
# --------

    # Unlimited traffic on the loopback interface.

    iptables -A INPUT  -i $LOOPBACK_INTERFACE  -j ACCEPT
    iptables -A OUTPUT -o $LOOPBACK_INTERFACE  -j ACCEPT

# ------------------------------------------------------------------------

# Network Ghouls
```

```
    # Deny access to jerks
    # --------------------
    # /etc/rc.d/rc.firewall.blocked contains a list of
    # iptables -A INPUT -i $EXTERNAL_INTERFACE -s address -j DROP
    # rules to block from any access.

    # Refuse any connection from problem sites
    if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    deny_file="/etc/rc.d/rc.firewall.blocked"
    temp_file="/tmp/temp.ip.addresses"
    cat $deny_file | sed -n -e "s/^[ ]*\([0-9.]*\).*$/\1/p" \
    | awk ' $1 ' > $temp_file
    while read ip_addy
    do
      case $ip_addy in
        *) iptables -A INPUT -i $EXTERNAL_INTERFACE -s $ip_addy -j DROP
           iptables -A INPUT -i $EXTERNAL_INTERFACE -d $ip_addy -j DROP
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -s $ip_addy -j REJECT
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -d $ip_addy -j REJECT
        ;;
     esac
    done < $temp_file
    rm -f $temp_file > /dev/null 2>&1
    unset temp_file
    unset deny_file
    fi

# --------------------------------------------------------------------------

# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

    # Refuse incoming packets pretending to be from the external address.
    iptables -A INPUT   -s $IPADDR -j DROP

    # Refuse incoming packets claiming to be from a Class A, B or C private
network
    iptables -A INPUT   -s $CLASS_A -j DROP
    iptables -A INPUT   -s $CLASS_B -j DROP
    iptables -A INPUT   -s $CLASS_C -j DROP

    # Refuse broadcast address SOURCE packets
    iptables -A INPUT   -s $BROADCAST_DEST -j DROP
    iptables -A INPUT   -d $BROADCAST_SRC -j DROP

    # Refuse Class D multicast addresses
    # Multicast is illegal as a source address.
    # Multicast uses UDP.
    iptables -A INPUT   -s $CLASS_D_MULTICAST -j DROP

    # Refuse Class E reserved IP  addresses
    iptables -A INPUT   -s $CLASS_E_RESERVED_NET -j DROP

    # Refuse special addresses defined as reserved by the IANA.
    # Note:  The remaining reserved addresses are not included
    # filtering them causes problems as reserved blocks are
    # being allocated more often now. The following are based on
    # reservations as listed by IANA as of 2001/01/04. Please regularly
    # check at http://www.iana.org/ for the latest status.

    # Note:  this list includes the loopback, multicast, & reserved addresses.
```

```
    # 0.*.*.*                   - Can't be blocked for DHCP users.
    # 127.*.*.*                 - LoopBack
    # 169.254.*.*               - Link Local Networks
    # 192.0.2.*                 - TEST-NET
    # 224-255.*.*.*             - Classes D & E, plus unallocated.

    iptables -A INPUT   -s 0.0.0.0/8 -j DROP
    iptables -A INPUT   -s 127.0.0.0/8 -j DROP
    iptables -A INPUT   -s 169.254.0.0/16 -j DROP
    iptables -A INPUT   -s 192.0.2.0/24 -j DROP
    iptables -A INPUT   -s 224.0.0.0/3 -j DROP

# ------------------------------------------------------------------------

    # UDP TRACEROUTE
    # --------------

    # traceroute usually uses -S 32769:65535 -D 33434:33523

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $TRACEROUTE_SRC_PORTS \
             -d $IPADDR --destination-port $TRACEROUTE_DEST_PORTS -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $TRACEROUTE_SRC_PORTS \
             --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------------

    # DNS forward-only nameserver
    # ---------------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             -s $PRIMARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $PRIMARY_NAMESERVER --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $PRIMARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $PRIMARY_NAMESERVER --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             -s $SECONDARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $SECONDARY_NAMESERVER --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $SECONDARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
```

```
            -d $SECONDARY_NAMESERVER --destination-port 53 -j ACCEPT

    # ------------------------------------------------------------------

    # POP server (110)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
            --source-port $UNPRIVPORTS \
            -d $IPADDR --destination-port 110 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
            -s $IPADDR --source-port 110 \
            --destination-port $UNPRIVPORTS -j ACCEPT

    # POP client (110)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
            --source-port 110 \
            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
            -s $IPADDR --source-port $UNPRIVPORTS \
            --destination-port 110 -j ACCEPT

    # ------------------------------------------------------------------
    # IMAP server (143)
    # -----------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
#             --source-port $UNPRIVPORTS \
#             -d $IPADDR --destination-port 143 -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
#             -s $IPADDR --source-port 143 \
#             --destination-port $UNPRIVPORTS -j ACCEPT

    # IMAP client (143)
    # -----------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
#             --source-port 143 \
#             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
#             -s $IPADDR --source-port $UNPRIVPORTS \
#             --destination-port 143 -j ACCEPT

    # IMAP server over SSL (993)
    # --------------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
            --source-port $UNPRIVPORTS \
            -d $IPADDR --destination-port 993 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
            -s $IPADDR --source-port 993 \
            --destination-port $UNPRIVPORTS -j ACCEPT

    # IMAP client over SSL (993)
    # --------------------------
```

```
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 993 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 993 -j ACCEPT

    # ------------------------------------------------------------------

    # SMTP server (25)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 25 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 25 \
             --destination-port $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SMTP client (25)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 25 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 25 -j ACCEPT

    # ------------------------------------------------------------------

    # SSH server (22)
    # ---------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $SSH_REMOTE_PORTS \
             -d $IPADDR --destination-port 22 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 22 \
             --destination-port $SSH_REMOTE_PORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SYSLOG server (514)
    # -------------------

    # Provides full remote logging. Using  this feature you're able to
    # control all syslog messages on one host.

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp \
#             -s $SYSLOG_CLIENT --source-port $UNPRIVPORTS \
#             -d $IPADDR --destination-port 514 -j ACCEPT

# ------------------------------------------------------------------------

# ICMP
# ----
```

```
    #    To prevent denial of service attacks based on ICMP bombs, filter
    #    incoming Redirect (5) and outgoing Destination Unreachable (3).
    #    Note, however, disabling Destination Unreachable (3) is not
    #    advisable, as it is used to negotiate packet fragment size.

    # For bi-directional ping.
    #     Message Types:  Echo_Reply (0),  Echo_Request (8)
    #     To prevent attacks, limit the src addresses to your ISP range.
    #
    # For outgoing traceroute.
    #     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
    #     default UDP base: 33434 to base+nhops-1
    #
    # For incoming traceroute.
    #     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
    #     To block this, deny OUTGOING 3 and 11

    #  0: echo-reply (pong)
    #  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
    #  4: source-quench
    #  5: redirect
    #  8: echo-request (ping)
    # 11: time-exceeded
    # 12: parameter-problem

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type echo-reply \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type destination-unreachable \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type source-quench \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type time-exceeded \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type parameter-problem \
             -d $IPADDR -j ACCEPT


    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type fragmentation-needed -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type source-quench -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type echo-request -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type parameter-problem -j ACCEPT

# ------------------------------------------------------------------------

# Enable logging for selected denied packets
```

```
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --destination-port $PRIVPORTS -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --destination-port $UNPRIVPORTS -j DROP


    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type 5 -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type 13/255 -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE  -j REJECT

# ------------------------------------------------------------------------------
        ;;
  stop)
        echo -n "Shutting Firewalling: "

    # Remove all existing rules belonging to this filter
    iptables -F

    # Delete all user-defined chain to this filter
    iptables -X

    # Reset the default policy of the filter to accept.
    iptables -P INPUT   ACCEPT
    iptables -P OUTPUT  ACCEPT
    iptables -P FORWARD ACCEPT

        ;;
  status)
        status iptables
        ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: iptables {start|stop|status|restart|reload}"
        exit 1
esac
echo "done"

exit 0
```

Step 2

Once the script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the script automatically for you at each boot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/iptables
[root@deep /]# chown 0.0 /etc/rc.d/init.d/iptables
```

- To create the symbolic `rc.d` links for your firewall, use the following command:
```
[root@deep /]# chkconfig --add iptables
[root@deep /]# chkconfig --level 2345 iptables on
```

- To manually stop the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables stop
Shutting Firewalling Services:           [OK]
```

- To manually start the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables start
Starting Firewalling Services:           [OK]
```

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time your server boots.

WARNING: Don't try to edit the above script with `MS Wordpad` or some similar program or strange characters will appear in the firewall script file under Linux. Instead use the `vi` editor of Linux to edit the file and everything will work fine for you. You have been warned.

`/etc/rc.d/init.d/iptables:` The Primary Domain Name Server File

This is the configuration script file for our Primary Domain Name Server. This secure configuration allows unlimited traffic on the Loopback interface, `ICMP`, `DNS` Full Server and Client (53), `SSH` Server (22), `SMTP` Client (25), and Outgoing Traceroute requests by default.

If you don't want some services listed in the firewall rules files for the Primary Domain Name Server that I make ON by default, comment them out with a "**#**" at the beginning of the line. If you want some other services that I commented out with a "**#**", then remove the "**#**" at the beginning of their lines. The text in bold are the parts of the configuration that must be customized and adjusted to satisfy your needs.

Step 1
Create the **`iptables`** script file (`touch /etc/rc.d/init.d/iptables`) on your Primary Domain Name Server and add the following lines:

```
#!/bin/sh
#
# ------------------------------------------------------------------------
# Copyright (C) 1999, 2001 OpenNA.com
# Last modified by Gerhard Mourani:  04-01-2001 <http://www.openna.com/>
# This firewall configuration is suitable for Primary/Master DNS Server.
# ------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/iptables.
# chkconfig: - 60 95
# description: Starts and stops the IPTABLES packet filter \
#              used to provide firewall network services.

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi

if [ ! -x /sbin/iptables ]; then
    exit 0
fi

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Firewalling: "

# ------------------------------------------------------------------------
#  Some definitions for easy maintenance.
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

IPADDR=`ifconfig eth0 | fgrep -i inet | cut -d : -f 2 | cut -d \  -f 1`
EXTERNAL_INTERFACE="eth0"                       # Internet connected interface
LOOPBACK_INTERFACE="lo"                         # Your local naming convention
SECONDARY_NAMESERVER="***.**.**.*"              # Your Secondary Name Server
#SYSLOG_SERVER="***.**.**.*"                    # Your Syslog Internal Server
SMTP_SERVER="***.**.**.*"                       # Your Central Mail Hub Server

LOOPBACK="127.0.0.0/8"                          # Reserved loopback addr range
CLASS_A="10.0.0.0/8"                            # Class A private networks
CLASS_B="172.16.0.0/12"                         # Class B private networks
CLASS_C="192.168.0.0/16"                        # Class C private networks
CLASS_D_MULTICAST="224.0.0.0/4"                 # Class D multicast addr
CLASS_E_RESERVED_NET="240.0.0.0/5"              # Class E reserved addr
BROADCAST_SRC="0.0.0.0"                         # Broadcast source addr
BROADCAST_DEST="255.255.255.255"                # Broadcast destination addr
PRIVPORTS="0:1023"                              # Privileged port range
UNPRIVPORTS="1024:"                             # Unprivileged port range

# ------------------------------------------------------------------------

# The SSH client starts at 1023 and works down to 513 for each
# additional simultaneous connection originating from a privileged port.
# Clients can optionally be configured to use only unprivileged ports.
SSH_LOCAL_PORTS="1022:65535"                    # Port range for local clients
SSH_REMOTE_PORTS="513:65535"                    # Port range for remote clients
```

```
# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# --------------------------------------------------------------------------

# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

    # Remove all existing rules belonging to this filter
    iptables -F

    # Remove any existing user-defined chains.
    iptables -X

    # Set the default policy of the filter to deny.
    iptables -P INPUT   DROP
    iptables -P OUTPUT  DROP
    iptables -P FORWARD DROP

# --------------------------------------------------------------------------

# LOOPBACK
# --------

    # Unlimited traffic on the loopback interface.

    iptables -A INPUT  -i $LOOPBACK_INTERFACE  -j ACCEPT
    iptables -A OUTPUT -o $LOOPBACK_INTERFACE  -j ACCEPT

# --------------------------------------------------------------------------

# Network Ghouls

    # Deny access to jerks
    # --------------------
    # /etc/rc.d/rc.firewall.blocked contains a list of
    # iptables -A INPUT -i $EXTERNAL_INTERFACE -s address -j DROP
    # rules to block from any access.

    # Refuse any connection from problem sites
    if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    deny_file="/etc/rc.d/rc.firewall.blocked"
    temp_file="/tmp/temp.ip.addresses"
    cat $deny_file | sed -n -e "s/^[ ]*\([0-9.]*\).*$/\1/p" \
    | awk ' $1 ' > $temp_file
    while read ip_addy
    do
      case $ip_addy in
        *) iptables -A INPUT -i $EXTERNAL_INTERFACE -s $ip_addy -j DROP
           iptables -A INPUT -i $EXTERNAL_INTERFACE -d $ip_addy -j DROP
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -s $ip_addy -j REJECT
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -d $ip_addy -j REJECT
        ;;
     esac
    done < $temp_file
    rm -f $temp_file > /dev/null 2>&1
    unset temp_file
    unset deny_file
    fi

# --------------------------------------------------------------------------
```

```
# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

    # Refuse incoming packets pretending to be from the external address.
    iptables -A INPUT   -s $IPADDR -j DROP

    # Refuse incoming packets claiming to be from a Class A, B or C private
network
    iptables -A INPUT   -s $CLASS_A -j DROP
    iptables -A INPUT   -s $CLASS_B -j DROP
    iptables -A INPUT   -s $CLASS_C -j DROP

    # Refuse broadcast address SOURCE packets
    iptables -A INPUT   -s $BROADCAST_DEST -j DROP
    iptables -A INPUT   -d $BROADCAST_SRC -j DROP

    # Refuse Class D multicast addresses
    # Multicast is illegal as a source address.
    # Multicast uses UDP.
    iptables -A INPUT   -s $CLASS_D_MULTICAST -j DROP

    # Refuse Class E reserved IP  addresses
    iptables -A INPUT   -s $CLASS_E_RESERVED_NET -j DROP

    # Refuse special addresses defined as reserved by the IANA.
    # Note:  The remaining reserved addresses are not included
    # filtering them causes problems as reserved blocks are
    # being allocated more often now. The following are based on
    # reservations as listed by IANA as of 2001/01/04. Please regularly
    # check at http://www.iana.org/ for the latest status.

    # Note:  this list includes the loopback, multicast, & reserved addresses.

    # 0.*.*.*                   - Can't be blocked for DHCP users.
    # 127.*.*.*                 - LoopBack
    # 169.254.*.*               - Link Local Networks
    # 192.0.2.*                 - TEST-NET
    # 224-255.*.*.*             - Classes D & E, plus unallocated.

    iptables -A INPUT   -s 0.0.0.0/8 -j DROP
    iptables -A INPUT   -s 127.0.0.0/8 -j DROP
    iptables -A INPUT   -s 169.254.0.0/16 -j DROP
    iptables -A INPUT   -s 192.0.2.0/24 -j DROP
    iptables -A INPUT   -s 224.0.0.0/3 -j DROP

# ------------------------------------------------------------------------------

    # UDP TRACEROUTE
    # --------------

    # Traceroute usually uses -S 32769:65535 -D 33434:33523

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $TRACEROUTE_SRC_PORTS \
             -d $IPADDR --destination-port $TRACEROUTE_DEST_PORTS -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $TRACEROUTE_SRC_PORTS \
             --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT
```

```
# ------------------------------------------------------------------------

    # DNS: full server (53)
    # ---------------------

    # server/client to server query or response

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 53 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port 53 \
             --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port 53 \
             -d $IPADDR --destination-port 53 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port 53 \
             --destination-port 53 -j ACCEPT


    # DNS client (53)
    # ---------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 53 -j ACCEPT


    # DNS Zone Transfers (53)
    # -----------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             -s $SECONDARY_NAMESERVER --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 53 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port 53 \
             -d $SECONDARY_NAMESERVER --destination-port $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SSH server (22)
    # ---------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $SSH_REMOTE_PORTS \
             -d $IPADDR --destination-port 22 -j ACCEPT
```

```
    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 22 \
             --destination-port $SSH_REMOTE_PORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SYSLOG client (514)
    # -------------------

#    iptables -A OUTPUT  -o $EXTERNAL_INTERFACE -p udp \
#             -s $IPADDR --source-port 514 \
#             -d $SYSLOG_SERVER --destination-port $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SMTP client (25)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $SMTP_SERVER --source-port 25 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $SMTP_SERVER --destination-port 25 -j ACCEPT

# ----------------------------------------------------------------------------

# ICMP
# ----

    #    To prevent denial of service attacks based on ICMP bombs, filter
    #    incoming Redirect (5) and outgoing Destination Unreachable (3).
    #    Note, however, disabling Destination Unreachable (3) is not
    #    advisable, as it is used to negotiate packet fragment size.

    # For bi-directional ping.
    #     Message Types:  Echo_Reply (0),  Echo_Request (8)
    #     To prevent attacks, limit the src addresses to your ISP range.
    #
    # For outgoing traceroute.
    #     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
    #     default UDP base: 33434 to base+nhops-1
    #
    # For incoming traceroute.
    #     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
    #     To block this, deny OUTGOING 3 and 11

    #  0: echo-reply (pong)
    #  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
    #  4: source-quench
    #  5: redirect
    #  8: echo-request (ping)
    # 11: time-exceeded
    # 12: parameter-problem

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type echo-reply \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type destination-unreachable \
```

```
            -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type source-quench \
            -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type time-exceeded \
            -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type parameter-problem \
            -d $IPADDR -j ACCEPT


    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type fragmentation-needed -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type source-quench -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type echo-request -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type parameter-problem -j ACCEPT

# --------------------------------------------------------------------------

# Enable logging for selected denied packets

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
            --destination-port $PRIVPORTS -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
            --destination-port $UNPRIVPORTS -j DROP


    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type 5 -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type 13/255 -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE  -j REJECT

# --------------------------------------------------------------------------

        ;;
  stop)
        echo -n "Shutting Firewalling: "

    # Remove all existing rules belonging to this filter
    iptables -F

    # Delete all user-defined chain to this filter
    iptables -X

    # Reset the default policy of the filter to accept.
    iptables -P INPUT   ACCEPT
    iptables -P OUTPUT  ACCEPT
```

```
    iptables -P FORWARD ACCEPT

        ;;
  status)
        status iptables
        ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: iptables {start|stop|status|restart|reload}"
        exit 1
esac
echo "done"

exit 0
```

Step 2
Once the script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the script automatically for you at each boot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/iptables
[root@deep /]# chown 0.0 /etc/rc.d/init.d/iptables
```

- To create the symbolic `rc.d` links for your firewall, use the following command:
```
[root@deep /]# chkconfig --add iptables
[root@deep /]# chkconfig --level 2345 iptables on
```

- To manually stop the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables stop
Shutting Firewalling Services:           [OK]
```

- To manually start the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables start
Starting Firewalling Services:           [OK]
```

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time your server boots.

WARNING: Don't try to edit the above script with `MS Wordpad` or some similar program or strange characters will appear in the firewall script file under Linux. Instead use the `vi` editor of Linux to edit the file and everything will work fine for you. You have been warned.

/etc/rc.d/init.d/iptables: The Secondary Domain Name Server File

This is the configuration script file for our Secondary Domain Name Server. This secure configuration allows unlimited traffic on the Loopback interface, ICMP, DNS Full Server and Client (53), SSH Server (22), SMTP Client (25), and Outgoing Traceroute requests by default.

If you don't want some services listed in the firewall rules files for the Secondary Domain Name Server that I make ON by default, comment them out with a "**#**" at the beginning of the line. If you want some other services that I commented out with a "**#**", then remove the "**#**" at the beginning of their lines. The text in bold are the parts of the configuration that must be customized and adjusted to satisfy your needs.

Step 1
Create the **iptables** script file (`touch /etc/rc.d/init.d/iptables`) on your Secondary Domain Name Server and add the following lines:

```
#!/bin/sh
#
# ------------------------------------------------------------------------
# Copyright (C) 1999, 2001 OpenNA.com
# Last modified by Gerhard Mourani:  04-01-2001 <http://www.openna.com/>
# This firewall configuration is suitable for Secondary/Slave DNS Server.
# ------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/iptables.
# chkconfig: - 60 95
# description: Starts and stops the IPTABLES packet filter \
#              used to provide firewall network services.

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi

if [ ! -x /sbin/iptables ]; then
    exit 0
fi

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Firewalling: "

# ------------------------------------------------------------------------
#  Some definitions for easy maintenance.
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

IPADDR=`ifconfig eth0 | fgrep -i inet | cut -d : -f 2 | cut -d \  -f 1`
EXTERNAL_INTERFACE="eth0"                       # Internet connected interface
LOOPBACK_INTERFACE="lo"                         # Your local naming convention
PRIMARY_NAMESERVER="***.**.**.*"                # Your Primary Name Server
#SYSLOG_SERVER="***.**.**.*"                    # Your Syslog Internal Server
SMTP_SERVER="***.**.**.*"                       # Your Central Mail Hub Server
```

```
LOOPBACK="127.0.0.0/8"                          # Reserved loopback addr range
CLASS_A="10.0.0.0/8"                            # Class A private networks
CLASS_B="172.16.0.0/12"                         # Class B private networks
CLASS_C="192.168.0.0/16"                        # Class C private networks
CLASS_D_MULTICAST="224.0.0.0/4"                 # Class D multicast addr
CLASS_E_RESERVED_NET="240.0.0.0/5"              # Class E reserved addr
BROADCAST_SRC="0.0.0.0"                         # Broadcast source addr
BROADCAST_DEST="255.255.255.255"                # Broadcast destination addr
PRIVPORTS="0:1023"                              # Privileged port range
UNPRIVPORTS="1024:"                             # Unprivileged port range

# ------------------------------------------------------------------------

# The SSH client starts at 1023 and works down to 513 for each
# additional simultaneous connection originating from a privileged port.
# Clients can optionally be configured to use only unprivileged ports.
SSH_LOCAL_PORTS="1022:65535"                    # Port range for local clients
SSH_REMOTE_PORTS="513:65535"                    # Port range for remote clients

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ------------------------------------------------------------------------

# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

    # Remove all existing rules belonging to this filter
    iptables -F

    # Remove any existing user-defined chains.
    iptables -X

    # Set the default policy of the filter to deny.
    iptables -P INPUT   DROP
    iptables -P OUTPUT  DROP
    iptables -P FORWARD DROP

# ------------------------------------------------------------------------

# LOOPBACK
# --------

    # Unlimited traffic on the loopback interface.

    iptables -A INPUT  -i $LOOPBACK_INTERFACE  -j ACCEPT
    iptables -A OUTPUT -o $LOOPBACK_INTERFACE  -j ACCEPT

# ------------------------------------------------------------------------

# Network Ghouls

    # Deny access to jerks
    # --------------------
    # /etc/rc.d/rc.firewall.blocked contains a list of
    # iptables -A INPUT -i $EXTERNAL_INTERFACE -s address -j DROP
    # rules to block from any access.

    # Refuse any connection from problem sites
    if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    deny_file="/etc/rc.d/rc.firewall.blocked"
    temp_file="/tmp/temp.ip.addresses"
```

```
    cat $deny_file | sed -n -e "s/^[ ]*\([0-9.]*\).*$/\1/p" \
    | awk ' $1 ' > $temp_file
    while read ip_addy
    do
      case $ip_addy in
        *) iptables -A INPUT -i $EXTERNAL_INTERFACE -s $ip_addy -j DROP
           iptables -A INPUT -i $EXTERNAL_INTERFACE -d $ip_addy -j DROP
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -s $ip_addy -j REJECT
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -d $ip_addy -j REJECT
        ;;
     esac
    done < $temp_file
    rm -f $temp_file > /dev/null 2>&1
    unset temp_file
    unset deny_file
    fi

# --------------------------------------------------------------------------

# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

    # Refuse incoming packets pretending to be from the external address.
    iptables -A INPUT   -s $IPADDR -j DROP

    # Refuse incoming packets claiming to be from a Class A, B or C private
network
    iptables -A INPUT   -s $CLASS_A -j DROP
    iptables -A INPUT   -s $CLASS_B -j DROP
    iptables -A INPUT   -s $CLASS_C -j DROP

    # Refuse broadcast address SOURCE packets
    iptables -A INPUT   -s $BROADCAST_DEST -j DROP
    iptables -A INPUT   -d $BROADCAST_SRC -j DROP

    # Refuse Class D multicast addresses
    # Multicast is illegal as a source address.
    # Multicast uses UDP.
    iptables -A INPUT   -s $CLASS_D_MULTICAST -j DROP

    # Refuse Class E reserved IP  addresses
    iptables -A INPUT   -s $CLASS_E_RESERVED_NET -j DROP

    # Refuse special addresses defined as reserved by the IANA.
    # Note:  The remaining reserved addresses are not included
    # filtering them causes problems as reserved blocks are
    # being allocated more often now. The following are based on
    # reservations as listed by IANA as of 2001/01/04. Please regularly
    # check at http://www.iana.org/ for the latest status.

    # Note:  this list includes the loopback, multicast, & reserved addresses.

    # 0.*.*.*                   - Can't be blocked for DHCP users.
    # 127.*.*.*                 - LoopBack
    # 169.254.*.*               - Link Local Networks
    # 192.0.2.*                 - TEST-NET
    # 224-255.*.*.*             - Classes D & E, plus unallocated.

    iptables -A INPUT   -s 0.0.0.0/8 -j DROP
    iptables -A INPUT   -s 127.0.0.0/8 -j DROP
    iptables -A INPUT   -s 169.254.0.0/16 -j DROP
```

```
    iptables -A INPUT   -s 192.0.2.0/24 -j DROP
    iptables -A INPUT   -s 224.0.0.0/3 -j DROP

# ------------------------------------------------------------------------

    # UDP TRACEROUTE
    # --------------

    # traceroute usually uses -S 32769:65535 -D 33434:33523

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $TRACEROUTE_SRC_PORTS \
             -d $IPADDR --destination-port $TRACEROUTE_DEST_PORTS -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $TRACEROUTE_SRC_PORTS \
             --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------------

    # DNS: full server (53)
    # ---------------------

    # server/client to server query or response

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 53 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port 53 \
             --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port 53 \
             -d $IPADDR --destination-port 53 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port 53 \
             --destination-port 53 -j ACCEPT


    # DNS client (53)
    # ---------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 53 -j ACCEPT


    # DNS Zone Transfers (53)
```

```
    # ------------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             -s $PRIMARY_NAMESERVER --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 53 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port 53 \
             -d $PRIMARY_NAMESERVER --destination-port $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SSH server (22)
    # ---------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $SSH_REMOTE_PORTS \
             -d $IPADDR --destination-port 22 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 22 \
             --destination-port $SSH_REMOTE_PORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SYSLOG client (514)
    # -------------------

#    iptables -A OUTPUT  -o $EXTERNAL_INTERFACE -p udp \
#             -s $IPADDR --source-port 514 \
#             -d $SYSLOG_SERVER --destination-port $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # SMTP client (25)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $SMTP_SERVER --source-port 25 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $SMTP_SERVER --destination-port 25 -j ACCEPT

# ------------------------------------------------------------------------

    # FTP server (21)
    # ---------------

    # incoming request
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 21 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 21 \
             --destination-port $UNPRIVPORTS -j ACCEPT


    # PORT MODE data channel responses
    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port 20 \
```

```
            --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
            --source-port $UNPRIVPORTS \
            -d $IPADDR --destination-port 20 -j ACCEPT


    # PASSIVE MODE data channel responses
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
            --source-port $UNPRIVPORTS \
            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
            -s $IPADDR --source-port $UNPRIVPORTS \
            --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------------------

# ICMP
# ----

    #    To prevent denial of service attacks based on ICMP bombs, filter
    #    incoming Redirect (5) and outgoing Destination Unreachable (3).
    #    Note, however, disabling Destination Unreachable (3) is not
    #    advisable, as it is used to negotiate packet fragment size.

    # For bi-directional ping.
    #     Message Types:  Echo_Reply (0),  Echo_Request (8)
    #     To prevent attacks, limit the src addresses to your ISP range.
    #
    # For outgoing traceroute.
    #     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
    #     default UDP base: 33434 to base+nhops-1
    #
    # For incoming traceroute.
    #     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
    #     To block this, deny OUTGOING 3 and 11

    #  0: echo-reply (pong)
    #  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
    #  4: source-quench
    #  5: redirect
    #  8: echo-request (ping)
    # 11: time-exceeded
    # 12: parameter-problem

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type echo-reply \
            -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type destination-unreachable \
            -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type source-quench \
            -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type time-exceeded \
            -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
```

```
            --icmp-type parameter-problem \
            -d $IPADDR -j ACCEPT


    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type fragmentation-needed -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type source-quench -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type echo-request -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
            -s $IPADDR --icmp-type parameter-problem -j ACCEPT

# --------------------------------------------------------------------------

# Enable logging for selected denied packets

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
            --destination-port $PRIVPORTS -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
            --destination-port $UNPRIVPORTS -j DROP


    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type 5 -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
            --icmp-type 13/255 -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE  -j REJECT

# --------------------------------------------------------------------------

      ;;
  stop)
      echo -n "Shutting Firewalling: "

    # Remove all existing rules belonging to this filter
    iptables -F

    # Delete all user-defined chain to this filter
    iptables -X

    # Reset the default policy of the filter to accept.
    iptables -P INPUT   ACCEPT
    iptables -P OUTPUT  ACCEPT
    iptables -P FORWARD ACCEPT

      ;;
  status)
      status iptables
      ;;
  restart|reload)
      $0 stop
      $0 start
      ;;
  *)
```

```
        echo "Usage: iptables {start|stop|status|restart|reload}"
        exit 1
esac
echo "done"

exit 0
```

Step 2
Once the script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the script automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/iptables
[root@deep /]# chown 0.0 /etc/rc.d/init.d/iptables
```

- To create the symbolic `rc.d` links for your firewall, use the following command:
```
[root@deep /]# chkconfig --add iptables
[root@deep /]# chkconfig --level 2345 iptables on
```

- To manually stop the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables stop
Shutting Firewalling Services:           [OK]
```

- To manually start the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables start
Starting Firewalling Services:           [OK]
```

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time your server boots.

WARNING: Don't try to edit the above script with `MS Wordpad` or some similar program or strange characters will appear in the firewall script file under Linux. Instead use the `vi` editor of Linux to edit the file and everything will work fine for you. You have been warned.

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

9 Networking - Firewall `Masquerading & Forwarding`

In this Chapter

Recommended RPM packages to be installed for a `Gateway` Server

A minimal configuration provides the basic set of packages required by the Linux operating system. A minimal configuration is a perfect starting point for building a secure operating system. Below is the list of all recommended RPM packages required to run your Linux server as a Gateway/Firewall Server. Remember that a Gateway Server is nothing other than a Linux server with a big firewall, which forwards all internal traffic to the Internet.

This configuration assumes that your kernel is a monolithic kernel. Also, I suppose that you will install `IPTABLES` by RPM package. Therefore, `IPTABLES` RPM package is already included in the list below as you can see. All security tools are not installed, it is yours to install them as you need by RPM since compiler packages are not installed and included in the list.

basesystem	e2fsprogs	**iptables**	openssh-clients	slang
bash	ed	kernel	openssh-server	slocate
bdflush	file	less	openssl	sysklogd
bind	filesystem	libstdc++	pam	syslinux
bzip2	fileutils	libtermcap	passwd	SysVinit
chkconfig	findutils	lilo	popt	tar
console-tools	gawk	logrotate	procps	termcap
cpio	gdbm	losetup	psmisc	textutils
cracklib	gettext	MAKEDEV	pwdb	tmpwatch
cracklib-dicts	glib	man	qmail	utempter
crontabs	glibc	mingetty	readline	util-linux
db1	glibc-common	mktemp	rootfiles	vim-common
db2	grep	mount	rpm	vim-minimal
db3	groff	ncurses	sed	vixie-cron
dev	gzip	net-tools	setup	words
devfsd	info	newt	sh-utils	which
diffutils	initscripts	openssh	shadow-utils	zlib

Tested and fully functional `on` OpenNA.com.

Linux Masquerading & Forwarding

Abstract
Unlike the firewall example configurations in the previous chapter, configuring a Linux Server to masquerade and forward traffic generally from the inside private network that has unregistered IP addresses (i.e. 192.168.1.0/24) to the outside network (i.e. the Internet) requires a special setup of your kernel and your firewall configuration scripts file. This kind of configuration is also known as a Gateway Server or Proxy Server (a machine that serves as a gateway for internal traffic to external traffic). This configuration must be set only if you have the intentions and the needs for this kind of service, and it's for this reason that the configuration of the script file for the Gateway Server is in its own chapter.

Masquerading means that if one of the computers on your local network for which your Linux machine (or Gateway/Proxy) acts as a firewall wants to send something to the outside, your machine can "masquerade" as that computer. In other words, it forwards the traffic to the intended outside destination, but makes it look like it came from the firewall machine itself. It works both ways: if the outside host replies, the Linux firewall will silently forward the traffic to the corresponding local computer. This way, the computers on your local network are completely invisible to the outside world, even though they can reach outside and can receive replies. This makes it possible to have the computers on the local network participate on the Internet even if they don't have officially registered IP addresses.

Building a kernel with Firewall Masquerading & Forwarding support
Once again, the first thing you need to do is ensure that your kernel has been built with the netfilter infrastructure in it: netfilter is a general framework inside the Linux kernel, which other things (such as the iptables module) can plug into.

Step 1
This means you need kernel 2.4.0 or beyond, and answer "**y**" or "**m**" to the following kernel configuration questions. Contrary to previous kernel generations (2.2.x) which only allow to build a modularized kernel with masquerading and forwarding support, the new generation of kernel (2.4.x) lets you build a Linux Gateway Server with features directly included in it by answering "**y**" to the related masquerading and forwarding networking options.

Below I assume that you want to build masquerading and forwarding support as well as the other firewall features as a modules into the kernel, of course if this is not the case, all you have to do is to answer to the related kernel options with "**y**" for yes intend of "**m**" for module. Personally, I prefer to build masquerading and forwarding support directly into the kernel by answering "**y**" to all the questions. But it is up to you to decide which way is the best for your needs. For some of you that aren't sure, I can say that in year 2000 some problems have been found in the tool responsible for loading modules in the system, these problems were related to some bugs in the code of the program, which allowed non root users to gain access to the system.

*** Networking options**
*
Packet socket (CONFIG_PACKET) [Y/m/n/?]
Packet socket: mmapped IO (CONFIG_PACKET_MMAP) [N/y/?] **y**
Kernel/User netlink socket (CONFIG_NETLINK) [N/y/?] **y**
Routing messages (CONFIG_RTNETLINK) [N/y/?] (NEW) **y**
Netlink device emulation (CONFIG_NETLINK_DEV) [N/y/m/?] (NEW) **y**
Network packet filtering (replaces ipchains) (CONFIG_NETFILTER) [N/y/?] **y**
Network packet filtering debugging (CONFIG_NETFILTER_DEBUG) [N/y/?] (NEW) **y**
Socket Filtering (CONFIG_FILTER) [N/y/?]
Unix domain sockets (CONFIG_UNIX) [Y/m/n/?]
TCP/IP networking (CONFIG_INET) [Y/n/?]
IP: multicasting (CONFIG_IP_MULTICAST) [Y/n/?] **n**
IP: advanced router (CONFIG_IP_ADVANCED_ROUTER) [N/y/?] **y**
IP: policy routing (CONFIG_IP_MULTIPLE_TABLES) [N/y/?] (NEW) **y**
IP: use netfilter MARK value as routing key (CONFIG_IP_ROUTE_FWMARK) [N/y/?] (NEW) **y**
IP: fast network address translation (CONFIG_IP_ROUTE_NAT) [N/y/?] (NEW) **y**
IP: equal cost multipath (CONFIG_IP_ROUTE_MULTIPATH) [N/y/?] (NEW) **y**
IP: use TOS value as routing key (CONFIG_IP_ROUTE_TOS) [N/y/?] (NEW) **y**
IP: verbose route monitoring (CONFIG_IP_ROUTE_VERBOSE) [N/y/?] (NEW) **y**
IP: large routing tables (CONFIG_IP_ROUTE_LARGE_TABLES) [N/y/?] (NEW) **y**
IP: kernel level autoconfiguration (CONFIG_IP_PNP) [N/y/?]
IP: tunneling (CONFIG_NET_IPIP) [N/y/m/?]
IP: GRE tunnels over IP (CONFIG_NET_IPGRE) [N/y/m/?]
IP: TCP Explicit Congestion Notification support (CONFIG_INET_ECN) [N/y/?]
IP: TCP syncookie support (disabled per default) (CONFIG_SYN_COOKIES) [N/y/?] **y**
*
*** IP: Netfilter Configuration**
*
Connection tracking (required for masq/NAT) (CONFIG_IP_NF_CONNTRACK) [N/y/m/?] (NEW) **m**
FTP protocol support (CONFIG_IP_NF_FTP) [N/m/?] (NEW) **m**
IP tables support (required for filtering/masq/NAT) (CONFIG_IP_NF_IPTABLES) [N/y/m/?] (NEW) **m**
limit match support (CONFIG_IP_NF_MATCH_LIMIT) [N/m/?] (NEW) **m**
MAC address match support (CONFIG_IP_NF_MATCH_MAC) [N/m/?] (NEW) **m**
netfilter MARK match support (CONFIG_IP_NF_MATCH_MARK) [N/m/?] (NEW) **m**
Multiple port match support (CONFIG_IP_NF_MATCH_MULTIPORT) [N/m/?] (NEW) **m**
TOS match support (CONFIG_IP_NF_MATCH_TOS) [N/m/?] (NEW) **m**
Connection state match support (CONFIG_IP_NF_MATCH_STATE) [N/m/?] (NEW) **m**
Packet filtering (CONFIG_IP_NF_FILTER) [N/m/?] (NEW) **m**
REJECT target support (CONFIG_IP_NF_TARGET_REJECT) [N/m/?] (NEW) **m**
Full NAT (CONFIG_IP_NF_NAT) [N/m/?] (NEW) **m**
MASQUERADE target support (CONFIG_IP_NF_TARGET_MASQUERADE) [N/m/?] (NEW) **m**
REDIRECT target support (CONFIG_IP_NF_TARGET_REDIRECT) [N/m/?] (NEW) **m**
Packet mangling (CONFIG_IP_NF_MANGLE) [N/m/?] (NEW) **m**
TOS target support (CONFIG_IP_NF_TARGET_TOS) [N/m/?] (NEW) **m**
MARK target support (CONFIG_IP_NF_TARGET_MARK) [N/m/?] (NEW) **m**
LOG target support (CONFIG_IP_NF_TARGET_LOG) [N/m/?] (NEW) **m**

If you enabled IP Masquerading and Forwarding support, then the following modules will automatically be compiled into the kernel if you have a monolithic kernel or compiled as modules if you have a modularized kernel. They are needed to make masquerading and other security features for these protocols to work:

```
ip_conntrack.o
ip_conntrack_ftp.o
ip_nat_ftp.o
ip_tables.o
ipt_LOG.o
ipt_MARK.o
ipt_MASQUERADE.o
ipt_REDIRECT.o
ipt_REJECT.o
ipt_TOS.o
ipt_limit.o
ipt_mac.o
ipt_mark.o
ipt_multiport.o
ipt_state.o
ipt_tos.o
iptable_filter.o
iptable_mangle.o
iptable_nat.o
```

WARNING: If you have followed the Linux Kernel chapter and have recompiled your kernel, these options as shown above are already set. Don't forget that only your **Gateway/Proxy Server** needs to have these kernel options (all features under IP: Netfilter Configuration) enabled. They are required to masquerade your Internal Network to the outside and to set ON some other features and security. Remember that other servers like the Web Server, Mail Server, Primary Domain Name Server and Secondary Domain Name Server examples don't need to have these options enabled since they either have a real IP address assigned or don't act as a Gateway for the inside network.

Step 2
The IP masquerading code will only work if IP forwarding is enabled on your system. This feature is by default disabled and you can enable it with the following command:

- To enable IPv4 forwarding on your Linux system, use the following command:
  Edit the **`sysctl.conf`** file (`vi /etc/sysctl.conf`) and add the following lines:

```
# Enable packet forwarding (required only for Gateway, VPN, Proxy, PPP)
net.ipv4.ip_forward = 1
```

You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

WARNING: The IP forwarding line above is only required if you answered "**`y`**" or "**`m`**" to all the kernel options under "**IP: Netfilter Configuration**" and choose to have a server act as a Gateway and masquerade for your inside network.

There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w net.ipv4.ip_forward=1
```

Some Points to Consider

You can safely assume that you are potentially at risk if you connect your system to the Internet. Your gateway to the Internet is your greatest exposure, so we recommend the following:

- ✓ The gateway should not run any more applications than are absolutely necessary.
- ✓ The gateway should strictly limit the type and number of protocols allowed to flow through it (protocols potentially provide security holes, such as FTP and telnet).
- ✓ Any system containing confidential or sensitive information should not be directly accessible from the Internet.

`/etc/rc.d/init.d/iptables`: The Gateway Server File

This is the configuration script file for our Gateway Server. This secure configuration allows unlimited traffic on the Loopback interface, Unlimited traffic within the local network, `ICMP`, `DNS` forward-only nameserver (53), `SSH` Server and Client (22), `HTTP` Client (80), `HTTPS` Client (443), `POP` Client (110), `IMAP` Client (143), `NNTP` NEWS Client (119), `SMTP` Client (25), `TELNET` client (23), `AUTH` client (113), `WHOIS` client (43), `FINGER` client (79), `IRC` Client (6667), `ICQ` Client (4000), `FTP` Client (20, 21), RealAudio / QuickTime Client, and Outgoing Traceroute requests by default.

If you don't want some services listed in the firewall rules files for the Gateway Server that I make ON by default, comment them out with a "**#**" at the beginning of the line. If you want some other services that I commented out with a "**#**", then remove the "**#**" at the beginning of their lines.

If you have a modularized kernel and have configured Masquerading on your server, don't forget to uncomment the modules necessary to masquerade their respective services under the "**`FIREWALL MODULES`**" section of the firewall script file. The text in bold are the parts of the configuration that must be customized and adjusted to satisfy your needs.

Step 1
Create the **`iptables`** script file (`touch /etc/rc.d/init.d/iptables`) on your Gateway Server and add the following lines:

```
#!/bin/sh
#
# ------------------------------------------------------------------------
# Copyright (C) 1999, 2001 OpenNA.com
# Last modified by Gerhard Mourani:  04-01-2001 <http://www.openna.com/>
# This firewall configuration is suitable for Gateway & Proxy Server.
# ------------------------------------------------------------------------
#
# Invoked from /etc/rc.d/init.d/iptables.
# chkconfig: - 60 95
# description: Starts and stops the IPTABLES packet filter \
#              used to provide firewall network services.

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi

if [ ! -x /sbin/iptables ]; then
    exit 0
fi

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Firewalling: "

# ------------------------------------------------------------------------
```

```
#  Some definitions for easy maintenance.
#  EDIT THESE TO SUIT YOUR SYSTEM AND ISP.

IPADDR=`ifconfig eth0 | fgrep -i inet | cut -d : -f 2 | cut -d \  -f 1`
EXTERNAL_INTERFACE="eth0"                       # Internet connected interface
LOOPBACK_INTERFACE="lo"                         # Your local naming convention
LOCAL_INTERFACE_1="eth1"                        # Your Internal LAN interface
INTRANET="***.**.**.*/24"                       # Your Private IP Addr Range
PRIMARY_NAMESERVER="***.**.**.*"                # Your Primary Name Server
SECONDARY_NAMESERVER="***.**.**.*"              # Your Secondary Name Server
#SYSLOG_SERVER="***.**.**.*"                    # Your Syslog Internal Server

LOOPBACK="127.0.0.0/8"                          # Reserved loopback address
range
CLASS_A="10.0.0.0/8"                            # Class A private networks
CLASS_B="172.16.0.0/12"                         # Class B private networks
CLASS_C="192.168.0.0/16"                        # Class C private networks
CLASS_D_MULTICAST="224.0.0.0/4"                 # Class D multicast addr
CLASS_E_RESERVED_NET="240.0.0.0/5"              # Class E reserved addr
BROADCAST_SRC="0.0.0.0"                         # Broadcast source addr
BROADCAST_DEST="255.255.255.255"                # Broadcast destination addr
PRIVPORTS="0:1023"                              # Privileged port range
UNPRIVPORTS="1024:"                             # Unprivileged port range

# ------------------------------------------------------------------------------

# The SSH client starts at 1023 and works down to 513 for each
# additional simultaneous connection originating from a privileged port.
# Clients can optionally be configured to use only unprivileged ports.
SSH_LOCAL_PORTS="1022:65535"                    # Port range for local clients
SSH_REMOTE_PORTS="513:65535"                    # Port range for remote clients

# traceroute usually uses -S 32769:65535 -D 33434:33523
TRACEROUTE_SRC_PORTS="32769:65535"
TRACEROUTE_DEST_PORTS="33434:33523"

# ------------------------------------------------------------------------------

# FIREWALL MODULES
# ----------------

    # Uncomment all of the following modules lines only
    # for modularized kernel system.

    # These modules are necessary to masquerade their respective services.
#    /sbin/modprobe ip_tables
#    /sbin/modprobe iptable_nat
#    /sbin/modprobe ip_conntrack
#    /sbin/modprobe ip_conntrack_ftp
#    /sbin/modprobe ip_tables
#    /sbin/modprobe ip_nat_ftp
#    /sbin/modprobe ipt_LOG
#    /sbin/modprobe ipt_MARK
#    /sbin/modprobe ipt_MASQUERADE
#    /sbin/modprobe ipt_REDIRECT
#    /sbin/modprobe ipt_REJECT
#    /sbin/modprobe ipt_TOS
#    /sbin/modprobe ipt_limit
#    /sbin/modprobe ipt_mac
#    /sbin/modprobe ipt_mark
#    /sbin/modprobe ipt_multiport
#    /sbin/modprobe ipt_state
#    /sbin/modprobe ipt_tos
```

```
#    /sbin/modprobe iptable_mangle

# --------------------------------------------------------------------------

# Default policy is DENY
# Explicitly accept desired INCOMING & OUTGOING connections

    # Remove all existing rules belonging to this filter
    iptables -F
    iptables -F -t nat

    # Remove any existing user-defined chains.
    iptables -X

    # Set the default policy of the filter to deny.
    iptables -P INPUT   DROP
    iptables -P OUTPUT  DROP
    iptables -P FORWARD DROP

# --------------------------------------------------------------------------

# LOOPBACK
# --------

    # Unlimited traffic on the loopback interface.

    iptables -A INPUT  -i $LOOPBACK_INTERFACE  -j ACCEPT
    iptables -A OUTPUT -o $LOOPBACK_INTERFACE  -j ACCEPT

# --------------------------------------------------------------------------

# Unlimited traffic within the local network.

    # All internal machines have access to the fireall machine.

    iptables -A INPUT  -i $LOCAL_INTERFACE_1 -s $INTRANET -j ACCEPT
    iptables -A OUTPUT -o $LOCAL_INTERFACE_1 -d $INTRANET -j ACCEPT

# --------------------------------------------------------------------------

# STATEFUL PART!
# --------------

    # Kill malformed XMAS packets
    iptables -A INPUT -p tcp --tcp-flags ALL ALL -j DROP
    iptables -A FORWARD -p tcp --tcp-flags ALL ALL -j DROP

    # Kill malformed NULL packets
    iptables -A INPUT -p tcp --tcp-flags ALL NONE -j DROP
    iptables -A FORWARD -p tcp --tcp-flags ALL NONE -j DROP

    # Block faked, or "spoofed," packets from getting through the firewall.
    iptables -A FORWARD -i $LOCAL_INTERFACE_1 -s ! $INTRANET -j DROP

    # Allow all internal packets out of our network.
    iptables -A FORWARD -m state --state NEW,ESTABLISHED \
             -i $LOCAL_INTERFACE_1 -s $INTRANET -j ACCEPT

    # Allow the associated packets with those connections back in.
    iptables -A FORWARD -m state --state ESTABLISHED,RELATED \
             -i $EXTERNAL_INTERFACE -s ! $INTRANET -j ACCEPT

    # All internal traffic is masqueraded externally.
```

```
    iptables -A POSTROUTING -t nat -o $EXTERNAL_INTERFACE -j MASQUERADE

# Blocks any forwards that come from Internet connection. Uncomment only for
# users with modem device like "ppp0".

#    iptables -A FORWARD -i $EXTERNAL_INTERFACE -m state \
#    --state NEW,INVALID -j REJECT

# ------------------------------------------------------------------------

# Network Ghouls

    # Deny access to jerks
    # --------------------
    # /etc/rc.d/rc.firewall.blocked contains a list of
    # iptables -A INPUT -i $EXTERNAL_INTERFACE -s address -j DROP
    # rules to block from any access.

    # Refuse any connection from problem sites
    if [ -f /etc/rc.d/rc.firewall.blocked ]; then
    deny_file="/etc/rc.d/rc.firewall.blocked"
    temp_file="/tmp/temp.ip.addresses"
    cat $deny_file | sed -n -e "s/^[ ]*\([0-9.]*\).*$/\1/p" \
    | awk ' $1 ' > $temp_file
    while read ip_addy
    do
      case $ip_addy in
        *) iptables -A INPUT -i $EXTERNAL_INTERFACE -s $ip_addy -j DROP
           iptables -A INPUT -i $EXTERNAL_INTERFACE -d $ip_addy -j DROP
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -s $ip_addy -j REJECT
           iptables -A OUTPUT -o $EXTERNAL_INTERFACE -d $ip_addy -j REJECT
        ;;
     esac
    done < $temp_file
    rm -f $temp_file > /dev/null 2>&1
    unset temp_file
    unset deny_file
    fi

# ------------------------------------------------------------------------

# SPOOFING & BAD ADDRESSES
# Refuse spoofed packets.
# Ignore blatantly illegal source addresses.
# Protect yourself from sending to bad addresses.

    # Refuse incoming packets pretending to be from the external address.
    iptables -A INPUT   -s $IPADDR -j DROP

    # Refuse incoming packets claiming to be from a Class A, B or C private
network
    iptables -A INPUT   -s $CLASS_A -j DROP
    iptables -A INPUT   -s $CLASS_B -j DROP
#    iptables -A INPUT   -s $CLASS_C -j DROP

    # Refuse broadcast address SOURCE packets
    iptables -A INPUT   -s $BROADCAST_DEST -j DROP
    iptables -A INPUT   -d $BROADCAST_SRC -j DROP

    # Refuse Class D multicast addresses
    # Multicast is illegal as a source address.
    # Multicast uses UDP.
    iptables -A INPUT   -s $CLASS_D_MULTICAST -j DROP
```

```
    # Refuse Class E reserved IP  addresses
    iptables -A INPUT   -s $CLASS_E_RESERVED_NET -j DROP

    # Refuse special addresses defined as reserved by the IANA.
    # Note:  The remaining reserved addresses are not included
    # filtering them causes problems as reserved blocks are
    # being allocated more often now. The following are based on
    # reservations as listed by IANA as of 2001/01/04. Please regularly
    # check at http://www.iana.org/ for the latest status.

    # Note:  this list includes the loopback, multicast, & reserved addresses.

    # 0.*.*.*                   - Can't be blocked for DHCP users.
    # 127.*.*.*                 - LoopBack
    # 169.254.*.*               - Link Local Networks
    # 192.0.2.*                 - TEST-NET
    # 224-255.*.*.*             - Classes D & E, plus unallocated.

    iptables -A INPUT   -s 0.0.0.0/8 -j DROP
    iptables -A INPUT   -s 127.0.0.0/8 -j DROP
    iptables -A INPUT   -s 169.254.0.0/16 -j DROP
    iptables -A INPUT   -s 192.0.2.0/24 -j DROP
    iptables -A INPUT   -s 224.0.0.0/3 -j DROP

# ------------------------------------------------------------------------

    # UDP TRACEROUTE
    # --------------

    # traceroute usually uses -S 32769:65535 -D 33434:33523

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $TRACEROUTE_SRC_PORTS \
             -d $IPADDR --destination-port $TRACEROUTE_DEST_PORTS -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $TRACEROUTE_SRC_PORTS \
             --destination-port $TRACEROUTE_DEST_PORTS -j ACCEPT

# ------------------------------------------------------------------------

    # DNS forward-only nameserver
    # ---------------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             -s $PRIMARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $PRIMARY_NAMESERVER --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $PRIMARY_NAMESERVER --source-port 53 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             -d $PRIMARY_NAMESERVER --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             -s $SECONDARY_NAMESERVER --source-port 53 \
```

```
            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
            -s $IPADDR --source-port $UNPRIVPORTS \
            -d $SECONDARY_NAMESERVER --destination-port 53 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
            -s $SECONDARY_NAMESERVER --source-port 53 \
            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
            -s $IPADDR --source-port $UNPRIVPORTS \
            -d $SECONDARY_NAMESERVER --destination-port 53 -j ACCEPT

    # ------------------------------------------------------------------

    # HTTP client (80)
    # ----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
            --source-port 80 \
            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
            -s $IPADDR --source-port $UNPRIVPORTS \
            --destination-port 80 -j ACCEPT

    # ------------------------------------------------------------------

    # HTTPS client (443)
    # ------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
            --source-port 443 \
            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
            -s $IPADDR --source-port $UNPRIVPORTS \
            --destination-port 443 -j ACCEPT

    # ------------------------------------------------------------------

    # WWW-CACHE client
    # ----------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
#            --source-port 3128 \
#            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
#            -s $IPADDR --source-port $UNPRIVPORTS \
#            --destination-port 3128 -j ACCEPT

    # ------------------------------------------------------------------

    # NNTP NEWS client (119)
    # ----------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
            --source-port 119 \
            -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
```

```
        -s $IPADDR --source-port $UNPRIVPORTS \
        --destination-port 119 -j ACCEPT

# ------------------------------------------------------------------

# POP client (110)
# ----------------

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
        --source-port 110 \
        -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
        -s $IPADDR --source-port $UNPRIVPORTS \
        --destination-port 110 -j ACCEPT

# ------------------------------------------------------------------

# IMAP client (143)
# -----------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
#             --source-port 143 \
#             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
#             -s $IPADDR --source-port $UNPRIVPORTS \
#             --destination-port 143 -j ACCEPT

# ------------------------------------------------------------------

# SMTP client (25)
# ----------------

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
        --source-port 25 \
        -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
        -s $IPADDR --source-port $UNPRIVPORTS \
        --destination-port 25 -j ACCEPT

# ------------------------------------------------------------------

# SSH server (22)
# ---------------

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
        --source-port $SSH_REMOTE_PORTS \
        -d $IPADDR --destination-port 22 -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
        -s $IPADDR --source-port 22 \
        --destination-port $SSH_REMOTE_PORTS -j ACCEPT


# SSH client (22)
# ---------------

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
        --source-port 22 \
        -d $IPADDR --destination-port $SSH_LOCAL_PORTS -j ACCEPT
```

```
    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $SSH_LOCAL_PORTS \
             --destination-port 22 -j ACCEPT

    # ------------------------------------------------------------------

    # TELNET client (23)
    # ------------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
#             --source-port 23 \
#             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
#             -s $IPADDR --source-port $UNPRIVPORTS \
#             --destination-port 23 -j ACCEPT

    # ------------------------------------------------------------------

    # AUTH server (113)
    # -----------------

    # Reject, rather than deny, the incoming auth port. (NET-3-HOWTO)
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 113 -j REJECT


    # AUTH client (113)
    # -----------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
#             --source-port 113 \
#             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
#             -s $IPADDR --source-port $UNPRIVPORTS \
#             --destination-port 113 -j ACCEPT

    # ------------------------------------------------------------------

    # WHOIS client (43)
    # -----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 43 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 43 -j ACCEPT

    # ------------------------------------------------------------------

    # FINGER client (79)
    # ------------------

#    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
#             --source-port 79 \
#             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

#    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
#             -s $IPADDR --source-port $UNPRIVPORTS \
```

```
#             --destination-port 79 -j ACCEPT

    # ------------------------------------------------------------------

    # FTP client (21)
    # ---------------

    # outgoing request
    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 21 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 21 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT


    # PORT mode data channel
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port 20 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 20 -j ACCEPT

    # ------------------------------------------------------------------

    # IRC client (6667)
    # -----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 6667 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 6667 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # RealAudio / QuickTime client
    # ----------------------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 554 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 554 -j ACCEPT


    # TCP is a more secure method:  7070:7071
```

```
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 7070:7071 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 7070:7071 -j ACCEPT


    # UDP is the preferred method:  6970:6999
    # For LAN machines, UDP requires the RealAudio masquerading module and
    # the ipmasqadm third-party software.

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 6970:6999 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port 6970:6999 \
             --destination-port $UNPRIVPORTS -j ACCEPT

    # ------------------------------------------------------------------

    # ICQ client (4000)
    # -----------------

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
             --source-port 2000:4000 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 2000:4000 -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --source-port 4000 \
             -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp  \
             -s $IPADDR --source-port $UNPRIVPORTS \
             --destination-port 4000 -j ACCEPT

    # ------------------------------------------------------------------

    # SYSLOG client (514)
    # -------------------

#    iptables -A OUTPUT  -o $EXTERNAL_INTERFACE -p udp \
#             -s $IPADDR --source-port 514 \
#             -d $SYSLOG_SERVER --destination-port $UNPRIVPORTS -j ACCEPT

# ------------------------------------------------------------------------

# ICMP

    #    To prevent denial of service attacks based on ICMP bombs, filter
    #    incoming Redirect (5) and outgoing Destination Unreachable (3).
    #    Note, however, disabling Destination Unreachable (3) is not
    #    advisable, as it is used to negotiate packet fragment size.

    # For bi-directional ping.
    #     Message Types:  Echo_Reply (0),  Echo_Request (8)
    #     To prevent attacks, limit the src addresses to your ISP range.
```

```
    #
    # For outgoing traceroute.
    #     Message Types:  INCOMING Dest_Unreachable (3), Time_Exceeded (11)
    #     default UDP base: 33434 to base+nhops-1
    #
    # For incoming traceroute.
    #     Message Types:  OUTGOING Dest_Unreachable (3), Time_Exceeded (11)
    #     To block this, deny OUTGOING 3 and 11

    #  0: echo-reply (pong)
    #  3: destination-unreachable, port-unreachable, fragmentation-needed, etc.
    #  4: source-quench
    #  5: redirect
    #  8: echo-request (ping)
    # 11: time-exceeded
    # 12: parameter-problem

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type echo-reply \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type destination-unreachable \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type source-quench \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type time-exceeded \
             -d $IPADDR -j ACCEPT

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type parameter-problem \
             -d $IPADDR -j ACCEPT


    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type fragmentation-needed -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type source-quench -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type echo-request -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p icmp  \
             -s $IPADDR --icmp-type parameter-problem -j ACCEPT

# ------------------------------------------------------------------------------

# Enable logging for selected denied packets

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --destination-port $PRIVPORTS -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp  \
             --destination-port $UNPRIVPORTS -j DROP
```

```
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type 5 -j DROP

    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p icmp  \
             --icmp-type 13/255 -j DROP

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE  -j REJECT

# ------------------------------------------------------------------------

        ;;
  stop)
        echo -n "Shutting Firewalling: "

    # Remove all existing rules belonging to this filter
    iptables -F

    # Delete all user-defined chain to this filter
    iptables -X

    # Reset the default policy of the filter to accept.
    iptables -P INPUT   ACCEPT
    iptables -P OUTPUT  ACCEPT
    iptables -P FORWARD ACCEPT

        ;;
  status)
        status iptables
        ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: iptables {start|stop|status|restart|reload}"
        exit 1
esac
echo "done"

exit 0
```

Step 2
Once the script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the script automatically for you at each boot.

- To make this script executable and to change its default permissions, use the command:
  ```
  [root@deep /]# chmod 700 /etc/rc.d/init.d/iptables
  [root@deep /]# chown 0.0 /etc/rc.d/init.d/iptables
  ```

- To create the symbolic `rc.d` links for your firewall, use the following command:
  ```
  [root@deep /]# chkconfig --add iptables
  [root@deep /]# chkconfig --level 2345 iptables on
  ```

- To manually stop the firewall on your system, use the following command:
  ```
  [root@deep /]# /etc/rc.d/init.d/iptables stop
  Shutting Firewalling Services:           [OK]
  ```

- To manually start the firewall on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables start
Starting Firewalling Services:           [OK]
```

Now, your firewall rules are configured to use System V init (System V init is in charge of starting all the normal processes that need to run at boot time) and it will be automatically started each time your server boots.

WARNING: Don't try to edit the above script with `MS Wordpad` or some similar program or strange characters will appear in the firewall script file under Linux. Instead use the `vi` editor of Linux to edit the file and everything will work fine for you. You have been warned.

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Deny access to some address

Sometimes you'll know an address that you would like to block from having any access at all to your server. You can do that by creating the **`rc.firewall.blocked`** file under `/etc/rc.d` directory. Instead of entering the entire `iptables` line per ip address for those jerks on the internet, Michael Brown has provided a bit of code, which is already included in all firewall scripts files in this book, that will take a listing of IP address, strip out any comments and run the resulting list through an `iptables` routine.

The net effect is to have the `/etc/rc.d/rc.firewall.blocked` file increase no more than needed, especially when one might have a large number of IP addresses to deny.

Modification released under the GNU. There is no other error checking done...nor is their any warranty implied or expressed. Use at your own discretion and RISK.

Here is a sample `rc.firewall.blocked` file listing:

```
#    333.444.555.666          # some comment about value one
#    #444.555.666.777         # some more text
#    555.666.777.888          # some comment on value three and line begins with spaces.
#    some text 666.777.888.999  # there might be text here too.

#    This will produce a temp.ipaddresses file listing of:
#    333.444.555.666
#    555.666.777.888

#    Then the case statement will put each IP address into the selected rules of deny/reject.
```

Step 1
Create the **`rc.firewall.blocked`** file (`touch /etc/rc.d/rc.firewall.blocked`) and add inside this file all the IP addresses that you want to block from having any access to your server at all. For example, I've put the following IP addresses in this file:

```
204.254.45.9    # Cracker site with priority 01.
187.231.11.5    # Spam site with priority 07.
#214.34.144.4   # Temporaly reactivated, please verify with log file.
```

Here we can see how this modified code can be useful. Now we can add the bad IP address, with some comments if necessary to remember actions taken for the specified IP address, into the `/etc/rc.d/rc.firewall.blocked` file.

Step 2
Once the "`rc.firewall.blocked`" file has been created, it is time to check its permission mode and change its owner to be the super-user 'root'.

- To make this file executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 644 /etc/rc.d/rc.firewall.blocked
[root@deep /]# chown 0.0 /etc/rc.d/rc.firewall.blocked
```

Further documentation
For more details, there is manual page you can read:

$ iptables-restore (8) - IP packet filter administration
$ iptables-save (8) - Save IP Tables
$ iptables (8) - Restore IP Tables

`IPTABLES` Administrative Tools
The commands listed below are some that we use often, but many more exist, and you should check the manual pages and documentation for more information.

`IPTABLES`
The `iptables` tool is used for the firewall packet filter administration of the Linux system. We can use it to set up a firewall rules file, as we are doing in this book. Once firewall rules have been created we can play with its many commands to maintain, and inspect the rules in the Linux kernel.

- To list all rules in the selected chain, use the command:
```
[root@deep /]# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
```

This command will list all rules in the selected chain. If no chain is selected, all chains are listed.

- To list all input rules in the selected chain, use the command:

```
[root@deep /]# iptables -L INPUT
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  192.168.1.0/24       anywhere
DROP       all  --  204.254.45.9         anywhere
DROP       all  --  187.231.11.5         anywhere
DROP       all  --  207.35.78.5          anywhere
```

This command will list all input rules we have configured in the selected chain.

- To list all output rules in the selected chain, use the command:

```
[root@deep /]# iptables -L OUTPUT
Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere
ACCEPT     all  --  anywhere             192.168.1.0/24
ACCEPT     udp  --  207.35.78.5          207.35.78.3        udp
spt:domain dpt:domain
ACCEPT     tcp  --  207.35.78.5          207.35.78.3        tcp
spts:1024:65535 dpt:domain
```

This command will list all output rules we have configured in the selected chain.

- To list all forward rules in the selected chain, use the command:

```
[root@deep /]# iptables -L FORWARD
Chain FORWARD (policy DROP)
target     prot opt source               destination
DROP       tcp  --  anywhere             anywhere           tcp
DROP       tcp  --  anywhere             anywhere           tcp
DROP       all  -- !192.168.0.0/24       anywhere
ACCEPT     all  --  192.168.0.0/24       anywhere           state NEW
ACCEPT     all  -- !192.168.0.0/24       anywhere           state
```

This command will list all forward rules in the selected chain. This of course works only if you have configured Masquerading on your server (for gateway servers in general).

- To list all rules in numeric output in the selected chain, use the command:

```
[root@deep /]# iptables -nL
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  192.168.1.0/24       0.0.0.0/0
DROP       all  --  204.254.45.9         0.0.0.0/0

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     all  --  0.0.0.0/0            192.168.1.0/24
ACCEPT     udp  --  207.35.78.5          207.35.78.5
```

This command will list all rules in numeric output. All the IP addresses and port numbers will be printed in numeric format.

Part IV Cryptography & Authentication Related Reference

In this Part

Cryptography & Authentication - `GnuPG`
Cryptography & Authentication - `OPENSSL`
Cryptography & Authentication - `OpenSSH`

In this part, we'll talk about three essential programs that I highly recommend you install in all of servers you may run. Those programs are vital to keep communications with your servers secure. Since the beginning of this book, we've been talking about network security and network security means secure communications; therefore it's important to keep all communication as secure as possible.

It's not a plus to install those programs, it's an absolute necessity.

10 Cryptography & Authentication - GnuPG

In this Chapter

Compiling - Optimizing & Installing GnuPG
GnuPG Administrative Tools

Linux GnuPG

Abstract

At this point of our reading, we are ready to compile, configure, optimize and install software on our Linux server. Yes it is time, and we will begin our adventure with the powerful and easy to install GnuPG tool. Why do we choose to begin with GnuPG? The answer is simple, we are playing with a highly secured server and the first action to take each time we want to install some new software on this secured machine is to be absolutely sure that the software in question comes from a trusted source and is unmodified. With the GnuPG tool we can verify the supplied signature and be sure that the software is original. So it is recommended that this program is installed before any others.

Encryption of data sources is an invaluable feature that gives us a high degree of confidentiality for our work. A tool like GnuPG does much more than just encryption of mail messages. It can be used for all kinds of data encryption, and its utilization is only limited by the imagination. The GnuPG RPM package comes already installed on you computer, but this version is not up to date and it's recommended you install the latest release available for your server and CPU architecture.

According to the official [GnuPG README] file:
GnuPG is GNU's tool for secure data communication and storage. It can be used to encrypt data and to create digital signatures. It includes an advanced key management facility and is compliant with the proposed OpenPGP Internet standard as described in RFC2440. Because GnuPG does not use any patented algorithm it is not compatible with PGP2 versions. PGP 2.x uses only IDEA (which is patented worldwide) and RSA (which is patented in the United States until Sep 20, 2000).

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest GnuPG version number is 1.0.6

Packages

The following are based on information as listed by GnuPG as of 2001/05/25. Please regularly check at www.gnupg.org for the latest status.

Pristine source code is available from:
GnuPG Homepage: http://www.gnupg.org/
GnuPG FTP Site: 134.95.80.189
You must be sure to download: gnupg-1.0.6.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install GnuPG, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > GnuPG1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > GnuPG2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff GnuPG1 GnuPG2 > GnuPG-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing GnuPG

Below are the required steps that you must make to configure, compile and optimize the GnuPG software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp gnupg-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf gnupg-version.tar.gz
```

Step 2
In order to check that the version of GnuPG, which you are going to install, is an original and unmodified one, use the commands described below and check the supplied signature. Since we don't have GnuPG already installed in the system, we have to verify the MD5 checksum of the program.

- To verify the MD5 checksum of GnuPG, use the following command:
```
[root@deep /]# md5sum gnupg-version.tar.gz
```

This should yield an output similar to this:
```
7c319a9e5e70ad9bc3bf0d7b5008a508     gnupg-1.0.6.tar.gz
```

Now check that this checksum is exactly the same as the one published on the GnuPG website at the following URL: http://www.gnupg.org/download.html

Step 3
After that, move into the newly created GnuPG directory then configure and optimize it.

- To move into the newly created GnuPG directory use the following command:
```
[root@deep tmp]# cd gnupg-1.0.6/
```

- To configure and optimize GnuPG use the following compile lines:
```
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
./configure \
--prefix=/usr \
--mandir=/usr/share/man \
--infodir=/usr/share/info \
--disable-nls
```

WARNING: Pay special attention to the compile CFLAGS line above. We optimize GnuPG for an i686 CPU architecture with the parameter "-march=i686 and -mcpu=i686". Please don't forget to adjust the CFLAGS line to reflect your own system.

Step 4
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install GnuPG in the server:

```
[root@deep gnupg-1.0.6]# make
[root@deep gnupg-1.0.6]# cd
[root@deep /root]# find /* > GnuPG1
[root@deep /root]# cd /var/tmp/gnupg-1.0.6/
[root@deep gnupg-1.0.6]# make check
[root@deep gnupg-1.0.6]# make install
[root@deep gnupg-1.0.6]# strip /usr/bin/gpg
[root@deep gnupg-1.0.6]# strip /usr/bin/gpgv
[root@deep gnupg-1.0.6]# cd
[root@deep /root]# find /* > GnuPG2
[root@deep /root]# diff GnuPG1 GnuPG2 > GnuPG-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

The **make check** will run any self-tests that come with the package and finally the **strip** command will reduce the size of the gpg and gpgv binaries to get the optimal performance of those programs.

Step 5
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete GnuPG and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf gnupg-version/
[root@deep tmp]# rm -f gnupg-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install GnuPG. It will also remove the GnuPG compressed archive from the /var/tmp/ directory.

GnuPG Administrative Tools

The commands listed below are ones that we use often, but many more exist. Check the manual page gpg (1) for more information.

Creating a key

First of all, we must create a new key-pair (public and private) if this is a first use of the GnuPG software to be able to use its encryption features.

- To create a new key-pair, use the following command:

```
[root@deep /]# gpg --gen-key
gpg (GnuPG) 1.0.6; Copyright (C) 2000 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: /root/.gnupg: directory created
gpg: /root/.gnupg/options: new options file created
gpg: you have to start GnuPG again, so it can read the new options file
```

- We start GnuPG again with the same command:

```
[root@deep /]# gpg --gen-key
gpg (GnuPG) 1.0.6; Copyright (C) 2000 Free Software Foundation, Inc.
This program comes with ABSOLUTELY NO WARRANTY.
This is free software, and you are welcome to redistribute it
under certain conditions. See the file COPYING for details.

gpg: /root/.gnupg/secring.gpg: keyring created
gpg: /root/.gnupg/pubring.gpg: keyring created
Please select what kind of key you want:
   (1) DSA and ElGamal (default)
   (2) DSA (sign only)
   (4) ElGamal (sign and encrypt)
Your selection? 1
DSA keypair will have 1024 bits.
About to generate a new ELG-E keypair.
              minimum keysize is  768 bits
              default keysize is 1024 bits
    highest suggested keysize is 2048 bits
What keysize do you want? (1024) 1024
Requested keysize is 1024 bits
Please specify how long the key should be valid.
         0 = key does not expire
      <n>  = key expires in n days
      <n>w = key expires in n weeks
      <n>m = key expires in n months
```

```
      <n>y = key expires in n years
Key is valid for? (0) 0
Key does not expire at all
Is this correct (y/n)? y

You need a User-ID to identify your key; the software constructs the user
id from Real Name, Comment and Email Address in this form:
    "Heinrich Heine (Der Dichter) <heinrichh@duesseldorf.de>"

Real name: Gerhard Mourani
Email address: gmourani@openna.com
Comment:
You selected this USER-ID:
    "Gerhard Mourani <gmourani@openna.com>"

Change (N)ame, (C)omment, (E)mail or (O)kay/(Q)uit? O
You need a Passphrase to protect your secret key.

Enter passphrase: mypassphrase
Repeat passphrase: mypassphrase

We need to generate a lot of random bytes. It is a good idea to perform
some other action (type on the keyboard, move the mouse, utilize the
disks) during the prime generation; this gives the random number
generator a better chance to gain enough entropy.
++++++++++++++++++++.+++++.+++++.+++++++++++++++.+++++++++++++++..+++++++
++++++++...+++++++++++++++.+++++.++++++++++++++++++++++++++++++++++++++++
..+++++
<+++++..>+++++.................<..+++++.................................
.........+++++^^^^
public and secret key created and signed.
```

A new key-pair is created (secret and public key) in the “root” home directory `~/root`.

Exporting a key/s for a user

Once your own key-pair is created, you can expand your horizons by exporting and distributing your public key over the world. This can be done by publishing it on your homepage, through an available key server on the Internet, or any other available method. `GnuPG` has some useful options to help you publish your public keys.

- To extract your public key in ASCII armored output, use the following command:
  `[root@deep /]#` **`gpg --export –ao UID`**

  As an example:
  `[root@deep /]#` **`gpg --export –ao Gerhard Mourani`**

Where “`--export`” is for extracting Public-key from your pubring encrypted file, “`a`” is to create ASCII armored output that you can mail, publish or put it on a web page, “`o`” to put the result in a file and `UID` represents the user key you want to export.

Importing a key/s

When you receive someone's public key (or some trusted third partly keys) you have to add them to your key database in order to be able to use his/her keys for future encryption, verification and authentication.

- To import Public Keys to your keyring database, use the following command:
  ```
  [root@deep /]# gpg --import filename
  ```

  As an example:
  ```
  [root@deep /]# gpg --import redhat2.asc
  gpg: key DB42A60E: public key imported
  gpg: /root/.gnupg/trustdb.gpg: trustdb created
  gpg: Total number processed: 1
  gpg:               imported: 1
  ```

The above command will append the new key `filename` into the keyring database and will update all already existing keys. It is important to note that `GnuPG` does not import keys that are not self-signed (`asc`).

Key signing

When you import keys into your public keyring database and are sure that the trusted third party is really the person they claim, you can start signing his/her keys. Signing a key certifies that you know the owner of the keys and this leads to the situation where the signature acknowledges that the user ID mentioned in the key is actually the owner of that key.

- To sign the key for company Red Hat that we have added into our keyring database above, use the following command:
  ```
  [root@deep /]# gpg --sign-key UID
  ```

  As an example:
  ```
  [root@deep /]# gpg --sign-key RedHat

  pub  1024D/DB42A60E  created: 1999-09-23 expires: never      trust: -/q
  sub  2048g/961630A2  created: 1999-09-23 expires: never
  (1)  Red Hat, Inc <security@redhat.com>


  pub  1024D/DB42A60E  created: 1999-09-23 expires: never      trust: -/q
               Fingerprint: CA20 8686 2BD6 9DFC 65F6  ECC4 2191 80CD DB42
  A60E

       Red Hat, Inc <security@redhat.com>

  Are you really sure that you want to sign this key
  with your key: "Gerhard Mourani <gmourani@openna.com>"

  Really sign? y

  You need a passphrase to unlock the secret key for
  user: "Gerhard Mourani <gmourani@openna.com>"
  1024-bit DSA key, ID 90883AB4, created 2000-10-24

  Enter passphrase:
  ```

WARNING: You should only sign a key as being authentic when you are ABSOLUTELY SURE that the key is really authentic! You should never sign a key based on any kind of assumption.

Checking the signature

We have shown above how to sign a key, now we will explain how people can verify if the signature is really the good one. Once you have extracted your public key and exported it, everyone who knows or gets your public key should be able to check whether encrypted data from you is also really signed by you.

- To check the signature of encrypted data, use the following command:
  ```
  [root@deep /]# gpg --verify Data
  ```

The "`--verify`" option will check the signature where `Data` is the encrypted data/file you want to verify.

Encrypt and decrypt

After installing, importing, signing and configuring everything in the way that we want, we can start encrypting and decrypting our files.

- To encrypt and sign data for the user Red Hat that we have added on our keyring database above, use the following command:
  ```
  [root@deep /]# gpg -sear RedHat file
  ```

  As an example:
  ```
  [root@deep /]# gpg -sear RedHat Message-to-RedHat.txt

  You need a passphrase to unlock the secret key for
  user: "Gerhard Mourani <gmourani@openna.com>"
  1024-bit DSA key, ID 90883AB4, created 2000-10-24
  Enter passphrase:
  ```

Of the arguments passed, the "`s`" is for signing (To avoid the risk that somebody else claims to be you, it is very useful to sign everything you encrypt), "`e`" for encrypting, "`a`" to create ASCII armored output (".asc" ready for sending by mail), "`r`" to encrypt the user id name and `file` is the message you want to encrypt.

- To decrypt data, use the following command:
  ```
  [root@deep /]# gpg -d file
  ```

  For example:
  ```
  [root@deep /]# gpg -d Message-from-GerhardMourani.asc
  You need a passphrase to unlock the secret key for
  user: "Gerhard Mourani (Open Network Architecture) <gmourani@openna.com>"
  1024-bit DSA key, ID 90883AB4, created 2000-10-24
  Enter passphrase:
  ```

Where "`d`" is for decrypting and `file` is the message you want to decrypt. It is important that the public key of the sender of the message we want to decrypt be in our public keyring database or of course nothing will work.

Some possible uses of GnuPG

GnuPG can be used to:

1. Encrypt data.
2. Create digital signatures.
3. Verify program source integrity.
4. Sign individual sensitive files.

List of installed GnuPG files in your system

> /usr/bin/gpg
> /usr/bin/gpgv
> /usr/lib/gnupg
> /usr/lib/gnupg/rndunix
> /usr/lib/gnupg/rndegd
> /usr/lib/gnupg/tiger
> /usr/share/man/man1/gpg.1
> /usr/share/man/man1/gpgv.1
> /usr/share/gnupg
> /usr/share/gnupg/options.skel
> /usr/share/gnupg/FAQ
> /usr/share/gnupg/faq.html
> /usr/share/info
> /usr/share/info/gpg.info
> /usr/share/info/gpgv.info

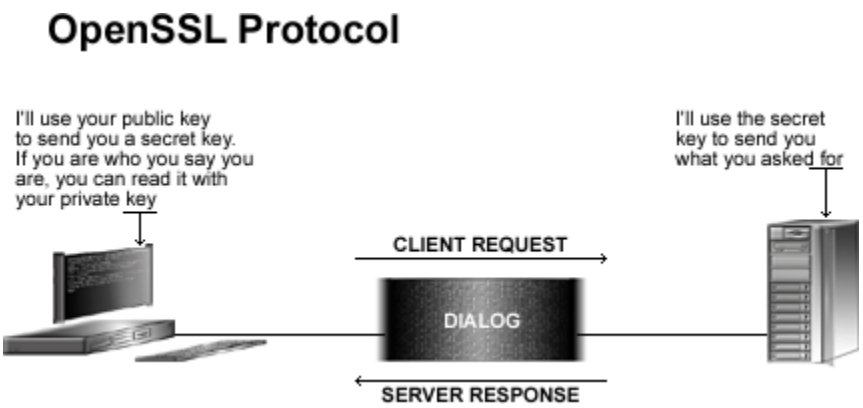
11 Cryptography & Authentication - OPENSSL

In this Chapter

Linux OPENSSL

Abstract
Most server software like IMAP & POP, SSH, Samba, Sendmail, OpenLDAP, FTP, Apache, and others that ask for users to authentice themselves before allowing access to services, by default they transmit the users' login id and password in plain text. Alternatively, encryption mechanisms like SSL ensure safe and secure transactions. With this technology, data going over the network is point-to-point encrypted. Once OpenSSL has been installed on your Linux server you can use it as a third party tool to enable other applications with SSL functionality.

As explained on the [OpenSSL web site]:
The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, fully featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols with full-strength cryptography. Worldwide communities of volunteers that use the Internet to communicate, plan, and develop the OpenSSL toolkit and its related documentation manages the project.



Summary of the Cryptographic Thechnology.

Cryptography Advantages
The main advantages gained by using encryption technology are:

Data Confidentiality
When a message is encrypted, an algorithm converts it into enciphered text that hides the meaning of the message, which can then be sent via any public mechanism, and transforms the input plain text. This process involves a secret key that is used to encrypt and later decrypt the data. Without the secret key, the encrypted data is meaningless.

Data Integrity
A cryptographic checksum, called a message authentication code (MAC), can be calculated on arbitrarily user-supplied text to protect the integrity of the data. The results (text and MAC) are then sent to the receiver who can verify the trial MAC appended to a message by recalculating the MAC for the message, using the appropriate secret key and verifying that it exactly matches the trial MAC.

Authentication
Personal identification is another use of cryptography, where the user/sender knows a secret, which can serve to authenticate his/her identity.

Electronic Signature
A digital signature assures the sender and receiver that the message is authentic and that only the owner of the key could have generated the digital signature.

Disclaimer

This software package uses strong cryptography, so even if it is created, maintained and distributed from liberal countries in Europe (where it is legal to do this), it falls under certain export/import and/or use restrictions in some other parts of the world.

PLEASE REMEMBER THAT EXPORT/IMPORT AND/OR USE OF STRONG CRYPTOGRAPHY SOFTWARE, PROVIDING CRYPTOGRAPHY HOOKS OR EVEN JUST COMMUNICATING TECHNICAL DETAILS ABOUT CRYPTOGRAPHY SOFTWARE IS ILLEGAL IN SOME PARTS OF THE WORLD. SO, WHEN YOU IMPORT THIS PACKAGE TO YOUR COUNTRY, RE-DISTRIBUTE IT FROM THERE OR EVEN JUST EMAIL TECHNICAL SUGGESTIONS OR EVEN SOURCE PATCHES TO THE AUTHOR OR OTHER PEOPLE YOU ARE STRONGLY ADVISED TO PAY CLOSE ATTENTION TO ANY EXPORT/IMPORT AND/OR USE LAWS WHICH APPLY TO YOU. THE AUTHORS OF OPENSSL ARE NOT LIABLE FOR ANY VIOLATIONS YOU MAKE HERE. SO BE CAREFUL, IT IS YOUR RESPONSIBILITY.

These installation instructions assume

Commands are Unix-compatible.
The source path `is /var/tmp` (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest `OpenSSL` version number is `0.9.6a`

Packages

The following are based on information as listed by `OpenSSL` as of 2001/04/13. Please regularly check at www.openssl.org for the latest status.

Pristine source code is available from:
`OpenSSL` Homepage: http://www.openssl.org/
`OpenSSL` FTP Site: `129.132.7.170`
You must be sure to download: `openssl-0.9.6a.tar.gz`

Pristine source

If you don’t use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `OpenSSL`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  `[root@deep /root]#` **`find /* > OpenSSL1`**
- And the following one after you install the software:
  `[root@deep /root]#` **`find /* > OpenSSL2`**

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff OpenSSL1 OpenSSL2 > OpenSSL-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `OpenSSL`

Below are the required steps that you must make to configure, compile and optimize the `OpenSSL` software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp openssl-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf openssl-version.tar.gz
```

Step 2
After that, move into the newly created `OpenSSL` directory then configure and optimize it.

- To move into the newly created `OpenSSL` directory use the following command:
```
[root@deep tmp]# cd openssl-0.9.6a/
```

Step 3
By default, `OpenSSL` source files suppose that your `perl` binary program is located under `/usr/local/bin/perl`. We must modify the "`#!/usr/local/bin/perl`" line in all scripts that rely on `perl` with `OpenSSL` to reflect our `perl` binary program under Linux to be `/usr/bin`.

- To point all `OpenSSL` script files to our `perl` binary, use the following command:
```
[root@deep openssl-0.9.6a]# perl util/perlpath.pl /usr/bin/perl
```

Step 4
At this stage, it is time to configure `OpenSSL` for our system.

- To configure and optimize `OpenSSL`, use the following compile lines:
```
./Configure linux-elf no-idea no-mdc2 no-rc5 no-md2 \
--prefix=/usr \
--openssldir=/usr/share/ssl
```

Step 5
After that there is one file to modify, this file is named `Makefile`. The changes we bring into it is to be adding our optimization flags to speed up the `OpenSSL` software. Also, we change the directory from where manual pages of `OpenSSL` will be installed to be under `/usr/share/man` directory.

a) Edit the **`Makefile`** file (`vi +60 Makefile`) and change the following line:

```
CFLAG= -fPIC -DTHREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -DNO_IDEA -DNO_MDC2
-DNO_RC5 -DNO_MD2 -DL_ENDIAN -DTERMIO -O3 -fomit-frame-pointer -m486 -Wall -
DSHA1_ASM -DMD5_ASM -DRMD160_ASM
```

To read:

```
CFLAG= -fPIC -DTHREADS -D_REENTRANT -DDSO_DLFCN -DHAVE_DLFCN_H -DNO_IDEA -DNO_MDC2
-DNO_RC5 -DNO_MD2 -DL_ENDIAN -DTERMIO -O3 -march=i686 -mcpu=i686 -funroll-loops -
fomit-frame-pointer -Wall -DSHA1_ASM -DMD5_ASM -DRMD160_ASM
```

WARNING: Please don't forget to adjust the above `CFLAG` line to reflect your own system and CPU. In the configure line, we disable support for old encryption mechanism like `MD2`, `MDC2`, `RC5`, and `IDEA` since there are rarely used and required now.

b) Edit the **`Makefile`** file (`vi +174 Makefile`) and change the following line:

```
MANDIR=$(OPENSSLDIR)/man
```

To read:

```
MANDIR=/usr/share/man
```

Step 6
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `OpenSSL` in the server:

```
[root@deep openssl-0.9.6a]# LD_LIBRARY_PATH=`pwd` make
[root@deep openssl-0.9.6a]# LD_LIBRARY_PATH=`pwd` make test
[root@deep openssl-0.9.6a]# cd
[root@deep /root]# find /* > OpenSSL1
[root@deep /root]# cd /var/tmp/openssl-0.9.6a/
[root@deep openssl-0.9.6a]# make install
[root@deep openssl-0.9.6a]# strip /usr/bin/openssl
[root@deep openssl-0.9.6a]# mkdir -p /usr/share/ssl/crl
[root@deep openssl-0.9.6a]# /sbin/ldconfig
[root@deep openssl-0.9.6a]# cd
[root@deep /root]# find /* > OpenSSL2
[root@deep /root]# diff OpenSSL1 OpenSSL2 > OpenSSL-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then test the `OpenSSL` libraries to finally install the binaries and any supporting files into the appropriate locations.

OpenSSL must know where to find the necessary OpenSSL source libraries to compile successfully its required files. With the command "LD_LIBRARY_PATH=`pwd`", as used in the compile lines, we set the PATH environment variable to the default directory where we have uncompressed the OpenSSL source files.

NOTE: It is important to kown that RSAREF is not more needed, because RSA Security Inc. released the RSA public key encryption algorithm into the public domain on September 6, 2000. There is no longer any need to use RSAREF, and since RSAREF is slower than OpenSSL's RSA routines there's good reason not to.

Step 7
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete OpenSSL and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf openssl-version/
[root@deep tmp]# rm -f openssl-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install OpenSSL. It will also remove the OpenSSL compressed archive from the /var/tmp directory

Configuring OpenSSL

After building OpenSSL, your next step is to verify or change, if necessary, options in your OpenSSL configuration files. Those files are:

- ✓ /usr/shared/ssl/openssl.cnf (The OpenSSL Configuration File)
- ✓ /usr/shared/ssl/misc/sign.sh (The mod_ssl CA scrip file to sign certificates)

/usr/shared/ssl/openssl.cnf: The OpenSSL Configuration File

This is the general configuration file for OpenSSL program where you can configure expiration date of your keys, the name of your organization, the address and so on. The most important parameters you may change will be in the **[CA_default]** and especially the **[req_distinguished_name]** sections. We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **openssl.cnf** file (vi /usr/share/ssl/openssl.cnf) and set your needs.

```
#
# OpenSSL example configuration file.
# This is mostly being used for generation of certificate requests.
#

# This definition stops the following lines choking if HOME isn't
# defined.
HOME                    = .
RANDFILE                = $ENV::HOME/.rnd
```

```
# Extra OBJECT IDENTIFIER info:
#oid_file               = $ENV::HOME/.oid
oid_section             = new_oids

# To use this configuration file with the "-extfile" option of the
# "openssl x509" utility, name here the section containing the
# X.509v3 extensions to use:
# extensions            =
# (Alternatively, use a configuration file that has only
# X.509v3 extensions in its main [= default] section.)

[ new_oids ]

# We can add new OIDs in here for use by 'ca' and 'req'.
# Add a simple OID like this:
# testoid1=1.2.3.4
# Or use config file substitution like this:
# testoid2=${testoid1}.5.6

####################################################################
[ ca ]
default_ca      = CA_default            # The default ca section

####################################################################
[ CA_default ]

dir             = /usr/share/ssl        # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/ca.db.index      # database index file.
new_certs_dir   = $dir/ca.db.certs      # default place for new certs.

certificate     = $dir/certs/ca.crt     # The CA certificate
serial          = $dir/ca.db.serial     # The current serial number
crl             = $dir/crl.pem          # The current CRL
private_key     = $dir/private/ca.key   # The private key
RANDFILE        = $dir/ca.db.rand       # private random number file

x509_extensions = usr_cert              # The extentions to add to the cert

# Extensions to add to a CRL. Note: Netscape communicator chokes on V2 CRLs
# so this is commented out by default to leave a V1 CRL.
# crl_extensions        = crl_ext

default_days    = 365                   # how long to certify for
default_crl_days= 30                    # how long before next CRL
default_md      = md5                   # which md to use.
preserve        = no                    # keep passed DN ordering

# A few difference way of specifying how similar the request should look
# For type CA, the listed attributes must be the same, and the optional
# and supplied fields are just that :-)
policy          = policy_match

# For the CA policy
[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional
```

```
# For the 'anything' policy
# At this point in time, you must list all acceptable 'object'
# types.
[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
default_bits            = 1024
default_keyfile         = privkey.pem
distinguished_name      = req_distinguished_name
attributes              = req_attributes
x509_extensions = v3_ca # The extentions to add to the self signed cert

# Passwords for private keys if not present they will be prompted for
# input_password = secret
# output_password = secret

# This sets a mask for permitted string types. There are several options.
# default: PrintableString, T61String, BMPString.
# pkix   : PrintableString, BMPString.
# utf8only: only UTF8Strings.
# nombstr : PrintableString, T61String (no BMPStrings or UTF8Strings).
# MASK:XXXX a literal mask value.
# WARNING: current versions of Netscape crash on BMPStrings or UTF8Strings
# so use this option with caution!
string_mask = nombstr

# req_extensions = v3_req # The extensions to add to a certificate request

[ req_distinguished_name ]
countryName                     = Country Name (2 letter code)
countryName_default             = CA
countryName_min                 = 2
countryName_max                 = 2

stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Quebec

localityName                    = Locality Name (eg, city)
localityName_default            = Montreal

0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = OpenNA.com

# we can do this but it is not needed normally :-)
#1.organizationName             = Second Organization Name (eg, company)
#1.organizationName_default     = World Wide Web Pty Ltd

organizationalUnitName          = Organizational Unit Name (eg, section)
organizationalUnitName_default  = Network Operation Center

commonName                      = Common Name (eg, YOUR name)
commonName_default              = www.openna.com
commonName_max                  = 64
```

```
emailAddress                    = Email Address
emailAddress_default            = noc@openna.com
emailAddress_max                = 40

# SET-ex3                       = SET extension number 3

[ req_attributes ]
challengePassword               = A challenge password
challengePassword_min           = 8
challengePassword_max           = 20

unstructuredName                = An optional company name

[ usr_cert ]

# These extensions are added when 'ca' signs a request.

# This goes against PKIX guidelines but some CAs do it and some software
# requires this to avoid interpreting an end user certificate as a CA.

basicConstraints=CA:FALSE

# Here are some examples of the usage of nsCertType. If it is omitted
# the certificate can be used for anything *except* object signing.

# This is OK for an SSL server.
# nsCertType                    = server

# For an object signing certificate this would be used.
# nsCertType = objsign

# For normal client use this is typical
# nsCertType = client, email

# and for everything including object signing:
# nsCertType = client, email, objsign

# This is typical in keyUsage for a client certificate.
# keyUsage = nonRepudiation, digitalSignature, keyEncipherment

# This will be displayed in Netscape's comment listbox.
nsComment                       = "OpenSSL Generated Certificate"

# PKIX recommendations harmless if included in all certificates.
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyid,issuer:always

# This stuff is for subjectAltName and issuerAltname.
# Import the email address.
# subjectAltName=email:copy

# Copy subject details
# issuerAltName=issuer:copy

#nsCaRevocationUrl              = http://www.domain.dom/ca-crl.pem
#nsBaseUrl
#nsRevocationUrl
#nsRenewalUrl
#nsCaPolicyUrl
#nsSslServerName

[ v3_req ]
```

```
# Extensions to add to a certificate request

basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment

[ v3_ca ]


# Extensions for a typical CA


# PKIX recommendation.

subjectKeyIdentifier=hash

authorityKeyIdentifier=keyid:always,issuer:always

# This is what PKIX recommends but some broken software chokes on critical
# extensions.
#basicConstraints = critical,CA:true
# So we do this instead.
basicConstraints = CA:true

# Key usage: this is typical for a CA certificate. However since it will
# prevent it being used as an test self-signed certificate it is best
# left out by default.
# keyUsage = cRLSign, keyCertSign

# Some might want this also
# nsCertType = sslCA, emailCA

# Include email address in subject alt name: another PKIX recommendation
# subjectAltName=email:copy
# Copy issuer details
# issuerAltName=issuer:copy

# DER hex encoding of an extension: beware experts only!
# obj=DER:02:03
# Where 'obj' is a standard or added object
# You can even override a supported extension:
# basicConstraints= critical, DER:30:03:01:01:FF

[ crl_ext ]

# CRL extensions.
# Only issuerAltName and authorityKeyIdentifier make any sense in a CRL.

# issuerAltName=issuer:copy
authorityKeyIdentifier=keyid:always,issuer:always
```

WARNING: You don't need to change all the default options set in the file `openssl.cnf`; The configurations you may usually change will be in the **`[ CA_default ]`** and **`[ req_distinguished_name ]`** sections.

/usr/share/ssl/misc/sign.sh: The Scrip File to Sign Certificates

The `openssl ca` commands has some strange requirements and the default `OpenSSL` config doesn't allow one easily to use `openssl ca` directly. It is for this reason that we don't use the files `CA.pl` or `CA.sh` to sign certificates. To solve the problem, we'll create and customize the `sign.sh` script file below to replace them. The text in bold are the parts of the script file that must be customized and adjusted to satisfy our needs.

Step 1
Create the **`sign.sh`** script file (`touch /usr/share/ssl/misc/sign.sh`) and add into it the following lines:

```
#!/bin/sh
##
##  sign.sh -- Sign a SSL Certificate Request (CSR)
##  Copyright (c) 1998-1999 Ralf S. Engelschall, All Rights Reserved.
##

#   argument line handling
CSR=$1
if [ $# -ne 1 ]; then
    echo "Usage: sign.sign <whatever>.csr"; exit 1
fi
if [ ! -f $CSR ]; then
    echo "CSR not found: $CSR"; exit 1
fi
case $CSR in
   *.csr ) CERT="`echo $CSR | sed -e 's/\.csr/.crt/'`" ;;
       * ) CERT="$CSR.crt" ;;
esac

#   make sure environment exists
if [ ! -d ca.db.certs ]; then
    mkdir ca.db.certs
fi
if [ ! -f ca.db.serial ]; then
    echo '01' >ca.db.serial
fi
if [ ! -f ca.db.index ]; then
    cp /dev/null ca.db.index
fi

#   create an own SSLeay config
cat >ca.config <<EOT
[ ca ]
default_ca          = CA_own
[ CA_own ]
dir                 = /usr/share/ssl
certs               = /usr/share/ssl/certs
new_certs_dir       = /usr/share/ssl/ca.db.certs
database            = /usr/share/ssl/ca.db.index
serial              = /usr/share/ssl/ca.db.serial
RANDFILE            = /usr/share/ssl/ca.db.rand
certificate         = /usr/share/ssl/certs/ca.crt
private_key         = /usr/share/ssl/private/ca.key
default_days        = 365
default_crl_days    = 30
default_md          = md5
preserve            = no
policy              = policy_anything
[ policy_anything ]
countryName         = optional
```

```
stateOrProvinceName = optional
localityName        = optional
organizationName    = optional
organizationalUnitName             = optional
commonName          = supplied
emailAddress        = optional
EOT

#  sign the certificate
echo "CA signing: $CSR -> $CERT:"
openssl ca -config ca.config -out $CERT -infiles $CSR
echo "CA verifying: $CERT <-> CA cert"
openssl verify -CAfile /usr/share/ssl/certs/ca.crt $CERT

#  cleanup after SSLeay
rm -f ca.config
rm -f ca.db.serial.old
rm -f ca.db.index.old

#  die gracefully
exit 0
```

Step 2
Once the script file has been created, it is important to make it executable and change its default permissions. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason.

- To make this script executable and to change its default permissions, use the command:
  `[root@deep /]# chmod 700 /usr/share/ssl/misc/sign.sh`
  `[root@deep /]# chown 0.0 /usr/share/ssl/misc/sign.sh`

WARNING: You can also find this program "`sign.sh`" in the `mod_ssl` distribution under the `mod_ssl-version/pkg.contrib/` subdirectory, or on our `floppy-2.0.tgz` archive file. Also note that the section **`[ CA_own ]`** must be changed to refect your own environment and don't forget to change the **`openssl verify -CAfile /usr/share/ssl/certs/ca.crt $CERT`** line too.

All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

OpenSSL Administrative Tools

After your desired configuration options have been set and the program is running, we can play with its utility. As an example, we'll show you how to create certificates for Apache Webserver and your own CA (**C**ertifying **A**uthority) to sign your "Certificate Signing Request" yourself. All commands listed below are assumed to be made in the **/usr/share/ssl** directory.

Apache Key & CSR Generation

The utility openssl that you use to generate the RSA Private Key (Key) and the **C**ertificate **S**igning **R**equest (CSR) comes with Openssl and is usually installed under the directory /usr/bin with our Linux distribution. Below is the step to create certificates for Apache with mod_ssl Web server.

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the website for which you want to request a certificate. When you want to access your website through https://www.mydomain.com/ then the FQDN of your website is www.mydomain.com.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key protected with a pass-phrase for your Apache Web server. The command below will generate 1024 bit RSA Private Key and stores it in the file www.mydomain.com.key. It will ask you for a pass-phrase: use something secure and remember it. Your certificate will be useless without the key. If you don't want to protect your key with a pass-phrase (only if you absolutely trust that server machine, and you make sure the permissions are carefully set so only you can read that key) you can leave out the -des3 option below.

- To generate the Key, use the following command:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -des3 –rand
random1:random2:random3:random4:random5 -out www.mydomain.com.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

WARNING: Please backup your **www.mydomain.com.key** file and remember the pass-phrase you had to enter at a secure location. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give the name www.mydomain.com when prompted for `Common Name'. Do not enter your personal name here. We are requesting a certificate for a Web server, so the Common Name has to match the FQDN of your website (a requirement of the browsers).

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key www.mydomain.com.key -out
www.mydomain.com.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) [Network Operation Centre]:
Common Name (eg, YOUR name) [www.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**CommonName**" (i.e. when you generate a CSR for a website which will be later accessed via https://www.mydomain.com/, enter www.mydomain.com here).

After generation of your **C**ertificate **S**igning **R**equest (CSR), you must send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed Certificate and store it into a www.mydomain.com.crt file. The result is then a real Certificate, which can be used for Apache.

CA Key & CRT Generation

If you don't want to pay a commercial **C**ertifying **A**uthority (CA) to sign you certificates, you can use your own CA and now have to sign the CSR yourself by this CA. This solution is economical, and allows an organization to host their own CA server and generate as many certificates as they need for internal use without paying any cent to a commercial CA. Unfortunately using your own CA to generate certificates cause problems in electronic commerce, because customers need to have some trust in your organization by the use of recognized commercial CA. See below on how to sign a CSR with your CA yourself.

Step 1
As for the Apache Web server above, the first step is to create the RSA private key protected with a pass-phrase for your CA. The command below will generate 1024 bit RSA Private Key and stores it in the file ca.key. It will ask you for a pass-phrase: use something secure and remember it. Your certificate will be useless without the key.

- To create the RSA private key for your (CA), use the following command:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -des3 -out ca.key 1024
Generating RSA private key, 1024 bit long modulus
.........................+++++
.............................................+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

WARNING: Please backup your **ca.key** file and remember the pass-phrase you had to enter at a secure location. A good choice is to backup this information onto a diskette or other removable media.

Step 2
Now, we must create a self-signed (CA) certificate (x509 structure) with the RSA key of the CA. The req command creates a self-signed certificate when the -x509 switch is used.

- To create a self-signed (CA) certificate, use the following command:

```
[root@deep ssl]# openssl req -new -x509 -days 365 -key ca.key -out ca.crt
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) [Network Operation
Centre]:Marketing Department
Common Name (eg, YOUR name) [www.openna.com]:
Email Address [noc@openna.com]:sales@openna.com
```

Step 3
Once the self-signed (CA) certificate has been created, we must place all certificates and CA files to their appropriate directory.

- To place the files into their appropriate directory, use the following commands:

```
[root@deep ssl]# mv www.mydomain.com.key private/
[root@deep ssl]# mv ca.key private/
[root@deep ssl]# mv ca.crt certs/
```

Step 4
Finally, you can use this CA to sign all servers CSR's in order to create real SSL Certificates for use inside an Apache Web server (assuming you already have a www.mydomain.com.csr at hand). We must prepare the script **sign.sh** for signing (which is needed because the openssl ca command has some strange requirements, and the default OpenSSL config doesn't allow one easily to use openssl ca directly). The script named **sign.sh** is distributed with the floppy disk under the OpenSSL directory. Use this script for signing.

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign.sh www.mydomain.com.csr
CA signing: www.mydomain.com.csr -> www.mydomain.com.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName               :PRINTABLE:'CA'
stateOrProvinceName       :PRINTABLE:'Quebec'
localityName              :PRINTABLE:'Montreal'
organizationName          :PRINTABLE:'Open Network Architecture'
organizationalUnitName    :PRINTABLE:'Network Operation Centre'
commonName                :PRINTABLE:'www.openna.com'
emailAddress              :IA5STRING:'noc@openna.com'
Certificate is to be certified until Oct  18 14:59:29 2001 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: www.mydomain.com.crt <-> CA cert
www.mydomain.com.crt: OK
```

This signs the CSR and results in a www.mydomain.com.crt file. Move this file to its appropriate directory.

- To move the CRT file to its appropriate directory, use the following command:

```
[root@deep ssl]# mv www.mydomain.com.crt certs/
```

Now you have two files: **www.mydomain.com.key** and **www.mydomain.com.crt**. These can now, for example, be used as follows, inside the virtual host section of your Apache server's httpd.conf file:

```
SSLCertificateFile      /usr/share/ssl/certs/www.mydomain.com.crt
SSLCertificateKeyFile   /usr/share/ssl/private/www.mydomain.com.key
```

In this example, www.mydomain.com.crt is our Web server Certificate Signing Request Public Key, and www.mydomain.com.key is our Web server RSA Private Key.

The **www.mydomain.com.csr** file is no longer needed, we can remove it safety from the system.

- To remove this file from the system, use the following command:
  ```
  [root@deep ssl]# rm -f www.mydomain.com.csr
  ```

WARNING: If you receive error message during signature of the certificate, it's probably because you've entered the wrong FQDN (Fully Qualified Domain Name) for the server when OpenSSL prompted you for the "**CommonName**"; the "**CommonName**" must be something like www.mydomain.com and not mydomain.com. Also, since you generate both the certificate and the CA certificate, it's important that at least one piece of information differs between both files, or you may encounter problems during the signature of the certificate request.

Securing OpenSSL

This small section deals especially with actions we can make to improve and tighten security under OpenSSL. It is important to note that we refer to the features available within the base installed program and not to any additional software.

Changing the default mode of OpenSSL keys

Make your keys "Read and Write" only by the super-user "root". This is important because no one needs to touch these files.

- To make your keys "read and Write" only by "root", use the following commands:
  ```
  [root@deep /]# chmod 750 /usr/share/ssl/private/
  [root@deep /]# chmod 400 /usr/share/ssl/certs/ca.crt
  [root@deep /]# chmod 400 /usr/share/ssl/certs/www.mydomain.com.crt
  [root@deep /]# chmod 400 /usr/share/ssl/private/ca.key
  [root@deep /]# chmod 400 /usr/share/ssl/private/www.mydomain.com.key
  ```

Some possible uses of OpenSSL software

OpenSSL can be used to:

1. Creation of your own Certifying Authority Server.
2. Creation of RSA, DH and DSA key parameters.
3. Creation of X.509 certificates, CSRs and CRLs.
4. Calculation of Message Digest.
5. Encryption and Descryptiion with Ciphers.
6. SSL/TLS Client and Server Tests.
7. Handling of S/MIME signed or encrypted mail.
8. Provide data confidentiality, integrity, authentication, and electronic signature in transmission for the users.
9. Secure electronic commerce transactions.

List of installed `OpenSSL` files in your system

> /usr/bin/openssl
> /usr/bin/c_rehash
> /usr/include/openssl
> /usr/include/openssl/e_os.h
> /usr/include/openssl/e_os2.h
> /usr/include/openssl/crypto.h
> /usr/include/openssl/tmdiff.h
> /usr/include/openssl/opensslv.h
> /usr/include/openssl/opensslconf.h
> /usr/include/openssl/ebcdic.h
> /usr/include/openssl/symhacks.h
> /usr/include/openssl/md2.h
> /usr/include/openssl/md4.h
> /usr/include/openssl/md5.h
> /usr/include/openssl/sha.h
> /usr/include/openssl/mdc2.h
> /usr/include/openssl/hmac.h
> /usr/include/openssl/ripemd.h
> /usr/include/openssl/des.h
> /usr/include/openssl/rc2.h
> /usr/include/openssl/rc4.h
> /usr/include/openssl/rc5.h
> /usr/include/openssl/idea.h
> /usr/include/openssl/blowfish.h
> /usr/include/openssl/cast.h
> /usr/include/openssl/bn.h
> /usr/include/openssl/rsa.h
> /usr/include/openssl/dsa.h
> /usr/include/openssl/dh.h
> /usr/include/openssl/dso.h
> /usr/include/openssl/buffer.h
> /usr/include/openssl/bio.h
> /usr/include/openssl/stack.h
> /usr/include/openssl/safestack.h
> /usr/include/openssl/lhash.h
> /usr/include/openssl/rand.h
> /usr/include/openssl/err.h
> /usr/include/openssl/objects.h
> /usr/include/openssl/obj_mac.h
> /usr/include/openssl/evp.h
> /usr/include/openssl/asn1.h
> /usr/include/openssl/asn1_mac.h
> /usr/include/openssl/pem.h
> /usr/include/openssl/pem2.h
> /usr/include/openssl/x509.h
> /usr/include/openssl/x509_vfy.h
> /usr/include/openssl/x509v3.h
> /usr/include/openssl/conf.h
> /usr/include/openssl/conf_api.h
> /usr/include/openssl/txt_db.h
> /usr/include/openssl/pkcs7.h
> /usr/include/openssl/pkcs12.h
> /usr/include/openssl/comp.h
> /usr/include/openssl/ssl.h
> /usr/include/openssl/ssl2.h
> /usr/include/openssl/ssl3.h
> /usr/include/openssl/ssl23.h
> /usr/include/openssl/tls1.h
> /usr/lib/libcrypto.a
> /usr/lib/libssl.a
> /usr/share/man/man1/ca.1
> /usr/share/man/man1/asn1parse.1
> /usr/share/man/man1/CA.pl.1
> /usr/share/man/man1/ciphers.1
> /usr/share/man/man1/crl2pkcs7.1
> /usr/share/man/man1/crl.1
> /usr/share/man/man1/dgst.1
> /usr/share/man/man3/BN_mod_mul_montgomery.3
> /usr/share/man/man3/BN_mod_mul_reciprocal.3
> /usr/share/man/man3/BN_new.3
> /usr/share/man/man3/BN_num_bytes.3
> /usr/share/man/man3/BN_rand.3
> /usr/share/man/man3/BN_set_bit.3
> /usr/share/man/man3/BN_zero.3
> /usr/share/man/man3/buffer.3
> /usr/share/man/man3/crypto.3
> /usr/share/man/man3/CRYPTO_set_ex_data.3
> /usr/share/man/man3/d2i_DHparams.3
> /usr/share/man/man3/d2i_RSAPublicKey.3
> /usr/share/man/man3/des.3
> /usr/share/man/man3/DH_generate_key.3
> /usr/share/man/man3/DH_generate_parameters.3
> /usr/share/man/man3/DH_get_ex_new_index.3
> /usr/share/man/man3/DH_new.3
> /usr/share/man/man3/dh.3
> /usr/share/man/man3/DH_set_method.3
> /usr/share/man/man3/DH_size.3
> /usr/share/man/man3/DSA_do_sign.3
> /usr/share/man/man3/DSA_dup_DH.3
> /usr/share/man/man3/DSA_generate_key.3
> /usr/share/man/man3/DSA_generate_parameters.3
> /usr/share/man/man3/DSA_get_ex_new_index.3
> /usr/share/man/man3/DSA_new.3
> /usr/share/man/man3/dsa.3
> /usr/share/man/man3/DSA_set_method.3
> /usr/share/man/man3/DSA_SIG_new.3
> /usr/share/man/man3/DSA_sign.3
> /usr/share/man/man3/DSA_size.3
> /usr/share/man/man3/err.3
> /usr/share/man/man3/ERR_clear_error.3
> /usr/share/man/man3/ERR_error_string.3
> /usr/share/man/man3/ERR_get_error.3
> /usr/share/man/man3/ERR_GET_LIB.3
> /usr/share/man/man3/ERR_load_crypto_strings.3
> /usr/share/man/man3/ERR_load_strings.3
> /usr/share/man/man3/ERR_print_errors.3
> /usr/share/man/man3/ERR_put_error.3
> /usr/share/man/man3/ERR_remove_state.3
> /usr/share/man/man3/EVP_DigestInit.3
> /usr/share/man/man3/EVP_EncryptInit.3
> /usr/share/man/man3/EVP_OpenInit.3
> /usr/share/man/man3/evp.3
> /usr/share/man/man3/EVP_SealInit.3
> /usr/share/man/man3/EVP_SignInit.3
> /usr/share/man/man3/EVP_VerifyInit.3
> /usr/share/man/man3/hmac.3
> /usr/share/man/man3/lhash.3
> /usr/share/man/man3/lh_stats.3
> /usr/share/man/man3/md5.3
> /usr/share/man/man3/mdc2.3
> /usr/share/man/man3/OpenSSL_add_all_algorithms.3
> /usr/share/man/man3/OPENSSL_VERSION_NUMBER.3
> /usr/share/man/man3/RAND_add.3
> /usr/share/man/man3/RAND_bytes.3
> /usr/share/man/man3/RAND_cleanup.3
> /usr/share/man/man3/RAND_egd.3
> /usr/share/man/man3/RAND_load_file.3
> /usr/share/man/man3/rand.3
> /usr/share/man/man3/RAND_set_rand_method.3
> /usr/share/man/man3/rc4.3
> /usr/share/man/man3/ripemd.3
> /usr/share/man/man3/RSA_blinding_on.3
> /usr/share/man/man3/RSA_check_key.3
> /usr/share/man/man3/RSA_generate_key.3

> /usr/share/man/man1/dhparam.1
> /usr/share/man/man1/dsaparam.1
> /usr/share/man/man1/dsa.1
> /usr/share/man/man1/enc.1
> /usr/share/man/man1/gendsa.1
> /usr/share/man/man1/genrsa.1
> /usr/share/man/man1/nseq.1
> /usr/share/man/man1/openssl.1
> /usr/share/man/man1/passwd.1
> /usr/share/man/man1/pkcs12.1
> /usr/share/man/man1/pkcs7.1
> /usr/share/man/man1/pkcs8.1
> /usr/share/man/man1/rand.1
> /usr/share/man/man1/req.1
> /usr/share/man/man1/rsa.1
> /usr/share/man/man1/rsautl.1
> /usr/share/man/man1/s_client.1
> /usr/share/man/man1/sess_id.1
> /usr/share/man/man1/smime.1
> /usr/share/man/man1/speed.1
> /usr/share/man/man1/spkac.1
> /usr/share/man/man1/s_server.1
> /usr/share/man/man1/verify.1
> /usr/share/man/man1/version.1
> /usr/share/man/man1/x509.1
> /usr/share/man/man3/bn.3
> /usr/share/man/man3/BIO_ctrl.3
> /usr/share/man/man3/BIO_f_base64.3
> /usr/share/man/man3/BIO_f_buffer.3
> /usr/share/man/man3/BIO_f_cipher.3
> /usr/share/man/man3/BIO_find_type.3
> /usr/share/man/man3/BIO_f_md.3
> /usr/share/man/man3/BIO_f_null.3
> /usr/share/man/man3/BIO_f_ssl.3
> /usr/share/man/man3/BIO_new_bio_pair.3
> /usr/share/man/man3/BIO_new.3
> /usr/share/man/man3/bio.3
> /usr/share/man/man3/BIO_push.3
> /usr/share/man/man3/BIO_read.3
> /usr/share/man/man3/BIO_s_accept.3
> /usr/share/man/man3/BIO_s_bio.3
> /usr/share/man/man3/BIO_s_connect.3
> /usr/share/man/man3/BIO_set_callback.3
> /usr/share/man/man3/BIO_s_fd.3
> /usr/share/man/man3/BIO_s_file.3
> /usr/share/man/man3/BIO_should_retry.3
> /usr/share/man/man3/BN_add.3
> /usr/share/man/man3/BIO_s_mem.3
> /usr/share/man/man3/BIO_s_null.3
> /usr/share/man/man3/BIO_s_socket.3
> /usr/share/man/man3/blowfish.3
> /usr/share/man/man3/BN_add_word.3
> /usr/share/man/man3/BN_bn2bin.3
> /usr/share/man/man3/BN_cmp.3
> /usr/share/man/man3/BN_copy.3
> /usr/share/man/man3/BN_CTX_new.3
> /usr/share/man/man3/BN_CTX_start.3
> /usr/share/man/man3/BN_generate_prime.3
> /usr/share/man/man3/bn_internal.3
> /usr/share/man/man3/BN_mod_inverse.3
> /usr/share/man/man3/RSA_get_ex_new_index.3
> /usr/share/man/man3/RSA_new.3
> /usr/share/man/man3/RSA_padding_add_PKCS1_type_1.3
> /usr/share/man/man3/rsa.3
> /usr/share/man/man3/RSA_print.3
> /usr/share/man/man3/RSA_private_encrypt.3
> /usr/share/man/man3/RSA_public_encrypt.3
> /usr/share/man/man3/RSA_set_method.3
> /usr/share/man/man3/RSA_sign_ASN1_OCTET_STRING.3
> /usr/share/man/man3/RSA_sign.3
> /usr/share/man/man3/RSA_size.3
> /usr/share/man/man3/sha.3
> /usr/share/man/man3/threads.3
> /usr/share/man/man3/SSL_accept.3
> /usr/share/man/man3/SSL_CIPHER_get_name.3
> /usr/share/man/man3/SSL_clear.3
> /usr/share/man/man3/SSL_connect.3
> /usr/share/man/man3/SSL_CTX_free.3
> /usr/share/man/man3/SSL_CTX_new.3
> /usr/share/man/man3/SSL_CTX_set_cipher_list.3
> /usr/share/man/man3/SSL_CTX_set_ssl_version.3
> /usr/share/man/man3/SSL_free.3
> /usr/share/man/man3/SSL_get_ciphers.3
> /usr/share/man/man3/SSL_get_current_cipher.3
> /usr/share/man/man3/SSL_get_error.3
> /usr/share/man/man3/SSL_get_fd.3
> /usr/share/man/man3/SSL_get_peer_cert_chain.3
> /usr/share/man/man3/SSL_get_peer_certificate.3
> /usr/share/man/man3/SSL_get_rbio.3
> /usr/share/man/man3/SSL_get_session.3
> /usr/share/man/man3/SSL_get_verify_result.3
> /usr/share/man/man3/SSL_library_init.3
> /usr/share/man/man3/SSL_new.3
> /usr/share/man/man3/SSL_pending.3
> /usr/share/man/man3/ssl.3
> /usr/share/man/man3/SSL_read.3
> /usr/share/man/man3/SSL_SESSION_free.3
> /usr/share/man/man3/SSL_set_bio.3
> /usr/share/man/man3/SSL_set_fd.3
> /usr/share/man/man3/SSL_set_session.3
> /usr/share/man/man3/SSL_set_verify_result.3
> /usr/share/man/man3/SSL_shutdown.3
> /usr/share/man/man3/SSL_write.3
> /usr/share/man/man5/config.5
> /usr/share/man/man7/des_modes.7
> /usr/share/ssl
> /usr/share/ssl/misc
> /usr/share/ssl/misc/CA.sh
> /usr/share/ssl/misc/CA.pl
> /usr/share/ssl/misc/sign.sh
> /usr/share/ssl/misc/der_chop
> /usr/share/ssl/misc/c_hash
> /usr/share/ssl/misc/c_info
> /usr/share/ssl/misc/c_issuer
> /usr/share/ssl/misc/c_name
> /usr/share/ssl/certs
> /usr/share/ssl/private
> /usr/share/ssl/lib
> /usr/share/ssl/openssl.cnf
> /usr/share/ssl/crl

12 Cryptography & Authentication - OpenSSH

In this Chapter

Linux OpenSSH

Abstract
As illustrated in the chapter related to Linux installation, many network services including, but not limited to, telnet, rsh, rlogin, or rexec are vulnerable to electronic eavesdropping. As a consequence, anyone who has access to any machine connected to the network can listen in on network communication and get your password, as well as any other private information that is sent over the network in plain text.

Currently the Telnet program is indispensable for daily administration tasks, but it is insecure since it transmits your password in plain text over the network and allows any listener to capture your password and then use your account to do anything he likes. To solve this problem we must find either another way, or another program, to replace it. Fortunately OpenSSH is a truly seamless and secure replacement of old, insecure and obsolete remote login programs such as telnet, rlogin, rsh, rdist, or rcp.

According to the official [OpenSSH README] file:
SSH (**S**ecure **Sh**ell) is a program to log into another computer over a network, to execute commands on a remote machine, and to move files from one machine to another. It provides strong authentication and secure communications over insecure channels. It is intended as a replacement for rlogin, rsh, rcp, and rdist.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest OpenSSH version number is 2.9p1

Packages
The following are based on information as listed by OpenSSH as of 2001/05/01. Please regularly check at www.openssh.com for the latest status.

Pristine source code is available from:
OpenSSH Homepage: http://www.openssh.com/
OpenSSH FTP Site: 129.128.5.191
You must be sure to download: openssh-2.9p1.tar.gz

> **NOTE**: Don't forget to download the portable version (the **p** suffix) of OpenSSH tarball for Linux. There is strictly OpenBSD-based development of this software and another one known as portable version, which run on many operating systems (these are known as the **p** releases, and named like "OpenSSH 2.9p1").

Prerequisites

OpenSSH requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive files. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL, which enables support for SSL functionality, must already be installed on your system to be able to use the OpenSSH software.

NOTE: For more information on OpenSSL software, see its related chapter in this book. Even if you don't need to use OpenSSL software to create or hold encrypted key files, it's important to note that OpenSSH program require its libraries files to be able to work properly on your system.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files onto the system when the program is updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install OpenSSH, and one afterwards, and then compare them using the Linux **diff** utility to find out what files have been placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > OpenSSH1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > OpenSSH2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff OpenSSH1 OpenSSH2 > OpenSSH-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were changed program and remove the files manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing OpenSSH

Below are the required steps that you must make to configure, compile and optimize the OpenSSH software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
  ```
  [root@deep /]# cp openssh-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf openssh-version.tar.gz
  ```

Step 2
After that, move into the newly created OpenSSH directory then configure and optimize it.

- To move into the newly created OpenSSH directory use the following command:
  ```
  [root@deep tmp]# cd openssh-2.9p1/
  ```

- To configure and optimize OpenSSH use the following compile lines:
  ```
  CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
  ./configure \
  --prefix=/usr \
  --sysconfdir=/etc/ssh \
  --libexecdir=/usr/libexec/openssh \
  --mandir=/usr/share/man \
  --with-pam \
  --with-ipaddr-display \
  --with-ipv4-default \
  --with-md5-passwords
  ```

This tells OpenSSH to set itself up for this particular configuration with:

- Enabled PAM support.
- Use the ip address instead of the hostname in $DISPLAY.
- Use IPv4 by connections unless '-6' specified.
- Enable use of MD5 passwords.

NOTE: Pay special attention to the compile CFLAGS line above. We optimize OpenSSH for an i686 CPU architecture with the parameter "-march=i686 and -mcpu=i686". Please don't forget to adjust this CFLAGS line to reflect your own system and architecture.

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install OpenSSH in the server:

```
[root@deep openssh-2.9p1]# make
[root@deep openssh-2.9p1]# cd
[root@deep /root]# find /* > OpenSSH1
[root@deep /root]# cd /var/tmp/openssh-2.9p1/
[root@deep openssh-2.9p1]# make install
[root@deep openssh-2.9p1]# cd
[root@deep /root]# find /* > OpenSSH2
[root@deep /root]# diff OpenSSH1 OpenSSH2 > OpenSSH-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 4
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory neededsince they are no longer needed.

- To delete OpenSSH and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf openssh-version/
[root@deep tmp]# rm -f openssh-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install OpenSSH. It will also remove the OpenSSH compressed archive from the /var/tmp directory.

Configuring OpenSSH

After building OpenSSH, your next step is to verify or change, if necessary, options in your OpenSSH configuration files. Those files are:

- ✓ /etc/ssh/ssh_config (The OpenSSH Client Configuration File)
- ✓ /etc/ssh/sshd_config (The OpenSSH Server Configuration File)
- ✓ /etc/pam.d/sshd (The OpenSSH PAM Support Configuration File)
- ✓ /etc/rc.d/init.d/sshd (The OpenSSH Initialization File)

/etc/ssh/ssh_config: The OpenSSH Client Configuration File

The ssh_config file is the system-wide configuration file for OpenSSH which allows you to set options that modify the operation of the client programs. The file contains keyword-value pairs, one per line, with keywords being case insensitive.

Here are the most important keywords to configure your ssh client for maximum security; a complete listing and/or special requirements are available in the manual page for ssh (1). We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy your needs.

- Edit the **ssh_config** file (vi /etc/ssh/ssh_config) and set your needs. Below is what we recommend you:

```
# Site-wide defaults for various options

 Host *
   ForwardAgent no
   ForwardX11 no
   RhostsAuthentication no
   RhostsRSAAuthentication no
   RSAAuthentication yes
   PasswordAuthentication no
   FallBackToRsh no
   UseRsh no
   BatchMode no
   CheckHostIP yes
   StrictHostKeyChecking yes
   IdentityFile ~/.ssh/identity
   IdentityFile ~/.ssh/id_dsa
   IdentityFile ~/.ssh/id_rsa
   Port 22
   Protocol 2,1
```

```
Cipher blowfish
EscapeChar ~
```

This tells `ssh_config` file to set itself up for this particular configuration with:

`Host *`
This option "`Host`" restricts all forwarded declarations and options in the configuration file to be only for those hosts that match one of the patterns given after the keyword. The pattern `*` mean for all hosts up to the next Host keyword. With this option you can set different declarations for different hosts in the same `ssh_config` file. In particular, I find it useful when you want to automate backup over the network with `SSH` and don't want to supplies the user password. In this way we can build a new section reserved to this meaning and disable function that ask for password for the specified host in question.

`ForwardAgent no`
This option "`ForwardAgent`" specifies which connection authentication agent (if any) should be forwarded to the remote machine.

`ForwardX11 no`
This option "`ForwardX11`" is for people that use the Xwindow `GUI` and want to automatically redirect `X11` sessions to the remote machine. Since we setup a server and don't have `GUI` installed on it, we can safely turn this option off.

`RhostsAuthentication no`
This option "`RhostsAuthentication`" specifies whether we can try to use `rhosts` based authentication. Because `rhosts` authentication is insecure you shouldn't use this option.

`RhostsRSAAuthentication no`
This option "`RhostsRSAAuthentication`" specifies whether or not to try `rhosts` authentication in concert with `RSA` host authentication. Evidently our answer is `no`.

`RSAAuthentication yes`
This option "`RSAAuthentication`" specifies whether to try `RSA` authentication. It is important to note that it is reserved for the `SSH1` protocol only. This option must be set to `yes` for better security in your sessions if you use `SSH1` and only `SSH1` since it doesn't applies for `SSH2` protocol (`SSH2` use `DSA` instead of `RSA`). `RSA` use public and private key pairs created with the `ssh-keygen` utility for authentication purposes.

`PasswordAuthentication no`
This option "`PasswordAuthentication`" specifies whether we should use password-based authentication. For strong security, this option must always be set to `no`. You should put '`PasswordAuthentication no`' in the `sshd_config` file, otherwise people might try to guess the password for the user. With '`PasswordAuthentication no`', your public key must be on the computer or no login is allowed: that's what we want. Take a note that with the Windows client program called "`putty`" you cannot set this option to `no` or you will not be able to log in the server using `putty`.

`FallBackToRsh no`
This option "`FallBackToRsh`" specifies that if a connection with `ssh` daemon fails `rsh` should automatically be used instead. Recalling that `rsh` service is insecure, this option must always be set to `no`.

`UseRsh no`

This option "UseRsh" specifies that rlogin/rsh services should be used on this host. As with the FallBackToRsh option, it must be set to no for obvious reasons.

BatchMode no
This option "BatchMode" specifies whether a username and password querying on connect will be disabled. This option is useful when you create scripts and don't want to supply the password. (e.g. Scripts that use the scp command to make backups over the network).

CheckHostIP yes
This option "CheckHostIP" specifies whether or not ssh will additionally check the host IP address that connect to the server to detect DNS spoofing. It's recommended that you set this option to yes but in the other side you can lose some performance.

StrictHostKeyChecking yes
This option "StrictHostKeyChecking" specifies whether or not ssh will automatically add new host keys to the $HOME/.ssh/known_hosts file, or never automatically add new host keys to the host file. This option, when set to yes, provides maximum protection against Trojan horse attacks. One interesting procedure with this option is to set it to no at the beginning, allow ssh to add automatically all common hosts to the host file as they are connected to, and then return to set it to yes to take advantage of its feature.

```
IdentityFile ~/.ssh/identity
IdentityFile ~/.ssh/id_dsa
IdentityFile ~/.ssh/id_rsa
```
These options specify alternate multiple authentication identity files to read.

Port 22
This option "Port" specifies on which port number ssh connects to on the remote host. The default port is 22.

Protocol 2,1
This option "Protocol" specifies the protocol versions ssh should support in order of preference. In our configuration the default is "2,1". This means that ssh tries version 2 and falls back to version 1 if version 2 is not available. Depending of the ssh client version you use to connect, you may need to invert this order but you can connect with ssh client version 1 even if the order is "2,1".

Cipher blowfish
This option "Cipher" specifies what cipher should be used for encrypting sessions. The blowfish use 64-bit blocks and keys of up to 448 bits.

EscapeChar ~
This option "EscapeChar" specifies the session escape character for suspension.

/etc/ssh/sshd_config: The OpenSSH Server Configuration File

The sshd_config file is the system-wide configuration file for OpenSSH which allows you to set options that modify the operation of the daemon. This file contains keyword-value pairs, one per line, with keywords being case insensitive.

Here are the most important keywords to configure your sshd server for maximum security; a complete listing and/or special requirements are available in the manual page for sshd (8). We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **sshd_config** file (vi /etc/ssh/sshd_config) and set your needs. Below is what we recommend you:

```
# This is ssh server systemwide configuration file.

Port 22
ListenAddress 207.35.78.3
HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_rsa_key
ServerKeyBits 768
LoginGraceTime 60
KeyRegenerationInterval 3600
PermitRootLogin no
IgnoreRhosts yes
IgnoreUserKnownHosts yes
StrictModes yes
X11Forwarding no
PrintMotd yes
KeepAlive yes
SyslogFacility AUTH
LogLevel INFO
RhostsAuthentication no
RhostsRSAAuthentication no
RSAAuthentication yes
PasswordAuthentication no
PermitEmptyPasswords no
AllowUsers gmourani
PAMAuthenticationViaKbdInst yes
Subsystem       sftp    /usr/libexec/openssh/sftp-server
```

This tells sshd_config file to set itself up for this particular configuration with:

Port 22
The option "Port" specifies on which port number ssh daemon listens for incoming connections. The default port is 22.

ListenAddress 207.35.78.3
The option "ListenAddress" specifies the IP address of the interface network on which the ssh daemon server socket is bound. The default is "0.0.0.0"; to improve security you may specify only the required ones to limit possible addresses.

HostKey /etc/ssh/ssh_host_key
HostKey /etc/ssh/ssh_host_dsa_key
HostKey /etc/ssh/ssh_host_rsa_key
These options specify the location containing the different private host keys. If you have compiled OpenSSH as described in this book, then the default ones is correct.

`ServerKeyBits 768`
The option "`ServerKeyBits`" specifies how many bits to use in the server key. These bits are used when the daemon starts to generate its `RSA` key.

`LoginGraceTime 60`
The option "`LoginGraceTime`" specifies how long in seconds after a connection request the server will wait before disconnecting, if the user has not successfully logged in. A low value is recommended for this setting. Imagine what 1024 simulated connections at the same time can do to the other processes on your server.

`KeyRegenerationInterval 3600`
The option "`KeyRegenerationInterval`" specifies how long in seconds the server should wait before automatically regenerated its key. This is a security feature to prevent decrypting captured sessions.

`PermitRootLogin no`
The option "`PermitRootLogin`" specifies whether root can log in using `ssh`. Never say `yes` to this option. It is better and safer to log in with a regular `UID` and then `su` to root, or better yet, use the sudo program.

`IgnoreRhosts yes`
The option "`IgnoreRhosts`" specifies whether the `rhosts` or `shosts` files should not be used in authentication. For security reasons it is recommended to NOT use `rhosts` or `shosts` files for authentication.

`IgnoreUserKnownHosts yes`
The option "`IgnoreUserKnownHosts`" specifies whether the `ssh` daemon should ignore the user's `$HOME/.ssh/known_hosts` file during `RhostsRSAAuthentication`. Since we don't allow `.rhosts` files in our server, it is safe to say `yes` here.

`StrictModes yes`
The option "`StrictModes`" specifies whether `ssh` should check user's permissions in their home directory and `rhosts` files before accepting login. This option must always be set to `yes` because sometimes users may accidentally leave their directory or files world-writable.

`X11Forwarding no`
The option "`X11Forwarding`" specifies whether `X11` forwarding should be enabled or not on this server. Since we setup a server without `GUI` installed on it, we can safely turn this option off.

`PrintMotd yes`
The option "`PrintMotd`" specifies whether the `ssh` daemon should print the contents of the `/etc/motd` file when a user logs in interactively. The `/etc/motd` file is also known as "the message of the day".

`SyslogFacility AUTH`
The option "`SyslogFacility`" specifies the facility code used when logging messages from `sshd`. The facility specifies the subsystem that produced the message--in our case, `AUTH`.

`LogLevel INFO`
The option "`LogLevel`" specifies the level that is used when logging messages from `sshd`. `INFO` is a good choice. See the manual page for `sshd` for more information on other possibilities.

`RhostsAuthentication no`

The option "RhostsAuthentication" specifies whether sshd can try to use rhosts based authentication. Because rhosts authentication is insecure you shouldn't use this option.

RhostsRSAAuthentication no
The option "RhostsRSAAuthentication" specifies whether to try rhosts authentication in concert with RSA host authentication.

RSAAuthentication yes
The option "RSAAuthentication" specifies whether to try RSA authentication. It is important to note that it is reserved for the SSH1 protocol only. This option must be set to yes for enhanced security in your sessions if you use SSH1 and only SSH1 since it doesn't apply for the SSH2 protocol (SSH2 use DSA instead of RSA). RSA uses public and private key pairs created with the ssh-keygen utility for authentication purposes.

PasswordAuthentication no
The option "PasswordAuthentication" specifies whether we should use password-based authentication. For strong security, this option must always be set to no. You should put 'PasswordAuthentication no' in the sshd_config file, otherwise people might try to guess the password for the user. With 'PasswordAuthentication no', your public key must be on the computer or no login is allowed: that's what we want. Note that with the Windows client program called "putty" you cannot set this option to no or you will not be able to log into the server using putty.

PermitEmptyPasswords no
This option "PermitEmptyPasswords" is closely related with the above option "PasswordAuthentication" and specifies whether, if password authentication is allowed, the server should allows logging in to accounts with a null password. Since we do not allow password authentication in the server, we can safety turn off this option.

AllowUsers admin
This option "AllowUsers" specifies and controls which users can access ssh services. Multiple users can be specified, separated by spaces.

/etc/pam.d/sshd: The OpenSSH PAM Support Configuration File

For better security of OpenSSH, we'll configure it to use PAM password authentication support. To do that, you must create the /etc/pam.d/sshd file and add the following parameters inside it.

- Create the **sshd** file (touch /etc/pam.d/sshd) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_access.so
account    required     /lib/security/pam_time.so
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_limits.so
session    optional     /lib/security/pam_console.so
```

/etc/rc.d/init.d/sshd: The Openssh Initialization File

The /etc/rc.d/init.d/sshd script file is responsible for automatically starting and stopping the OpenSSH daemon on your server. Loading the sshd daemon, as a standalone daemon, will eliminate load time and will even reduce swapping since non-library code will be shared. This is the best way to start sshd daemon on the system, never use programs like Xinetd or inet to start it.

Step 1
Create the **sshd** script file (touch /etc/rc.d/init.d/sshd) and add the following lines inside it:

```
#!/bin/bash

# Init file for OpenSSH server daemon
#
# chkconfig: 2345 55 25
# description: OpenSSH server daemon
#
# processname: sshd
# config: /etc/ssh/ssh_host_key
# config: /etc/ssh/ssh_host_key.pub
# config: /etc/ssh/ssh_random_seed
# config: /etc/ssh/sshd_config
# pidfile: /var/run/sshd.pid

# source function library
. /etc/rc.d/init.d/functions

RETVAL=0

function start()
{
        if [ ! -s /etc/ssh/ssh_host_key ]; then
                /usr/bin/ssh-keygen -b 1024 -f /etc/ssh/ssh_host_key -N ""
        fi
        if [ ! -s /etc/ssh/ssh_host_dsa_key ]; then
                /usr/bin/ssh-keygen -d -f /etc/ssh/ssh_host_dsa_key -N ""
        fi

        action "Starting sshd:" /usr/sbin/sshd
        RETVAL=$?
        [ "$RETVAL" = 0 ] && touch /var/lock/subsys/sshd
}

function stop()
{
        echo -n "Stopping sshd:"
        killproc /usr/sbin/sshd
        RETVAL=$?
        echo
        [ "$RETVAL" = 0 ] && rm -f /var/lock/subsys/sshd
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
```

```
        stop
        start
        ;;
  reload)
        killproc /usr/sbin/sshd -HUP
        ;;
  condrestart)
        if [ -f /var/lock/subsys/sshd ] ; then
                stop
                start
        fi
        ;;
  status)
        status /usr/sbin/sshd
        ;;
  *)
        echo "Usage: sshd {start|stop|restart|reload|condrestart|status}"
        RETVAL=1
esac
exit $RETVAL
```

Step 2
Once the `openssh` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/sshd
[root@deep /]# chown 0.0 /etc/rc.d/init.d/sshd
```

- To create the symbolic `rc.d` links for `OpenSSH`, use the following command:
```
[root@deep /]# chkconfig --add sshd
[root@deep /]# chkconfig --level 2345 sshd on
```

- To start `OpenSSH` software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/sshd start
Starting sshd:            [OK]
```

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Further documentation

For more details, there are several manual pages about OpenSSH that you can read:

```
$ man ssh (1)            - OpenSSH secure shell client (remote login program)
$ man ssh [slogin] (1)   - OpenSSH secure shell client (remote login program)
$ man ssh-add (1)        - Adds identities for the authentication agent
$ man ssh-agent (1)      - Authentication agent
$ man ssh-keygen (1)     - Authentication key generation
$ man sshd (8)           - Secure shell daemon
$ sftp-server (8)        - SFTP server subsystem
```

OpenSSH Per-User Configuration

After your desired configuration options have been set and sshd daemon is running, it is time to create new private & public keys for our users to establish the secure connection.

Related to manual page for sshd (8):
There are cryptosystems where encryption and decryption are done using separate keys, and it is not possible to derive the decryption key from the encryption key. The idea is that each user creates a public/private key pair for authentication purposes. The server knows the public key, and only the user knows the private key.

The file $HOME/.ssh/authorized_keys2 for SSH2 or $HOME/.ssh/authorized_keys for SSH1 lists the public keys that are permitted for logging in. When the user logs in, the ssh program tells the server which key pair it would like to use for authentication. The server checks if this key is permitted, and if so, send the user (actually the ssh program running on behalf of the user) a challenge, a random number, encrypted by the user's public key. The challenge can only be decrypted using the proper private key. The user's client then decrypts the challenge using the private key, proving that he/she knows the private key but without disclosing it to the server.

Step 1
I'll show you below how to create a new SSH private & public key for one user. This example assumes that secure encrypted connections will be made between Linux servers.

- To create your (DSA) private & public keys for SSH2 of LOCAL, use the commands:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ ssh-keygen -d
Generating DSA parameter and key.
Enter file in which to save the key (/home/gmourani/.ssh/id_dsa):
Created directory '/home/gmourani/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/gmourani/.ssh/id_dsa.
Your public key has been saved in /home/gmourani/.ssh/id_dsa.pub.
The key fingerprint is:
1f:af:aa:22:0a:21:85:3c:07:7a:5c:ae:c2:d3:56:64 gmourani@deep
```

WARNING: The above example assumes that you want to generate (DSA) private & public keys for SSH protocol 2 (highly recommended). If you want to generate (RSA) private & public keys for SSH protocol 1, then you must remove the '-d' option to the key generation command as follow:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ ssh-keygen
```

Removing the '-d' option will generate SSH1 instead of SSH2 private & public keys. The SSH1 private key will be named "identity" and the public key will be "identity.pub"

If you have multiple accounts you might want to create a separate key on each of them. You may want to have separate keys for:

- Your server (1)
- Your server (2)
- Your server (3)

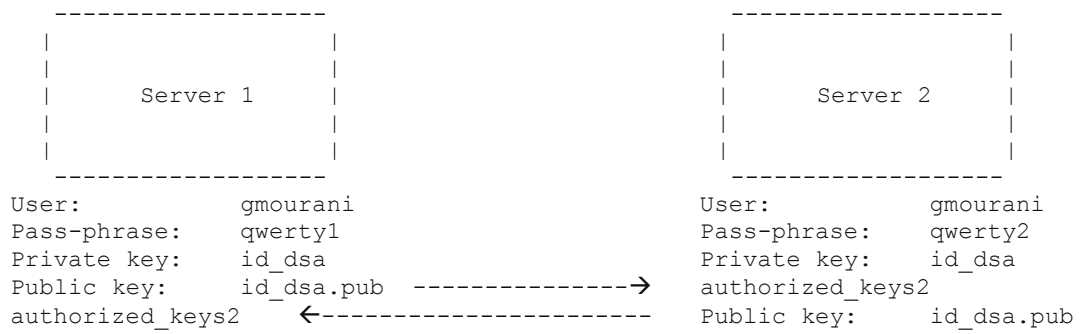
This allows you to limit access between these servers, e.g. not allowing the first server (1) account to access your second server (2) account or the third server (3). This enhances the overall security in the case any of your authentication keys are compromised for any reason.

Step 2
Copy your local public key **id_dsa.pub** for SSH2 or **identity.pub** for SSH1 from the /home/gmourani/.ssh directory remotely under the name, say, "authorized_keys2" for SSH2 or "authorized_keys" for SSH1. One way to copy the file is to use the ftp command or you might need to send your public key in electronic mail to the administrator of the system. Just include the contents of the ~/.ssh/id_dsa.pub or ~/.ssh/identity.pub file in the message.

To resume the required steps:

1) The user creates his/her DSA or RSA keys pair by running ssh-keygen. This stores the private key in $HOME/.ssh/id_dsa (SSH2) or in $HOME/.ssh/identity (SSH1) and the public key in $HOME/.ssh/id_dsa.pub (SSH2) or in $HOME/.ssh/identity.pub (SSH1) into the user's home directory on the LOCAL machine.

2) The user should then copy the id_dsa.pub key (SSH2) or identity.pub key (SSH1) to $HOME/.ssh/authorized_keys2 for SSH2 or to $HOME/.ssh/authorized_keys for SSH1 into his/her home directory on the REMOTE machine (the authorized_keys2 or authorized_keys files corresponds to the conventional $HOME/.rhosts file, and has one key per line, though the lines can be very long).



Public key of user `gmourani` on the first server (1) is sending to the second server (2) under the $HOME directory of user `gmourani` and become '`authorized_keys2`'; the same action is made on the second server (2). The public key of user `gmourani` on server (2) is sending to server (1) under the $HOME directory of user `gmourani` and become '`authorized_keys2`'.

NOTE: `OpenSSH`'s public key is a one-line string. Adding public keys from commercial `SSH` tools which stretch the public key over several lines will not be recognized by `OpenSSH`.

Changing your pass-phrase

You can change the pass-phrase at any time by using the `-p` option of `ssh-keygen`.

- To change the pass-phrase, use the command:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ ssh-keygen -d -p
Enter file in which the key is (/home/gmourani/.ssh/id_dsa):
Enter old passphrase:
Key has comment 'dsa w/o comment'
Enter new passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved with the new passphrase.
```

If you want to change the pass-phrase of a user running `SSH1` protocol then omit the "`-d`" option in the above example.

`OpenSSH` Users Tools

The commands listed belows are some that we use regularly, but many more exist, and you should check the manual pages and documentation of `OpenSSH` for more details.

ssh

The ssh (**S**ecure **Sh**ell) command provides secure encrypted communications between two untrusted hosts over an insecure network. It is a program for securely logging into a remote machine and executing commands from there. It is a suitable replacement for insecure programs like telnet, rlogin, rcp, rdist, and rsh.

- To login to a remote machine, use the command:

```
[root@deep /]# ssh -l <login_name> <hostname>
```

For example:

```
[root@deep /]# ssh -l gmourani deep.openna.com
gmourani@deep.openna.com's password:
Last login: Tue Oct 19 1999 18:13:00 -0400 from deep.openna.com
No mail.
[gmourani@deep gmourani]$
```

Where <login_name> is the name you use to connect to the ssh server and <hostname> is the remote address (you can use IP address here) of your ssh server.

scp

The scp (Secure Copy) utility copies files from the local system to a remote system or vice versa, or even between two remote systems using the scp command.

- To copy files from remote to local system, use the following command:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ scp -p <login_name@hostname>:/dir/for/file
localdir/to/filelocation
```

For example:

```
[gmourani@deep /]$ scp -p gmourani@mail:/etc/test1 /tmp
Enter passphrase for RSA key 'gmourani@mail.openna.com':
test1            |          2 KB |   2.0 kB/s | ETA: 00:00:00 | 100%
```

- To copy files from local to remote system, use the following command:

```
[root@deep /]# su gmourani
[gmourani@deep /]$ scp -p localdir/to/filelocation
<username@hostname>:/dir/for/file
```

For example:

```
[gmourani@deep /]$ scp1 -p /usr/bin/test2 gmourani@mail:/var/tmp
gmourani@mail's password:
test2            |          7 KB |   7.9 kB/s | ETA: 00:00:00 | 100%
```

WARNING: The "-p" option indicates that the modification and access times, as well as modes of the source file, should be preserved on the copy. This is usually desirable. Please check under chapter related to backup in this book for more information about other possible use of SSH technology with Linux.

Some possible uses of OpenSSH

OpenSSH can be used to:

1. Replace telnet, rlogin, rsh, rdist, and rcp programs.
2. Make secure backups over the network.
3. Execute remote commands.
4. Access to corporate resources over the Internet.
5. Transfer files remotely in a secure manner.

List of installed OpenSSH files in your system

> /etc/rc.d/init.d/sshd
> /etc/ssh
> /etc/ssh/ssh_config
> /etc/ssh/sshd_config
> /etc/ssh/ssh_host_key
> /etc/ssh/ssh_host_key.pub
> /etc/ssh/ssh_host_dsa_key
> /etc/ssh/ssh_host_dsa_key.pub
> /etc/ssh/ssh_host_rsa_key
> /etc/ssh/ssh_host_rsa_key.pub
> /etc/ssh/primes
> /etc/pam.d/sshd
> /usr/bin/ssh
> /usr/bin/scp
> /usr/bin/ssh-add
> /usr/bin/ssh-agent
> /usr/bin/ssh-keygen
> /usr/bin/sftp
> /usr/bin/ssh-keyscan
> /usr/bin/slogin
> /usr/sbin/sshd
> /usr/share/man/man1/ssh.1
> /usr/share/man/man1/scp.1
> /usr/share/man/man1/sftp.1
> /usr/share/man/man1/ssh-keyscan.1
> /usr/share/man/man1/ssh-add.1
> /usr/share/man/man1/ssh-agent.1
> /usr/share/man/man1/ssh-keygen.1
> /usr/share/man/man1/slogin.1
> /usr/share/man/man8/sshd.8
> /usr/share/man/man8/sftp-server.8
> /usr/libexec/openssh
> /usr/libexec/openssh/sftp-server

Free SSH Server for Linux

FreSSH

FreSSH Homepage: http://www.fressh.org/

Free SSH Client for MS Windows

Putty

Putty Homepage: http://www.chiark.greenend.org.uk/~sgtatham/putty.html

Tera Term Pro and TTSSH

Tera Term Pro Homepage: http://hp.vector.co.jp/authors/VA002416/teraterm.html
TTSSH Homepage: http://www.zip.com.au/~roca/download.html

Part V Monitoring & System Integrity Related Reference

In this Part

Part V of the book, deals with security tools that we must use to administer our Linux server. These tools are very important to us in our daily work of preventing and checking for possible attacks, holes, etc that will surely come to our network. They will automate many tasks and will help us to administer and keep our Linux servers secure.

Therefore, I highly recommend you install them and once again, these tools are not a bonus, but a requirement that you must have installed on each server on your network. One exception is for `Xinetd`, which is optional and depends of what server and services you have configured. Generally, you don't have to install it, but if you use `IMAP & POP` servers then you must install it or `they` will not work. For me this is the only reason to install `Xinetd`.

13 Monitoring & System Integrity - `sXid`

In this Chapter

Linux sXid

Abstract

SUID/SGID files can be a security hazard. To reduce the risks, we have previously removed the 's' bits from root-owned programs that won't require such privileges (See chapter related to General System Security), but future and existing files may be set with these 's' bits enabled without you being notified.

sXid is an all in one suid/sgid monitoring program designed to be run by cron on a regular basis. Basically it tracks any changes in your s[ug]id files and folders. If there are any new ones, ones that aren't set any more, or they have changed bits or other modes then it reports the changes in an easy to read format via email or on the command line. sXid will automate the task to find all SUID/SGID on your server and report them to you. Once installed you can forget it and it will do the job for you.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest sXid version number is 4.0.1

Packages

The following are based on information as listed by sXid as of 2001/03/25. Please regularly check at ftp://marcus.seva.net/pub/sxid/ for the latest status.

Pristine source code is available from:
sXid Homepage: ftp://marcus.seva.net/pub/sxid/
sXid FTP Site: 137.155.111.51
You must be sure to download: sxid_4.0.1.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install sXid, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > sXid1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > sXid2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff sXid1 sXid2 > sXid-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `sXid`

Below are the required steps that you must make to configure, compile and optimize the `sXid` software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp sxid_version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf sxid_version.tar.gz
```

Step 2
After that, move into the newly created `sXid` directory then configure and optimize it.

- To move into the newly created `sXid` directory use the following command:

```
[root@deep tmp]# cd sxid-4.0.1/
```

- To configure and optimize `sXid` use the following compile lines:

```
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--mandir=/usr/share/man
```

WARNING: Pay special attention to the compile `CFLAGS` line above. We optimize `sXid` for an i686 CPU architecture with the parameter "`-march=i686` and `-mcpu=i686`". Please don't forget to adjust this `CFLAGS` line to reflect your own system.

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `sXid` in the server:

```
[root@deep sXid-4.0.1]# cd
[root@deep /root]# find /* > sXid1
[root@deep /root]# cd /var/tmp/sxid-4.0.1/
[root@deep sxid-4.0.1]# make install
[root@deep sxid-4.0.1]# cd
[root@deep /root]# find /* > sXid2
[root@deep /root]# diff sXid1 sXid2 > sXid-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 4
Once the compilation, optimization and installation of the software has been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `sXid` and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf sxid-version/
[root@deep tmp]# rm -f sxid_version_tar.gz
```

The `rm` command as used above will remove all the source files we have used to compile and install `sXid`. It will also remove the `sXid` compressed archive from the `/var/tmp` directory.

Configuring `sXid`

After building `sXid`, your next step is to verify or change, if necessary, options in your `sXid` configuration files. These files are:

- ✓ `/etc/sxid.conf` (The `sXid` Configuration File)
- ✓ `/etc/cron.daily/sxid` (The `sXid` Cron File)

`/etc/sxid.conf`: The `sXid` Configuration File

The configuration file for `sXid` allows you to set options that modify the operation of the program. It is well commented and very basic. We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

Step 1
Edit the **`sxid.conf`** file (`vi /etc/sxid.conf`) and set your needs. Below is what we recommend you.

```
# Configuration file for sXid
# Note that all directories must be absolute with no trailing /'s

# Where to begin our file search
SEARCH = "/"

# Which subdirectories to exclude from searching
EXCLUDE = "/proc /mnt /cdrom /floppy"

# Who to send reports to
EMAIL = "noc@openna.com"

# Always send reports, even when there are no changes?
ALWAYS_NOTIFY = "no"

# Where to keep interim logs. This will rotate 'x' number of
# times based on KEEP_LOGS below
LOG_FILE = "/var/log/sxid.log"

# How many logs to keep
```

```
KEEP_LOGS = "5"

# Rotate the logs even when there are no changes?
ALWAYS_ROTATE = "no"

# Directories where +s is forbidden (these are searched
# even if not explicitly in SEARCH), EXCLUDE rules apply
FORBIDDEN = "/home /tmp"

# Remove (-s) files found in forbidden directories?
ENFORCE = "yes"

# This implies ALWAYS_NOTIFY. It will send a full list of
# entries along with the changes
LISTALL = "no"

# Ignore entries for directories in these paths
# (this means that only files will be recorded, you
# can effectively ignore all directory entries by
# setting this to "/"). The default is /home since
# some systems have /home g+s.
IGNORE_DIRS = "/home"

# File that contains a list of (each on it's own line)
# of other files that sxid should monitor. This is useful
# for files that aren't +s, but relate to system
# integrity (tcpd, inetd, apache...).
# EXTRA_LIST = "/etc/sxid.list"

# Mail program. This changes the default compiled in
# mailer for reports. You only need this if you have changed
# it's location and don't want to recompile sxid.
MAIL_PROG = "/bin/mail"
```

Step 2
Now, for security reasons, change the mode of this file to be 0400.

- This procedure can be accomplished with the following command:
  `[root@deep /]# chmod 400 /etc/sxid.conf`

`/etc/cron.daily/sxid`: The `sXid` Cron File

The `sxid` file is a small script executed automatically by the `crond` program of your server each day to tracks any changes in your s[ug]id files and folders. If there are any new ones, ones that aren't set any more, or they have changed bits or other modes then it reports the changes. If you intend to automate this task, follow the simple steps below.

Step 1
Create the **`sxid`** script file (`touch /etc/cron.daily/sxid`) and add the following lines:

```
#!/bin/sh

SXID_OPTS=

if [ -x /usr/bin/sxid ]; then
        /usr/bin/sxid ${SXID_OPTS}
fi
```

Step2
Now, make this script executable and change its permission mode to be 0700.

- This procedure can be accomplished with the following command:
  `[root@deep /]# chmod 700 /etc/cron.daily/sxid`

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Further documentation

For more details, there are some manual pages you can read:

```
$ man sxid.conf (5) - Configuration settings for sxid
$ man sxid (1)      - Check for changes in s[ug]id files and directories
```

`sXid` Administrative Tools

After your desired configuration options have been set and the program is running, we can play with its utility. The `sXid` software is meant to run as a cronjob. It must run once a day, but busy shell boxes may want to run it twice a day. You can also run this manually for spot-checking.

- To run `sxid` manually, use the command:

```
[root@deep /]# sxid -k
sXid Vers  : 4.0.1
Check run  : Wed Oct 4 15:42:20 2000
This host  : deep.openna.com
Spotcheck  : /root
Excluding  : /proc /mnt /cdrom /floppy
Ignore Dirs: /home
Forbidden  : /home /tmp
  (enforcing removal of s[ug]id bits in forbidden paths)


No changes found
```

This checks for changes by recursing the current working directory. Log files will not be rotated and no email sent. All output will go to stdout.

List of installed `sXid` files in your system

> /etc/sxid.conf
> /usr/bin/sxid
> /usr/share/man/man1/sxid.1
> /usr/share/man/man5/sxid.conf.5

14 Monitoring & System Integrity - `Logcheck`

In this Chapter

Linux Logcheck

Abstract
One of the most important tasks in the security world is to regularly check the log files. Often the daily activities of an administrator don't allow them the time to do this task and this can bring about problems.

As explained in the [Logcheck abstract]:
Don't let the media image fool you, most hackers you'll run across are not very crafty and make a lot of noise ratting your system's door knob…then again they can be as noisy as they want really because there is a 99.99% chance the sysadmins won't know anyway <Craig>.

Auditing and logging system events is important! What is more important is that system administrators be aware of these events so they can prevent problems that will inevitably occur if you have a system connected to the Internet. Unfortunately for most Unices it doesn't matter how much you log activity if nobody ever checks the logs, which is often the case. This is where logcheck will help.

Logcheck automates the auditing process and weeds out "normal" log information to give you a condensed look at problems and potential troublemakers mailed to wherever you please. Logcheck is a software package that is designed to automatically run and check system log files for security violations and unusual activity. Logcheck utilizes a program called logtail that remembers the last position it read from in a log file and uses this position on subsequent runs to process new information.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Logcheck version number is 1.1.1

Packages
The following are based on information as listed by Abacus as of 2001/03/25. Please regularly check at http://www.psionic.com/abacus/logcheck/ for the latest status.

Pristine source code is available from:
Logcheck Homepage Site: http://www.psionic.com/abacus/logcheck/
You must be sure to download: logcheck-1.1.1.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `Logcheck`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > Logcheck1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > Logcheck2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff Logcheck1 Logcheck2 > Logcheck-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `Logcheck`

Below are the required steps that you must make to configure, compile and optimize the `logcheck` software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp logcheck-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf logcheck-version.tar.gz
```

Step 2
After that, move into the newly created `Logcheck` directory and modify some of its files as shown below to specify the installation paths, configuration, compilation and optimizations flags for your Linux system. We must hack those files to be compliant with Linux file system structure and install/optimize `Logcheck` under our PATH Environment variable.

- To move into the newly created `Logcheck` directory use the following command:
```
[root@deep tmp]# cd logcheck-1.1.1/
```

Step 2.1
The first file that we will work on it is named `logcheck.sh` located under the `/systems/linux` subdirectory of the `Logcheck` source directory. Into this file, we will change the default location of different `Logcheck` configuration files.

- Edit the **`logcheck.sh`** file (`vi +34 systems/linux/logcheck.sh`) and change all of the targeted lines in the order shown below:

a) `vi +34 systems/linux/logcheck.sh` and change the line:

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/ucb:/usr/local/bin
```

To read:

```
PATH=/bin:/sbin:/usr/bin:/usr/sbin
```

b) `vi +47 systems/linux/logcheck.sh` and change the line:

```
LOGTAIL=/usr/local/bin/logtail
```

To read:

```
LOGTAIL=/usr/sbin/logtail
```

c) `vi +55 systems/linux/logcheck.sh` and change the line:

```
TMPDIR=/usr/local/etc/tmp
```

To read:

```
TMPDIR=/tmp/logcheck$$-$RANDOM
```

d) `vi +92 systems/linux/logcheck.sh` and change the line:

HACKING_FILE=/usr/local/etc/logcheck.hacking

To read:

HACKING_FILE=/etc/logcheck/logcheck.hacking

e) `vi +101 systems/linux/logcheck.sh` and change the line:

```
VIOLATIONS_FILE=/usr/local/etc/logcheck.violations
```

To read:

```
VIOLATIONS_FILE=/etc/logcheck/logcheck.violations
```

f) `vi +118 systems/linux/logcheck.sh` and change the line:

```
VIOLATIONS_IGNORE_FILE=/usr/local/etc/logcheck.violations.ignore
```

To read:

```
VIOLATIONS_IGNORE_FILE=/etc/logcheck/logcheck.violations.ignore
```

g) `vi +125 systems/linux/logcheck.sh` and change the line:

```
IGNORE_FILE=/usr/local/etc/logcheck.ignore
```

To read:

```
IGNORE_FILE=/etc/logcheck/logcheck.ignore
```

h) `vi +148 systems/linux/logcheck.sh` and add the following two lines between:

```
rm -f $TMPDIR/check.$$ $TMPDIR/checkoutput.$$ $TMPDIR/checkreport.$$
rm -rf $TMPDIR
mkdir $TMPDIR
if [ -f $TMPDIR/check.$$ -o -f $TMPDIR/checkoutput.$$ -o -f
$TMPDIR/checkreport.$$ ]; then
        echo "Log files exist in $TMPDIR directory that cannot be
removed. This
may be an attempt to spoof the log checker." \
        | $MAIL -s "$HOSTNAME $DATE ACTIVE SYSTEM ATTACK!" $SYSADMIN
        exit 1
fi
```

i) `vi +224 systems/linux/logcheck.sh` and add the following one line between:

```
if [ ! -s $TMPDIR/check.$$ ]; then
     rm -f $TMPDIR/check.$$
     rm -rf $TMPDIR
     exit 0
fi
```

j) `vi +274 systems/linux/logcheck.sh` and add the following one line between:

```
# Clean Up
rm -f $TMPDIR/check.$$ $TMPDIR/checkoutput.$$ $TMPDIR/checkreport.$$
rm -rf $TMPDIR
```

Step 2.2
The second and final file that we must modify is the `Makefile` of `Logcheck`. As for the `logcheck.sh` file above, we will change the default location of some `Logcheck` files and binaries. Also we will be adding our optimization flags to this `Makefile` file to speed up our `Logcheck` software.

- Edit the **`Makefile`** file (`vi +9 Makefile`) and change all of the targeted lines in the order shown below:

a) `vi +9 Makefile` and change the line:

```
CC = cc
```

To read:

```
CC = gcc
```

b) `vi +14 Makefile` and change the line:

```
CFLAGS = -O
```

To read:

```
CFLAGS = -O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer
```

WARNING: Pay special attention to the compile `CFLAGS` line above. We optimize `Logcheck` for an i686 CPU architecture with the parameter "`-march=i686` and `-mcpu=i686`". Please don't forget to adjust this `CFLAGS` line to reflect your own system and architecture.

c) `vi +22 Makefile` and change the line:

```
INSTALLDIR = /usr/local/etc
```

To read:

```
INSTALLDIR = /etc/logcheck
```

d) `vi +25 Makefile` and change the line:

```
INSTALLDIR_BIN = /usr/local/bin
```

To read:

```
INSTALLDIR_BIN = /usr/sbin
```

e) `vi +30 Makefile` and change the line:

```
INSTALLDIR_SH = /usr/local/etc
```

To read:

```
INSTALLDIR_SH = /usr/sbin
```

f) `vi +66 Makefile` and change/remove the lines:

```
@echo "Creating temp directory $(TMPDIR)"
@if [ ! -d $(TMPDIR) ]; then /bin/mkdir $(TMPDIR); fi
@echo "Setting temp directory permissions"
chmod 700 $(TMPDIR)
```

To read:

```
#@echo "Creating temp directory $(TMPDIR)"
#@if [ ! -d $(TMPDIR) ]; then /bin/mkdir $(TMPDIR); fi
#@echo "Setting temp directory permissions"
#chmod 700 $(TMPDIR)
```

g) `vi +75 Makefile` and change the line:

```
cp ./systems/$(SYSTYPE)/logcheck.sh $(INSTALLDIR_SH)
```

To read:

```
cp ./systems/$(SYSTYPE)/logcheck.sh $(INSTALLDIR_SH)/logcheck
```

h) `vi +78 Makefile` and change the line:

```
chmod 700 $(INSTALLDIR_SH)/logcheck.sh
```

To read:

```
chmod 700 $(INSTALLDIR_SH)/logcheck
```

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `Logcheck` in the server:

```
[root@deep logcheck-1.1.1]# cd
[root@deep /root]# find /* > Logcheck1
[root@deep /root]# cd /var/tmp/logcheck-1.1.1/
[root@deep logcheck-1.1.1]# mkdir -m700 /etc/logcheck
[root@deep logcheck-1.1.1]# make linux
[root@deep logcheck-1.1.1]# cd
[root@deep /root]# find /* > Logcheck2
[root@deep /root]# diff Logcheck1 Logcheck2 > Logcheck-Installed
```

The above commands will configure the software for the Linux operating system, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations. The changes made to the Logcheck files will configure the software to use the compiler optimization flags specific to our system, and locate all files related to Logcheck software to the destination target directories we have chosen to be compliant with the Linux file system structure.

Step 4
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Logcheck and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf logcheck-version/
[root@deep tmp]# rm -f logcheck-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Logcheck. It will also remove the Logcheck compressed archive from the /var/tmp directory.

Configuring Logcheck

After building Logcheck, your next step is to verify or change, if necessary, the options in your Logcheck configuration files. Those files are:

- ✓ /etc/logcheck/logcheck.hacking
- ✓ /etc/logcheck/logcheck.ignore
- ✓ /etc/logcheck/logcheck.violations
- ✓ /etc/logcheck/logcheck.violations.ignore

From the default install, there is no Logcheck configuration files to modify, the default entries look fine and if you want to make some personal adjustment, all you have to do is to edit the related Logcheck configuration file. More information about the operation of each one is contained into the INSTALL file of Logcheck under its uncompressed source directory.

Step 1
Although the fact that there is no Logcheck configuration files to change, the last action to make before using the program is to automate it, to do that, create a file named logcheck under the /etc/cron.daily directory and add the following lines to set Logcheck to run once per day.

- To create the logcheck file under /etc/cron.daily directory with its required lines to run once per day, type the following lines in your terminal (as root):

```
cat <<EOF > /etc/cron.daily/logcheck
# !/bin/sh
# Daily check Log files for security violations and unusual activity
/usr/sbin/logcheck
EOF
```

Step 2
Now, make this script executable and change its mode to be 0700.

- This procedure can be accomplished with the following command:
  `[root@deep /]# chmod 700 /etc/cron.daily/logcheck`

WARNING: Remember, in our configuration and installation, `Logcheck` does not report anything via email if it has nothing useful to say.

List of installed `Logcheck` files in your system

> /etc/cron.daily/logcheck
> /etc/logcheck
> /etc/logcheck/logcheck.hacking
> /etc/logcheck/logcheck.ignore
> /etc/logcheck/logcheck.violations
> /etc/logcheck/logcheck.violations.ignore
> /usr/sbin/logcheck
> /usr/sbin/logtail

15 Monitoring & System Integrity - `PortSentry`

In this Chapter

Linux PortSentry

Abstract
Firewalls help us to protect our network from intruders. With them we can choose which ports we want to open and which ones we don't. This information is kept private by your organization. Nobody on the outside knows this information, but attackers, as well as spammers, know that for some kinds of attacks you can use a special program to scan all the ports on a server to gleam this valuable information (what is open and what is not).

As explained in the [PortSentry abstract]:
A port scan is a symptom of a larger problem coming your way. It is often the pre-cursor for an attack and is a critical piece of information for properly defending your information resources. PortSentry is a program designed to detect and respond to port scans against a target host in real-time and has a number of options to detect port scans. When it finds one it can react in the following ways:

- ✓ A log indicating the incident is made via syslog().
- ✓ The target host is automatically dropped.
- ✓ The local host is automatically re-configured to route all traffic to the target to a dead host to make the target system disappear.
- ✓ The local host is automatically re-configured to drop all packets from the target via a local packet filter.

The purpose of this is to give an admin a heads up that their host is being probed.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest PortSentry version number is 1.0

Packages
The following is based on information as listed by Abacus as of 2001/03/25. Please regularly check at http://www.psionic.com/abacus/portsentry/ for the latest status.

Pristine source code is available from:
PortSentry Homepage Site: http://www.psionic.com/abacus/portsentry/
You must be sure to download: portsentry-1.0.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `PortSentry`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > PortSentry1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > PortSentry2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff PortSentry1 PortSentry2 > PortSentry-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `PortSentry`

Below are the required steps that you must make to configure, compile and optimize the `PortSentry` software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp portsentry-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf portsentry-version.tar.gz
```

Step 2
After that, move into the newly created `PortSentry` directory and modify some of its files as shown below to specify the installation paths, configuration, compilation and optimizations flags for your Linux system. We must hack those files to be compliant with Linux file system structure and install/optimize `PortSentry` under our `PATH` Environment variable.

- To move into the newly created `PortSentry` directory use the following command:
```
[root@deep tmp]# cd portsentry-1.0/
```

Step 2.1
The first file that we will work on it is named `portsentry.conf` located under the source directory of `PortSentry`. In this file, we will change the default location of different `PortSentry` configuration files.

- Edit the **`portsentry.conf`** file (`vi +83 portsentry.conf`) and change all of the targeted lines in the order shown below:

  ```
  IGNORE_FILE="/usr/local/psionic/portsentry/portsentry.ignore"
  ```

  To read:

  ```
  IGNORE_FILE="/etc/portsentry/portsentry.ignore"
  ```

`vi +85 portsentry.conf` and change the line:

```
HISTORY_FILE="/usr/local/psionic/portsentry/portsentry.history"
```

To read:

```
HISTORY_FILE="/var/log/portsentry/portsentry.history"
```

`vi +87 portsentry.conf` and change the line:

```
BLOCKED_FILE="/usr/local/psionic/portsentry/portsentry.blocked"
```

To read:

```
BLOCKED_FILE="/var/log/portsentry/portsentry.blocked"
```

Step 2.2
The second file that we will modify is the `portsentry_config.h` header file. Under this file, we will change the default install location of the configuration file for `PortSentry`.

- Edit the **`portsentry_config.h`** file (`vi +34 portsentry_config.h`) and change the following line:

  ```
  #define CONFIG_FILE "/usr/local/psionic/portsentry/portsentry.conf"
  ```

  To read:

  ```
  #define CONFIG_FILE "/etc/portsentry/portsentry.conf"
  ```

Step 2.3
The final file that we must modify is the `Makefile` of `PortSentry`. The changes we make to this file is to addi our optimization flags to speed up our `PortSentry` software.

- Edit the **`Makefile`** file (`vi +24 Makefile`) and change all of the targeted lines in the order shown below:

  ```
  CC = cc
  ```

  To read:

  ```
  CC = gcc
  ```

`vi +29 Makefile` and change the line:

```
CFLAGS = -O -Wall
```

To read:

```
CFLAGS = -O3 -march=i686 -mcpu=i686 –funroll-loops –fomit-frame-pointer
-Wall
```

`vi +38 Makefile` and change the line:

```
INSTALLDIR = /usr/local/psionic
```

To read:

```
INSTALLDIR = /etc
```

WARNING: Pay special attention to the compile `CFLAGS` line above. We optimize `Portsentry` for an i686 CPU architecture with the parameter "`-march=i686` and `-mcpu=i686`". Please don't forget to adjust this `CFLAGS` line to reflect your own system.

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `PortSentry` in the server:

```
[root@deep portsentry-1.0]# cd
[root@deep /root]# find /* > PortSentry1
[root@deep /root]# cd /var/tmp/portsentry-1.0/
[root@deep portsentry-1.0]# make linux
[root@deep portsentry-1.0]# install -m700 -s portsentry /usr/sbin/
[root@deep portsentry-1.0]# mkdir -p -m700 /etc/portsentry
[root@deep portsentry-1.0]# install -m600 portsentry.conf /etc/portsentry/
[root@deep portsentry-1.0]# install -m600 portsentry.ignore /etc/portsentry/
[root@deep portsentry-1.0]# touch /etc/portsentry/portsentry.modes
[root@deep portsentry-1.0]# chmod 600 /etc/portsentry/portsentry.modes
[root@deep portsentry-1.0]# mkdir -p -m700 /var/log/portsentry
[root@deep portsentry-1.0]# touch /var/log/portsentry/portsentry.blocked
[root@deep portsentry-1.0]# touch /var/log/portsentry/portsentry.history
[root@deep portsentry-1.0]# cd
[root@deep /root]# find /* > PortSentry2
[root@deep /root]# diff PortSentry1 PortSentry2 > PortSentry-Installed
```

The above commands will configure the software for the Linux operating system, compile all source files into executable binaries, and then install the binaries and all files related to `Portsentry` software to the destination target directories we have chosen.

Step 4
Once configuration, compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `PortSentry` and its related source directory, use the following commands:
  ```
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# rm -rf portsentry-version/
  [root@deep tmp]# rm -f portsentry-version.tar.gz
  ```

The `rm` command as used above will remove all the source files we have used to compile and install `PortSentry`. It will also remove the `PortSentry` compressed archive from the `/var/tmp` directory.

Configuring `PortSentry`

After building `PortSentry`, your next step is to verify or change, if necessary, options in your `PortSentry` configuration files. Those files are:

- ✓ `/etc/portsentry/portsentry.conf` (The `PortSentry` Configuration File)
- ✓ `/etc/portsentry/portsentry.ignore` (The `PortSentry` Ignore File)
- ✓ `/etc/portsentry/portsentry.modes` (The `PortSentry` Modes File)
- ✓ `/etc/rc.d/init.d/portsentry` (The `PortSentry` Initialization File)
- ✓ `/etc/logrotate.d/portsentry` (The `PortSentry` Log Rotation File)

`/etc/portsentry/portsentry.conf`: The `PortSentry` Config File

The `portsentry.conf` file is the main configuration file for `PortSentry`, which allows you to set options that modify the operation of the program. It is well commented and very basic. We must change the default one to fit our requirements and operating system.

From this configuration file you can specify which ports you want `PortSentry` to listen to, which IP addresses are denied, monitor, ignore, disables automatic responses, and so on. For more information read the `README.install` file under the `PortSentry` uncompressed source directory. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **`portsentry.conf`** file (`vi /etc/portsentry/portsentry.conf`) and set your needs. Below is what we recommend you.

```
# PortSentry Configuration
#
# $Id: portsentry.conf,v 1.13 1999/11/09 02:45:42 crowland Exp crowland $
#
# IMPORTANT NOTE: You CAN NOT put spaces between your port arguments.
#
# The default ports will catch a large number of common probes
#
# All entries must be in quotes.


########################
# Port Configurations #
########################
#
#
# Some example port configs for classic and basic Stealth modes
#
# I like to always keep some ports at the "low" end of the spectrum.
# This will detect a sequential port sweep really quickly and usually
# these ports are not in use (i.e. tcpmux port 1)
#
# ** X-Windows Users **: If you are running X on your box, you need to be
sure
# you are not binding PortSentry to port 6000 (or port 2000 for
OpenWindows users).
# Doing so will prevent the X-client from starting properly.
#
# These port bindings are *ignored* for Advanced Stealth Scan Detection
Mode.
#

# Un-comment these if you are really anal:
#TCP_PORTS="1,7,9,11,15,70,79,80,109,110,111,119,138,139,143,512,513,514,
515,540,635,1080,1524,2000,2001,4000,4001,5742,6000,6001,6
667,12345,12346,20034,30303,32771,32772,32773,32774,31337,40421,40425,497
24,54320"
#UDP_PORTS="1,7,9,66,67,68,69,111,137,138,161,162,474,513,517,518,635,640
,641,666,700,2049,32770,32771,32772,32773,32774,31337,5432
1"
#
# Use these if you just want to be aware:
TCP_PORTS="1,11,15,79,111,119,143,540,635,1080,1524,2000,5742,6667,12345,
12346,20034,31337,32771,32772,32773,32774,40421,49724,54320"
UDP_PORTS="1,7,9,69,161,162,513,635,640,641,700,32770,32771,32772,32773,32774,31337,5
4321"
#
# Use these for just bare-bones
#TCP_PORTS="1,11,15,110,111,143,540,635,1080,524,2000,12345,12346,20034,3
2771,32772,32773,32774,49724,54320"
#UDP_PORTS="1,7,9,69,161,162,513,640,700,32770,32771,32772,32773,32774,31
337,54321"
```

```
##########################################
# Advanced Stealth Scan Detection Options #
##########################################
#
# This is the number of ports you want PortSentry to monitor in Advanced
mode.
# Any port *below* this number will be monitored. Right now it watches
# everything below 1023.
#
# On many Linux systems you cannot bind above port 61000. This is because
# these ports are used as part of IP masquerading. I don't recommend you
# bind over this number of ports. Realistically: I DON'T RECOMMEND YOU
MONITOR
# OVER 1023 PORTS AS YOUR FALSE ALARM RATE WILL ALMOST CERTAINLY RISE.
You've
# been warned! Don't write me if you have have a problem because I'll
only tell
# you to RTFM and don't run above the first 1023 ports.
#
#
ADVANCED_PORTS_TCP="1023"
ADVANCED_PORTS_UDP="1023"
#
# This field tells PortSentry what ports (besides listening daemons) to
# ignore. This is helpful for services like ident that services such
# as FTP, SMTP, and wrappers look for but you may not run (and probably
# *shouldn't* IMHO).
#
# By specifying ports here PortSentry will simply not respond to
# incoming requests, in effect PortSentry treats them as if they are
# actual bound daemons. The default ports are ones reported as
# problematic false alarms and should probably be left alone for
# all but the most isolated systems/networks.
#
# Default TCP ident and NetBIOS service
ADVANCED_EXCLUDE_TCP="113,139"
# Default UDP route (RIP), NetBIOS, bootp broadcasts.
ADVANCED_EXCLUDE_UDP="520,138,137,67"


######################
# Configuration Files#
######################
#
# Hosts to ignore
IGNORE_FILE="/etc/portsentry/portsentry.ignore"
# Hosts that have been denied (running history)
HISTORY_FILE="/var/log/portsentry/portsentry.history"
# Hosts that have been denied this session only (temporary until next
restart)
BLOCKED_FILE="/var/log/portsentry/portsentry.blocked"

####################
# Response Options#
####################
# Options to dispose of attacker. Each is an action that will
# be run if an attack is detected. If you don't want a particular
# option then comment it out and it will be skipped.
#
# The variable $TARGET$ will be substituted with the target attacking
# host when an attack is detected. The variable $PORT$ will be
substituted
# with the port that was scanned.
```

```
#
###################
# Ignore Options #
###################
# These options allow you to enable automatic response
# options for UDP/TCP. This is useful if you just want
# warnings for connections, but don't want to react for
# a particular protocol (i.e. you want to block TCP, but
# not UDP). To prevent a possible Denial of service attack
# against UDP and stealth scan detection for TCP, you may
# want to disable blocking, but leave the warning enabled.
# I personally would wait for this to become a problem before
# doing though as most attackers really aren't doing this.
# The third option allows you to run just the external command
# in case of a scan to have a pager script or such execute
# but not drop the route. This may be useful for some admins
# who want to block TCP, but only want pager/e-mail warnings
# on UDP, etc.
#
#
# 0 = Do not block UDP/TCP scans.
# 1 = Block UDP/TCP scans.
# 2 = Run external command only (KILL_RUN_CMD)

BLOCK_UDP="1"
BLOCK_TCP="1"

####################
# Dropping Routes:#
####################
# This command is used to drop the route or add the host into
# a local filter table.
#
# The gateway (333.444.555.666) should ideally be a dead host on
# the *local* subnet. On some hosts you can also point this at
# localhost (127.0.0.1) and get the same effect. NOTE THAT
# 333.444.555.66 WILL *NOT* WORK. YOU NEED TO CHANGE IT!!
#
# All KILL ROUTE OPTIONS ARE COMMENTED OUT INITIALLY. Make sure you
# uncomment the correct line for your OS. If you OS is not listed
# here and you have a route drop command that works then please
# mail it to me so I can include it. ONLY ONE KILL_ROUTE OPTION
# CAN BE USED AT A TIME SO DON'T UNCOMMENT MULTIPLE LINES.
#
# NOTE: The route commands are the least optimal way of blocking
# and do not provide complete protection against UDP attacks and
# will still generate alarms for both UDP and stealth scans. I
# always recommend you use a packet filter because they are made
# for this purpose.
#

# Generic
#KILL_ROUTE="/sbin/route add $TARGET$ 333.444.555.666"

# Generic Linux
#KILL_ROUTE="/sbin/route add -host $TARGET$ gw 333.444.555.666"

# Newer versions of Linux support the reject flag now. This
# is cleaner than the above option.
KILL_ROUTE="/sbin/route add -host $TARGET$ reject"

# Generic BSD (BSDI, OpenBSD, NetBSD, FreeBSD)
#KILL_ROUTE="/sbin/route add $TARGET$ 333.444.555.666"
```

```
# Generic Sun
#KILL_ROUTE="/usr/sbin/route add $TARGET$ 333.444.555.666 1"

# NEXTSTEP
#KILL_ROUTE="/usr/etc/route add $TARGET$ 127.0.0.1 1"

# FreeBSD (Not well tested.)
#KILL_ROUTE="route add -net $TARGET$ -netmask 255.255.255.255 127.0.0.1 -
blackhole"

# Digital UNIX 4.0D (OSF/1 / Compaq Tru64 UNIX)
#KILL_ROUTE="/sbin/route add -host -blackhole $TARGET$ 127.0.0.1"

# Generic HP-UX
#KILL_ROUTE="/usr/sbin/route add net $TARGET$ netmask 255.255.255.0
127.0.0.1"

##
# Using a packet filter is the preferred method. The below lines
# work well on many OS's. Remember, you can only uncomment *one*
# KILL_ROUTE option.
##

# For those of you running Linux with ipfwadm installed you may like
# this better as it drops the host into the packet filter.
# You can only have one KILL_ROUTE turned on at a time though.
# This is the best method for Linux hosts.
#
#KILL_ROUTE="/sbin/ipfwadm -I -i deny -S $TARGET$ -o"
#
# This version does not log denied packets after activation
#KILL_ROUTE="/sbin/ipfwadm -I -i deny -S $TARGET$"
#
# New ipchain support for Linux kernel version 2.102+
#KILL_ROUTE="/sbin/ipchains -I input -s $TARGET$ -j DENY -l"
#
# For those of you running FreeBSD (and compatible) you can
# use their built in firewalling as well.
#
#KILL_ROUTE="/sbin/ipfw add 1 deny all from $TARGET$:255.255.255.255 to
any"

################
# TCP Wrappers#
################
# This text will be dropped into the hosts.deny file for wrappers
# to use. There are two formats for TCP wrappers:
#
# Format One: Old Style - The default when extended host processing
# options are not enabled.
#
#KILL_HOSTS_DENY="ALL: $TARGET$"
#
# Format Two: New Style - The format used when extended option
# processing is enabled. You can drop in extended processing
# options, but be sure you escape all '%' symbols with a backslash
# to prevent problems writing out (i.e. \%c \%h )
#
#KILL_HOSTS_DENY="ALL: $TARGET$ : DENY"

###################
# External Command#
```

```
####################
# This is a command that is run when a host connects, it can be whatever
# you want it to be (pager, etc.). This command is executed before the
# route is dropped. I NEVER RECOMMEND YOU PUT IN RETALIATORY ACTIONS
# AGAINST THE HOST SCANNING YOU. TCP/IP is an *unauthenticated protocol*
# and people can make scans appear out of thin air. The only time it
# is reasonably safe (and I *never* think it is reasonable) to run
# reverse probe scripts is when using the "classic" -tcp mode. This
# mode requires a full connect and is very hard to spoof.
#
#KILL_RUN_CMD="/some/path/here/script $TARGET$ $PORT$"


#####################
# Scan trigger value#
#####################
# Enter in the number of port connects you will allow before an
# alarm is given. The default is 0 which will react immediately.
# A value of 1 or 2 will reduce false alarms. Anything higher is
# probably not necessary. This value must always be specified, but
# generally can be left at 0.
#
# NOTE: If you are using the advanced detection option you need to
# be careful that you don't make a hair trigger situation. Because
# Advanced mode will react for *any* host connecting to a non-used
# below your specified range, you have the opportunity to really
# break things. (i.e someone innocently tries to connect to you via
# SSL [TCP port 443] and you immediately block them). Some of you
# may even want this though. Just be careful.
#

SCAN_TRIGGER="0"

######################
# Port Banner Section#
######################
#
# Enter text in here you want displayed to a person tripping the
PortSentry.
# I *don't* recommend taunting the person as this will aggravate them.
# Leave this commented out to disable the feature
#
# Stealth scan detection modes don't use this feature
#
PORT_BANNER="** UNAUTHORIZED ACCESS PROHIBITED *** YOUR CONNECTION
ATTEMPT HAS BEEN LOGGED. GO AWAY."

# EOF
```

/etc/portsentry/portsentry.ignore:

The `portsentry.ignore` file is where you add any host you want to be ignored if it connects to a tripwired port. This should always contain at least the `localhost (127.0.0.1)` and the IP's of the local interfaces `(lo)`. It is not recommend that you put in every IP on your network. It is well commented and very simple to understand.

- Edit the **`portsentry.ignore`** file (`vi /etc/portsentry/portsentry.ignore`) and add in any host you want to be ignored if it connects to a tripwired port. Below is what we recommend.

```
# Put hosts in here you never want blocked. This includes the IP
addresses of all local interfaces on the protected host (i.e virtual
host, mult-home) Keep 127.0.0.1 and 0.0.0.0 to keep people from playing
games.

127.0.0.1
0.0.0.0
```

/etc/portsentry/portsentry.modes: The PortSentry Modes File

The PortSentry program can be configured in six different modes of operation, but be aware that only one protocol mode type can be started at a time. To be more accurate, you can start one TCP mode and one UDP mode, so two TCP modes and one UDP mode, for example, won't work.

- The available PortSentry modes are:

- ✓ portsentry –tcp (Basic port-bound TCP mode)
- ✓ portsentry –udp (Basic port-bound UDP mode)
- ✓ portsentry –stcp (Stealth TCP scan detection mode)
- ✓ portsentry –sudp ("Stealth" UDP scan detection mode)
- ✓ portsentry –atcp (Advanced "Stealth" TCP scan detection mode)
- ✓ portsentry –audp (Advanced "Stealth" UDP scan detection mode)

For the best use of this software it is preferable to start PortSentry in **Advanced TCP stealth scan detection mode** and **stealth UDP scan detection mode**. For information about the other modes available, please refer to the README.install and README.stealth file under the PortSentry source directory.

With the **Advanced TCP stealth scan detection mode** "-atcp", PortSentry will first check to see what ports you have running on your server, then remove these ports from monitoring and will begin watching the remaining ports. This is very powerful and reacts exceedingly quickly for port scanners. It also uses very little CPU time. This mode is the most sensitive and the most effective of all the protection options. With the **stealth UDP scan detection mode** "-sudp", the PortSentry UDP ports will be listed and then monitored.

The six different modes of operation under which PortSentry can operate must be specified in the configuration file named portsentry.modes located in the /etc/portsentry/ directory. We can add inside this file all the six possible modes of PortSentry, then uncomment the two you want to use for the Linux server.

- Edit the **portsentry.modes** file (vi /etc/portsentry/portsentry.modes) and add the following lines inside it. Below is what we recommend you.

```
# Place whitespace dilineated modes below.
# Blank lines and pound deliniated comments are ignored.

# tcp
# udp
# stcp
atcp
sudp
# audp
```

/etc/rc.d/init.d/portsentry: The PortSentry Initialization File

The /etc/rc.d/init.d/portsentry script file is responsible to automatically start and stop the PortSentry daemon on your Server.

Step 1
Create the **portsentry** script file (touch /etc/rc.d/init.d/portsentry) and add the following lines inside it:

```
#! /bin/sh
#
# portsentry    Start the portsentry Port Scan Detector
#
# Author:       Craig Rowland <crowland@psionic.com>
#
# chkconfig: 345 98 05
# description: PortSentry Port Scan Detector is part of the Abacus Project \
#              suite of tools. The Abacus Project is an initiative to release \
#              low-maintenance, generic, and reliable host based intrusion \
#              detection software to the Internet community.
# processname: portsentry
# configfile: /etc/portsentry/portsentry.conf
# pidfile: /var/run/portsentry.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Get config.
. /etc/sysconfig/network

# Check that networking is up.
if [ ${NETWORKING} = "no" ]
then
        exit 0
fi

[ -f /usr/sbin/portsentry ] || exit 0

# See how we were called.
case "$1" in
  start)
        echo -n "Starting Port Scan Detector: "
        if [ -s /etc/portsentry/portsentry.modes ] ; then
                modes=`cut -d "#" -f 1 /etc/portsentry/portsentry.modes`
        else
                modes="tcp udp"
        fi
        for i in $modes ; do
                portsentry -$i
                echo -n "$i "
        done
        echo
        touch /var/lock/subsys/portsentry
        ;;
  stop)
        echo -n "Stopping Port Scan Detector: "
        killproc portsentry
        echo
        rm -f /var/lock/subsys/portsentry
        ;;
  status)
        status portsentry
```

```
        ;;
  restart|reload)
        $0 stop
        $0 start
        ;;
  *)
        echo "Usage: portsentry {start|stop|status|restart|reload}"
        exit 1
esac

exit 0
```

Step 2
Once the `portsentry` script file has been created, it is important to make it executable and change its default permissions. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons.

- To make this script executable and to change its default permissions, use the command:
  ```
  [root@deep /]# chmod 700 /etc/rc.d/init.d/portsentry
  [root@deep /]# chown 0.0 /etc/rc.d/init.d/portsentry
  ```

- To create the symbolic `rc.d` links for `PortSentry`, use the following command:
  ```
  [root@deep /]# chkconfig --add portsentry
  [root@deep /]# chkconfig --level 345 portsentry on
  ```

- To start `PortSentry` software manually, use the following command:
  ```
  [root@deep /]# /etc/rc.d/init.d/portsentry start
  Starting Port Scan Detector:         [OK]
  ```

`/etc/logrotate.d/portsentry`: The `PortSentry` Log Rotation File

The `/etc/logrotate.d/portsentry` file is responsible to rotate log files related to `PortSentry` software automatically each week via `syslog`. If you are not familiar with `syslog`, look at the `syslog.conf (5)` manual page for a description of the `syslog` configuration file, or the `syslogd (8)` manual page for a description of the `syslogd` daemon.

- Create the **`portsentry`** file (`touch /etc/logrotate.d/portsentry`) and add the following lines inside it:

  ```
  /var/log/portsentry/portsentry.blocked {
      postrotate
          /usr/bin/killall -HUP portsentry
      endscript
  }

  /var/log/portsentry/portsentry.blocked.atcp {
      postrotate
          /usr/bin/killall -HUP portsentry
      endscript
  }

  /var/log/portsentry/portsentry.blocked.sudp {
      postrotate
          /usr/bin/killall -HUP portsentry
      endscript
  ```

```
}

/var/log/portsentry/portsentry.history {
    postrotate
        /usr/bin/killall -HUP portsentry
    endscript
}
```

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

List of installed `PortSentry` files in your system

> /etc/logrotate.d/portsentry
> /etc/portsentry
> /etc/portsentry/portsentry.conf
> /etc/portsentry/portsentry.ignore
> /etc/portsentry/portsentry.modes
> /usr/sbin/portsentry
> /var/log/portsentry
> /etc/rc.d/init.d/portsentry
> /var/log/portsentry/portsentry.blocked
> /var/log/portsentry/portsentry.history

16 Monitoring & System Integrity - `Tripwire`

In this Chapter

Linux Tripwire

Abstract

Tripwire ASR 1.3.1 is the "**A**cademic **S**ource **R**elease (ASR)" of Tripwire software. Personally, I prefer the 1.3.1 version of the software rather than the 2.2.1 version because it can be compiled and installed without any compatibility problems on most popular Unix based operating systems.

Tripwire data and network integrity software was originally developed in 1992 at Purdue University by world-renowned computer security expert, Dr. Eugene Spafford, and by master's degree student, Gene Kim. The resulting academic source release (ASR) was quickly embraced by computer security experts and actively used by thousands of corporate, government, and educational organizations worldwide.

As explained in the [Tripwire ASR goals]:
With the advent of increasingly sophisticated and subtle account break-ins on Unix systems, the need for tools to aid in the detection of unauthorized modification of files becomes clear.

Tripwire is a tool that aids system administrators and users in monitoring a designated set of files for any changes. Used with system files on a regular (e.g., daily) basis, Tripwire can notify system administrators of corrupted or tampered files, so damage control measures can be taken in a timely manner.

Tripwire is a file and directory integrity checker, a utility that compares a designated set of files and directories against information stored in a previously generated database. Any differences are flagged and logged, including added or deleted entries. When run against system files on a regular basis, any changes in critical system files will be spotted -- and appropriate damage control measures can be taken immediately. With Tripwire, system administrators can conclude with a high degree of certainty that a given set of files remain free of unauthorized modifications if Tripwire reports no changes.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Tripwire version number is 1.3.1-1

Packages

The following is based on information as listed by Tripwire as of 2001/03/25. Please regularly check at www.tripwire.com for the latest status.

Source code is available from:
Tripwire Homepage: http://www.tripwire.com/
You must be sure to download: Tripwire-1.3.1-1.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `Tripwire`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > Tripwire1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > Tripwire2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff Tripwire1 Tripwire2 > Tripwire-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `Tripwire`

Below are the required steps that you must make to configure, compile and optimize the `Tripwire` software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp Tripwire-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf Tripwire-version.tar.gz
```

Step 2
After that, move into the newly created `Tripwire` directory and modify some of its files as shown below to specify the installation paths, compilation and optimizations flags for your Linux system.

- To move into the newly created `Tripwire` directory use the following command:
```
[root@deep tmp]# cd tw_ASR_1.3.1_src/
```

Step 2.1
The first file we will work on it is named **utils.c** located under the source directory of Tripwire.

- Edit the **utils.c** file (vi +462 src/utils.c) and change the line:

```
else if (iscntrl(*pcin)) {
```

To read:

```
else if (!(*pcin & 0x80) && iscntrl(*pcin)) {
```

Step 2.2
The second file we must modify is the **config.parse.c** file.

- Edit the **config.parse.c** file (vi +356 src/config.parse.c) and change the line:

```
rewind(fpout);
return;
```

To read:

```
else {
rewind(fpin);
    }
return;
```

Step 2.3
The third file to modify is the **config.h** header file of Tripwire. Into this file, we will change the default location of different Tripwire directories files.

- Edit the **config.h** file (vi +106 include/config.h) and change all of the targeted lines in the order shown below:

```
#define CONFIG_PATH     "/usr/local/bin/tw"
#define DATABASE_PATH   "/var/tripwire"
```

To read:

```
#define CONFIG_PATH     "/etc"
#define DATABASE_PATH   "/var/spool/tripwire"
```

vi +165 include/config.h and change the line:

```
#define TEMPFILE_TEMPLATE "/tmp/twzXXXXXX"
```

To read:

```
#define TEMPFILE_TEMPLATE "/var/tmp/.twzXXXXXX"
```

Step 2.4
The next file we must modify is the **config.pre.y** file of this program.

- Edit the **config.pre.y** file (`vi +66 src/config.pre.y`) and change the line:

```
#ifdef TW_LINUX
```

To read:

```
#ifdef TW_LINUX_UNDEF
```

Step 2.5
The last file to modify is the **Makefile** of `Tripwire`. The changes we make to this file is to add our optimization flags to speed up our `Tripwire` software and to change the default location of different `Tripwire` binaries and directories files.

- Edit the **Makefile** file (`vi +13 Makefile`) and change all of the targeted lines in the order shown below:

```
DESTDIR = /usr/local/bin/tw
DATADIR = /var/tripwire
```

To read:

```
DESTDIR = /usr/sbin
DATADIR = /var/spool/tripwire
```

```
MANDIR  = /usr/man
```

To read:

```
MANDIR  = /usr/share/man
```

```
LEX     = lex
```

To read:

```
LEX     = flex
```

```
CFLAGS = -O
```

To read:

```
CFLAGS = -O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-
pointer
```

WARNING: Pay special attention to the compile `CFLAGS` line above. We optimize `Tripwire` for an i686 CPU architecture with the parameter "`-march=i686` and `-mcpu=i686`". Please don't forget to adjust this `CFLAGS` line to reflect your own system and architecture.

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install Tripwire in the server:

```
[root@deep tw_ASR_1.3.1_src]# make
[root@deep tw_ASR_1.3.1_src]# cd
[root@deep /root]# find /* > Tripwire1
[root@deep /root]# cd /var/tmp/tw_ASR_1.3.1_src/
[root@deep tw_ASR_1.3.1_src]# make install
[root@deep tw_ASR_1.3.1_src]# chmod 700 /var/spool/tripwire/
[root@deep tw_ASR_1.3.1_src]# chmod 500 /usr/sbin/tripwire
[root@deep tw_ASR_1.3.1_src]# chmod 500 /usr/sbin/siggen
[root@deep tw_ASR_1.3.1_src]# mv /usr/sbin/tw.config /etc/
[root@deep tw_ASR_1.3.1_src]# strip /usr/sbin/tripwire
[root@deep tw_ASR_1.3.1_src]# strip /usr/sbin/siggen
[root@deep tw_ASR_1.3.1_src]# cd
[root@deep /root]# find /* > Tripwire2
[root@deep /root]# diff Tripwire1 Tripwire2 > Tripwire-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

The chmod command will change the default permission mode of Tripwire directory to be 700 (drwx------) only readable, writable, and executable by the super-user "root". It will also make the binaries program /usr/sbin/tripwire and /usr/sbin/siggen only readable, and executable by the super-user "root" (-r-x------). The mv command as used above will move the file tw.config under /usr/sbin to /etc directory and finally the strip command will reduce the size of the tripwire and siggen binaries to get the optimal performance of those programs.

Step 4
Once configuration, compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Tripwire and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf tw_ASR_version/
[root@deep tmp]# rm -f Tripwire-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Tripwire. It will also remove the Tripwire compressed archive from the /var/tmp directory.

Configuring Tripwire

After building Tripwire, your next step is to verify or change, if necessary options in your Tripwire configuration files. Those files are:

- ✓ /etc/tw.config (The Tripwire Configuration File)
- ✓ /etc/cron.daily/Tripwire (The Tripwire Cron File)

/etc/tw.config: The Tripwire Configuration File

The tw.config file is the Tripwire configuration file where you decide and set which system files and directories that you want monitored. Note that extensive testing and experience are necessary when editing this file before you get working file reports. The following is a working example from where you can start you own customization. We must create, edit or change it to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

Step 1
Edit the **tw.config** file (vi /etc/tw.config) and add into this file all the files and directories that you want monitored. The format of the configuration file is described in its header and in the manual page tw.config (5). Below is what we recommend:

```
# Gerhard Mourani: gmourani@openna.com
# last updated: 04/01/2001

#  First, root's "home"
/root                   R
!/root/.bash_history
/                       R

# OS itself and critical boot resources
/boot                   R

# Critical directories and configuration files
/bin                    R
/chroot                 R
/etc                    R
/lib                    R
/sbin                   R

# Critical devices
/dev/kmem               R
/dev/mem                R
/dev/null               R
/dev/zero               R
/proc/devices           R
/proc/net               R
/proc/tty               R
/proc/sys               R
/proc/cpuinfo           R
/proc/mounts            R
/proc/dma               R
/proc/filesystems       R
/proc/ide               R
/proc/interrupts        R
/proc/ioports           R
/proc/scsi              R
/proc/kcore             R
/proc/self              R
/proc/kmsg              R
/proc/stat              R
/proc/fs                R
/proc/bus               R
/proc/loadavg           R
/proc/uptime            R
/proc/locks             R
/proc/version           R
/proc/meminfo           R
/proc/cmdline           R
/proc/misc              R
```

```
# Other popular filesystems
/usr                    R
/dev                    L-am

# Truncate home
=/home                  R

# var tree
=/var/spool             L
/var/db                 L
/var/lib                L
/var/local              L
!/var/lock
/var/log                L
/var/preserve           L
/var/spool/cron         L
/var/spool/mqueue       L
/var/spool/mail         L
/var/spool/tripwire     L

# Unusual directories
=/proc                  E
=/tmp
=/mnt/cdrom
```

Step 2
Now, for security reasons, change the mode of this file to be 0400.

- This procedure can be accomplished with the following command:
  `[root@deep /]# chmod 400 /etc/tw.config`

`/etc/cron.daily/tripwire`: The `Tripwire` Cron File

The `tripwire` file is a small script executed automatically by the `crond` program of your server each day to scan your hard disk for possible changed files or directories and mail the results to the system administrator. This script will automate the procedure of integrity checking for you. If you intend to automate this task, follow the simple steps below.

Step 1
Create the **`tripwire`** script file (`touch /etc/cron.daily/tripwire`) and add the lines:

```
#!/bin/sh
/usr/sbin/tripwire -loosedir -q | (cat <<EOF
This is an automated report of possible file integrity changes, generated by
the Tripwire integrity checker. To tell Tripwire that a file or entire
directory tree is valid, as root run:

/usr/sbin/tripwire -update [pathname|entry]

If you wish to enter an interactive integrity checking and verification
session, as root run:

/usr/sbin/tripwire -interactive

Changed files/directories include:
EOF
cat
) | /bin/mail -s "File integrity report" root
```

Step 2
Now, make this script executable and change its permission mode to be 0700.

- This procedure can be accomplished with the following command:
  `[root@deep /]# chmod 700 /etc/cron.daily/tripwire`

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Securing `Tripwire`

It is recommended that the database (`tw.db_[hostname]`) file of `Tripwire` be moved someplace (e.g. floppy) where it cannot be modified. This is important because data from `Tripwire` is only as trustworthy as its database.

It is also recommend that you make a hardcopy printout of the database contents right away. In the event that you become suspicious of the integrity of the database, you will be able to manually compare information against this hardcopy.

Further documentation

For more details, there are several manual pages you can read:

```
$ man siggen (8)          - Signature generation routine for Tripwire
$ man tripwire (8)        - A file integrity checker for UNIX systems
$ man tw.config (5)       - Configuration file for Tripwire
```

`Tripwire` Administrative Tools

The commands listed below are some of the most used of this software, but many more exist. Check the `Tripwire` manual pages for more details.

Running `Tripwire` in Interactive Checking Mode

In “Interactive Checking Mode” feature, `Tripwire` verifies files or directories that have been added, deleted, or changed from the original database and asks the user whether the database entry should be updated. This mode is the most convenient way of keeping your database up-to-date, but it requires that the user be "at the console". If you intend to use this mode, then follow the simple steps below.

Step 1
Tripwire must have a database to compare against, so we first create the file information database. This action will create a file called "tw.db_[hostname]" in the directory you specified to hold your databases (where [hostname] will be replaced with your machine hostname).

- To create the file information database for Tripwire, use the following command:

```
[root@deep /]# cd /var/spool/tripwire/
[root@deep tripwire]# /usr/sbin/tripwire –initialize
Tripwire(tm) ASR (Academic Source Release) 1.3.1
File Integrity Assessment Software
(c) 1992, Purdue Research Foundation, (c) 1997, 1999 Tripwire
Security Systems, Inc. All Rights Reserved. Use Restricted to
Authorized Licensees.
### Warning:   creating ./databases directory!
###
### Phase 1:   Reading configuration file
### Phase 2:   Generating file list
### Phase 3:   Creating file information database
```

We move to the directory we specified to hold our database, and then we create the file information database, which is used for all subsequent Integrity Checking. This command is used only one time to create the information database of all files and directories that must be checked by the program. Once your information database is created you don't have to retype this command again.

Step 2
Once the file information database of Tripwire has been created, we can now run Tripwire in "Interactive Checking Mode". This mode will prompt the user for whether or not each changed entry on the system should be updated to reflect the current state of the file.

- To run Tripwire in Interactive Checking Mode, use the following command:

```
[root@deep /]# cd /var/spool/tripwire/database/
[root@deep database]# cp tw.db_myserverhostname /var/spool/tripwire/
[root@deep database]# cd ..
[root@deep tripwire]# /usr/sbin/tripwire --interactive
Tripwire(tm) ASR (Academic Source Release) 1.3.1
File Integrity Assessment Software
(c) 1992, Purdue Research Foundation, (c) 1997, 1999 Tripwire
Security Systems, Inc. All Rights Reserved. Use Restricted to
Authorized Licensees.
### Phase 1:     Reading configuration file
### Phase 2:     Generating file list
### Phase 3:     Creating file information database
### Phase 4:     Searching for inconsistencies
###
###          Total files scanned:               15722
###          Files added:                       34
###          Files deleted:                     42
###          Files changed:                     321
###
###          Total file violations:             397
###
added:   -rwx------ root        22706 Dec 31 06:25:02 1999
/root/tmp/firewall
---> File: '/root/tmp/firewall'
---> Update entry?  [YN(y)nh?]
```

NOTE: In interactive mode, `Tripwire` first reports all added, deleted, and changed files, then allows the user to update the entry in the database.

Running `Tripwire` in Database Update Mode

Running `Tripwire` in “Database Update Mode” mixed with the `tripwire.verify` script file that mails the results to the system administrator will reduce the time of scanning the system. Instead of running `Tripwire` in “Interactive Checking Mode” and waiting for the long scan to finish, the script file `tripwire.verify` will scan the system and report via mail the result, then you run `Tripwire` in “Database Update Mode” and update only single files or directories that have changed (if needed).

As an example:
If a single file has changed, you can:

```
[root@deep /]# tripwire -update /etc/newly.installed.file
```

Or, if an entire set of files or directories has changed, you can run:

```
[root@deep /]# tripwire -update /usr/lib/Package_Dir
```

In either case, `Tripwire` regenerates the database entries for every specified file. A backup of the old database is created in the `./databases` directory.

Some possible uses of `Tripwire` software

`Tripwire` can be used to:

1. Check the integrity of your files system.
2. Get a list of new installed or removed files on your system.

List of installed `Tripwire` files on your system

> /etc/tw.config
> /usr/sbin/tripwire
> /usr/sbin/siggen
> /usr/share/man/man5/tw.config.5
> /usr/share/man/man8/siggen.8
> /usr/share/man/man8/tripwire.8
> /var/spool/tripwire
> /var/spool/tripwire/tw.db_TEST

17 Monitoring & System Integrity - `Xinetd`

In this Chapter

Compiling - Optimizing & Installing `Xinetd`
Configuring `Xinetd`
Securing `Xinetd`

Linux Xinetd - The Super Servers

Abstract

Xinetd is a secure, powerful and efficient replacement for the old Internet services daemons named inetd and tcp_wrappers (inetd does not provide effective resource management. It will happily use up all your memory if you are running a popular service. It is unreliable under high loads and will cuts off service for 10 minutes if it receives too many connections in 1 minute). This security tool can control denial-of-access attacks by providing access control mechanisms for all services based on the address of the remote client that want to connect to the server as well as the ability to make services available based on time of access, extensive logging, and the ability to bind services to specific interfaces.

But wait, Xinetd is NOT efficient or adequate for all services, and especially for services like FTP and SSH. It is far better to run these services as standalone daemons. Loading the FTP or SSH daemons, as standalone daemons will eliminate load time and will even reduce swapping since non-library code will be shared. Also FTP and SSH have very good access control mechanisms, therefore, don't think that if you run these services through Xinetd you will gain security.

A few security features of Xinetd are:

- ✓ Provide access control mechanisms
- ✓ Prevent denial of service attacks
- ✓ Extensive logging abilities
- ✓ Offload services to a remote host
- ✓ Make services available based on time
- ✓ Limits on the number of servers that can be started
- ✓ IPv6 support
- ✓ User interaction

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Xinetd version number is 2.1.8.9pre15

Packages

The following are based on information as listed by Xinetd as of 2001/05/20. Please regularly check at www.xinetd.org for the latest status.

Pristine source code is available from:
Xinetd Homepage: http://www.xinetd.org/
You must be sure to download: xinetd-2.1.8.9pre15.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `Xinetd`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > Xinetd1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > Xinetd2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff Xinetd1 Xinetd2 > Xinetd-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `Xinetd`

Below are the required steps that you must make to configure, compile and optimize the `Xinetd` software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp xinetd-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf xinetd-version.tar.gz
```

Step 2
After that, move into the newly created `Xinetd` directory then configure and optimize it.

- To move into the newly created `Xinetd` directory use the following command:
```
[root@deep tmp]# cd xinetd-2.1.8.9pre15/
```

- To configure and optimize `Xinetd` use the following compile lines:
```
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--with-loadavg \
--mandir=/usr/share/man
```

This tells `Xinetd` to set itself up for this particular hardware with:

- '`--with-loadavg`' allows to deactivate some services when the machine is overloaded.

WARNING: Pay special attention to the compile `CFLAGS` line above. We optimize `Xinetd` for an i686 CPU architecture with the parameter "`-march=i686` and `-mcpu=i686`". Please don't forget to adjust this `CFLAGS` line to reflect your own system and CPU architecture.

The "`-fomit frame pointer`" flag is an optimization dealing with the stack and cannot be used with `Xinetd`. This is the reason why we don't use it here.

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `Xinetd` in the server:

```
[root@deep xinetd-2.1.8.9pre15]# make
[root@deep xinetd-2.1.8.9pre15]# cd
[root@deep /root]# find /* > Xinetd1
[root@deep /root]# cd /var/tmp/xinetd-2.1.8.9pre15/
[root@deep xinetd-2.1.8.9pre15]# make install
[root@deep xinetd-2.1.8.9pre15]# rm -f /usr/sbin/itox
[root@deep xinetd-2.1.8.9pre15]# rm -f /usr/share/man/man8/itox.8
[root@deep xinetd-2.1.8.9pre15]# strip /usr/sbin/xinetd
[root@deep xinetd-2.1.8.9pre15]# cd
[root@deep /root]# find /* > Xinetd2
[root@deep /root]# diff Xinetd1 Xinetd2 > Xinetd-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Take special attention to the `rm` command, we use it to remove `itox` binary and `itox.8` manual page from the system because this utility is now replaced by `xconv.pl` perl script. The `strip` command will reduce the size of the `xinetd` binary program and will make it faster again.

Step 4
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `Xinetd` and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf xinetd-version/
[root@deep tmp]# rm -f xinetd-version.tar.gz
```

The `rm` command as used above will remove all the source files we have used to compile and install `Xinetd`. It will also remove the `Xinetd` compressed archive from the `/var/tmp` directory.

Step 5
One last thing to do is to remove `/etc/hosts.allow` and `/etc/hosts.deny` files (if this is not been done already) from your system. Yes, you can remove them safety, those files are related to `TCP WRAPPERS` program, which is not installed anymore the system because `Xinetd` does the same job better and can run well without it. The files `hosts.allow` and `hosts.deny` are installed by other Linux RPM packages during install. So we can remove them with the following commands.

- To delete `hosts.allow` and `hosts.deny` files from your system, use the commands:
```
[root@deep /]# rm -f /etc/hosts.allow
[root@deep /]# rm -f /etc/hosts.deny
```

Configuring `Xinetd`

After building `Xinetd`, your next step is to verify or change, if necessary options in your `Xinetd` configuration files. Those files are:

- ✓ `/etc/xinetd.conf` (The `Xinetd` Configuration File)
- ✓ `/etc/rc.d/init.d/xinetd` (The `Xinetd` Initialization File)

`/etc/xinetd.conf`: The `xinetd` Configuration File

The `xinetd.conf` file which determines the services provided by `xinetd`. It basically contains a list of IP services to listen to and tells `xinetd` daemon (also known as the super-servers) which ports to listen to. related by those listed in its configuration file, and what server to start for each port among other things. When it receives a connection on a port it checks to see if it has a service for it, and if services exist, then it attempts to start the appropriate server. The first thing to look at as soon as you put your Linux system on ANY network is what `Xinetd` services you need to offer and enable via the configuration file `/etc/xinetd.conf`.

Below are some of the default services handled by this secure and powerful program, that you can run through its configuration file. For easy interpretation, we have separated them by group related to their nature.

Group 1: BSD services
- ✓ `login`
- ✓ `shell`
- ✓ `exec`
- ✓ `comsat`
- ✓ `talk`
- ✓ `ntalk`

Group 2: Standard Internet services
- ✓ `telnet`
- ✓ `ftp`

Group 3: Other services
- ✓ `name`
- ✓ `uucp`
- ✓ `tftp`

Group 4: Information services
- ✓ `finger`
- ✓ `systat`
- ✓ `netstat`

Group 5: Internal services
- ✓ `echo`
- ✓ `chargen`
- ✓ `daytime`
- ✓ `time`
- ✓ `servers`
- ✓ `services`

Group 6: RPC services
- ✓ `rstatd`
- ✓ `rquotad`
- ✓ `rusersd`
- ✓ `sprayd`
- ✓ `walld`

Group 7: User Mail Agent services
- ✓ `imap`
- ✓ `imaps`
- ✓ `pop2`
- ✓ `pop3`
- ✓ `pop3s`

As you can imagine, for a secure server, most of the group services, which are available through Xinetd, are insecure by their nature and must be disabled if you don't use them. Of course Xinetd exists because those services are insecure. It tries to make more secure by having control of them, but since we don't use many of those risky services it is better to have a program that can monitor and control the ones we may need, such as IMAP or POP and exclude all of the rest. In this manner, we can be reassured that even the small amount of services that we could offer are monitored, controlled, logged, etc and stay in our control. It is important to note that, services, which you do not need to offer, should be uninstalled so that you have one less thing to worry about, and attackers have one less place to look for a hole.

Understanding /etc/xinetd.conf

OK, now it is time to talk and understand a bit more about the format of the /etc/xinetd.conf file. The services listed in xinetd.conf can be separated into two major sections which are called the "defaults section" and the "services sections". Below is an explanation and configuration of each one:

The defaults section of Xinetd configuration file:

The defaults section, as its name implies, states default settings for the services specified elsewhere in the file (attributes in this section will be used by every service Xinetd manages). The defaults section can contain a number of attributes as shown below (each attribute defined in this section keeps the provided value(s) for all the next described services). There can be only one defaults section in a xinetd.conf file. Here, are the most important attributes in the default section of your xinetd.conf file for maximum security; a complete listing and/or special requirements are available in the man page for xinetd (8) and xinetd.conf (5) and it is preferable to not talk about all of them to keep this tutorial as simple as possible.

If you need some special services that are not described here to run through Xinetd, refer to the appropriate manual page, in this manner you will have the opportunity to become familiar with the software and to add new or needed services when time will arrive. From now, we must create, check or change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to meet our needs.

- Create the **xinetd.conf** file (touch /etc/xinetd.conf) and set your needs for the default section of this file. Below is just an example:

```
defaults
{
        instances       = 60
        log_type        = SYSLOG authpriv
        log_on_success  = HOST PID
        log_on_failure  = HOST
        only_from       =
        per_source      = 5
        enabled         = pop3s imaps
}
```

This tells the default section of xinetd.conf file to set itself up for this particular configuration with:

instance = 60
The option instance specifies the maximum number of requests that can be simultaneously active for a service. It also says that for any service that doesn't specify it's own instances attribute, that service will be limited to 60 connections. The special value "UNLIMITED" can by used to specify an unlimited number of connections. This attribute will protect from Denial of Service (DoS) attacks.

log_type = SYSLOG authpriv
The option log_type specifies the log type formats you want to use (you may choose FILE or SYSLOG) to capture the output service generated by the program. For the FILE log type, this means the full path to the log file, and for the SYSLOG log type, the syslog facility of the system.

log_on_success = HOST PID
The option log_on_success specifies what is to be logged when a server is started. This attribute accepts five different values: PID (log of the pid xinetd uses to spawn the server), HOST (to logs the remote host's IP address), USERID (to logs the userid of the remote user as returned by remote identd daemon service if available), EXIT (logs the exit status of the server when it exits), and DURATION (logs the duration of the server session).

log_on_failure = HOST
The option log_on_failure specifies what is to be logged when either the server could not be started due to lack of resources, or access was denied via the rules in the configuration file. This attribute accepts four valid values: HOST (to logs the remote host's IP address), USERID (to logs the userid of the remote user as returned by remote identd daemon service if available), ATTEMPT (to acknowledge that a failed attempt was made), and RECORD (to grabs as much info as is possible about the remote end).

only_from =
This attribute only_from specifies which remote hosts are allowed to connect to the server and use this service. By default denying access to every one, is the first step of a reliable security policy. Not giving a value to this attribute makes every connection fail. This is the same principle as for the IPTABLES Firewall rules. In our example we deny access to all connection then, allows access by means of this same attribute for specific service under the services sections of Xinetd. Other combination for the value of "only_from" attribute exists; please consult the manual page xinetd.conf (5) for more information.

per_source = 5
The option per_source specifies the maximum number of connections a specific remote IP address can have to a specific local service. It can either be an integer, or the special value "UNLIMITED" for an unlimited number of connections. This attribute will protect from Denial of Service (DoS) attacks.

enabled = pop3s imaps
The option enabled takes a list of service names to enable with the super-server. The most interesting part is that it will enable only the services listed as arguments to this attribute and the rest will be disabled. Each service names you add to this attribute line can be listed and setup in the services sections of the Xinetd configuration file. If you forget to add the service names you want to run through xinetd to the attribute “enabled”, then this service name will not work even if you add its required configuration lines in the services sections, therefore don’t forget to check for the existence of this attribute line (enabled) and set all services you want to be available with Xinetd for filtering (in our example we only enable at this time service pop3s and imaps).

The services sections of Xinetd configuration file:

Now the default section attributes are complete, we'll move on to the service sections. Contrary to the default section, the services sections defines individual services to be started by the Xinetd daemon and how they'll be started. This is important to note, if the service names we want to offer and enable via the xinetd configuration file are not specified in the "enabled" attribute line of the previous default section, (see the default section of Xinetd configuration file for more information) then they are considered to be disabled by default and we don't need to worry about them.

The services sections have a number of attributes that can be specified, most are required and are the same for all available services, others are optional or are a security feature and depends on what services you want to run and include in your xinetd.conf file. Below we will show you different configuration options for pop3s, time, chargen, echo, daytime, and imaps services. In this way you will have a good idea of specific parameters available for different services, which can run through Xinetd and how to play with them.

If you remember, we said at the beginning of this tutorial that we don't need to install TCP WRAPPER anymore with Xinetd on Linux. TCP WRAPPER is a program that controls who can or cannot log in to the server and from where. Contrary to its predecessor (inetd), Xinetd has two powerful features already built on it, which allow you to have the same and even better control as the TCP WRAPPER program could offer you.

The first feature is named "only_from", this attribute with its list of IP addresses determines the remote host to which the particular service is available.

The second attribute is named "no_access" and determines the remote hosts to which the particular service is unavailable.

The use of these two attributes can determine the location access control enforced by Xinetd. One very interesting part of these two attributes is the possibility to build a very restrictive but flexible access control program.

For each service, we must check or change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy your needs.

For pop3s service:

- Edit the **xinetd.conf** file (vi /etc/xinetd.conf) and set your needs under the services sections of this file. The first thing you'll probably notice here are that contrary to the old inetd software, the services sections are now split into individual service configurations. Below is just an example for an pop3s service:

```
service pop3s
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/ipop3d
        only_from       = 0.0.0.0/0 #allows every client
        no_access       = 207.35.78.10
        instances       = 30
        log_on_success  += USERID
        log_on_failure  += USERID
        nice            = -2
}
```

This tells the `services sections` of `xinetd.conf` file to set itself up for this particular configuration with:

`service pop3s`
The option `service` specifies a unique name for the service you wish to configure. This name is what the program uses to look up the service information in the `/etc/services` file. Be aware that you cannot use any name you want to set this attribute, protocols exist for this purpose and if you don't know exactly the correct name to use to enable your needed services, then edit the `/etc/services` file and look inside it for the appropriate name for your requirements.

`socket_type = stream`
The option `socket_type` specifies the type of socket to be used for the specific service. The available values are: "`stream`", "`dgram`", "`raw`", "`rdm`", or "`seqpacket`", depending on whether the socket is a stream, datagram, raw, reliably delivered message, or sequenced packet socket. For `pop3s` service we must choose and set this attribute to the value "`stream`".

`wait = no`
The option `wait` specifies if a datagram server connected to its peer allows the `xinetd` daemon to receive further messages on the socket or not. If the answer is yes (`xinetd` can receive further messages on the socket with this program) then this program should use the "`nowait`" entry and we will set the value of `wait` to `no` to indicate the "`nowait`" entry. This is the default for most services under `Xinetd`.

`user = root`
The option `user` contains the user name of the user the server should run as. Usually this value is set and uses the super-user account named "root" but it is preferable to verify the software you want to run through `Xinetd` if other values are possible for better security.

`server = /usr/sbin/ipop3d`
The option `server` contains the pathname of the program, which is to be executed by `xinetd` when a request is found on its socket.

`only_from = 0.0.0.0/0`
This attribute `only_from` specifies which remote hosts are allowed to connect to the server and use this service. By default we have denied access to everyone in the default section of `Xinetd`, therefore we must allow access for the specific service in question in this section of the configuration file. For a public mail server that runs `IMAP` or `POP` server it is important to set the value of this line to `0.0.0.0/0` in your configuration since connections can come from different places.

`no_access = 207.35.78.10`
The attribute `no_access` specifies which remote hosts are not allowed to connect to the server and use this service. In our example, we don't allow the machine with IP address of 207.35.78.10 to connect with `pop3s`. As you can see, the combination of both attributes (`only_from` and `no_access`) allows us to tie and have a full control of what can pass through our network.

`instance = 30`
As noted in the previous `defaults section`, the option `instance` specifies the maximum number of requests any service may handle at one once. Setting this attribute in the service definition should override whatever is in the `defaults section` (`instance = 60`).

`log_on_success += USERID`
As noted in the previous `defaults section`, the option `log_on_success` specifies what is to be logged when a server is started. For a `pop3s` connection we choose to log the userid of the remote user as returned by remote `identd` daemon service if available (`USERID`). Take a special note to the assignment operator in this case `'+='` which means to add the value to the set.

`log_on_failure += USERID`
As noted in the previous `defaults section`, the option `log_on_failure` specifies what is to be logged when either the server could not be started due to lack of resources, or access was denied via the rules in the configuration file. For an `pop3s` connection we choose to log the userid of the remote user as returned by remote `identd` daemon service if available.

`nice = -2`
The option `nice` specifies the services process priority of Unix by modifying the default scheduling priority of the process. The default priority for a normal program, like `pop3,` is 10 and is related to `nice man (1)` the range goes from -20 (highest priority) to 19 (lowest). By increasing the priority of the `pop3s` process the connection time will be faster. This hack can by applied to any other processes running on Unix, see the manual page about the command `nice (1)` for more information in this feature.

For `time` service:

- Edit the **`xinetd.conf`** file (`vi /etc/xinetd.conf`) and set your needs under the `services sections` of this file. Below is just an example for a `time server` service which is used by the `rdate` program:

```
# description: An RFC 868 time server. This is the tcp \
# version, which is used by rdate.

service time
{
        socket_type     = stream
        wait            = no
        user            = root
        type            = INTERNAL
        id              = time-stream
        protocol        = tcp
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
}

# description: An RFC 868 time server. This is the udp \
# version.

service time
{
        socket_type     = dgram
        wait            = yes
        user            = root
        type            = INTERNAL
        id              = time-dgram
        protocol        = udp
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
        port            = 37
}
```

This tells the `services sections` of `xinetd.conf` file to set itself up for this particular configuration with:

`socket_type = stream` and `socket_type = dgram`
As described previously, the option `socket_type` specifies the type of socket to be used for the specific service. The available values are: "`stream`", "`dgram`", "`raw`", "`rdm`", or "`seqpacket`", depending on whether the socket is a stream, datagram, raw, reliably delivered message, or sequenced packet socket. For `time` service we must choose "`stream`" for TCP connection and "`dgram`" for UDP connection.

`wait = no` and `wait = yes`
As described previously, the option `wait` specifies if a datagram server connected to its peer allow `xinetd` daemon to receive further messages on the socket or not. If the answer is yes (`xinetd` can receive further message on the socket with this program) then this program should use the "`nowait`" entry and we will set the value of `wait` to `no` to indicate the "`nowait`" entry. It important to note that UDP protocol in its nature do not allow peer daemon to receive further message and it is for the reason that we set the `wait` attribute for UDP version of the `time server` to `yes` (`xinetd` cannot receive further message on the socket with this program).

`type = INTERNAL`
Well, here we see a new attribute; the option `type` specifies the type of service. The available values are: "`RPC`", "`INTERNAL`", and "`UNLISTED`", depending on whether the specific program is an RPC service (`type = RPC`), or a service provided by `Xinetd` (`type = INTERNAL`) or if it is a service not listed in a standard system file like `/etc/rpc` for RPC services, or `/etc/services` for non-RPC services (`type = UNLISTED`). In our case `time server` is provided by `xinetd`.

`id = time-stream` and `id = time-dgram`
Ok, here is another new attribute; By default with `Xinetd` the attribute `id` is the same as the `service` name, but some time (as in our example `time server`) there exist same services that can use different protocols (TCP or UDP) and need to be described with different entries in the configuration file for `Xinetd` be able to distinguish them. With this attribute (`id`), we can uniquely identify a same service, which use different protocol of communication like TCP and UDP.

`protocol = tcp` and `protocol = udp`
We continue our discovery with the new attribute named "`protocol`", this option determines the type of protocol that is employed by the specific service. In our example `time server` use TCP and UDP protocol and we specify those information with the "`protocol`" attribute of `Xinetd`.

`only_from = 207.35.78.0/24 192.168.1.0/24`
The attribute `only_from` specifies which remote hosts are allowed to connect to the server and use this service. By default we have denied access to everyone in the default section of `Xinetd`, therefore we must allow access for the specific service in question in this section of the configuration file of `Xinetd`. In our example we allow all machines under the 207.35.78.0 and 192.168.1.0 IP addresses class range to connect with `time server`.

`no_access = 207.35.78.10`
The attribute `no_access` specifies which remote hosts are not allowed to connect to the server and use this service. In our example we don't allow the machine with IP address of 207.35.78.10 under the 207.35.78.0 IP addresses class range to connect with `time server`. As you can see, the combination of both attributes (`only_from` and `no_access`) allows us to tie and have a full control of what can pass through our network.

`port = 37`
Sometimes, and especially with the UDP protocol, it is preferable to specify to the program on which port we want the connection to be established. This option "`port`" makes it possible by determining the service port.

For `chargen` service:

- Edit the **`xinetd.conf`** file (`vi /etc/xinetd.conf`) and set your needs under the `services sections` of this file. Below is just an example for a `chargen` service:

```
# description: A chargen server. This is the tcp \
# version.

service chargen
{
        socket_type     = stream
        wait            = no
        user            = root
        type            = INTERNAL
        id              = chargen-stream
        protocol        = tcp
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
}

# description: A chargen server. This is the udp \
# version.

service chargen-udp
{
        socket_type     = dgram
        wait            = yes
        user            = root
        type            = INTERNAL
        id              = chargen-dgram
        protocol        = udp
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
        port            = 19
}
```

Here, you are supposed to know and understand every attribute as shown above. If you have problems, then refer to the previous `time server` configuration parameters for more information.

For `echo` service:

- Edit the **`xinetd.conf`** file (`vi /etc/xinetd.conf`) and set your needs under the `services sections` of this file. Below is just an example for an `echo` service:

```
# description: An echo server. This is the tcp \
# version.

service echo
{
        socket_type     = stream
        wait            = no
        user            = root
        type            = INTERNAL
        id              = echo-stream
        protocol        = tcp
```

```
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
}

# description: An echo server. This is the udp \
# version.

service echo-udp
{
        socket_type     = dgram
        wait            = yes
        user            = root
        type            = INTERNAL
        id              = echo-dgram
        protocol        = udp
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
        port            = 7
}
```

For `daytime` service:

- Edit the **`xinetd.conf`** file (`vi /etc/xinetd.conf`) and set your needs under the `services sections` of this file. Below is just an example for a `daytime` service:

```
# description: A daytime server. This is the tcp \
# version.

service daytime
{
        socket_type     = stream
        wait            = no
        user            = root
        type            = INTERNAL
        id              = daytime-stream
        protocol        = tcp
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
}

# description: A daytime server. This is the udp \
# version.

service daytime-udp
{
        socket_type     = dgram
        wait            = yes
        user            = root
        type            = INTERNAL
        id              = daytime-dgram
        protocol        = udp
        only_from       = 207.35.78.0/24 192.168.1.0/24
        no_access       = 207.35.78.10
        port            = 13
}
```

For `imaps` service:

At this stage of your reading, you know the most important attributes and values for `Xinetd` but be aware that many others exist, like the "`redirect`" attribute, which allows a TCP service to be redirected to another host in your network. This option is useful when your internal machines are not visible to the outside world and you want to connect to it outside the network. The "`bind`" attribute is another one, which allows a service to be bound to a specific interface of your choice on the server for maximum security.

- Edit the **`xinetd.conf`** file (`vi /etc/xinetd.conf`) and set your needs under the `services sections` of this file. Below is just an example for an `imaps` service:

```
service imaps
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/imapd
        only_from       = 0.0.0.0/0 #allows every client
        no_access       = 207.35.78.10
        instances       = 30
        log_on_success  += DURATION USERID
        log_on_failure  += USERID
        nice            = -2
        redirect        = 192.168.1.14 993
        bind            = 207.35.78.3
}
```

This tells the `services sections` of `xinetd.conf` file to set itself up for this particular configuration with:

`redirect = 192.168.1.14 993`
The attribute `redirect` allows a TCP service received on the specified port (in our example the port 993) to be redirected to another host (192.168.1.14) by forwarding all data between the two hosts.

`bind = 207.35.78.3`
The attribute `bind` allows a service of your choice to be bound to a specific interface on the server. In our case `imaps` service is bound to the interface 207.35.78.3. Therefore, if someone from the allowed hosts tries to bind to another interface on the server, then `Xinetd` will refuse the connection. This is a security feature.

Sample `/etc/xinetd.conf`: The `Xinetd` Configuration File

All of the interesting options we've shown you previously can easily be applied to the majority of services you want to run. Now, it is up to you and only you to decide how to mix and apply these attributes features to fit you personal configuration and needs. Below we show you a sample `xinetd.conf` file that you can use to begin with a secure server.

- Edit the **`xinetd.conf`** file (`vi /etc/xinetd.conf`) and set your needs. Below is what we recommend you to enable at this time:

```
defaults
{
        instances       = 60
        log_type        = SYSLOG authpriv
        log_on_success  = HOST PID
        log_on_failure  = HOST
```

```
                only_from       =
                per_source      = 5
                enabled         = pop3s imaps
        }
```

NOTE: More service examples exist under the subdirectory named `xinetd` of the `Xinetd` source archive. Check for file with name like `sample.conf` into this subdirectory (`xinetd`) if you need services, which are not explained in this tutorial.

`/etc/rc.d/init.d/xinetd`: The `Xinetd` Initialization File

The `/etc/rc.d/init.d/xinetd` script file is responsible to automatically start and stop the `Xinetd` daemon on your server.

Step 1
Create the **`xinetd`** script file (`touch /etc/rc.d/init.d/xinetd`) and add the following lines inside it:

```
#! /bin/sh
#
# xinetd        This starts and stops xinetd.
#
# chkconfig: 345 50 50
# description: xinetd is a powerful replacement for inetd. \
#              xinetd has access control machanisms, extensive \
#              logging capabilities, the ability to make services \
#              available based on time, and can place \
#              limits on the number of servers that can be started, \
#              among other things.
#
# processname: /usr/sbin/xinetd
# config: /etc/sysconfig/network
# config: /etc/xinetd.conf
# pidfile: /var/run/xinetd.pid

PATH=/sbin:/bin:/usr/bin:/usr/sbin

# Source function library.
. /etc/init.d/functions

# Get config.
test -f /etc/sysconfig/network && . /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "yes" ] || exit 0

[ -f /usr/sbin/xinetd ] || exit 1
[ -f /etc/xinetd.conf ] || exit 1

RETVAL=0

start(){
    echo -n "Starting xinetd: "
    daemon xinetd -reuse -pidfile /var/run/xinetd.pid
    RETVAL=$?
    echo
    touch /var/lock/subsys/xinetd
    return $RETVAL
```

```
}

stop(){
    echo -n "Stopping xinetd: "
    killproc xinetd
    RETVAL=$?
    echo
    rm -f /var/lock/subsys/xinetd
    return $RETVAL

}

reload(){
    echo -n "Reloading configuration: "
    killproc xinetd -USR2
    RETVAL=$?
    echo
    return $RETVAL
}

restart(){
    stop
    start
}

condrestart(){
    [ -e /var/lock/subsys/xinetd ] && restart
    return 0
}


# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status xinetd
        ;;
    restart)
        restart
        ;;
    reload)
        reload
        ;;
    condrestart)
        condrestart
        ;;
    *)
        echo "Usage: xinetd {start|stop|status|restart|condrestart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

Step 2
Once the xinetd script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and the creation of symbolic links will let the processes that control the initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/xinetd
[root@deep /]# chown 0.0 /etc/rc.d/init.d/xinetd
```

- To create the symbolic rc.d links for Xinetd, use the following command:
```
[root@deep /]# chkconfig --add xinetd
[root@deep /]# chkconfig --level 345 xinetd on
```

- To start Xinetd software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/xinetd start
Starting xinetd:          [OK]
```

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, floppy-2.0.tgz for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example /var/tmp, assuming you have done this your directory structure will be /var/tmp/floppy-2.0. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Securing Xinetd

Xinetd is a small and efficiently security tool that run on your system with just some little files installed. The only component of Xinetd that can be secured is its configuration file, below we show you some steps to secure /etc/xinetd.conf for optimal security.

Step 1
Make your /etc/xinetd.conf file “Read” only by the super-user “root” by changing its default permission. This is important because no one needs to touch this file.

- To make your xinetd.conf file “read” only by “root”, use the command:
```
[root@deep /]# chmod 400 /etc/xinetd.conf
```

Step 2
One more security measure you can take to secure **xinetd.conf** is to set it immutable, using the **chattr** command.

- To set the file immutable simply, execute the following command:
```
[root@deep /]# chattr +i /etc/xinetd.conf
```

This will prevent any changes (accidental or otherwise) to the `xinetd.conf` file. A file with the immutable attribute set "`i`" cannot be modified, deleted or renamed, no link can be created to this file and no data can be written to it. The only person that can set or clear this attribute is the super-user root.

If you wish later to modify the `xinetd.conf` file you will need to unset the immutable flag:

- To unset the immutable flag, simply execute the following command:
  `[root@deep /]#` **`chattr -i /etc/xinetd.conf`**

Further documentation

For more details, there are some man pages you can read:

```
$ man xinetd.conf (5)        - configuration settings for xinetd
$ man xinetd.log (8)         - xinetd service log format
$ man xinetd (8)             - the extended Internet services daemon
```

List of installed `Xinetd` files on your system

> /usr/sbin/xinetd
> /usr/sbin/xconv.pl
> /etc/xinetd.conf
> /etc/rc.d/init.d/xinetd
> /usr/share/man/man5/xinetd.conf.5
> /usr/share/man/man8/xinetd.log.8
> /usr/share/man/man8/xinetd.8

Part VI Management & Limitation Related Reference

In this Part

Management & Limitation - `Quota`

Here we will talk about a tool, which can be used to control users directories sizes. This part of the book is optional and will be interesting only for companies who provide Mail, Web, or FTP services to their customers and want to control amount of MB allowed for each users on the system for the specific service.

`Quota` falls into a security tool since it allows you to limit disk space that users may consume on the system, without a program like `quota`, users may fill as much disk space as they want and as you can imagine this will bring a big problems for you.

18 Management & Limitation - Quota

In this Chapter

Building a kernel with Quota support enable
Modifying the /etc/fstab file
Creating the quota.user and quota.group files
Assigning Quota for Users and Groups
Quota Administrative Tools

Set Quota on your Linux system

Abstract
Quota is a system administration tool for monitoring and limiting users' and/or groups' disk usage, per file system. Two features of disk storage with the Quota tool are available to set limits: the first is the number of inodes (number of files) a user or a group of users may possess and the second is the number of disk blocks (amount of space in kilobytes) that may be allocated to a user or a group of users. With Quota, users are forced by the system administrator to not consume an unlimited amount disk space on a system. This program is handled on per user and per file system basis and must be set separately for each file system.

It is useful for Mail, Web, and FTP Servers where limitations must be applied on the users, but can be used for any other purposes. It is your to decide where and how to use it.

Build a kernel with Quota support enable
The first thing you need to do is ensure that your kernel has been built with Quota support enabled. In the 2.4 kernel version you need ensure that you have answered **Y** to the following questions:

```
*Filesystems
*
Quota support (CONFIG_QUOTA) [N/y/?] y
```

Prerequisites
The Quota tool must be already installed on your system. If this is not the case, you must install it from your Linux CD-ROM or source archive files.

- To verify if Quota package is installed on your system, use the command:
```
[root@deep /]# rpm -q quota
package quota is not installed
```

- To mount your CD-ROM drive before installing the require package, use the command:
```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
mount: block device /dev/cdrom is write-protected, mounting read-only
```

- To install the quota package on your Linux system, use the following command:
```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh quota-version.i386.rpm
quota    ##################################################
```

- To unmount your CD-ROM drive, use the following command:
```
[root@deep RPMS]# cd /; umount /mnt/cdrom/
```

Modifying the /etc/fstab file
The /etc/fstab file contains information about various file systems installed on your Linux server. Quota must be enabled in the fstab file before you can use it. Since Quota must be set for each file system separately, and because in the fstab file, each file system is described on a separate line, Quota must be set on each of the separate lines in the fstab for which you want to enable Quota support.

Step 1
With the program Quota, depending on your needs, etc, you can enable Quota for users, groups or both (users and groups). For all examples below, we'll use the /home/ directory and shows you the three possibilities.

Possibility 1:

- To enable user Quota support on a specific file system, edit your fstab file (vi /etc/fstab) and add the "**usrquota**" option to the fourth field after the word "defaults" or any other options you may have set for this specific file system.

As an example change:

```
LABEL=/home /home   ext2    defaults     1 2 (as an example: the word "defaults")
LABEL=/home /home   ext2    nosuid,nodev     1 2 (as an example: any other options you have set)
```

To read:

```
LABEL=/home /home   ext2    defaults,usrquota    1 2
LABEL=/home /home   ext2    nosuid,nodev,usrquota    1 2
```

Possibility 2:

- To enable group Quota support on a file system, edit your fstab file (vi /etc/fstab) and add "**grpquota**" to the fourth field after the word "defaults" or any other options you may have set for this specific file system.

As an example change:

```
LABEL=/home /home   ext2    defaults     1 2 (as an example: the word "defaults")
LABEL=/home /home   ext2    nosuid,nodev     1 2 (as an example: any other options you have set)
```

To read:

```
LABEL=/home /home   ext2    defaults,grpquota    1 2
LABEL=/home /home   ext2    nosuid,nodev,grpquota    1 2
```

Possibility 3:

- To enable both users Quota and group Quota support on a file system, edit your fstab file (vi /etc/fstab) and add "**usrquota,grpquota**" to the fourth field after the word "defaults" or any other options you may have set for this specific file system.

Change:

```
LABEL=/home /home   ext2    defaults     1 2 (as an example: the word "defaults")
LABEL=/home /home   ext2    nosuid,nodev     1 2 (as an example: any other options you have set)
```

To read:

```
LABEL=/home /home   ext2    defaults,usrquota,grpquota   1 2
LABEL=/home /home   ext2    nosuid,nodev,usrquota,grpquota   1 2
```

Step 2
Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the system about the modification.

- This can be accomplished with the following commands:
```
[root@deep /]# mount -oremount /home/
```

Each file system that has been modified must be remounted with the command shown above. In our example we have modified the /home/ file.

Creating the quota.user and quota.group files

After the modification of your /etc/fstab file, in order for Quotas to be established on a file system, the root directory of the file system on which you want to enable Quota feature (i.e. /home/ in our example) must contain a file, owned by root, called "**quota.user**" if you want to use and set user Quota, and/or "**quota.group**" if you want to use and set group Quota, or both if you want users and group Quota.

Step 1
We must create, in the directory in which we want to have Quota feature enabled, the required files. In our example, we will create under the /home/ directory the file for user and group restrictions as shown below.

- To create the quota.user and/or quota.group files, as "root" go to the root of the partition you wish to enable Quota (i.e. /home/) and create quota.user and/or quota.group by doing:

```
[root@deep /]# touch /home/quota.user
[root@deep /]# touch /home/quota.group
[root@deep /]# chmod 600 /home/quota.user
[root@deep /]# chmod 600 /home/quota.group
```

The **touch** command will create new empty files under the /home/ directory named quota.user and quota.group. The **chmod** command will set the mode of these files to be read-write only by the super-user "root".

> **WARNING:** Both Quota record files, quota.user and quota.group, should be owned by root, with read-write permission for "root" only (0600/-rw-------).

Assigning Quota for Users and Groups

After the required files have been created, you can assign Quotas to users or groups of users on your system. This operation is performed with the edquota tool.

The edquota tool

The edquota program is a Quota editor that creates a temporary file of the current disk Quotas used by the super-user "root" to set Quotas for users or group of users in the system. The example below shows you how to setup Quotas for users or groups on your system.

Assigning `quota` for a particular user

Consider, for example, that you have a user with the login id "`gmourani`" on your system. The following command takes you into the editor (`vi`) to edit and set `Quotas` for user "`gmourani`" on each partition that has `Quotas` enabled:

Step 1

- To edit and modify `Quota` for user "`gmourani`", use the following command:

```
[root@deep /]# edquota -u gmourani
Quotas for user gmourani:
/dev/sda8: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
```

After the execution of the above command, you will see the following lines related to the example user "`gmourani`" appear on the screen. The "**`blocks in use:`**" display the total number of blocks (in kilobytes) the user has presently consumed on a partition. The "**`inodes in use:`**" value displays the total number of files the user has presently on a partition. These parameters ("`blocks in use`, and `inodes in use`") are controlled and set automatically by the system and you don't need to touch them.

Step 2

- To assign 5MB of quota for user "`gmourani`", change the following parameters:

```
Quotas for user gmourani:
/dev/sda6: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
```

To read:

```
Quotas for user gmourani:
/dev/sda6: blocks in use: 0, limits (soft = 5000, hard = 6000)
        inodes in use: 0, limits (soft = 0, hard = 0)
```

The **`soft limit`** (`soft = 5000`) specifies the maximum amount of disk usage a `Quota` user is allowed to have (in our example this amount is fixed to 5MB). The **`hard limit`** (`hard = 6000`) specifies the absolute limit on the disk usage a `Quota` user can't go beyond it. Take a note that the "`hard limit`" value works only when the "`grace period`" parameter is set.

The `grace period` parameter

The "`grace period`" parameter allows you to set a time limit before the `soft limit` value is enforced on a file system with `Quota` enabled (see the `soft limit` above for more information). For example, this parameter can be used to warn your users about a new policy that will set a `Quota` of 5MB of disk space in their home directory in 7 days. You can set the 0 days default part of this parameter to any length of time that you feel reasonable. The change of this setting requires two steps as follows (in my example I assume 7 days).

Step 1

- Edit the default `grace period` parameter, by using the following command:

```
[root@deep /]# edquota -t
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/sda8: block grace period: 0 days, file grace period: 0 days
```

Step 2

- To modify the grace period to 7 days. Change or set the following default parameters:

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/sda8: block grace period: 0 days, file grace period: 0 days
```

To read:

```
Time units may be: days, hours, minutes, or seconds
Grace period before enforcing soft limits for users:
/dev/sda8: block grace period: 7 days, file grace period: 7 days
```

NOTE: The command "`edquota -t`" edits the `soft time` limits for each file system with `Quotas` enabled.

Assigning `quota` for a particular group

Consider, for example, you have a group with the group id "`users`" on your system. The following command takes you into the `vi` editor to edit `Quotas` for the group "`users`" on each partition that has `Quotas` enabled:

- To edit and modify `Quota` for group "`users`", use the following command:

```
[root@deep /]# edquota -g users
Quotas for group users:
/dev/sda8: blocks in use: 0, limits (soft = 0, hard = 0)
        inodes in use: 0, limits (soft = 0, hard = 0)
```

The procedure is the same as for assigning `Quotas` for a particular user; as described previously, you must modify the parameter of "`soft =` and `hard =`" then save your change.

Assigning `quota` for groups of users with the same value

The `edquota` tool has a special option `(-p)` that assign `Quotas` for groups of users with the same value assigned to an initial user. Assuming that you want to assign users starting at UID 500 on the system the same value as the user "`gmourani`", we would first edit and set gmourani's `Quota` information, then execute:

- To assign `Quota` for group of users with the same value, use the following command:

```
[root@deep /]# edquota -p gmourani `awk -F: '$3 > 499 {print $1}'
/etc/passwd`
```

The `edquota` program will duplicate the `Quota` that we have set for the user "`gmourani`" to all users in the `/etc/passwd` file that begin after UID 499.

NOTE : You can use the `quota` utility to set a maximun size to a mail box for your mail users. For example: set `quota` to users at 10M in your `/var` partition and put the min and max inodes parameter of quota to 1. Then a user will be able to keep in his `/var/spool/$LOGNAME` only 10M.

Further documentation

For more details, there are several man pages you can read:

```
$ man edquota (8)            - edit user quotas
$ man quota (1)              - display disk usage and limits
$ man quotacheck (8)         - scan a file system for disk usages
$ man quotactl (2)           - manipulate disk quotas
$ man quotaon, quotaoff (8)  - turn file system quotas on and off
$ man repquota (8)           - summarize quotas for a file system
$ man rquota (3)             - implement quotas on remote machines
```

`Quota` Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual page for more information.

`Quota`

`Quota` displays users' disk usage and limits on a file system.

- To display user disk usage and limits, use the following command:

```
[root@deep /]# quota -u gmourani
Disk quotas for user gmourani (uid 500):
Filesystem  blocks   quota   limit   grace   files   quota   limit  grace
/dev/sda8    4692    5000    6000               9       0       0
```

- To display group `Quotas` for the group of which the user is a member, use the following command:

```
[root@deep /]# quota -g gmourani
Disk quotas for group gmourani (gid 500):
Filesystem  blocks   quota   limit   grace   files   quota   limit  grace
/dev/sda8    4660    5000    6000               1       0       0
```

NOTE: If the `group quota` is not set for the user specified, you will receive the following message: `Disk quotas for group gmourani (gid 501): none`

`Repquota`

The `Repquota` utility produces summarized `quota` information of the disk usage and `quotas` for the specified file systems. Also, it prints the current number of files and amount of space used (in kilobytes) for each user.

- Here is a sample output repquota gives (you results may vary):

```
[root@deep /]# repquota -a

                        Block limits               File limits
User            used    soft    hard  grace    used  soft  hard  grace
gmourani  --    4660    5000    6000              1     0     0

                        Block limits               File limits
User            used    soft    hard  grace    used  soft  hard  grace
root      --    4980       0       0              6     0     0
gmourani  --    4692    5000    6000              9     0     0
```

Part VII Domain Name System Related Reference

In this Part

Domain Name System - ISC BIND/DNS

Every time you send an electronic mail, surf the net, connect to another server, or talk with someone for example, you rely on the Domain Name System. It is rare that you don't have to pass through DNS in a networking environment. The Domain Name System is essential even if you don't run a Domain Name Server since it is the program (the directory to the Internet) that handles mapping between host names. Without it you cannot retrieve information remotely from everywhere on the network.

ISC BIND & DNS is very important and must be installed in every kind of server since many of services described in this book rely on it to work properly. Without DNS servers no one on the Internet will be able to find your servers.

19 Domain Name System - ISC BIND/DNS

In this Chapter

Linux ISC BIND & DNS Server

Abstract

Once we have installed all the necessary security software in our Linux server, it's time to improve and tune the networking part of our server. Domain Name System (DNS) is one of the **MOST** important network services for IP network communication, and for this reason, all Linux **client** machines should be configured to perform caching functions as a minimum. Setting up a caching server for client local machines will reduce the load on the site's primary server. A caching only name server will find the answer to name queries and remember the answer the next time we need it. This will shorten the waiting time the next time significantly.

A Name Server (NS) is a program that stores information about named resources and responds to queries from programs called *resolvers*, which act as client processes. The basic function of an NS is to provide information about network objects by answering queries. Linux is a perfect platform to run and deploy the BIND DNS server, a number of Linux DNS servers in the Internet are listed as authoritative DNS servers for Microsoft's domains. Yes, Microsoft has partially outsourced the management of its Domain Name System (DNS) servers to Linux for the job. Oops

BIND (**B**erkeley **I**nternet **N**ame **D**omain) is a widely used, free implementation of the Domain Name System for Unix and Windows NT. It provides a server, a client library, and several utility programs. It is estimated to be the DNS software in use in over 90% of the hosts on the Internet and this is the one that we will describe further down in this chapter.

To separate your internal Domain Name Services from external DNS, it is better to use Split DNS also known and referred to as "shadow namespaces". A Split DNS or "shadow namespace" is a name server that can answer queries from one source one way, and queries from another source another way. A Split DNS allow the Names, addresses and the topology of the secure network to be not available to the insecure external network. With Split DNS the external DNS only reveals public addresses and the internal DNS reveals internal IP addresses to the secure network. This is the recommended DNS configuration to use between hosts on the corporate network and external hosts.

To do split DNS, you must have two independent name servers for the same zone. One server and one copy of the zone are presented to the outside world. The other name server has a probably different bunch of contents for that zone which it makes available to the inside.

In our configuration and installation we'll run ISC BIND & DNS as non root-user and in a chrooted environment. We also provide you with three different configurations; one for a simple Caching Name Server Only (client), one for a Slave Name Server (Secondary DNS Server) and another one for a Master Name Server (Primary DNS Server).

The simple Caching Name Server configuration will be used for your servers that don't act as a Master or Slave Name Server, and the Slave and Master configurations will be used for your servers that act as a Master Name Server and Slave Name Server. Usually one of your servers acts as Primary/Master, another one acts as Secondary/Slave and the rest act as simple Caching client Name Server.

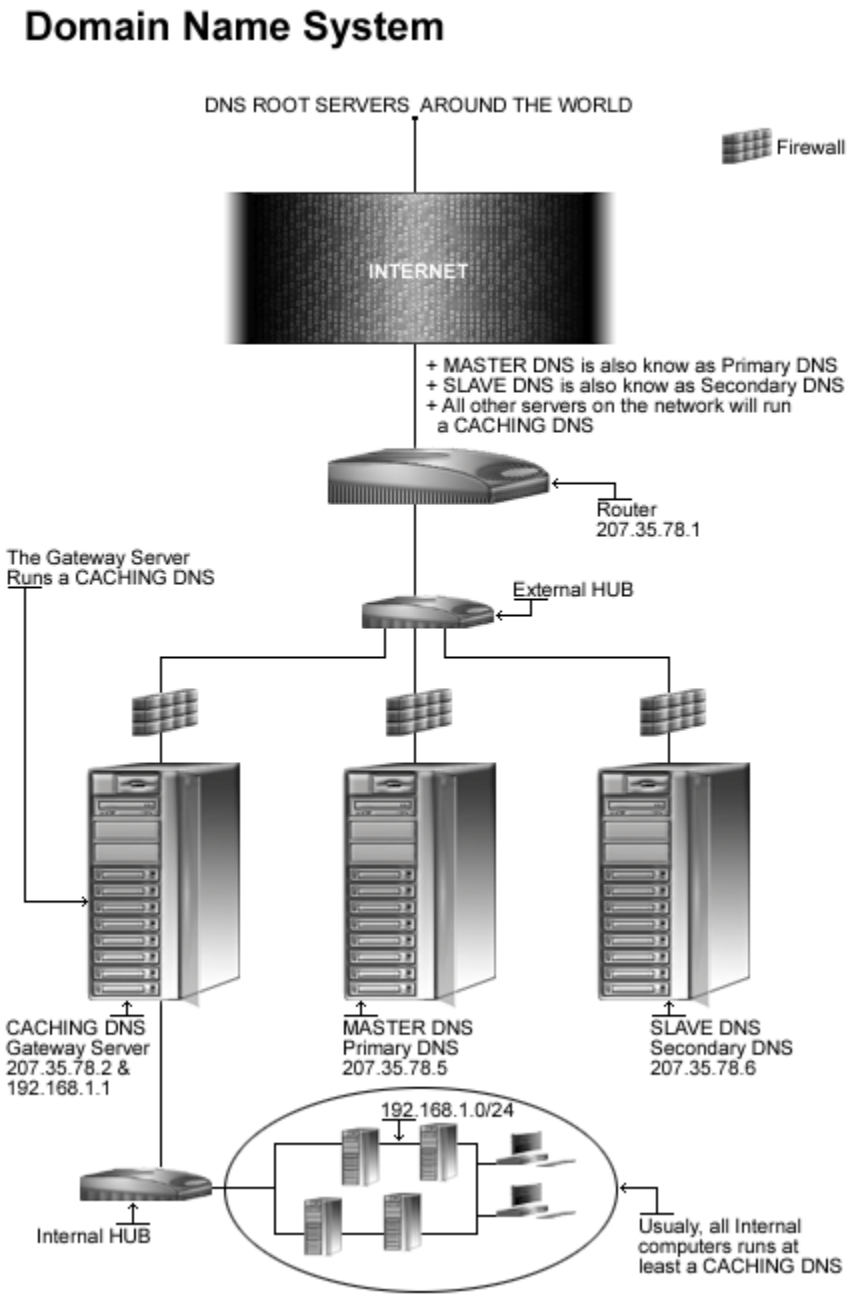
Recommended RPM packages to be installed for a DNS Server

A minimal configuration provides the basic set of packages required by the Linux operating system. A minimal configuration is a perfect starting point for building a secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a Primary/Master or Secondary/Slave Domain Name Server (DNS).

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install ISC BIND & DNS by RPM package. Therefore, bind and bind-utils RPM packages are already included in the list below as you can see. Not all security tools are installed, it is up to you to install them as you see fit, by RPM packages since compilers are not installed or included in the list.

basesystem	diffutils	initscripts	openssh	slang
bash	e2fsprogs	iptables	openssh-server	slocate
bdflush	ed	kernel	openssl	sysklogd
bind	file	less	pam	syslinux
bind-utils	filesystem	libstdc++	passwd	SysVinit
bzip2	fileutils	libtermcap	popt	tar
chkconfig	findutils	lilo	procps	termcap
console-tools	gawk	logrotate	psmisc	textutils
cpio	gdbm	losetup	pwdb	tmpwatch
cracklib	gettext	MAKEDEV	qmail	utempter
cracklib-dicts	glib	man	readline	util-linux
crontabs	glibc	mingetty	rootfiles	vim-common
db1	glibc-common	mktemp	rpm	vim-minimal
db2	grep	mount	sed	vixie-cron
db3	groff	ncurses	setup	words
dev	gzip	net-tools	sh-utils	which
devfsd	info	newt	shadow-utils	zlib

Tested and fully functional on OpenNA.com.



This is a graphical representation of the `DNS` configuration we use in this book. We try to show you different settings (Caching Only `DNS`, Primary/Master `DNS`, and Secondary/Slave `DNS`) on different servers. Please note that lot possibilities exist, and depend of your needs, and network architecture design.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest ICS BIND & DNS version number is 9.1.2

Packages

The following are based on information as listed by ISC BIND & DNS as of 2001/05/05. Please regularly check at www.isc.org for the latest status.

Source code is available from:
ISC BIND & DNS Homepage: http://www.isc.org/
ISC BIND & DNS FTP Site: 204.152.184.27
You must be sure to download: bind-9.1.2.tar.gz

Prerequisites

ICS BIND & DNS requires that the software below is already installed on your system to be able to compile successfully. If this is not the case, you must install it. Please make sure you have all of these programs installed on your machine before you proceed with this chapter.

- ✓ To improve signing and verification speed of BIND9, OpenSSL library that uses hand-optimized assembly language routines should be already installed on your system.
- ✓ Kernel 2.4 is required to set up BIND9 in your system.

NOTE: For more information on OpenSSL software, see its related chapter in this book.

Pristine source

If you don’t use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install ISC BIND & DNS, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > DNS1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > DNS2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff DNS1 DNS2 > ISC-BIND-DNS-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing ISC BIND & DNS

Below are the required steps that you must make to configure, compile and optimize the ISC BIND & DNS software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp bind-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf bind-version.tar.gz
```

Step 2
In order to check that the version of ISC BIND & DNS, which you are going to install, is an original and unmodified one, please check the supplied signature with the PGP key of ISC BIND & DNS. Unfortunately, ISC BIND & DNS don't provide a MD5 signature for verification. But a PGP key is available on the ISC BIND & DNS website.

To get a PGP key copy of ISC BIND & DNS, please point your browser to the following URL: http://www.isc.org/. For more information about how to use this key for verification, see the GnuPG chapter in this book.

Step 3
ISC BIND & DNS cannot run as super-user root; for this reason we must create a special user with no shell privileges on the system for running ISC BIND & DNS daemon.

- To create this special ISC BIND & DNS user, use the following command:

```
[root@deep tmp]# useradd -c "Named" -u 25 -s /bin/false -r -d /var/named
named 2>/dev/null || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID for the program.

Step 4
After that, move into the newly created ISC BIND & DNS directory and perform the following steps before compiling and optimizing it. The modifications we bring to the ISC BIND & DNS source files below are necessary to relocate some default files as well as to fix a small bug with the software.

- To move into the newly created ISC BIND & DNS directory, use the following command:

```
[root@deep tmp]# cd bind-9.1.2/
```

Step 4.1
The first file that we must modify is called **dighost.c** located under the source directory of ISC BIND & DNS. In this file, we will add a missing code line related to the reverse function of the program.

- Edit the **dighost.c** file (vi +224 bin/dig/dighost.c) and change the lines:

```
if (n == 0) {
        return (DNS_R_BADDOTTEDQUAD);
}
for (i = n - 1; i >= 0; i--) {
        snprintf(working, MXNAME/8, "%d.",
                 adrs[i]);
```

To read:

```
if (n == 0) {
        return (DNS_R_BADDOTTEDQUAD);
}
reverse[0] = 0;
for (i = n - 1; i >= 0; i--) {
        snprintf(working, MXNAME/8, "%d.",
                 adrs[i]);
```

Step 4.2
The second source file to modify is called **globals.h** and one of its functions is to specify the location of the named.pid and lwresd.pid files. We'll change the default location for these files to be compliant with our Linux operating system.

- Edit the **globals.h** file (vi +101 bin/named/include/named/globals.h) and change the lines:

```
"/run/named.pid");
```

To read:

```
"/run/named/named.pid");
```

and

```
"/run/lwresd.pid");
```

To read:

```
"/run/named/lwresd.pid");
```

Step 5
Once the required modifications have been made into the source files of ISC BIND & DNS, it is time configure and optimize it for our system.

- To configure and optimize ISC BIND & DNS use the following commands:

```
CFLAGS="-O3 -funroll-loops -fomit-frame-pointer" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--with-openssl=/usr/include/openssl \
--with-libtool \
--disable-ipv6 \
--enable-threads
```

This tells ISC BIND & DNS to set itself up for this particular system with:

- Build shared libraries.
- Use original OpenSSL rather than using bind-9 internal OpenSSL.
- Disable Ipv6 support.
- Use nultithreading

WARNING: Pay special attention to the above CFLAGS line. As you can see I voluntary omitted to include the option "-march=i686 and -mcpu=i686". I don't know why but with these options BIND compile successfully but never start on the system. Therefore I highly recommend to not include any "-march or -mcpu" options to compile BIND or nothing will work. Also if you have added this option into your /usr/lib/gcc-lib/i386-redhat-linux/2.96/specs or any equivalent file, remove it temporally the time to compile this program and add it after successful compilation of BIND.

Step 6
At this stage of our work the program is ready to be built and installed. We build ISC BIND & DNS with the 'make' command and produce a list of files on the system before we install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install ISC BIND & DNS.

```
[root@deep bind-9.1.2]# make
[root@deep bind-9.1.2]# cd
[root@deep /root]# find /* > DNS1
[root@deep /root]# cd /var/tmp/bind-9.1.2/
[root@deep bind-9.1.2]# make install
[root@deep bind-9.1.2]# cd doc/man/bin/
[root@deep bin]# install -c -m 444 named.8 /usr/share/man/man8/
[root@deep bin]# install -c -m 444 rndc.8 /usr/share/man/man8/
[root@deep bin]# install -c -m 444 lwresd.8 /usr/share/man/man8/
[root@deep bin]# install -c -m 444 nsupdate.8 /usr/share/man/man8/
[root@deep bin]# install -c -m 444 named-checkconf.1 /usr/share/man/man1/
[root@deep bin]# install -c -m 444 named-checkzone.1 /usr/share/man/man1/
[root@deep bin]# install -c -m 444 host.1 /usr/share/man/man1/
[root@deep bin]# install -c -m 444 dig.1 /usr/share/man/man1/
[root@deep bin]# install -c -m 444 rndc.conf.5 /usr/share/man/man5/
[root@deep bin]# cd ../../../
[root@deep bind-9.1.2]# strip /usr/sbin/named
[root@deep bind-9.1.2]# mkdir -p /var/named
```

```
[root@deep bind-9.1.2]# mkdir -p /var/run/named
[root@deep bind-9.1.2]# install -c -m 640 bin/rndc/rndc.conf /etc/
[root@deep bind-9.1.2]# chown named.named /etc/rndc.conf
[root@deep bind-9.1.2]# chown named.named /var/named/
[root@deep bind-9.1.2]# chown named.named /var/run/named/
[root@deep bind-9.1.2]# /sbin/ldconfig
[root@deep bind-9.1.2]# cd
[root@deep /root]# find /* > DNS2
[root@deep /root]# diff DNS1 DNS2 > ISC-BIND-DNS-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

Step 7
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete ISC BIND & DNS and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf bind-version/
[root@deep tmp]# rm -f bind-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install ISC BIND & DNS. It will also remove the ISC BIND & DNS compressed archive from the /var/tmp directory.

Configuring ISC BIND & DNS

After ISC BIND & DNS has been built and installed successfully in your system, your next step is to configure and customize all the required parameters in your different ISC BIND & DNS configuration files. Depending of the kind of Domain Name System you want to run in your Linux server, there are different configuration files to set up, these files are:

For running ISC BIND & DNS as a Caching-Only Name Server:

- ✓ /etc/named.conf (The ISC BIND & DNS Configuration File)
- ✓ /var/named/db.127.0.0 (The ISC BIND & DNS reverse mapping File)
- ✓ /var/named/db.cache (The ISC BIND & DNS Root server hints File)
- ✓ /etc/sysconfig/named (The ISC BIND & DNS System Configuration File)
- ✓ /etc/rc.d/init.d/named (The ISC BIND & DNS Initialization File)

For running ISC BIND & DNS as a Master/Primary Name Server:

- ✓ /etc/named.conf (The ISC BIND & DNS Configuration File)
- ✓ /var/named/db.127.0.0 (The ISC BIND & DNS reverse mapping File)
- ✓ /var/named/db.cache (The ISC BIND & DNS Root server hints File)
- ✓ /var/named/db.207.35.78 (The ISC BIND & DNS host names to addr mapping File)
- ✓ /var/named/db.openna (The ISC BIND & DNS addr to host names mapping File)
- ✓ /etc/sysconfig/named (The ISC BIND & DNS System Configuration File)
- ✓ /etc/rc.d/init.d/named (The ISC BIND & DNS Initialization File)

For running ISC BIND & DNS as a Slave/Secondary Name Server:

- /etc/named.conf (The ISC BIND & DNS Configuration File)
- /var/named/db.127.0.0 (The ISC BIND & DNS reverse mapping File)
- /var/named/db.cache (The ISC BIND & DNS Root server hints File)
- /etc/sysconfig/named (The ISC BIND & DNS System Configuration File)
- /etc/rc.d/init.d/named (The ISC BIND & DNS Initialization File)

WARNING: It is important to note that some of the configuration files mentioned above are the same for all types of Domain Name System and for this reason, files that are common for all configuration are described after all specific Domain Name System configurations. Please read all information contained in this chapter to be sure to not forget something.

Caching-Only Name Server

This section applies only if you chose to install and use ISC BIND & DNS as a Caching Name Server in your system. Caching-only name servers are servers not authoritative for any domains except 0.0.127.in-addr.arpa (the localhost). A Caching-Only Name Server can look up names inside and outside your zone, as can Primary and Slave Name Servers. The difference is that when a Caching-Only Name Server initially looks up a name within your zone, it ends up asking one of the Primary or Slave Names Servers for your zone for the answer.

/etc/named.conf: The ISC BIND & DNS Configuration File

Use this configuration file for all servers on your network that don't act as a Master or Slave Name Server. Setting up a simple Caching Server for local client machines will reduce the load on the network's primary server.

Step 1
Many users on dialup connections may use this configuration along with ISC BIND & DNS for such a purpose. With this configuration for a Caching-Only Name Server, all queries from outside clients are refused. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **named.conf** file (touch /etc/named.conf) and add the following lines in the file. Below is what we recommend you:

```
options {
        directory "/var/named";
        allow-transfer { none; };
        allow-query { 192.168.1.0/24; localhost; };
        allow-recursion { 192.168.1.0/24; localhost; };
        forwarders { 207.35.78.5; 207.35.78.6; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
```

```
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};
```

This tells `named.conf` file to set itself up for this particular configuration with:

`Options {};`
The `options` statement sets up global options to be used by `ISC BIND & DNS` and may appear only once in a configuration file.

`directory "/var/named";`
The `directory` statement indicates the working directory of the server and should be an absolute path. The working directory is where all configuration files related to `ISC BIND & DNS` resides.

`allow-transfer { none; };`
The `allow-transfer` statement specifies which hosts are allowed to receive zone transfers from the Primary/Master Name Server. The default setting of `ISC BIND & DNS` is to allow transfers from all hosts. Since zone transfer requests are only required for Secondary/Slave Name Server and since the configuration we are trying to do here is for a Caching-Only Name Server, we can completely disable this directive with the parameter "`allow-transfer { none; };`". This is a security feature.

`allow-query { 192.168.1.0/24; localhost; };`
The `allow-query` statement specifies which hosts are allowed to ask ordinary questions to the Caching Name Server. The default setting in the `options` block is to allow queries from all hosts. In our configuration, we wish to allow queries from one corporate subnet only. This is a security feature.

`allow-recursion { 192.168.1.0/24; localhost; };`
The `allow-recursion` statement specifies which hosts are allowed to make recursive queries through this server. With the configuration as shown above, we allow recursive queries only from internal hosts since allowing every external hosts on the Internet to ask your name server to answer recursive queries can open you up to certain kinds of cache poisoning attacks. This is a security feature.

`forwarders { 207.35.78.5; 207.35.78.6; };`
The `forwarders` statement specifies the IP addresses to be used for forwarding. Servers that do not have direct access to the Internet can use this option to create a large site-wide cache, reducing traffic over links to external name servers and to allow queries. It occurs only on those queries for which the server is not authoritative and does not have the answer in its cache. In the "`forwarders`" line, `207.35.78.5` and `207.35.78.6` are the IP addresses of the Primary (Master) and Secondary (Slave) DNS servers. They can also be the IP addresses of your ISP's DNS server and another DNS server, respectively.

Why would one assume that what's in one's ISP's name server's cache is any more "secure" than what one gets from the authoritative servers directly? That makes no sense at all. ISP's are often lazy about upgrades, which means that there's a substantial risk that their name servers may be compromised or cache-poisoned. Another downside of forwarding, of course, is that it introduces an extra hop for *every* query which can't be satisfied from the local server's cache or authoritative data.

Now, sometimes that hop is worth it (because the answer is in your forwarder's cache, so you don't need to expend other "hops" over the Internet trying to resolve it yourself), but at other times (when the answer *doesn't* happen to be in the forwarders cache), it just adds latency. So forwarding can *sometimes* be justified in terms of query performance. But in this case, it should be configured as "forward first" to provide redundancy in case the forwarders are unavailable. This is the default value "forward first" into BIND9, and causes the server to query the IP addresses as specified in the forwarders statement (the forwarders first), and if that doesn't answer the question, the server will then look for the answer itself. This is a performance feature.

```
version "Go away!";
```

The version statement allows us to hide the real version number of our ISC BIND & DNS server. This can be useful when some one from the Internet try to scan our Domain Name Server for possible vulnerable version of the software. You can change the string "Go away!" to whatever you want. Note doing this will not prevent attacks and may impede people trying to diagnose problems with your server. This is a security feature.

```
notify no;
```

DNS Notify is a mechanism that allows Master Name Servers to notify their Slave servers of changes to a zone's data. In response to a NOTIFY from a Master server, the Slave will check to see that its version of the zone is the current version and, if not, initiate a transfer. The notify statement by default is set to "yes" but since the loopback address 127.0.0.1 is the same to each system, we must avoid to transfer this localhost configuration file to Secondary/Slave Name Server.

NOTE: You can configure logging so that lame server messages aren't logged, which will reduce the overhead on your DNS and syslog servers. Lame server messages are report hosts that are believed to be name servers for the given domains, but which do not believe themselves to be such. This is often due to a configuration error on the part of that hostmaster.

You can disable "Lame server" messages by using the logging statement into your named.conf file:

```
logging {
         category lame-servers { null; };
};
```

By the way, some of us also like to disable message like "... points to a CNAME" by adding in the logging statement the following line:

```
category cname { null; };
```

Step 2
Finally, we must set the mode permissions of this file to be (0600/-rw-------) and owned by the user 'named' for security reason.

- To change the mode permissions and ownership of the named.conf file, use the following commands:

```
[root@deep /]# chmod 600 /etc/named.conf
[root@deep /]# chown named.named /etc/named.conf
```

/var/named/db.127.0.0: The reverse mapping File

Use this configuration file for all servers on your network that don't act as a Master or Slave Name Server. The "db.127.0.0" file covers the loopback network by providing a reverse mapping for the loopback address on your system.

Step 1
Create the following file in /var/named.

- Create the **db.127.0.0** file (touch /var/named/db.127.0.0) and add the following lines in the file:

```
; Revision History: March 01, 2001 - root@openna.com
; Start of Authority (SOA) records.
$TTL 86400
@       IN      SOA     localhost.      root.localhost. (
                                        00      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        86400 ) ; Minimum

        IN      NS      localhost.
1       IN      PTR     localhost.
```

Step 2
Now, we must set the mode permissions of this file to be (0644/-rw-r--r--) and owned by the user 'named' for security reason.

- To change the mode permissions and ownership of this file, use the following commands:

```
[root@deep /]# chmod 644 /var/named/db.127.0.0
[root@deep /]# chown named.named /var/named/db.127.0.0
```

Primary Master Name Server

This section applies only if you chose to install and use ISC BIND & DNS as a Primary Name Server in your system. The Primary Master Server is the ultimate source of information about a domain. The Primary Master is an authoritative server configured to be the source of zone transfer for one or more Secondary servers. The Primary Master Server obtains data for the zone from a file on disk.

/etc/named.conf: The ISC BIND & DNS Configuration File

Use this configuration for the server on your network that acts as a Master Name Server. In every respectable networking environment, you need to set up at least a Primary Domain Name Server for your network. We'll use "openna.com" as an example domain, and assume you are using IP network address of 207.35.78.0.

Step 1
To do this, add the following lines to your `/etc/named.conf` file. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **`named.conf`** file (`touch /etc/named.conf`) and add the following lines in the file. Below is what we recommend you:

```
options {
        directory "/var/named";
        allow-transfer { 207.35.78.6; };
        allow-query { 192.168.1.0/24; 207.35.78.0/24; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/24; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are the master server for OpenNA.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
        type master;
        file "db.207.35.78";
        allow-query { any; };
};
```

This tells `named.conf` file to set itself up for this particular configuration with:

`Options {};`
The `options` statement sets up global options to be used by `ISC BIND & DNS` and may appear only once in a configuration file.

`directory "/var/named";`
The `directory` statement indicates the working directory of the server and should be an absolute path. The working directory is where all configuration files related to `ISC BIND & DNS` resides.

`allow-transfer { 207.35.78.6; };`
The `allow-transfer` statement specifies which hosts are allowed to receive zone transfers from the Primary/Master Name Server. The default setting of `ISC BIND & DNS` is to allow transfers from all hosts. In the `allow-transfer` line as shown above, `207.35.78.6` (our Secondary/Slave Name Server) is the only IP address allowed to receive zone transfers from the

Primary/Master Name Server. You should configure your server to respond to zone transfers requests only from authorized IP addresses. In most cases, you'll only authorize your known Slave servers to transfer zones from your Primary/Master Name Server. As the information provided is often used by spammers and IP spoofers. This is a security feature.

`allow-query { 192.168.1.0/24; 207.35.78.0/24; localhost; };`
The `allow-query` statement specifies which hosts are allowed to ask ordinary questions to the Primary Name Server. The default setting in the `options` block is to allow queries from all hosts. In our configuration, we wish to allow queries from our subnets (`192.168.1.0/24`, `207.35.78.0/32`, and `localhost`). With this restriction, everyone form the Internet can query us for the zones that we administer and its reverse, but only internal hosts that we have specified in the "`allow-query`" statement can make other queries. Take a note that we add the "`allow-query { any; };`" option in each zone statement into the `named.conf` file to make effective this protection. This is a security feature.

`allow-recursion { 192.168.1.0/24; 207.35.78.0/24; localhost; };`
The `allow-recursion` statement specifies which hosts are allowed to make recursive queries through this server. With the configuration as shown above, we allow recursive queries only from internal hosts since allowing every external hosts on the Internet (external hosts will have their own name servers) to ask your name server to answer recursive queries can open you up to certain kinds of cache poisoning attacks. This is a security feature.

`version "Go away!";`
The `version` statement allows us to hide the real version number of our `ISC BIND & DNS` server. This can be useful when some one from the Internet try to scan our Domain Name Server for possible vulnerable version of the software. You can change the string "`Go away!`" to whatever you want. Note doing this will not prevent attacks and may impede people trying to diagnose problems with your server. This is a security feature.

`notify no;`
DNS Notify is a mechanism that allows Master Name Servers to notify their Slave servers of changes to a zone's data. In response to a NOTIFY from a Master Server, the Slave will check to see that its version of the zone is the current version and, if not, initiate a transfer. The `notify` statement by default is set to "`yes`" but since the `loopback address 127.0.0.1` is the same to each system, we must avoid to transfer this `localhost` configuration file to Secondary/Slave Name Server.

NOTE: You can configure logging so that lame server messages aren't logged, which will reduce the overhead on your `DNS` and `syslog` servers. Lame server messages are report hosts that are believed to be name servers for the given domains, but which do not believe themselves to be such. This is often due to a configuration error on the part of that hostmaster.

You can disable "Lame server" messages by using the logging statement into your `named.conf` file:

```
logging {
         category lame-servers { null; };
};
```

By the way, some of us also like to disable message like "... points to a CNAME" by adding in the logging statement the following line:

```
category cname { null; };
```

Step 2
Finally, we must set the mode permission of this file to be (0600/-rw-------) and owned by the user 'named' for security reason.

- To change the mode permission and ownership of the named.conf file, use the following commands:

```
[root@deep /]# chmod 600 /etc/named.conf
[root@deep /]# chown named.named /etc/named.conf
```

/var/named/db.127.0.0: The reverse mapping File

Use this configuration file for the server on your network that acts as a Master Name Server. The "db.127.0.0" file covers the loopback network by providing a reverse mapping for the loopback address on your system.

Step 1
Create the following file in /var/named.

- Create the **db.127.0.0** file (touch /var/named/db.127.0.0) and add the following lines in the file:

```
; Revision History: March 01, 2001 - root@openna.com
; Start of Authority (SOA) records.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        00      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        172800 ); Minimum TTL of 1 day

; Name Server (NS) records.
        IN      NS      ns1.openna.com.
        IN      NS      ns2.openna.com.

; only One PTR record.
1       PTR     localhost.
```

Step 2
Now, we must set the mode permissions of this file to be (0644/-rw-r--r--) and owned by the user 'named' for security reason.

- To change the mode permission and ownership of this file, use the following commands:

```
[root@deep /]# chmod 644 /var/named/db.127.0.0
[root@deep /]# chown named.named /var/named/db.127.0.0
```

/var/named/db.207.35.78: The host names to addresses mapping File

Use this configuration file for the server on your network that acts as a Master Name Server. The "db.207.35.78" file maps host names to addresses.

Step 1
Create the following file in /var/named.

- Create the **db.207.35.78** file (touch /var/named/db.207.35.78) and add the following lines in the file:

```
; Revision History: March 01, 2001 - root@openna.com
; Start of Authority (SOA) records.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        00      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        172800 ); Minimum TTL of 1 day

; Name Server (NS) records.
        IN      NS      ns1.openna.com.
        IN      NS      ns2.openna.com.

; Addresses Point to Canonical Names (PTR) for Reverse lookups
1       IN      PTR     router.openna.com.
2       IN      PTR     portal.openna.com.
3       IN      PTR     www.openna.com.
4       IN      PTR     smtp.openna.com.
```

Step 2
Now, we must set the mode permission of this file to be (0644/-rw-r--r--) and owned by the user 'named' for security reason.

- To change the mode permission and ownership of this file, use the following commands:

```
[root@deep /]# chmod 644 /var/named/db.207.35.78
[root@deep /]# chown named.named /var/named/db.207.35.78
```

/var/named/db.openna: The addresses to host names mapping File

Use this configuration file for the server on your network that acts as a Master Name Server. The "db.openna" file maps addresses to host names.

Step 1
Create the following file in /var/named.

- Create the **db.openna** file (touch /var/named/db.openna) and add the following lines in the file:

```
; Revision History: March 01, 2001 - root@openna.com
; Start of Authority (SOA) records.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        00      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
```

```
                                        172800 ); Minimum TTL of 1 day

; Name Server (NS) records.
        IN      NS      ns1.openna.com.
        IN      NS      ns2.openna.com.

; Mail Exchange (MX) records.
        MX      0       smtp.openna.com.

; Address (A) records.
localhost       IN      A       127.0.0.1
router          IN      A       207.35.78.1
portal          IN      A       207.35.78.2
www             IN      A       207.35.78.3
smtp            IN      A       207.35.78.4
```

Step 2
Now, we must set the mode permission of this file to be (`0644/-rw-r--r--`) and owned by the user '`named`' for security reason.

- To change the mode permission and ownership of this file, use the following commands:
```
[root@deep /]# chmod 644 /var/named/db.openna
[root@deep /]# chown named.named /var/named/db.openna
```

Secondary Slave Name Server

This section applies only if you chose to install and use `ISC BIND & DNS` as a Secondary Name Server in your system. The purpose of a Slave Name Server is to share the load with the Master Name Server, or handle the entire load if the Master Name Server is down. A Slave Name Server, which is an authoritative server, loads its data over the network from another Name Server (usually the Master Name Server, but it can load from another Slave Name Server too). This process is called a zone transfer. Slave servers provide necessary redundancy on the network.

`/etc/named.conf`: The `ISC BIND & DNS` Configuration File

Use this configuration for the server on your network that acts as a Slave Name Server. You must modify the "`named.conf`" file on the Slave Name Server host.

Step 1
Change every occurrence of primary to secondary except for "`0.0.127.in-addr.arpa`" and add a masters line with the IP address of the Master Server as shown below. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **`named.conf`** file (`touch /etc/named.conf`) and add the following lines in the file. Below is what we recommend you:

```
options {
        directory "/var/named";
        allow-transfer { none; };
        allow-query { 192.168.1.0/24; 207.35.78.0/24; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/24; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
```

```
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are a slave server for OpenNA.com
zone "openna.com" {
        type slave;
        file "db.openna";
        masters { 207.35.78.5; };
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
        type slave;
        file "db.207.35.78";
        masters { 207.35.78.5; };
        allow-query { any; };
};
```

The above `named.conf` file tells the Secondary Name Server that it is a Slave Server for the zone "`openna.com`" and should track the version of this zone that is being kept on the host "`207.35.78.5`", which is the Master Name Server in the network.

NOTE: A Slave Name Server doesn't need to retrieve all of its database (`db`) files over the network because these `db` files "`db.127.0.0`" and "`db.cache`" are the same as on a Primary Master, so you can keep a local copy of these files on the Slave Name Server.

You can configure logging so that lame server messages aren't logged, which will reduce the overhead on your `DNS` and `syslog` servers. Lame server messages are report hosts that are believed to be name servers for the given domains, but which do not believe themselves to be such. This is often due to a configuration error on the part of that hostmaster.

You can disable "Lame server" messages by using the logging statement into your `named.conf` file:

```
logging {
         category lame-servers { null; };
};
```

By the way, some of us also like to disable message like "... points to a CNAME" by adding in the logging statement the following line:

```
category cname { null; };
```

Step 2
Finally, we must set the mode permissions of this file to be (0600/-rw-------) and owned by the user 'named' for security reason.

- To change the mode permission and ownership of the named.conf file, use the following commands:
```
[root@deep /]# chmod 600 /etc/named.conf
[root@deep /]# chown named.named /etc/named.conf
```

/var/named/db.127.0.0: The reverse mapping File

Use this configuration file for the server on your network that acts as a Slave Name Server. The "db.127.0.0" file covers the loopback network by providing a reverse mapping for the loopback address on your system.

Step 1
Create the following file in /var/named.

- Create the **db.127.0.0** file (touch /var/named/db.127.0.0) and add the following lines in the file:

```
; Revision History: March 01, 2001 - root@openna.com
; Start of Authority (SOA) records.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        00      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        172800 ); Minimum TTL of 1 day

; Name Server (NS) records.
        IN      NS      ns1.openna.com.
        IN      NS      ns2.openna.com.

; only One PTR record.
1       PTR     localhost.
```

Step 2
Now, we must set the mode permissions of this file to be (0644/-rw-r--r--) and owned by the user 'named' for security reason.

- To change the mode permission and ownership of this file, use the following commands:
```
[root@deep /]# chmod 644 /var/named/db.127.0.0
[root@deep /]# chown named.named /var/named/db.127.0.0
```

/var/named/db.cache: The Root server hints File

This section applies for all type of Name Server (Caching, Master or Slave) that you may want to install in your system. The **db.cache** file is also know as the "Root server hints file" and tells your server (Caching, Master or Slave) where the servers for the "root" zone are, you must get a copy of **db.cache** file and copy this file into the /var/named directory.

Step 1
Use the following commands on another Unix computer in your organization to query a new **db.cache** file for your Name Servers or pick one from your Linux CD-ROM source distribution:

- To query a new db.cache file, use the following command:
  [root@deep]# dig @a.root-servers.net . ns > db.cache

- To query a new db.cache file by IP address, use the following command:
  [root@deep]# dig @198.41.0.4 . ns > db.cache

Don't forget to copy the db.cache file to the /var/named directory on your Name Server after retrieving it over the Internet.

NOTE: The root name servers do not change very often, but they do change. A good practice is to update your db.cache file every month or two.

Step 2
Now, we must set the mode permission of this file to be (0644/-rw-r--r--) and owned by the user 'named' for security reason.

- To change the mode permission and ownership of this file, use the following commands:
  [root@deep /]# **chmod 644 /var/named/db.cache**
  [root@deep /]# **chown named.named /var/named/db.cache**

/etc/sysconfig/named: The ISC BIND & DNS System Configuration File

This section applies for all type of Name Server (Caching, Master or Slave) that you may want to install in your system. The /etc/sysconfig/named file is used to specify ISC BIND & DNS system configuration information, such as if ISC BIND & DNS should run in a chroot environment, and if additional options are required to be passed to named daemon at startup.

- Create the **named** file (touch /etc/sysconfig/named) and add the following lines:

```
# Currently, you can use the following options:
#ROOTDIR=""
#OPTIONS=""
```

The "ROOTDIR="" option instructs ISC BIND & DNS where its root directory should be located, this line is useful when you want to run ISC BIND & DNS in an chroot jail environment for more security. For now, this line must be commented out since we'll see later in this chapter how to run ISC BIND & DNS in a chroot environment and how to use this option.

As usual with many daemons under Unix, we can add special options to the command line before starting the daemons. With the new system V feature of Linux most of command line options can now be specified in config files like the one above. The "OPTIONS=""" parameter in the /etc/sysconfig/named file is for this use for ISC BIND & DNS. We can for example add the "-d" option for debug level of ISC BIND & DNS but in most cases we don't need to use it.

/etc/rc.d/init.d/named: The ISC BIND & DNS Initialization File

This section applies for all type of Name Server (Caching, Master or Slave) that you may want to install in your system. The /etc/rc.d/init.d/named script file is responsible to automatically starting and stopping the ISC BIND & DNS daemon on your server. Loading the named daemon, as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Step 1
Create the **named** script file (touch /etc/rc.d/init.d/named) and add the following lines inside it:

```
#!/bin/bash
#
# named           This shell script takes care of starting and stopping
#                 named (BIND DNS server).
#
# chkconfig: - 55 45
# description: named (BIND) is a Domain Name Server (DNS) \
# that is used to resolve host names to IP addresses.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ "${NETWORKING}" = "no" ] && exit 0

[ -f /etc/sysconfig/named ] && . /etc/sysconfig/named

[ -f /usr/sbin/named ] || exit 0

[ -f "${ROOTDIR}"/etc/named.conf ] || exit 0

RETVAL=0

start() {
        # Start daemons.
        echo -n "Starting named: "
        if [ -n "${ROOTDIR}" -a "x${ROOTDIR}" != "x/" ]; then
                OPTIONS="${OPTIONS} -t ${ROOTDIR}"
        fi
        daemon named -u named ${OPTIONS}
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/named
        echo
        return $RETVAL
}
stop() {
        # Stop daemons.
        echo -n "Shutting down named: "
```

```
        killproc named
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/named
        echo
        return $RETVAL
}
restart() {
        stop
        start
}
reload() {
        /usr/sbin/rndc reload
        return $?
}
probe() {
        /usr/sbin/rndc reload >/dev/null 2>&1 || echo start
        return $?
}

# See how we were called.
case "$1" in
        start)
                start
                ;;
        stop)
                stop
                ;;
        restart)
                restart
                ;;
        condrestart)
                [ -f /var/lock/subsys/named ] && restart
                ;;
        reload)
                reload
                ;;
        probe)
                probe
                ;;
        *)
                echo "Usage: named
{start|stop|restart|condrestart|reload|probe}"
                exit 1
esac

exit $?
```

Step 2
Once the named script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and the creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/named
[root@deep /]# chown 0.0 /etc/rc.d/init.d/named
```

- To create the symbolic rc.d links for ISC BIND & DNS, use the following commands:
```
[root@deep /]# chkconfig --add named
[root@deep /]# chkconfig --level 2345 named on
```

- To start ISC BIND & DNS software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/named start
Starting named:           [OK]
```

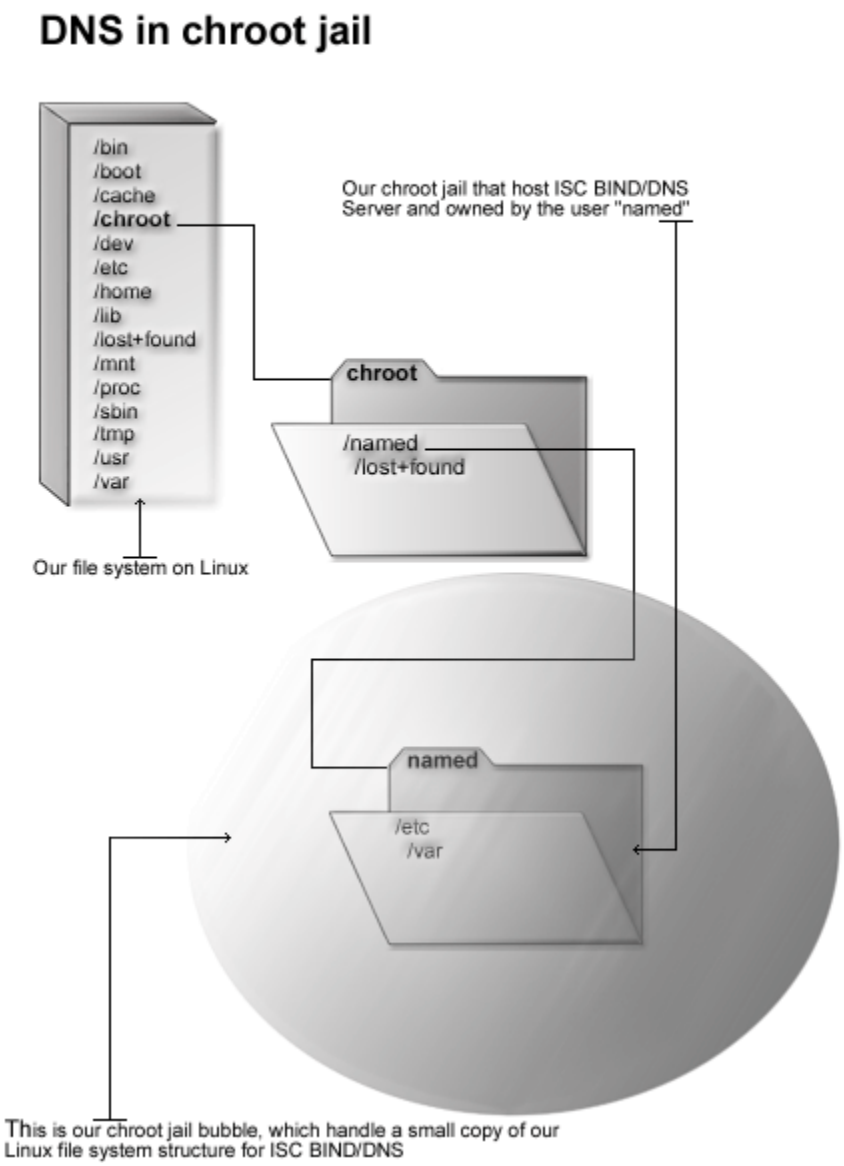
NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, floppy-2.0.tgz for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example /var/tmp, assuming you have done this your directory structure will be /var/tmp/floppy-2.0. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Running ISC BIND & DNS in a chroot jail

This part focuses on preventing ISC BIND & DNS from being used as a point of break-in to the system hosting it. Since ISC BIND & DNS performs a relatively large and complex function, the potential for bugs that affect security is rather high with this software. In fact, there have been many exploitable bugs in the past that allowed a remote attacker to obtain root access to hosts running ISC BIND & DNS.

To minimize this risk, ISC BIND & DNS can be run **as a non-root user**, which will limit any damage to what can be done as a normal user with a local shell. Of course, this is not enough for the security requirements of most DNS servers, so an additional step can be taken - that is, **running ISC BIND & DNS in a chroot jail.**

The main benefit of a chroot jail is that the jail will limit the portion of the file system the DNS daemon program can see to the root directory of the jail. Additionally, since the jail only needs to support DNS, the programs related to ISC BIND & DNS available in the jail can be extremely limited. Most importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail.


DNS in chroot jail
/bin
/boot
/cache
/chroot
/dev
/etc
/home
/lib
/lost+found
/mnt
/proc
/sbin
/tmp
/usr
/var
Our chroot jail that host ISC BIND/DNS
Server and owned by the user "named"
chroot
/named
/lost+found
Our file system on Linux
named
/etc
/var
This is our chroot jail bubble, which handle a small copy of our
Linux file system structure for ISC BIND/DNS

Necessary steps to run ISC BIND & DNS software in a chroot jail:

What you're essentially doing is creating a skeleton root file system with enough components necessary (directories, files, etc.) to allow Unix to do a chroot when the ISC BIND & DNS daemon starts. Contrary to its predecessor (Bind8), Bind9 is far more easily to setup in a chroot jail environment. Now there is no need to copy shared library dependencies of named binary as well as binaries programs to the jail. All you have to do is to copy its configuration file with its zone files and instruct its daemon process to chroot to the appropriate chroot directory before starting.

Step 1

The first step to do for running ISC BIND & DNS in a chroot jail will be to set up the chroot environment, and create the root directory of the jail. We've chosen /chroot/named for this purpose because we want to put this on its own separate file system to prevent file system attacks. Early in our Linux installation procedure we created a special partition /chroot for this exact purpose.

```
[root@deep /]# /etc/rc.d/init.d/named stop ← Only if named daemon already run.
Shutting down named:              [OK]

[root@deep /]# mkdir -p /chroot/named
[root@deep /]# mkdir -p /chroot/named/etc
[root@deep /]# mkdir -p /chroot/named/var/run/named
[root@deep /]# mkdir -p /chroot/named/var/named
[root@deep /]# chown -R named.named /chroot/named/var/run/named/
[root@deep /]# chown -R named.named /chroot/named/var/named/
```

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible.

> **WARNING:** The owner of the /chroot/named/var/named directory and all files into this directory must be owned by the process called "named".

Step 2

After that, we must move the main configuration files of ISC BIND & DNS into the appropriate places in the chroot jail. This includes the named.conf file and all zone files.

```
[root@deep /]# mv /etc/named.conf /chroot/named/etc/
[root@deep /]# cd /var/named; mv * /chroot/named/var/named/
[root@deep named]# chown named.named /chroot/named/etc/named.conf
[root@deep named]# chown -R named.named /chroot/named/var/named/*
```

Step 3

You will also need the **/etc/localtime** file in your chroot jail structure so that log entries are adjusted for your local time zone properly.

```
[root@deep named]# cp /etc/localtime /chroot/named/etc/
```

Step 4
Now we must set the named.conf file in the chroot jail directory immutable bit for better security.

- This procedure can be accomplished with the following commands:
```
[root@deep named]# cd /chroot/named/etc/
[root@deep etc]# chattr +i named.conf
```

WARNING: Don't forget to remove the immutable bit on these files if you have some modifications to bring to them with the command "chattr -i".

Step 5
Once the required files to run ISC BIND & DNS in the chroot jail environment have been relocated, we can remove the unnecessary directories related to ISC BIND & DNS from the system since the ones we'll work with now on a daily basis are located under the chroot directory. These directories are /var/named and /var/run/named.

```
[root@deep /]# rm -rf /var/named/
[root@deep /]# rm -rf /var/run/named/
```

Step 6
After that, it is time to instruct ISC BIND & DNS to start in the chroot jail environment. The /etc/sysconfig/named file is used for this purpose.

- Edit the **named** file (vi /etc/sysconfig/named) and change the following lines:

```
# Currently, you can use the following options:
#ROOTDIR=""
#OPTIONS=""
```

To read:

```
# Currently, you can use the following options:
ROOTDIR="/chroot/named/"
```

The "ROOTDIR="/chroot/named/"" option instructs ISC BIND & DNS where the chroot directory is located. Therefore the named daemon reads this line in the /etc/sysconfig/named file and chroot to the specified directory before starting.

Step 7
Finally, we must test the new chrooted jail configuration of our ISC BIND & DNS server.

- Start the new chrooted jail ISC BIND & DNS with the following command:
```
[root@deep /]# /etc/rc.d/init.d/named start
Starting named:                            [OK]
```

- If you don't get any errors, do a **ps ax | grep named** and see if we're running:
```
[root@deep /]# ps ax | grep named
4278 ?        S      0:00 named -u named -t /chroot/named/
4279 ?        S      0:00 named -u named -t /chroot/named/
4280 ?        S      0:00 named -u named -t /chroot/named/
4281 ?        S      0:00 named -u named -t /chroot/named/
```

If so, lets check to make sure it's chrooted by picking out one of its process numbers and doing **ls -la /proc/that_process_number/root/**.

```
[root@deep /]# ls -la /proc/4278/root/
```

If you see something like:

```
total 4
drwxrwxr-x    4 root     root         1024 Feb 22 16:23 .
drwxr-xr-x    4 root     root         1024 Feb 22 14:33 ..
drwxrwxr-x    2 root     root         1024 Feb 22 15:52 etc
drwxrwxr-x    4 root     root         1024 Feb 22 14:34 var
```

Congratulations! Your ISC BIND & DNS in chroot jail is working.

Securing ISC BIND & DNS

This section deals especially with actions we can make to improve and tighten security under ISC BIND & DNS. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

TSIG based transaction security with BIND

This section applies only if you chose to install and use ISC BIND & DNS as a Master or Slave Name Server in your system. The new BIND9 which is a major rewrite of almost all aspects of the underlying BIND architecture allows us to create transaction keys and use **T**ransaction **SIG**natures (TSIG) with ISC BIND & DNS (TSIG is used for signed DNS requests).

This means that if the server receives a message signed by this key, it can verify the signature. If the signature succeeds, the same key signs the response.

This new feature of BIND will allow us to have a better control about who can make a zone transfer, notify, and recursive query messages on the DNS server. It might be useful for dynamic updates too. Below, we show you all the required steps to generate this key and how to use it in your named.conf file.

Step 1
The first step will be to generate shared keys for each pair of hosts. This shared secret will be shared between Primary Domain Name Server and Secondary Domain Name Server and an arbitrary key name must be chosen like in our example "ns1-ns2". It is also important that the key name be the same on both hosts.

- To generate shared keys, use the following command:
  ```
  [root@deep /]# dnssec-keygen -a hmac-md5 -b 128 -n HOST ns1-ns2
  Kns1-ns2.+157+49406
  ```

Step 2
The above command will generate a 128 bit (16 byte) HMAC-MD5 key and the result will be in a file called "Kns1-ns2.+157+49406.private" with a base-64 encoded string following the word "Key:", which must be extracted from the file and used as a shared secret.

- Edit the **Kns1-ns2.+157+49406.private** file (vi Kns1-ns2.+157+49406.private), and extract the base-64 encoded string:

```
Private-key-format: v1.2
Algorithm: 157 (HMAC_MD5)
Key: ps1jy3f7czVa1VNZkYaLfw==
```

The string "ps1jy3f7czVa1VNZkYaLfw==" in the above example is the part of this file that must be extracted and used as the shared secret.

Step 3
Once the required base-64 encoded string has been extracted from the generated file, we can remove the files from our system and copy the shared secret to both machines via a secure transport mechanism like ssh, telephone, etc.

- To remove the generated files from the system, use the following commands:

```
[root@deep /]# rm -f Kns1-ns2.+157+49406.key
[root@deep /]# rm -f Kns1-ns2.+157+49406.private
```

Step 4
After that, it is time to inform the servers (Primary & Secondary) of the Key's existence by adding to each server's named.conf file the following parameters.

- Edit the **named.conf** file (vi /chroot/named/etc/named.conf) on both DNS servers, and add the following lines:

```
key ns1-ns2 {
  algorithm hmac-md5;
  secret "ps1jy3f7czVa1VNZkYaLfw==";
};
```

Once the above lines have been added, your named.conf file on both DNS servers (Primary & Secondary) should look something like:

For Primary/Master server:

```
key ns1-ns2 {
  algorithm hmac-md5;
  secret "ps1jy3f7czVa1VNZkYaLfw==";
};

options {
        directory "/var/named";
        allow-transfer { 207.35.78.6; };
        allow-query { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are the master server for openna.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
        type master;
        file "db.207.35.78";
        allow-query { any; };
};
```

For Secondary/Slave server:

```
        key ns1-ns2 {
          algorithm hmac-md5;
          secret "ps1jy3f7czVa1VNZkYaLfw==";
        };

        options {
                directory "/var/named";
                allow-transfer { none; };
                allow-query { 192.168.1.0/24; 207.35.78.0/32; localhost; };
                allow-recursion { 192.168.1.0/24; 207.35.78.0/32; localhost; };
                version "Go away!";
        };

        logging {
                 category lame-servers { null; };
        };

        // Root server hints
        zone "." { type hint; file "db.cache"; };

        // Provide a reverse mapping for the loopback address 127.0.0.1
        zone "0.0.127.in-addr.arpa" {
                type master;
                file "db.127.0.0";
                notify no;
        };

        // We are a slave server for openna.com
        zone "openna.com" {
                type slave;
                file "db.openna";
                masters { 207.35.78.5; };
                allow-query { any; };
        };

        zone "78.35.207.in-addr.arpa" {
                type slave;
                file "db.207.35.78";
                masters { 207.35.78.5; };
                allow-query { any; };
};
```

Step 5
One of the last steps is to instruct the both servers (Primary & Secondary) to Use the Key. The servers must be told when keys are to be used. Adding another parameter into the `named.conf` file on both DNS servers does this. Into this parameter, on `ns1` we add the IP address of `ns2` and on `ns2` we add the IP address of `ns1`.

- Edit the **`named.conf`** file (`vi /chroot/named/etc/named.conf`) on both `DNS` servers, and add the following lines:

```
server x.x.x.x {
  keys { ns1-ns2 ;};
};
```

Where `x.x.x.x` is the IP address.

Once the above lines have been added, your `named.conf` file on both `DNS` servers (Primary & Secondary) should look something like:

For Primary/Master server:

```
key ns1-ns2 {
  algorithm hmac-md5;
  secret "ps1jy3f7czVa1VNZkYaLfw==";
};

server 207.35.78.6 {
  keys { ns1-ns2 ;};
};

options {
        directory "/var/named";
        allow-transfer { 207.35.78.6; };
        allow-query { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are the master server for openna.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
```

```
        type master;
        file "db.207.35.78";
        allow-query { any; };
};
```

For Secondary/Slave server:

```
key ns1-ns2 {
  algorithm hmac-md5;
  secret "ps1jy3f7czVa1VNZkYaLfw==";
};

server 207.35.78.5 {
  keys { ns1-ns2 ;};
};

options {
        directory "/var/named";
        allow-transfer { none; };
        allow-query { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are a slave server for openna.com
zone "openna.com" {
        type slave;
        file "db.openna";
        masters { 207.35.78.5; };
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
        type slave;
        file "db.207.35.78";
        masters { 207.35.78.5; };
        allow-query { any; };
};
```

Step 6
Finally, since the named.conf file now handles a secret key, it is recommended that named.conf (on Primary & Secondary servers) be non-world readable.

- This procedure can be accomplished with the following command on Primary server:
  ```
  [root@deep /]# chmod 600 /chroot/named/etc/named.conf
  ```
- This procedure can be accomplished with the following command on Secondary server:
  ```
  [root@deep /]# chmod 600 /chroot/named/etc/named.conf
  ```

Restart your DNS server on both sides for the changes to take effect.

- Restart ISC BIND & DNS with the following command on both DNS servers:
  ```
  [root@deep /]# /etc/rc.d/init.d/named restart
  Shutting down named:                       [OK]
  Starting named:                            [OK]
  ```

WARNING: With TSIG feature enable on your DNS server, it is important to be sure that the clocks on the client and server are synchronized. TSIGs include a timestamp to reduce the potential for replay attacks. If the client and server's clocks are out by too much, TSIG validations will inevitably fail.

Using TSIG key based access control to make a zone transfer

This section applies only if you chose to install and use ISC BIND & DNS as a Master or Slave Name Server in your system. Once the TSIG feature has been configured and enabled in your DNS server, we can use it to improve security on the system.

One improvement can be made with the allow-transfer statement of ISC BIND & DNS. Usually, we configure our Primary/Master Domain Name Server to respond to zone transfers requests only from authorized IP addresses. In most cases, we'll only authorize our known Secondary/Slave Domain Name Servers. The same technique as described here can also be used for dynamic update, notify, and recursive query messages.

With BIND9, we do that within a zone phrase in the Primary Name Server with a directive like "allow-transfer { 207.35.78.6; };", but with the sharing of keys between ns1 and ns2 like we previously did, we have extended the possibility of our named.conf file to allow TSIG keys and can use this feature to modify the allow-transfer directive, which will improve security of zone transfer between ns1 and ns2.

- To use TSIG key based access control to make a zone transfer between Primary DNS & Secondary DNS, edit your **named.conf** file on the Primary/Master Domain Name Server (vi /chroot/named/etc/named.conf) and change the line:

  ```
  allow-transfer { 207.35.78.6; };
  ```

  To Read:

  ```
  allow-transfer { key ns1-ns2; };
  ```

This allows zone transfer to succeed only if a key named “ns1-ns2” signed the request, which only your Primary & Secondary `DNS` known and handle in their `named.conf` file.

Once the above line has been modified, your `named.conf` file on the Primary/Master server should look something like:

```
key ns1-ns2 {
  algorithm hmac-md5;
  secret "ps1jy3f7czVa1VNZkYaLfw==";
};

server 207.35.78.6 {
  keys { ns1-ns2 ;};
};

options {
        directory "/var/named";
        allow-transfer { key ns1-ns2; };
        allow-query { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are the master server for openna.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
        type master;
        file "db.207.35.78";
        allow-query { any; };
};
```

WARNING: If you use BIND9's dynamic update functionality, you'll also want to restrict zone updates to authorized IP addresses and you'd probably do this in the zone phrase. Note that if you don't specify an allow-update option, updates are not allowed for that zone, so you'll only need to do this if you actually use dynamic update.

```
zone "openna.com" {
        type master;
        file "db.openna";
        allow-update { key ns1-ns2; };
        allow-query { any; };
};
```

Using encryption algorithm for the name server control utility rndc

This section applies for all type of ISC BIND & DNS. The BIND9 utility for controlling the name server, rndc, has its own configuration file /etc/rndc.conf, which also required a TSIG key to work. The name server must be configured to accept rndc connections and to recognize the key specified in the rndc.conf file, using the controls statement in named.conf. Below are the procedures to follow before using rndc on your system.

Step 1
The first step will be to generate shared keys. This shared secret key will be included into /etc/rndc.conf file and /chroot/named/etc/named.conf file.

- To generate a random shared key, use the following command:
  ```
  [root@deep /]# dnssec-keygen -a hmac-md5 -b 128 -n user rndc
  Krndc.+157+36471
  ```

Step 2
The above command will generate a 128 bit (16 byte) HMAC-MD5 key and the result will be in a file called "Krndc.+157+36471.private" with a base-64 encoded string following the word "Key:", which must be extracted from the file and used as a shared secret.

- Edit the **Krndc.+157+36471.private** file (vi Krndc.+157+36471.private), and extract the base-64 encoded string:

  ```
  Private-key-format: v1.2
  Algorithm: 157 (HMAC_MD5)
  Key: 9kMjEIB5ikRJ6NSwtXWWVg==
  ```

The string "9kMjEIB5ikRJ6NSwtXWWVg==" in the above example is the part of this file that must be extracted and used as the shared secret.

Step 3
Once the required base-64 encoded string has been extracted from the generated file, we can remove the files from our system and copy the shared secret to both rndc.conf and named.conf files.

- To remove the generated files from the system, use the following commands:
  ```
  [root@deep /]# rm -f Krndc.+157+36471.key
  [root@deep /]# rm -f Krndc.+157+36471.private
  ```

Step 4
After that, we must edit the `rndc.conf` file and configure it with the key.

- Edit the **`rndc.conf`** file (`vi /etc/rndc.conf`), and add the following lines:

```
options {
        default-server  localhost;
        default-key     "localkey";
};

server localhost {
        key     "localkey";
};

key "localkey" {
        algorithm       hmac-md5;
        secret "9kMjEIB5ikRJ6NSwtXWWVg==";
};
```

In the above example, `rndc` will by default use the server at localhost (`127.0.0.1`) and the key called `localkey`. Commands to the localhost server will use the `localkey` key. The key statement indicates that `localkey` uses the `HMAC-MD5` algorithm and its secret clause contains the base-64 encoding of the `HMAC-MD5` secret enclosed in double quotes.

Step 5
Also don't forget to edit the `named.conf` file and configure it with the key.

- Edit the **`named.conf`** file (`vi /chroot/named/etc/named.conf`), and add the lines:

```
key ns1-ns2 {
  algorithm hmac-md5;
  secret "ps1jy3f7czVa1VNZkYaLfw==";
};

server 207.35.78.6 {
  keys { ns1-ns2 ;};
};

key localkey {
  algorithm hmac-md5;
  secret "9kMjEIB5ikRJ6NSwtXWWVg==";
};

controls {
  inet 127.0.0.1 allow { 127.0.0.1; } keys { localkey; };
};

options {
        directory "/var/named";
        allow-transfer { key ns1-ns2; };
        allow-query { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
```

```
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are the master server for openna.com
zone "openna.com" {
        type master;
        file "db.openna";
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
        type master;
        file "db.207.35.78";
        allow-query { any; };
};
```

In the above example, rndc connection will only be accepted at localhost (127.0.0.1).

Step 6
Finally, it is important to restart your DNS server for the changes to take effect.

- Restart ISC BIND & DNS with the following command:

```
[root@deep /]# /etc/rc.d/init.d/named restart
Shutting down named:                       [OK]
Starting named:                            [OK]
```

WARNING: Using the encryption algorithm for the name server control utility rndc doesn't work with the lwresd.conf file. It only work with named.conf file and not with lwresd.conf.

DNSSEC Cryptographic authentication of DNS information

This section applies only if you chose to install and use ISC BIND & DNS as a Master or Slave Name Server in your system. The BIND9 release of ISC BIND & DNS includes and support validation of DNSSEC (**DNS Sec**urity) signatures in responses but should still be considered experimental. The DNSSEC feature of BIND9 is used for signed zones, what DNSSEC do is to make sure that the DNS communication taking place is with the correct server, and that the information has not been tampered with during transport. This allows protection of Internet-wide DNS transfers, cache pollution, and will protect you from someone trying to spoof your DNS servers.

But be aware that DNSSEC is NOT for every kind of Name Server. DNSSEC verifies that the data received by a resolver is the same as the data published. For it to do anything, your resolver must be configured to verify data. Signing a localhost zone like for Caching-Only or Secondary/Slave Name Server is not useful, since it's not traveling over an insecure network. Signing data in general doesn't help you; it guarantees that anyone that gets data from your server can verify its correctness, if they've configured their resolver to do so.

Each zone (domain) in the DNS will need to have a key pair. The zone's public key will be included in its resource records. The zone's private key will be kept securely by the administrator of the zone, and never given to anyone outside your organization. Below, I show you all the required steps for the creation and use of DNSSEC signed zones.

In our example we assume that you want to use the DNSSEC feature for your Primary/Master Name Server with your parent zone (i.e. .COM) over the Internet. All commands listed below are assumed to be made in the **/chroot/named/var/named** directory since the DNSSEC tools require that the generated key files will be in the working directory.

Step 1
As usual in the cryptography area, the first step will be to generate a key pair. The generated zone keys here, will produce a private and public key to be used to sign records for the related zones in question and the zone keys must have the same name as the zone like in our example "openna.com". The resulting public keys should later be inserted into the related zone file with the **$INCLUDE** statements.

- To generate a 1024 bit DSA key for the openna.com zone, use the following command:
```
[root@deep /]# cd /chroot/named/var/named/
[root@deep named]# dnssec-keygen -a DSA -b 1024 -n ZONE openna.com
Kopenna.com.+003+28448
```

The above command will generate a 1024 bit DSA key for the openna.com zone and two output files will be produced: "Kopenna.com.+003+28448.key" and "Kopenna.com.+003+28448.private". The private key will be used to generate signatures, and the public key will be used for signature verification.

Step 2
Once the zone keys have been generated as shown previously, a keyset must be built and transmitted to the administrator of the parent zone in question to sign the keys with its own zone key. It is important that when building a keyset, at least the following information be included in the generation of the key: the TTL (**T**ime **T**o **L**ive) of the keyset must be specified, and the desired signature validity period of the parent's signature may also be specified.

- To generate a keyset containing the previous key, use the following command:
```
[root@deep named]# dnssec-makekeyset -t 3600 -e +864000 \
Kopenna.com.+003+28448
keyset-openna.com.
```

The above command generates a keyset containing the previous key with a TTL of 3600 and a signature validity period of 10 days (864000) starting from now to an output file called "keyset-openna.com.". This file should be transmitted to the parent to be signed. It includes the keys, as well as signatures over the keyset generated by the zone keys themselves, which are used to prove ownership of the private keys and encode the desired validity period.

Step 3
After that, the administrator on the parent zone (in our case .COM since our zone is openna.com) should receive the keyset files for each of your secure zones (in our example: keyset-openna.com.) and must sign the keys with its own private key. This is the step that permits others on the net to determine that the resource records that they receive from your zone are really from you.

- The administrator of your parent zone will sign the keyset with its zone keys by using something like the following command:

```
[root@internic named]# dnssec-signkey keyset-openna.com. \
KA.COM.+003+31877
signedkey-openna.com.
```

One output file called "signedkey-openna.com." will be produced. This file should be both transmitted back to the destinataire and retained. It will include all keys from the keyset file and signatures generated by this zone's zone keys.

WARNING: Take a note that in our example "KA.COM.+003+31877" is the key for the "A.COM" zone file, which is our parent zone. Olafur Gudmundsson <ogud@ogud.com> has informed me that .COM is not there yet, but what you SHOULD do is to contact your registrar and notify them that you MUST have your key set signed by .COM ASAP and when they expect that to happen. Verisign Global Registry has indicated that they want to start signing .COM sometime this year, but check with them what the current plans are.

To summarize our procedures :

- ✓ We have generated a key pair for our zone file in step 1.
- ✓ We have build and transmit a keyset to our parent zone for signing in step 2.
- ✓ Administrator in the parent zone signs our keyset with its private key.
- ✓ Administrator in the parent zone transmits back our ketset after singing it.

Step 4
Ok, from now if we recall what we said before is that the public keys should be inserted into the related zone file with the **$INCLUDE** statements, then at this step, we must insert the public key (Kopenna.com.+003+28448.key) into the related zone file, which is in our example the zone file called db.openna located under /chroot/named/var/named directory.

- Edit the **db.openna** zone file (vi /chroot/named/var/named/db.openna), and add the following line to your default zone file:

```
; Revision History: March 01, 2001 - root@openna.com
; Start of Authority (SOA) records.
$TTL 172800
@  IN  SOA  ns1.openna.com. root.openna.com.  (
                                        00      ; Serial
                                        10800   ; Refresh after 3 hours
                                        3600    ; Retry after 1 hour
                                        604800  ; Expire after 1 week
                                        172800 ); Minimum TTL of 1 day

$INCLUDE Kopenna.com.+003+28448.key

; Name Server (NS) records.
        IN      NS      ns1.openna.com.
        IN      NS      ns2.openna.com.

; Mail Exchange (MX) records.
        MX      0       smtp.openna.com.

; Address (A) records.
localhost       IN      A       127.0.0.1
router          IN      A       207.35.78.1
```

```
portal          IN      A       207.35.78.2
www             IN      A       207.35.78.3
smtp            IN      A       207.35.78.4
```

Don't forget to restart your DNS server for the change to take effect.

- Restart ISC BIND & DNS with the following command:
```
[root@deep /]# /etc/rc.d/init.d/named restart
Shutting down named:                       [OK]
Starting named:                            [OK]
```

NOTE: Please, check that everything looks right in your log files (/var/log/messages) before continuing with the step below. It is important to be sure that there is nothing wrong with your configuration.

Step 5
Once the keyset has been signed and approved by the parent zone (.COM), the final step will be to sign our zone. The result will produce one output file called "db.openna.signed". This file should be referenced by named.conf as the input file for the zone instead of the default one called "db.openna".

- To sign the zone file, use the following command:
```
[root@deep named]# dnssec-signzone -o openna.com db.openna
db.openna.signed
```

- One last requirement will be to change the owner of the db.openna.signed file to be the user under which our named daemon runs. In our case the named daemon runs under the user called "named":
```
[root@deep named]# chown named.named db.openna.signed
```

NOTE: If a zone doesn't publish a key, then BIND will accept any plausible-looking records, without a digital signature, just like in the original DNS. This provides compatibility with existing DNS zones, allowing Secure DNS to be gradually introduced throughout the Internet.

Step 6
The result of signing the zone will produce one output file called "db.openna.signed". Recall that this file should be referenced by named.conf as the input file for the zone.

- Edit the **named.conf** file (vi /chroot/named/etc/named.conf), and change the following line:

```
key ns1-ns2 {
  algorithm hmac-md5;
  secret "ps1jy3f7czVa1VNZkYaLfw==";
};

server 207.35.78.6 {
  keys { ns1-ns2 ;};
};
```

```
key localkey {
  algorithm hmac-md5;
  secret "JpWopXTHbRel32xLP9x7rg==";
};

controls {
  inet 127.0.0.1 allow { 127.0.0.1; } keys { localkey; };
};

options {
        directory "/var/named";
        allow-transfer { key ns1-ns2; };
        allow-query { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        allow-recursion { 192.168.1.0/24; 207.35.78.0/32; localhost; };
        version "Go away!";
};

logging {
         category lame-servers { null; };
};

// Root server hints
zone "." { type hint; file "db.cache"; };

// Provide a reverse mapping for the loopback address 127.0.0.1
zone "0.0.127.in-addr.arpa" {
        type master;
        file "db.127.0.0";
        notify no;
};

// We are the master server for openna.com
zone "openna.com" {
        type master;
        file "db.openna.signed";
        allow-query { any; };
};

zone "78.35.207.in-addr.arpa" {
        type master;
        file "db.207.35.78";
        allow-query { any; };
};
```

Step 7
Don't forget to restart your DNS server for the changes to take effect.

- Restart ISC BIND & DNS with the following command on both DNS servers:

```
[root@deep /]# /etc/rc.d/init.d/named restart
Shutting down named:                       [OK]
Starting named:                            [OK]
```

Optimizing ISC BIND & DNS

This section deals with actions we can make to further improve and tighten performance of ISC BIND & DNS. Note that we refer to the features available within the base installed program.

The BIND9 Lightweight Resolver

The new release of Bind comes with a new daemon program called "lwresd". The lwresd daemon is essentially a Caching-Only Name Server that answers requests using the lightweight resolver protocol rather than the DNS protocol. Because it needs to run on each host, it is designed to require no or minimal configuration. In our configuration we'll run lwresd in a chrooted environment.

On all Caching-Only Name Servers that you may have in your network, it can be interesting to run this daemon "lwresd" instead of the full "named" daemon. If we remember that a Caching-Only Name Server is not authoritative for any domains except 0.0.127.in-addr.arpa. It can look up names inside and outside your zone, as can Primary and Slave Name Servers but the difference is that when it initially looks up a name within your zone, it ends up asking one of the Primary or Slave Names Servers for your zone for the answer and nothing else. Therefore we can run the "lwresd" daemon in this kind of Name Server and everything will run, as we want.

Below, are the required steps to run your Caching-Only Name Server with the "lwresd" daemon instead of the "named" daemon in a chrooted environment.

Step 1
By default, the lwresd daemon listens on the loopback address (127.0.0.1). With a firewall on the system it is important to instruct the lwresd daemon to listen to the external interface of the server. This can be made with an "lwserver" statement lines in the /etc/resolv.conf file.

- Edit the **resolv.conf** file (vi /etc/resolv.conf), and add the following line:

```
lwserver 207.35.78.2
```

Where 207.35.78.2 is the IP address of the external interface in the firewall script file.

Step 2
Since lwresd will run in a chroot jail environment, we must copy the /etc/resolv.conf file to our chrooted environment for the lwresd daemon to be able to find the resolv.conf file and start.

- To copy the resolv.conf file to your chroot jail, use the following command:
  `[root@deep /]# cp /etc/resolv.conf /chroot/named/etc/`

Step 3
Now, we must create an initialization script file for the lwresd daemon to automatically start and stop on your server.

- Create the **lwresd** script file (touch /etc/rc.d/init.d/lwresd) and add the following lines inside it:

```
#!/bin/bash
#
# lwresd           This shell script takes care of starting and stopping
#                 lwresd (The lightweight resolver library).
#
# chkconfig: - 55 45
# description: lwresd is essentially a Caching-Only Name Server \
# that answers requests using the lightweight resolver \
# protocol rather than the DNS protocol.
# probe: true

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ "${NETWORKING}" = "no" ] && exit 0

[ -f /etc/sysconfig/named ] && . /etc/sysconfig/named

[ -f /usr/sbin/lwresd ] || exit 0

[ -f "${ROOTDIR}"/etc/resolv.conf ] || exit 0

RETVAL=0

start() {
        # Start daemons.
        echo -n "Starting lwresd: "
        if [ -n "${ROOTDIR}" -a "x${ROOTDIR}" != "x/" ]; then
                OPTIONS="${OPTIONS} -t ${ROOTDIR}"
        fi
        daemon lwresd -P 53 -u named ${OPTIONS}
        RETVAL=$?
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/lwresd
        echo
        return $RETVAL
}
stop() {
        # Stop daemons.
        echo -n "Shutting down lwresd: "
        killproc lwresd
        RETVAL=$?
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/lwresd
        echo
        return $RETVAL
}
restart() {
        stop
        start
}

# See how we were called.
case "$1" in
        start)
                start
                ;;
        stop)
                stop
                ;;
        restart)
```

```
                restart
                ;;
        *)
                echo "Usage: lwresd {start|stop|restart}"
                exit 1
esac

exit $?
```

Step 4
Once the `lwresd` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/lwresd
[root@deep /]# chown 0.0 /etc/rc.d/init.d/lwresd
```

- To create the symbolic `rc.d` links for `lwresd`, use the following commands:
```
[root@deep /]# chkconfig --add lwresd
[root@deep /]# chkconfig --level 2345 lwresd on
```

Step 5
Because we run `lwresd` instead of `named` daemon in our Caching-Only Name Server, it is important to deactivate and uninstall the `named` initialization script file in our system.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# chkconfig --del named
[root@deep /]# chkconfig --level 2345 named off
[root@deep /]# rm -f /etc/rc.d/init.d/named
```

Step 6
The `lwresd` daemon read its configuration file from `/etc/lwresd.conf`. This file is optional and the program can run without it and just with the `resolv.conf` file but it is preferable to create and use this configuration file with `lwresd` to reduce possible messages in the log file.

The format of `lwresd.conf` file is identical to `named.conf`. Therefore all you have to do is to rename your existing `named.conf` file configured for a Caching Name Server to become `lwresd.conf` file.

- This procedure can be accomplished with the following command:
```
[root@deep /]# cd /chroot/named/etc/
[root@deep etc]# mv named.conf lwresd.conf
```

Step 7
Now it is time to start your `DNS` server with the `lwresd` daemon.

- To start `lwressd` manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/lwresd start
Starting lwresd:                   [OK]
```

Further documentation

For more details, there are several manual pages you can read:

```
$ man named-checkconf (1)        - Configuration file syntax checking tool
$ man named-checkzone (1)        - Zone validity checking tool
$ man host (1)                   - DNS lookup utility
$ man dig (1)                    - DNS lookup utility
$ man rndc.conf (5)              - rndc configuration file
$ man named (8)                  - Internet domain name server
$ man rndc (8)                   - name server control utility
$ man lwresd (8)                 - lightweight resolver daemon
$ man nsupdate (8)               - Dynamic DNS update utility
```

ISC BIND & DNS Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages of ISC BIND & DNS and documentation for more information.

dig

The dig command DNS lookup utility (**d**omain **i**nformation **g**roper) is a tool for interrogating DNS name servers by performing DNS lookups and displays the answers that are returned from. It can also be used to update your db.cache file by telling your server where the servers for the "root" zone are. Dig is a useful tool to use when you want to troubleshoot DNS problems.

- Use the following command to query an address:

```
[root@deep /]# dig @www.openna.com

; <<>> DiG 9.1.0 <<>> @www.openna.com
;; global options:  printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 20994
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;.                               IN      NS

;; Query time: 3 msec
;; SERVER: 207.35.78.5#53(ns1.openna.com)
;; WHEN: Fri Feb 23 19:16:51 2001
;; MSG SIZE  rcvd: 17
```

Where **@www.openna.com** is the address of the server. Many options exist for this tool and I recommend you to read the dig manual page dig(1) for a complete list.

rndc

The rndc command utility allows the system administrator to control the operation of a name server. It replace the ndc(8) utility that was provided in old BIND8 releases. You can use this tool to reload configuration files and zones, schedule immediate maintenance for a zone, write server statistics, toggle query logging, stop the DNS server, and many other functions. The rndc tool prints a short summary of the supported commands and the available options if invoked on command line without options.

- Type `rndc` on your terminal to get a short summary of all available and supported commands:

```
[root@deep /]# rndc
Usage: rndc [-c config] [-s server] [-p port] [-y key] [-z zone] [-v
view]
        command [command ...]

command is one of the following:

  reload        Reload configuration file and zones.
  reload zone [class [view]]
                Reload a single zone.
  refresh zone [class [view]]
                Schedule immediate maintenance for a zone.
  stats         Write server statistics to the statistics file.
  querylog      Toggle query logging.
  dumpdb        Dump cache(s) to the dump file (named_dump.db).
  stop          Save pending updates to master files and stop the server.
  halt          Stop the server without saving pending updates.
  *status       Display ps(1) status of named.
  *trace        Increment debugging level by one.
  *notrace      Set debugging level to 0.
  *restart      Restart the server.

* == not yet implemented
Version: 9.1.0
```

ISC BIND & DNS Users Tools

The commands listed below are some that we often use, but many more exist. Check the manual pages of `ISC BIND & DNS` and documentation for more information.

nslookup

The `nslookup` program allows the user to query Internet domain name servers interactively or non-interactively. In interactive mode the user can query name servers for information about various hosts and domains, and print a list of hosts in a domain. In non-interactive mode the user can just print the name and request information for a host or domain.

Interactive mode has a lot of options and commands; it is recommended that you see the manual page for `nslookup`.

- To enter under `nslookup` Interactive mode, use the command:

```
[root@deep /]# nslookup
> www.openna.com
Server:         207.35.78.5
Address:        207.35.78.5#53

Name:   www.openna.com
Address: 207.35.78.3
> exit
```

- To run in non-interactive mode, use the command:

```
[root@deep /]# nslookup www.openna.com
Server:         207.35.78.5
Address:        207.35.78.5#53

Name:   www.openna.com
Address: 207.35.78.3
```

Where <www.openna.com> is the host name or Internet address of the name server to be looked up.

host

The host tool is a simple utility for performing DNS lookups. It is normally used to convert names to IP addresses and vice versa. When no arguments or options are given, host prints a short summary of its command line arguments and options.

- To print host command line arguments and options, use the command:

```
[root@deep /]# host
Usage: host [-aCdlrTwv] [-c class] [-n] [-N ndots] [-t type] [-W time]
            [-R number] hostname [server]
       -a is equivalent to -v -t *
       -c specifies query class for non-IN data
       -C compares SOA records on authorative nameservers
       -d is equivalent to -v
       -l lists all hosts in a domain, using AXFR
       -n Use the nibble form of IPv6 reverse lookup
       -N changes the number of dots allowed before root lookup is done
       -r disables recursive processing
       -R specifies number of retries for UDP packets
       -t specifies the query type
       -T enables TCP/IP mode
       -v enables verbose output
       -w specifies to wait forever for a reply
       -W specifies how long to wait for a reply
```

- To look up host names using the domain server, use the command:

```
[root@deep /]# host openna.com
openna.com. has address 207.35.78.3
```

List of installed ISC BIND & DNS files on your system

> /etc/rc.d/init.d/named
> /etc/sysconfig/named
> /etc/rndc.conf
> /usr/bin/dig
> /usr/bin/host
> /usr/bin/nslookup
> /usr/bin/nsupdate
> /usr/bin/isc-config.sh
> /usr/include/isc
> /usr/include/isc/assertions.h
> /usr/include/isc/base64.h
> /usr/include/isc/bitstring.h
> /usr/include/isc/boolean.h
> /usr/include/isc/buffer.h
> /usr/include/isc/bufferlist.h
> /usr/include/dns/journal.h
> /usr/include/dns/keyflags.h
> /usr/include/dns/keytable.h
> /usr/include/dns/keyvalues.h
> /usr/include/dns/lib.h
> /usr/include/dns/log.h
> /usr/include/dns/master.h
> /usr/include/dns/masterdump.h
> /usr/include/dns/message.h
> /usr/include/dns/name.h
> /usr/include/dns/namedconf.h
> /usr/include/dns/ncache.h
> /usr/include/dns/nxt.h
> /usr/include/dns/peer.h
> /usr/include/dns/rbt.h

> /usr/include/isc/commandline.h
> /usr/include/isc/entropy.h
> /usr/include/isc/error.h
> /usr/include/isc/event.h
> /usr/include/isc/eventclass.h
> /usr/include/isc/file.h
> /usr/include/isc/formatcheck.h
> /usr/include/isc/fsaccess.h
> /usr/include/isc/heap.h
> /usr/include/isc/hex.h
> /usr/include/isc/hmacmd5.h
> /usr/include/isc/interfaceiter.h
> /usr/include/isc/lang.h
> /usr/include/isc/lex.h
> /usr/include/isc/lfsr.h
> /usr/include/isc/lib.h
> /usr/include/isc/list.h
> /usr/include/isc/log.h
> /usr/include/isc/magic.h
> /usr/include/isc/md5.h
> /usr/include/isc/mem.h
> /usr/include/isc/msgcat.h
> /usr/include/isc/msgs.h
> /usr/include/isc/mutexblock.h
> /usr/include/isc/netaddr.h
> /usr/include/isc/ondestroy.h
> /usr/include/isc/os.h
> /usr/include/isc/print.h
> /usr/include/isc/quota.h
> /usr/include/isc/random.h
> /usr/include/isc/ratelimiter.h
> /usr/include/isc/refcount.h
> /usr/include/isc/region.h
> /usr/include/isc/resource.h
> /usr/include/isc/result.h
> /usr/include/isc/resultclass.h
> /usr/include/isc/rwlock.h
> /usr/include/isc/serial.h
> /usr/include/isc/sha1.h
> /usr/include/isc/sockaddr.h
> /usr/include/isc/socket.h
> /usr/include/isc/stdio.h
> /usr/include/isc/string.h
> /usr/include/isc/symtab.h
> /usr/include/isc/task.h
> /usr/include/isc/taskpool.h
> /usr/include/isc/timer.h
> /usr/include/isc/types.h
> /usr/include/isc/util.h
> /usr/include/isc/platform.h
> /usr/include/isc/app.h
> /usr/include/isc/dir.h
> /usr/include/isc/int.h
> /usr/include/isc/net.h
> /usr/include/isc/netdb.h
> /usr/include/isc/offset.h
> /usr/include/isc/stdtime.h
> /usr/include/isc/time.h
> /usr/include/isc/condition.h
> /usr/include/isc/mutex.h
> /usr/include/isc/once.h
> /usr/include/isc/thread.h
> /usr/include/dns
> /usr/include/dns/a6.h
> /usr/include/dns/acl.h
> /usr/include/dns/adb.h
> /usr/include/dns/byaddr.h
> /usr/include/dns/cache.h
> /usr/include/dns/callbacks.h
> /usr/include/dns/cert.h
> /usr/include/dns/rcode.h
> /usr/include/dns/rdata.h
> /usr/include/dns/rdataclass.h
> /usr/include/dns/rdatalist.h
> /usr/include/dns/rdataset.h
> /usr/include/dns/rdatasetiter.h
> /usr/include/dns/rdataslab.h
> /usr/include/dns/rdatatype.h
> /usr/include/dns/request.h
> /usr/include/dns/resolver.h
> /usr/include/dns/result.h
> /usr/include/dns/rootns.h
> /usr/include/dns/sdb.h
> /usr/include/dns/secalg.h
> /usr/include/dns/secproto.h
> /usr/include/dns/ssu.h
> /usr/include/dns/tcpmsg.h
> /usr/include/dns/time.h
> /usr/include/dns/tkey.h
> /usr/include/dns/tsig.h
> /usr/include/dns/ttl.h
> /usr/include/dns/types.h
> /usr/include/dns/validator.h
> /usr/include/dns/view.h
> /usr/include/dns/xfrin.h
> /usr/include/dns/zone.h
> /usr/include/dns/zt.h
> /usr/include/dns/enumclass.h
> /usr/include/dns/enumtype.h
> /usr/include/dns/rdatastruct.h
> /usr/include/dst
> /usr/include/dst/dst.h
> /usr/include/dst/lib.h
> /usr/include/dst/result.h
> /usr/include/lwres
> /usr/include/lwres/context.h
> /usr/include/lwres/lwbuffer.h
> /usr/include/lwres/lwpacket.h
> /usr/include/lwres/lwres.h
> /usr/include/lwres/result.h
> /usr/include/lwres/int.h
> /usr/include/lwres/lang.h
> /usr/include/lwres/list.h
> /usr/include/lwres/net.h
> /usr/include/lwres/ipv6.h
> /usr/include/lwres/netdb.h
> /usr/include/lwres/platform.h
> /usr/include/omapi
> /usr/include/omapi/compatibility.h
> /usr/include/omapi/lib.h
> /usr/include/omapi/omapi.h
> /usr/include/omapi/private.h
> /usr/include/omapi/result.h
> /usr/include/omapi/types.h
> /usr/lib/libisc.so.3.0.0
> /usr/lib/libisc.so.3
> /usr/lib/libisc.so
> /usr/lib/libisc.la
> /usr/lib/libisc.a
> /usr/lib/libdns.so.4.0.0
> /usr/lib/libdns.so.4
> /usr/lib/libdns.so
> /usr/lib/libdns.la
> /usr/lib/libdns.a
> /usr/lib/liblwres.so.1.1.0
> /usr/lib/liblwres.so.1
> /usr/lib/liblwres.so
> /usr/lib/liblwres.la
> /usr/lib/liblwres.a
> /usr/lib/libomapi.so.3.0.0

> /usr/include/dns/compress.h
> /usr/include/dns/confacl.h
> /usr/include/dns/confcache.h
> /usr/include/dns/confcommon.h
> /usr/include/dns/confctl.h
> /usr/include/dns/confctx.h
> /usr/include/dns/confip.h
> /usr/include/dns/confkeys.h
> /usr/include/dns/conflog.h
> /usr/include/dns/conflsn.h
> /usr/include/dns/conflwres.h
> /usr/include/dns/confparser.h
> /usr/include/dns/confresolv.h
> /usr/include/dns/confrrset.h
> /usr/include/dns/confview.h
> /usr/include/dns/confzone.h
> /usr/include/dns/db.h
> /usr/include/dns/dbiterator.h
> /usr/include/dns/dbtable.h
> /usr/include/dns/diff.h
> /usr/include/dns/dispatch.h
> /usr/include/dns/dnssec.h
> /usr/include/dns/events.h
> /usr/include/dns/fixedname.h
> /usr/lib/libomapi.so.3
> /usr/lib/libomapi.so
> /usr/lib/libomapi.la
> /usr/lib/libomapi.a
> /usr/sbin/named
> /usr/sbin/lwresd
> /usr/sbin/rndc
> /usr/sbin/dnssec-keygen
> /usr/sbin/dnssec-makekeyset
> /usr/sbin/dnssec-signkey
> /usr/sbin/dnssec-signzone
> /usr/sbin/named-checkconf
> /usr/sbin/named-checkzone
> /usr/share/man/man1/named-checkconf.1
> /usr/share/man/man1/named-checkzone.1
> /usr/share/man/man1/host.1
> /usr/share/man/man1/dig.1
> /usr/share/man/man5/rndc.conf.5
> /usr/share/man/man8/named.8
> /usr/share/man/man8/rndc.8
> /usr/share/man/man8/lwresd.8
> /usr/share/man/man8/nsupdate.8
> /var/named
> /var/run/named

Part VIII Mail Transfer Agent Related Reference

In this Part

Mail Transfer Agent - `Sendmail`
Mail Transfer Agent - `qmail`

Here we come to the part where we'll talk about mail and the necessity of having a mail server installed on our secure Linux server. On every kind of machine that runs a Unix operating system it's necessary and NOT optional to have a mail server. Even if you don't set-up your system to send or receive mail for users, you'll always have possible log messages that need to be delivered to root user, postmaster, daemons program, etc. Here is where a mail server is vital or you may lose some important messages like errors, attacks, intrusions etc, if you decide to not install a mail server on your system. The next two chapters of this book will deal extensively with mail transport agents you may want to install. We will begin our reading with `Sendmail` and finish with `qmail` software. It's yours to choose which MTA you prefer to use, `Sendmail` or `qmail`.

There is one question that most of you will ask often and this question is:
What are the differences between `Sendmail` and `qmail`?
Why should I use `Sendmail` instead of `qmail` or `qmail` instead of `Sendmail`?

You have to decide which features you want/need and then select the appropriate MTA. Some of you can decide to use `Sendmail` because of the many features and the support for most e-mail related RFCs.

For example, quoting the operations guide:
`Sendmail` is based on RFC821 (Simple Mail Transport Protocol), RFC822 (Internet Mail Headers Format), RFC974 (MX routing), RFC1123 (Internet Host Requirements), RFC2045 (MIME), RFC1869 (SMTP Service Extensions), RFC1652 (SMTP 8BITMIME Extension), RFC1870 (SMTP SIZE Extension), RFC1891 (SMTP Delivery Status Notifications), RFC1892 (Multipart/Report), RFC1893 (Mail System Status Codes), RFC1894 (Delivery Status Notifications), RFC1985 (SMTP Service Extension for Remote Message Queue Starting), RFC2033 (Local Message Transmission Protocol), RFC2034 (SMTP Service Extension for Returning Enhanced Error Codes), RFC2476 (Message Submission), RFC2487 (SMTP Service Extension for Secure SMTP over TLS), and RFC2554 (SMTP Service Extension for Authentication).

In the other part and from the point of view of security consider this:
With `Sendmail` the entire `sendmail` system is setuid and with `qmail` only one `qmail` program is setuid: `qmail-queue`. Its only purpose is to add a new mail message to the outgoing queue. Also five of the most important `qmail` programs are not security-critical. Even if all of these programs are completely compromised, so that an intruder has full control over the program accounts and the mail queue, he still can't take over your system. Finally, the stralloc concept and getln() of `qmail` which comes from a basic C library make it very easy to avoid buffer overruns, memory leaks, and artificial line length limits. `qmail` is based on RFC 822, RFC 1123, RFC 821, RFC 1651, RFC 1652, RFC 1854, RFC 1893, RFC 974, RFC 1939.

20 Mail Transfer Agent - Sendmail

In this Chapter

Linux Sendmail Mail Transfer Agent Server

Abstract

The Sendmail program is one of the most widely used Internet **M**ail **T**ransfer **A**gents (MTAs) in the world. The purpose of an MTA is to send mail from one machine to another, and nothing else. Sendmail is not a client program, which you use to read your e-mail. Instead, it actually moves your email over networks, or the Internet, to where you want it to go. Sendmail has been an easy target for system crackers to exploit in the past, but with the advent of Sendmail version 8, this has become much more difficult.

In our configuration and installation we'll provide you with two different configurations that you can set up for Sendmail; One for a Central Mail Hub Relay, and another for the local or neighbor clients and servers. We'll use the m4 macro of Linux to generate all ".mc" configuration files of Sendmail, since that makes maintenance much easier for people who don't understand sendmail re-write rules.

The Central Mail Hub Relay Server configuration will be used for your server where the assigned task is to send, receive and relay all mail for all local or neighbor client and server mail machines you may have on your network. A local or neighbor client and server refers to all other local server or client machines on your network that run Sendmail and send all outgoing mail to the Central Mail Hub for future delivery. You can configure the neighbor Sendmail so that it accepts only mail that is generated locally, thus insulating neighbor machines for easier security. This kind of internal client never receives mail directly via the Internet; Instead, all mail from the Internet for those computers is kept on the Mail Hub server. It is a good idea to run at least one Central Mail Hub Server for all computers on your network; this architecture will limit the management task on the server and client machines, and improve the security of your site.

If you decide to install and use Sendmail as your Central Mail Hub Server, it will be important to refer to the part that talk about **I**nternet **M**essage **A**ccess **P**rotocol in this book. Recall that Sendmail is just a program to send and receive mail and cannot be used to read mail. Therefore in a Central Mail Hub environment you need to have a program which allows users to connect to the Sendmail Mail Hub to get and read their mail, this is where a program like UW IMAP also know as a **I**nternet **M**essage **A**ccess **P**rotocol (IMAP) or **P**ost **O**ffice **P**rotocol (POP) is required and must be installed if you run Sendmail as your Mail Hub Server and only in this case. If you run Sendmail as a standalone local/neighbor client Mail Server, then you don't need to install a Internet Message Access Protocol like UW IMAP. If you decide to skip this chapter about Sendmail because you'd prefer to install qmail as your MTA, then you don't need to install UW IMAP even if you configure qmail as a Mail Hub Server since qmail already come with its own fast, small and secure POP program know as qmail-popd3.

Disclaimer

PLEASE REMEMBER THAT EXPORT/IMPORT AND/OR USE OF STRONG CRYPTOGRAPHY SOFTWARE, PROVIDING CRYPTOGRAPHY HOOKS OR EVEN JUST COMMUNICATING TECHNICAL DETAILS ABOUT CRYPTOGRAPHY SOFTWARE IS ILLEGAL IN SOME PARTS OF THE WORLD. SO, WHEN YOU IMPORT THIS PACKAGE TO YOUR COUNTRY, RE-DISTRIBUTE IT FROM THERE OR EVEN JUST EMAIL TECHNICAL SUGGESTIONS OR EVEN SOURCE PATCHES TO THE AUTHOR OR OTHER PEOPLE YOU ARE STRONGLY ADVISED TO PAY CLOSE ATTENTION TO ANY EXPORT/IMPORT AND/OR USE LAWS WHICH APPLY TO YOU. THE AUTHORS ARE NOT LIABLE FOR ANY VIOLATIONS YOU MAKE HERE. SO BE CAREFUL, IT IS YOUR RESPONSIBILITY.

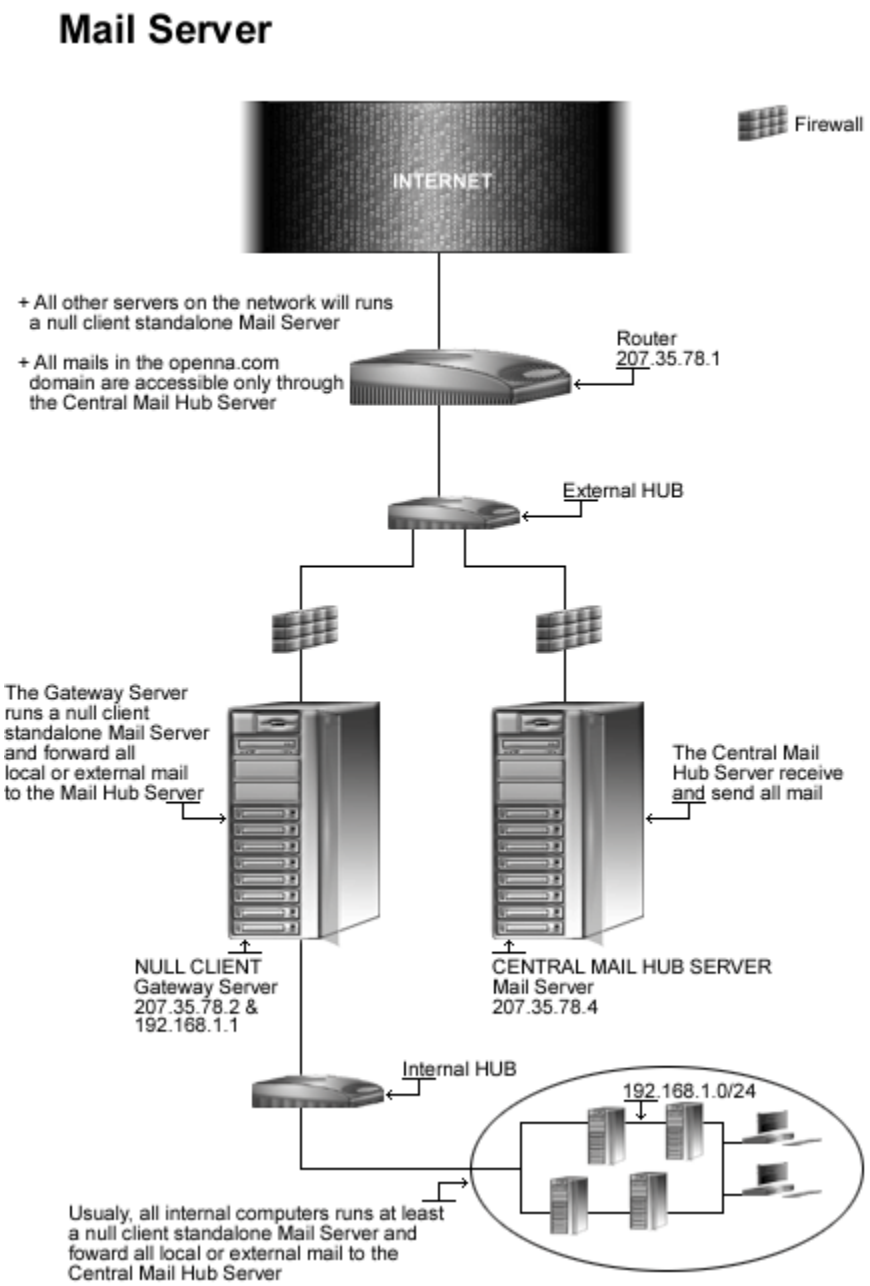
Recommended RPM packages to be installed for a Mail Server

A minimal configuration provides the basic set of packages required by the Linux operating system. Minimal configuration is a perfect starting point for building secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a Mail Server (SMTP) running on Sendmail software.

This configuration assumes that your kernel is a monolithic kernel. Also I assume that you will install Sendmail by it's RPM package. Therefore, sendmail RPM package is already included in the list below as you can see. Since IMAP is directly related to Sendmail software and especially in a Mail Hub Server environment, it is also included in the list below as well as xinetd, which allow imapd and popd protocols to run on the system. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

basesystem	ed	kernel	openssl	slocate
bash	file	less	pam	sysklogd
bdflush	filesystem	libstdc++	passwd	syslinux
bind	fileutils	libtermcap	popt	SysVinit
bzip2	findutils	lilo	**procmail**	tar
chkconfig	gawk	logrotate	procps	termcap
console-tools	gdbm	losetup	psmisc	textutils
cpio	gettext	**mailx**	pwdb	tmpwatch
cracklib	glib	MAKEDEV	**quota**	utempter
cracklib-dicts	glibc	man	readline	util-linux
crontabs	glibc-common	mingetty	rootfiles	vim-common
db1	grep	mktemp	rpm	vim-minimal
db2	groff	mount	sed	vixie-cron
db3	gzip	ncurses	**sendmail**	words
dev	**imap**	net-tools	setup	which
devfsd	info	newt	sh-utils	**xinetd**
diffutils	initscripts	openssh	shadow-utils	zlib
e2fsprogs	iptables	openssh-server	slang	

Tested and fully functional on OpenNA.com.



This is a graphical representation of the `Mail` Server configuration we use in this book. We try to show you different settings (Central Mail Hub Relay, and local or neighbor null client server) on different servers. Lots of possibilities exist, and depend on your needs and network architecture.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest Sendmail version number is 8.11.4

Packages

The following are based on information as listed by Sendmail as of 2001/05/25. Please regularly check at www.sendmail.org for the latest status.

Source code is available from:
Sendmail Homepage: http://www.sendmail.org/
Sendmail FTP Site: 209.246.26.20
You must be sure to download: sendmail.8.11.4.tar.gz

Prerequisites

Sendmail requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive file. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ OpenSSL, which enables support for SSL functionality, must already be installed on your system if you want to run Sendmail with SSL features.

Pristine source

If you don’t use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Sendmail, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > Sendmail1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > Sendmail2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff Sendmail1 Sendmail2 > Sendmail-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing Sendmail

Below are the required steps that you must make to compile and optimize the Sendmail software before installing it into your Linux system. Contrary to the majority of programs that we have already installed in this book, you'll find further down in this chapter a difference with the method used to compile and install this program.

Sendmail use a different procedure to install in the system, instead of using the default GNU autoconf build like many open source program use, it go with a script named Build which allow it to compile an appropriate Makefile for your specific system, and create an appropriate obj.* subdirectory before installing on the system. This method allows the program to works easily with build of multiplatform support.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp sendmail-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf sendmail-version.tar.gz
```

Step 2
In order to check that the version of Sendmail, which you are going to install, is an original and unmodified one, use the commands described below and check the supplied signature.

- To verify the MD5 checksum of Sendmail, use the following command:

```
[root@deep tmp]# md5sum sendmail-version.tar.gz
```

This should yield an output similar to this:

```
5e224eeb0aab63b7c178728ae42f26a5 sendmail.8.11.4.tar.gz
```

Now check that this checksum is exactly the same as the one published on the Sendmail website at the following URL: http://www.sendmail.org/8.11.html

Step 3
We must create a special user called mailnull, which will be the default UID for running mailers. Sendmail does a getpwnam() on mailnull during startup, and if that's defined in /etc/passwd file of Linux, it uses that UID:GID. In addition, if Sendmail sees that it's about to do something as root, it does it as this special user (mailnull) instead - so that if root has a .forward file executing a program, that simply will not run as root.

- To create this special Sendmail user, use the following command:

```
[root@deep tmp]# useradd -u 47 -d /var/spool/mqueue -r -s /bin/false
mailnull >/dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID.

Step 4
After that, move into the newly created `Sendmail` directory and perform the following steps before compiling and optimizing it. The modifications we bring to the `Sendmail` file below are necessary to be compliant with our Linux file system structure.

- To move into the newly created `Sendmail` directory, use the following command:
  `[root@deep tmp]#` **`cd sendmail-8.11.4/`**

Step 4.1
The file that we must modify is named **`smrsh.c`** located under the source directory of `Sendmail`. In this file, we will specify the directory in which all "`smrsh`" program commands must reside.

- Edit the **`smrsh.c`** file (`vi +80 smrsh/smrsh.c`) and change the line:

```
# else /* HPUX10 || HPUX11 || SOLARIS >= 20800 */
#  define CMDDIR        "/usr/adm/sm.bin"
```

To read:

```
# else /* HPUX10 || HPUX11 || SOLARIS >= 20800 */
#  define CMDDIR        "/etc/smrsh"
```

Step 4.2
The last modification to this file (`smrsh.c`) will be to specify the default search path for commands runs by the "`smrsh`" program. It allows us to limit the location where these programs may reside.

- Edit the **`smrsh.c`** file (`vi +89 smrsh/smrsh.c`) and change the line:

```
# define PATH          "/bin:/usr/bin:/usr/ucb"
```

To read:

```
# define PATH           "/bin:/usr/bin"
```

Step 4.3
Finally, edit the **`daemon.c`** file located under `sendmail` subdirectory of the `Sendmail` source archive and modify it as follow:

- Edit the **`daemon.c`** file (`vi +2765 sendmail/daemon.c`) and change the line:

```
/* get result */
p = &ibuf[0];
nleft = sizeof ibuf - 1;
while ((i = read(s, p, nleft)) > 0)
{
```

To read:

```
/* get result */
p = &ibuf[0];
nleft = sizeof (ibuf) - 1;
while ((i = read(s, p, nleft)) > 0)
{
```

Step 5
The Build script of Sendmail by default uses an operating system file that corresponds to your operating system type to get information about definitions for system installation and various compilation values. This file is located under the subdirectory named 'devtools/OS' of the Sendmail source archive and, if you're running a Linux system, it'll be named 'Linux'.

We'll rebuild and recreate this operating system file to suit our Linux system installation features and put it in the default devtools/OS subdirectory of the Sendmail source distribution, since the Build script will look for the default operating system file in this directory during compile time of Sendmail. In summary, the operating system file is read by Sendmail to get default information about how to compile the program for your specific system.

- Edit the **Linux** file (vi devtools/OS/Linux), and remove all predefined lines then add the following lines inside the file.

```
define(`confDEPEND_TYPE', `CC-M')
define(`confMANROOT', `/usr/share/man/man')
define(`confLIBS', `-ldl')
define(`confEBINDIR', `/usr/sbin')
define(`confLD', `ld')
define(`confMTLDOPTS', `-lpthread')
define(`confLDOPTS_SO', `-shared')
define(`confSONAME',`-soname')
```

This tells the Linux file to set itself up for this particular configuration with:

`define(`confDEPEND_TYPE', `CC-M')`
This macro option specifies how to build dependencies with Sendmail.

`define(`confMANROOT', `/usr/share/man/man')`
This macro option defines the location to install the Sendmail manual pages.

`define(`confLIBS', `-ldl')`
This macro option defines the -l flags passed to ld for selecting libraries during linking. Recall that ld is the GNU linker program which is used in the last step of a compiled program to link standard Unix object files on a standard, supported Unix system.

`define(`confEBINDIR', `/usr/sbin')`
This macro option defines where to install binaries executed from other binaries. On Linux the path must be set to the /usr/sbin directory.

`define(`confLD', `ld')`
This macro option simply defines the linker program to use with Sendmail program. In our case the linker we use is named ld.

`define(`confMTLDOPTS', `-lpthread')`
This macro option defines the additional linker options to use for linking multithread binaries with Sendmail program.

`define(`confLDOPTS_SO', `-shared')`
This macro option defines the additional linker options to use for linking shared object libraries with Sendmail program.

`define(`confSONAME', `-soname')`
This macro option defines the `ld` flag to use for recording the shared object name into shared object with `Sendmail` program.

Step 6
The `Build` script of `Sendmail` can use a default local configuration file specific to your operating system type to get additional information about various compilation values. `Sendmail` will include the following file if it is present in the subdirectory named 'devtools/Site'. The proper way of doing local configuration is by creating a file 'site.config.m4' in the directory 'devtools/Site'.

To summarize, if the local configuration file 'site.config.m4' exists under the subdirectory 'devtools/Site', then `Sendmail` will read and include it in its compilation to get additional information about how to compile and adjust various parts of the program for your specific system.

- Create the **site.config.m4** file (`touch devtools/Site/site.config.m4`), and add the following lines inside the file.

```
define(`confMAPDEF', `-DMAP_REGEX')
define(`confENVDEF', `-DPICKY_QF_NAME_CHECK -DXDEBUG=0')
define(`confCC', `gcc')
define(`confOPTIMIZE', `-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer')
define(`confNO_HELPFILE_INSTALL')
```

This tells the site.config.m4 file to set itself up for this particular configuration with:

`define(`confMAPDEF', `-DMAP_REGEX')`
This macro option specifies the database type to be included with `Sendmail`. The "`-DMAP_REGEX`" argument enables regular expression support for the operating system. Note that the new Berkeley DB package (`NEWDB`); `NEWDB` is the one we need is included automatically by the Build script of `Sendmail` during compile time. Therefore we don't need to include it in our definition.

`define(`confENVDEF', `-DMAP_REGEX -DPICKY_QF_NAME_CHECK -DXDEBUG=0')`
This macro option is used primarily to specify other environment information and code that should either be specially included or excluded. With "`-DPICKY_QF_NAME_CHECK`" defined, `Sendmail` will log an error if the name of the "`qf`" file is incorrectly formed and will rename the "`qf`" file into a "`Qf`" file. The "`-DXDEBUG=0` " argument disables the step of additional internal checking during compile time.

`define(`confCC', `gcc')`
This macro option defines the C compiler to use for compilation of `Sendmail`. In our case we use the default Linux "`gcc`" C compiler for better optimization.

`define(`confOPTIMIZE', `-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer')`
This macro option defines the flags passed to CC for optimization of the program related to our specific CPU processor architecture. In conjunction to the flags used here, this program will be compiled to run on an i686 CPU architecture and above. This is also where we'll get approximately 30% more speed for our `Sendmail` program comparatively to the majority of available `Sendmail` packages on the Internet.

```
define(`confNO_HELPFILE_INSTALL')
```

This macro option specifies to not install the Sendmail help file by default. Some experienced administrators recommend it, for better security.

WARNING: To enable SSL support with Sendmail, refer later in the section called "Running Sendmail with SSL support" for the appropriate instructions before continuing. This is important.

Step 7

Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install Sendmail in the server.

```
[root@deep sendmail-8.11.4]# cd sendmail/
[root@deep sendmail]# sh Build
[root@deep sendmail]# cd ../mailstats/
[root@deep mailstats]# sh Build
[root@deep mailstats]# cd ../smrsh/
[root@deep smrsh]# sh Build
[root@deep smrsh]# cd ../makemap/ (Required only for Mail Hub configuration)
[root@deep makemap]# sh Build (Required only for Mail Hub configuration)
[root@deep makemap]# cd ../praliases/ (Required only for Mail Hub configuration)
[root@deep praliases]# sh Build (Required only for Mail Hub configuration)
[root@deep praliases]# cd ..
[root@deep sendmail-8.11.4]# cd
[root@deep /root]# find /* > Sendmail1
[root@deep /root]# cd /var/tmp/sendmail-8.11.4/
[root@deep sendmail-8.11.4]# cd sendmail/
[root@deep sendmail]# sh Build install
[root@deep sendmail]# cd ../mailstats/
[root@deep mailstats]# sh Build install
[root@deep mailstats]# cd ../smrsh/
[root@deep smrsh]# sh Build install
[root@deep smrsh]# cd ../makemap/ (Required only for Mail Hub configuration)
[root@deep makemap]# sh Build install (Required only for Mail Hub configuration)
[root@deep makemap]# cd ../praliases/ (Required only for Mail Hub configuration)
[root@deep praliases]# sh Build install (Required only for Mail Hub configuration)
[root@deep praliases]# cd ..
[root@deep sendmail-8.11.4]# ln -fs /usr/sbin/sendmail /usr/lib/sendmail
[root@deep sendmail-8.11.4]# chmod 511 /usr/sbin/smrsh
[root@deep sendmail-8.11.4]# install -d -m700 /var/spool/mqueue
[root@deep sendmail-8.11.4]# chown 0.mail /var/spool/mail/
[root@deep sendmail-8.11.4]# chmod 1777 /var/spool/mail/
[root@deep sendmail-8.11.4]# mkdir /etc/smrsh
[root@deep sendmail-8.11.4]# strip /usr/sbin/sendmail
[root@deep sendmail-8.11.4]# cd
[root@deep /root]# find /* > Sendmail2
[root@deep /root]# diff Sendmail1 Sendmail2 > Sendmail-Installed
```

The **sh Build** command would build and make the necessary dependencies for the different binary files required by Sendmail before installation. The **sh Build install** command would install sendmail, mailstats, makemap, praliases, smrsh binaries as well as the corresponding manual pages on your system if compiled with this command. The **ln -fs** command would make a symbolic link of the sendmail binary to the /usr/lib directory. This is required, since some programs expect to find the sendmail binary in this directory (/usr/lib).

The **install** command will create the directory "mqueue" with permission 700 under /var/spool. A mail message can be temporarily undeliverable for a wide variety of reasons. To ensure that such messages are eventually delivered, Sendmail stores them in its queue directory until they can be delivered successfully. The **mkdir** command would create the /etc/smrsh directory on your system. This directory is where we'll put all program mailers that we allow Sendmail to be able to run.

WARNING: The programs "makemap", and "praliases" must only be installed on the Central Mail Hub Server". The "makemap" utility allow you to create and regenerate a database map like the /etc/mail/aliases.db or /etc/mail/access.db files, for Sendmail. The "praliases" display the system mail aliases (the content of /etc/mail/aliases file). Since it is better to only have one place (like our Central Mail Hub) to handle and manage all the db files in our network, then it is not necessary to use the "makemap", and "praliases" programs and build db files on your other hosts in the network.

Configuring Sendmail

After Sendmail has been built and installed successfully on your system, your next step is to configure and customize all the options into your different Sendmail configuration files. Depending of the kind of Mail server you want to run in your Linux server, there are different configuration files to set up, those files are:

For running Sendmail as a Central Mail Hub Server:

- /etc/mail/sendmail.mc (The Sendmail Macro Configuration File)
- /etc/mail/access (The Sendmail access Configuration File)
- /etc/mail/access.db (The Sendmail access DB Hash Table)
- /etc/mail/relay-domains (The Sendmail Relay Configuration File)
- /etc/mail/aliases (The Sendmail aliases Configuration File)
- /etc/mail/aliases.db (The Sendmail aliases DB Hash Table)
- /etc/mail/virtusertable (The Sendmail virtusertable Configuration File)
- /etc/mail/virtusertable.db (The Sendmail virtusertable DB Hash Table)
- /etc/mail/domaintable (The Sendmail domaintable Configuration File)
- /etc/mail/domaintable.db (The Sendmail domaintable DB Hash Table)
- /etc/mail/mailertable (The Sendmail mailertable Configuration File)
- /etc/mail/mailertable.db (The Sendmail mailertable DB Hash Table File)
- /etc/mail/local-host-names (The Sendmail Local Host Configuration File)
- /etc/sysconfig/sendmail (The Sendmail System Configuration File)
- /etc/rc.d/init.d/sendmail (The Sendmail Initialization File)

For running Sendmail as a Standalone Mail Server:

- /etc/mail/null.mc (The Sendmail null client Macro Configuration File)
- /etc/mail/local-host-names (The Sendmail Local Host Configuration File)
- /etc/sysconfig/sendmail (The Sendmail System Configuration File)
- /etc/rc.d/init.d/sendmail (The Sendmail Initialization File)

/etc/mail/sendmail.mc: The Sendmail Macro Configuration File

This section applies only if you chose to install and use Sendmail as a Central Mail Hub Server in your system. Instead of having each individual server or workstation in a network handle its own mail, it can be advantageous to have powerful central server that handles all mail. Such a server is called a Mail Hub. The advantage of a Central Mail Hub is:

- ✓ All incoming mail is sent to the Mail Hub, and no mail is sent directly to a client machine.
- ✓ All outgoing mail from clients is sent to the Mail Hub, and the Hub then forwards that mail to its ultimate destination.
- ✓ All outgoing mail appears to come from a single server and no client's name needs to be known to the outside world.
- ✓ No client needs to run a sendmail daemon to listen for mail.

Step 1
The "sendmail.cf" is the first configuration file reading by Sendmail when it runs and one of the most important for Sendmail. Among the many items contained in that file are the locations of all the other files, the default permissions for those files and directories that Sendmail needs. The m4 macro preprocessor program of Linux is used by Sendmail V8 to produce a Sendmail configuration file. This macro program will produce the /etc/mail/sendmail.cf configuration file by processing a file whose name ends in ".mc".

For this reason, we'll create this file (sendmail.mc) and put the necessary macro values in it to allow the m4 program to process (read) its input and gather definitions of macros, and then replaces those macros with their values and output the result to create our "sendmail.cf" file. Please refer to the Sendmail documentation and README file under the "cf" subdirectory of the V8 Sendmail source distribution for more information. We must change the sendmail.mc macro configuration file below to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **sendmail.mc** file (touch /etc/mail/sendmail.mc) and add the lines:

```
VERSIONID(`linux setup for LINUX OpenNA Boreas')dnl
OSTYPE(`linux')dnl
DOMAIN(`generic')dnl
define(`confTRY_NULL_MX_LIST',true)dnl
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')dnl
define(`confPRIVACY_FLAGS',
`authwarnings,goaway,restrictmailq,restrictqrun')dnl
define(`confSAFE_FILE_ENV',`/home')dnl
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
FEATURE(`mailertable',`hash -o /etc/mail/mailertable')dnl
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable')dnl
FEATURE(`redirect')dnl
FEATURE(`always_add_domain')dnl
FEATURE(`relay_hosts_only')dnl
FEATURE(`use_cw_file')dnl
FEATURE(`local_procmail')dnl
FEATURE(`access_db')dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`dnsbl')dnl
MAILER(`local')dnl
MAILER(`smtp')dnl
MAILER(`procmail')dnl
```

This tells the `sendmail.mc` file to set itself up for this particular configuration with:

`OSTYPE(`linux')dnl`
This configuration option specifies the default operating system `Sendmail` will be running on; in our case the "`linux`" operating system. This item is one of the minimum pieces of information required by the "`mc`" file.

`DOMAIN(`generic')dnl`
This configuration option will specify and describe a particular domain appropriate for your environment.

`define(`confTRY_NULL_MX_LIST',true)dnl`
This configuration option specifies whether the receiving server is the best MX for a host and if so, try connecting to that host directly. In our configuration we say yes (`true`) to this option.

`define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')dnl`
This configuration option sets the path to the `procmail` program installed in your server. Since the path in Red Hat Linux differs from other Linux versions, we must specify the new path with this macro. It's important to note that this macro is also used by the option `FEATURE(`local_procmail')` as defined later in this file.

`define(`confPRIVACY_FLAGS',`authwarnings,goaway,restrictmailq,restrictqrun')dnl`
This configuration option is one of the most important for the security of `Sendmail`. Setting the "`goaway`" option causes `Sendmail` to disallow all `SMTP "EXPN"` commands, it also causes it to reject all `SMTP "VERB"` commands and to disallow all `SMTP "VRFY"` commands. These changes prevent spammers from using the `"EXPN"` and `"VRFY"` commands in `Sendmail`. Ordinarily, anyone may examine the mail queue's contents by using the "`mailq`" command. To restrict who may examine the queue's contents, you must specify the "`restrictmailq`" option as shown above. With this option, `Sendmail` allows only users who are in the same group as the group ownership of the queue directory (`root`) to examine the contents. This allows the queue directory to be fully protected with mode `0700`, while selected users are still able to see the contents. Ordinarily, anyone may process the queue with the "`-q`" switch. To limit queue processing to "`root`" and the owner of the queue directory, you must specify the "`restrictqrun`" option too.

`define(`confSAFE_FILE_ENV',`/home')dnl`
This configuration option limit where in the file system, mailbox files can be written. It does a chroot into the specified portion of the file system and adds some other restrictions. In this example, we restrict and let people write only in their home directories with the value "`/home`".

`FEATURE(`smrsh',`/usr/sbin/smrsh')dnl`
This `m4` macro enables the use of "`smrsh`" (the **s**end**m**ail **r**estricted **sh**ell) instead of the default `/bin/sh` for mailing programs to provide increased security control. With this feature you can control what program gets run via e-mail through the `/etc/mail/aliases` and `~/.forward` files. The default location for the "`smrsh`" program is `/usr/libexec/smrsh`; since we have installed "`smrsh`" in another location, we need to add an argument to the `smrsh` feature to indicate the new placement `/usr/sbin/smrsh`. The use of "`smrsh`" is recommended by CERT, so you are encouraged to use this feature as often as possible.

FEATURE(`mailertable')dnl
This m4 macro enables the use of "mailertable" (database selects new delivery agents). A mailertable is a database that maps "host.domain" names to special delivery agent and new domain name pairs. With this feature, mail can be delivered through the use of a specified or particular delivery agent to a new domain name. Usually, this feature must be available only on a Central Mail Hub server.

FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable')dnl
This m4 macro enables the use of "virtusertable" (support for virtual domains), which allow multiple virtual domains to be hosted on one machine. A virtusertable is a database that maps virtual domains into new addresses. With this feature, mail for virtual domains can be delivered to a local, remote, or single user address. Usually this feature must be available only on a Central Mail Hub server.

FEATURE(`redirect')dnl
This m4 macro enables the use of "redirect" (support for address.REDIRECT). With this feature, mail addressed to a retired user account "gmourani", for example, will be bounced with an indication of the new forwarding address. The retired accounts must be set up in the aliases file on the mail server. Usually this feature must be available only on a Central Mail Hub server.

FEATURE(`always_add_domain')dnl
This m4 macro enables the use of "always_add_domain" (add the local domain even on local mail). With this feature, all addresses that are locally delivered will be fully qualified. It is safe and recommended to set this feature for security reasons.

FEATURE(`relay_hosts_only')dnl
This m4 macro enables the use of "relay_hosts_only". Normally domains are listed in /etc/mail/relay-domains and any domain names listed in this file are accepted for relaying. With this feature, each host in a domain must be listed. This mean that domain name like "openna.com" listed in this file is not enough to be accepted for relaying and you must add the host name like "host1.openna.com" to the domain name for the system to accept relaying.

FEATURE(`use_cw_file')dnl
This m4 macro enables the use of "use_cw_file" (use /etc/mail/local-host-names file for local hostnames). With this feature you can declare a list of hosts in the /etc/mail/local-host-names file for which the local host is acting as the MX recipient. In other word this feature causes the file /etc/mail/local-host-names to be read to obtain alternative names for the local host.

FEATURE(`local_procmail')dnl
This m4 macro enables the use of "local_procmail" (use procmail as local delivery agent). With this feature you can use procmail as a Sendmail delivery agent.

FEATURE(`access_db')dnl
This m4 macro enables the access database feature. With this feature you have the ability through the access db to allow or refuse to accept mail from specified domains. Usually this feature must be available only in a Central Mail Hub server.

FEATURE(`blacklist_recipients')dnl
This m4 macro enables the ability to block incoming mail for certain recipient usernames, hostnames, or addresses. With this feature you can, for example, block incoming mail to user nobody, host foo.mydomain.com, or guest@bar.mydomain.com.

`FEATURE(`dnsbl')dnl`
This `m4` macro enables `Sendmail` to automatically reject mail from any site in the Realtime Blackhole List database "`rbl.maps.vix.com`". The DNS based rejection is a database maintained in DNS of spammers. For details, see "`http://maps.vix.com/rbl/`".

`MAILER(`local'), MAILER(`smtp'), and MAILER(`procmail')dnl`
This `m4` macro enables the use of "`local`", "`smtp`", and "`procmail`" as delivery agents (in `Sendmail` by default, delivery agents are not automatically declared). With this feature, you can specify which ones you want to support and which ones to ignore. The `MAILER(`local')`, `MAILER(`smtp')`, and `MAILER(`procmail')` options cause support for `local`, `smtp`, `esmtp`, `smtp8`, relay delivery agents and `procmail` to be included. It's important to note that `MAILER(`smtp')` should always precede `MAILER(`procmail')`.

WARNING: Sometimes, a domain with which you wish to continue communications may end up in the RBL list. In this case, `Sendmail` allows you to override these domains to allow their e-mail to be received. To do this, simply edit the `/etc/mail/access` file and add the appropriate domain information.

For example:
```
blacklisted.domain      OK
```

Step 2
Now that our macro configuration file "`sendmail.mc`" is configured and created to correspond to our specific needs, we can build the `Sendmail` configuration file "`sendmail.cf`" with these commands.

- To build the `sendmail.cf` configuration file, use the following commands:
```
[root@deep /]# cd /etc/mail/
[root@deep mail]# m4 /var/tmp/sendmail-8.11.4/cf/m4/cf.m4 sendmail.mc >
/etc/mail/sendmail.cf
```

NOTE: Here, the `/var/tmp/sendmail-8.11.4/cf/m4/cf.m4` tells the `m4` program where to look for its default configuration file information. Please note that the `Sendmail` version may change and in this case don't forget to update the above command line to reflect the change.

Step 3
Finally, we must set the mode permission of this file to be (`0600/-rw-------`) and owned by the super-user 'root' for security reason.

- To change the mode permissions and ownership of the `sendmail.cf` file, use the following commands:
```
[root@deep mail]# chmod 600 sendmail.cf
[root@deep mail]# chown 0.0 sendmail.cf
```

/etc/mail/access: The Sendmail Access Configuration File

This section applies only if you chose to install and use Sendmail as a Central Mail Hub Server in your system. The "access" database file can be created to accept or reject mail from selected domains. For example, you may choose to reject all mail originating from known spammers. In our configuration, we use this file to list all email addresses from which we don't want to accept mails. This is useful to block unsolicited mails coming in our mailbox.

Step 1
The files "access" and "access.db" are not required for Local or Neighbor Client setups. It is required only if you decide to set up a Central Mail Hub to handle all your mail. Also note that the use of a Central Mail Hub will improve the security and the management of other servers and clients on your network that run Sendmail. We must change the access configuration file below to fit your requirement. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **access** file (touch /etc/mail/access) and add the following lines:

```
# Description showing below for the format of this file comes from
# the Sendmail source distribution under "cf/README" file.
#
# The table itself uses e-mail addresses, domain names, and network
# numbers as keys. For example,
#
#       spammer@aol.com         REJECT
#       cyberspammer.com        REJECT
#       192.168.212             REJECT
#
# would refuse mail from spammer@aol.com, any user from cyberspammer.com
# (or any host within the cyberspammer.com domain), and any host on the
# 192.168.212.* network.
#
# The value part of the map can contain:
#
#       OK              Accept mail even if other rules in the
#                       running ruleset would reject it, for example,
#                       if the domain name is unresolvable.
#
#       RELAY           Accept mail addressed to the indicated domain
#                       or received from the indicated domain for
#                       relaying through your SMTP server.  RELAY also
#                       serves as an implicit OK for the other checks.
#
#       REJECT          Reject the sender or recipient with a general
#                       purpose message.
#
#       DISCARD         Discard the message completely using the
#                       $#discard mailer.  This only works for sender
#                       addresses (i.e., it indicates that you should
#                       discard anything received from the indicated
#                       domain).
#
# For example:
#
#       cyberspammer.com        550 We don't accept mail from spammers
#       okay.cyberspammer.com   OK
#       sendmail.org            OK
#       128.32                  RELAY
#
# would accept mail from okay.cyberspammer.com, but would reject mail
# from all other hosts at cyberspammer.com with the indicated message.
```

```
# It would allow accept mail from any hosts in the sendmail.org domain,
# and allow relaying for the 128.32.*.* network.
#
# You can also use the access database to block sender addresses based on
# the username portion of the address. For example:
#
#       FREE.STEALTH.MAILER@    550 Spam not accepted
#
# Note that you must include the @ after the username to signify that
# this database entry is for checking only the username portion of the
# sender address.
#
# If you use like we do in our "sendmail.mc macro configuration:
#
#       FEATURE(`blacklist_recipients')
#
# then you can add entries to the map for local users, hosts in your
# domains, or addresses in your domain which should not receive mail:
#
#    badlocaluser                550 Mailbox disabled for this username
#    host.mydomain.com           550 That host does not accept mail
#    user@otherhost.mydomain.com 550 Mailbox disabled for this recipient
#
# This would prevent a recipient of badlocaluser@mydomain.com, any
# user at host.mydomain.com, and the single address
# user@otherhost.mydomain.com from receiving mail. Enabling this
# feature will keep you from sending mails to all addresses that
# have an error message or REJECT as value part in the access map.
# Taking the example from above:
#
#       spammer@aol.com         REJECT
#       cyberspammer.com        REJECT
#
# Mail can't be sent to spammer@aol.com or anyone at cyberspammer.com.
#
home.com                                DISCARD
china.com                               DISCARD
```

WARNING: Don't forget to specify in this file "access" all unsolicited email addresses that you don't want to receive email from.

Step 2
Once the access file has been configured to fit our requirement, we must use the "makemap" utility program of Sendmail to create the database map of this file.

- To create the "access database map", use the following command:
  `[root@deep /]# makemap hash /etc/mail/access.db < /etc/mail/access`

NOTE: Each time you add or modify information in the access configuration file of Sendmail, it's important to rerun the makemap utility as shown above to regenerate the access.db file and to update its internal information. Also don't forget to restart Sendmail for the changes to take effect.

Step 3
Finally, we must set the mode permission of these files to be (0600/-rw-------) and owned by the super-user 'root' for security reason.

- To change the mode permission and ownership of the access and access.db files, use the following commands:

```
[root@deep mail]# chmod 600 access
[root@deep mail]# chmod 600 access.db
[root@deep mail]# chown 0.0 access
[root@deep mail]# chown 0.0 access.db
```

/etc/mail/relay-domains: The Sendmail Relay Configuration File

This section applies only if you chose to install and use Sendmail as a Central Mail Hub Server in your system. With the new release of Sendmail, now relaying is denied by default (this is an Anti-Spam feature) and if you want to allow some hosts in your network to relay through your mail server, you must create and use the "relay-domains" file to list each FQND of servers allowed to relay through your Mail Server.

Step 1

- Create the **relay-domains** file (touch /etc/mail/relay-domains) and add the following lines:

```
localhost
www.openna.com
ns1.openna.com
ns2.openna.com
```

In the above example, we allow localhost, www.openna.com, ns1.openna.com, and ns2.openna.com to relay through our Mail Server.

Step 2
Finally, we must set the mode permission of this file to be (0600/-rw-------) and owned by the super-user 'root' for security reason.

- To change the mode permission and ownership of the relay-domains file, use the following commands:

```
[root@deep mail]# chmod 600 relay-domains
[root@deep mail]# chown 0.0 relay-domains
```

/etc/mail/aliases: The Sendmail Aliases Configuration File

This section applies only if you chose to install and use Sendmail as a Central Mail Hub Server in your system. Aliasing is the process of converting one local recipient name on the system into another (aliasing occurs only on local names). Example uses are to convert a generic name (such as root) into a real username on the system, or to convert one name into a list of many names (for mailing lists).

Step 1
For every envelope that lists a local user as a recipient, Sendmail looks up that recipient's name in the "aliases" file. Because Sendmail may have to search through thousands of names in the "aliases" file, a copy of the file is stored in a separate "db" database format file to significantly improve lookup speed.

If you configure your Sendmail to use a Central Server (Mail Hub) to handles all mail, you don't need to create the "aliases" and "aliases.db" files on the neighbor server or client machines. We must change the aliases configuration file below to fit our requirement.

- Create the **aliases** file (touch /etc/mail/aliases) and add the following lines by default:

```
# Basic system aliases -- these MUST be present.
MAILER-DAEMON:   postmaster
postmaster:      root

# General redirections for pseudo accounts.
bin:             root
daemon:          root
nobody:          root
mailnull:        root

# Person who should get root's mail
#root:           gmourani
```

NOTE: Your aliases file will be probably far more complex, but even so, note how the example shows the minimum form of aliases.

Step 2
Since /etc/mail/aliases is a database, after creating the text file as described above, you must use the "makemap" program of Sendmail to create its database map.

- To create the "aliases database map", use the following command:
  `[root@deep /]# **makemap hash /etc/mail/aliases.db < /etc/mail/aliases**`

NOTE: Don't forget to run the newaliases utility of Sendmail after each modification of the aliases file or your changes will not take effect.

When you start having a lot of aliases in your /etc/mail/aliases file, it's sometimes better to put huge alias lists in seperate files. Sendmail allows you to tell it to read email addresses for a file, instead of listing them in /etc/mail/aliases. Use parameter like `:include:/path/to/file` in place of email addresses in /etc/mail/aliases to do it.

As an example you can put the following in your /etc/mail/aliases file:
list: :include:/etc/mail/openna.txt

By adding for example the above line into your /etc/mail/alliases file, all email address for alias list will be read from /etc/mail/openna.txt. The format of file openna.txt is email addresses separated by comma.

Step 3
Finally, we must set the mode permission of these files to be (0600/-rw-------) and owned by the super-user 'root' for security reason.

- To change the mode permission and ownership of the aliases and aliases.db files, use the following commands:

```
[root@deep mail]# chmod 600 aliases
[root@deep mail]# chmod 600 aliases.db
[root@deep mail]# chown 0.0 aliases
[root@deep mail]# chown 0.0 aliases.db
```

/etc/mail/virtusertable, domaintable, mailertable: The Sendmail DB Hash Table Files

This section applies only if you chose to install and use Sendmail as a Central Mail Hub Server in your system. All of these files relate to particular features of Sendmail that can be tuned by the system administrator. Once again, these features are usually required only in the Central Mail Hub server. The following is the explanation of each one.

The virtusertable & virtusertable.db files:
A virtusertable is a database that maps virtual domains into news addresses. With this feature, mail for virtual domain on your network can be delivered to local, remote, or a single user address.

The domaintable & domaintable.db files:
A domaintable is a database that maps old domain to a new one. With this feature, multiple domain names on your network can be rewritten from the old domain to the new.

The mailertable & mailertable.db files:
A mailertable is a database that maps "host.domain" names to special delivery agent and new domain name pairs. With this feature mail on your network can be delivered through the use of a particular delivery agent to a new local or remote domain name.

- To create the **virtusertable, domaintable, mailertable,** and their corresponding "**.db**" files into /etc/mail directory, use the following commands:

```
[root@deep /]# for map in virtusertable domaintable mailertable
> do
> touch /etc/mail/${map}
> chmod 0600 /etc/mail/${map}
> makemap hash /etc/mail/${map}.db < /etc/mail/${map}
> chmod 0600 /etc/mail/${map}.db
> done
```

/etc/mail/null.mc: The Sendmail Null Client Macro File

This section applies only if you chose to install and use Sendmail as a Standalone Mail Server in your system. Since our local clients machines never receive mail directly from the outside world, and relay (send) all their mail through the Mail Hub server, we will create a special file called "null.mc", which, when later processed, will create a customized "sendmail.cf" configuration file that responds to this special setup for our neighbor or local server client machines.

Step 1
This m4 macro file is simple to create and configure because it doesn't need a lot of features, as the macro configuration file (sendmail.mc) for the Central Mail Hub server did. We must change the null.mc macro configuration file below to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **null.mc** file (touch /etc/mail/null.mc) and add the following lines:

```
VERSIONID(`linux setup for LINUX OpenNA Boreas')dnl
OSTYPE(`linux')dnl
DOMAIN(`generic')dnl
FEATURE(`nullclient',`boreas.openna.com')dnl
define(`confPRIVACY_FLAGS',
`authwarnings,goaway,restrictmailq,restrictqrun')dnl
define(`confSAFE_FILE_ENV',`/home')dnl
undefine(`ALIAS_FILE')dnl
```

This tells null.mc file to set itself up for this particular configuration setup with:

OSTYPE(`linux')
This configuration option specifies the default operating system Sendmail will be running on, in our case, the "linux" system. This item is one of the minimal pieces of information required by the "mc" file.

DOMAIN(`generic')
This configuration option will specify and describe a particular domain appropriate for your environment.

FEATURE(`nullclient',`boreas.openna.com')
This m4 macro sets your clients machines to never receive mail directly, to send their mail to a Central Mail Hub, and relay all mail through that server rather than sending directly. This feature creates a stripped down configuration file containing nothing but support for forwarding all mail to a Mail Hub via a local SMTP-based network. The argument `**boreas.openna.com**' included in this feature is the canonical name of that Mail Hub. You should, of course, **CHANGE** this canonical name to reflect your Mail Hub Server for example: FEATURE(`nullclient',`**my.mailhub.com**').

define(`confPRIVACY_FLAGS',`authwarnings,goaway,restrictmailq,restrictqrun')dnl
As for the previous sendmail.mc file, this configuration option is one of the most important for the security of Sendmail. Setting the "goaway" option causes Sendmail to disallow all SMTP "EXPN" commands, it also causes it to reject all SMTP "VERB" commands and to disallow all SMTP "VRFY" commands. These changes prevent spammers from using the "EXPN" and "VRFY" commands in Sendmail. Ordinarily, anyone may examine the mail queue's contents by using the "mailq" command. To restrict who may examine the queue's contents, you must specify the "restrictmailq" option as shown above. With this option, Sendmail allows only users who are in the same group as the group ownership of the queue directory (root) to examine the contents. This allows the queue directory to be fully protected with mode 0700, while selected users are still able to see the contents. Ordinarily, anyone may process the queue with the "-q" switch. To limit queue processing to "root" and the owner of the queue directory, you must specify the "restrictqrun" option too.

`define(`confSAFE_FILE_ENV',`/home')dnl`
This configuration option limit where in the file system, mailbox files can be written. It does a chroot into the specified portion of the file system and adds some other restrictions. In this example, we restrict and let people write only in their home directories with the value "`/home`".

`undefine(`ALIAS_FILE')`
This configuration option prevents the nullclient version of `Sendmail` from trying to access `/etc/mail/aliases` and `/etc/mail/aliases.db` files. With the adding of this line in the "`.mc`" file, you don't need to have an "`aliases`" file on all your internal neighbor client `Sendmail` machines. Aliases files are required only on the Mail Hub Server for all server and client aliases on the network.

WARNING: Don't forget to change the example canonical name '**`boreas.openna.com`**' to reflect your own Mail Hub Server canonical name for example: `FEATURE(`nullclient',`` **`my.mailhub.com'`**`)`. With this kind of configuration, we remark that no mailers should be defined, and no aliasing or forwarding is done.

Step 2
Now that our macro configuration file "`null.mc`" is created, we can build the `Sendmail` configuration file "`sendmail.cf`" from these statements in all our neighbor servers, and client machines with the following commands:

- To build the `sendmail.cf` configuration file, use the following commands:
  `[root@deep /]#` **`cd /etc/mail/`**
  `[root@deep mail]#` **`m4 /var/tmp/sendmail-8.11.4/cf/m4/cf.m4 null.mc > /etc/mail/sendmail.cf`**

Step 3
No mail should ever again be delivered to your local machine. Since there will be no incoming mail connections, you no longer needed to run `Sendmail` as a full daemon on your neighbor or local server, client machines.

- To stop the `Sendmail` daemon from running on your neighbor or local server, or client machines, edit or create the `/etc/sysconfig/sendmail` file and change/add the lines that read:

  `DAEMON=yes`

  To read:

  `DAEMON=no`

  And:

  `QUEUE=1h`

NOTE: The "`QUEUE=1h`" under `/etc/sysconfig/sendmail` file causes `Sendmail` to process the queue once every 1 hour. We leave that line in place because `Sendmail` still needs to process the queue periodically in case the Central Mail Hub Server is down.

Step 4
Local machines never use `aliases`, `access`, or other maps database. Since all map file databases are located and used on the Central Mail Hub Server for all local machines we may have on the network, we can safety remove the following binary and manual pages from all our local machines.

```
/usr/bin/newaliases
/usr/share/man/man1/newaliases.1
/usr/share/man/man5/aliases.5
```

- To remove the following files from your system, use the commands:
```
[root@client /]# rm -f /usr/bin/newaliases
[root@client /]# rm -f /usr/share/man/man1/newaliases.1
[root@client /]# rm -f /usr/share/man/man5/aliases.5
```

Step 5
Remove the unnecessary `Procmail` program from your entire local `Sendmail` server or client. Since local machines send all internal and outgoing mail to the Mail Hub Server for future delivery, we don't need to use a complex local delivery agent program like `Procmail` to do the job. Instead we can use the default `/bin/mail` program of Linux.

- To remove `Procmail` from your system, use the following command:
```
[root@client]# rpm -e procmail
```

Step 6
Finally, we must set the mode permission of the `sendmail.cf` file to be (`0600/-rw-------`) and owned by the super-user 'root' for security reason.

- To change the mode permission and ownership of the `sendmail.cf` file, use the following commands:
```
[root@deep mail]# chmod 600 /etc/mail/sendmail.cf
[root@deep mail]# chown 0.0 /etc/mail/sendmail.cf
```

`/etc/mail/local-host-names`: The `Sendmail` Local Configuration File

This section applies to every kind of `Sendmail` servers that you may want to run in your system. The `/etc/mail/local-host-names` file is read to obtain alternative names for the local host. One use for such a file might be to declare a list of hosts in your network for which the local host is acting as the MX recipient. On that machine we simply need to add the names of machines for which it (i.e. `boreas.openna.com`) will handle mail to `/etc/mail/local-host-names`. Here is an example:

- Create the **`local-host-names`** file (`touch /etc/mail/local-host-names`) and add the following lines:

```
# local-host-names - include all aliases for your machine here.
openna.com
smtp.openna.com
gurukuan.com
domain.com
```

With this type of configuration, all mail sent will appear as if it were sent from "openna.com", "gurukuan.com" or "domain.com".

Please be aware that if you configure your system to masquerade as another, any e-mail sent from your system to your system will be sent to the machine you are masquerading as. For example, in the above illustration, log files that are periodically sent to root@cronus.openna.com or the other hosts in the example above by the cron daemon of Linux would be sent to root@boreas.openna.com our Central Mail Hub Server.

WARNING: Do not use the local-host-names file of Sendmail on any Mail Server that run as a Standalone Server or the masquerading feature of your Sendmail Server which is configured, as a Central Mail Hub Server will not work. Only your Mail Hub Server must have in local-host-names file all the names of the host on your LAN. Every local-host-names file on internal servers where you've used the null.mc client macro file must be empty. After that, all you have to do is to point all your internal clients to the Central Mail Hub Server to get email.

/etc/sysconfig/sendmail: The Sendmail System Configuration File

This section applies to every kind of Sendmail servers that you may want to run in your system. The /etc/sysconfig/sendmail file is used to specify SENDMAIL system configuration information, such as if Sendmail should run as a daemon, if it should listen for mail or not, and how much time to wait before sending a warning if messages in the queue directory have not been delivered.

- Create the **sendmail** file (touch /etc/sysconfig/sendmail) and add the lines:

```
DAEMON=yes
QUEUE=1h
```

The "DAEMON=yes" option instructs Sendmail to run as a daemon. This line is useful when Sendmail client machines are configured to not accept mail directly from outside in favor of forwarding all local mail to a Central Hub; not running a daemon also improves security. If you have configured your server or client machines in this way, all you have to do is to replace the "DAEMON=yes" option to "DAEMON=no".

Mail is usually placed into the queue because it could not be transmitted immediately. The "QUEUE=1h" sets the time interval before sends a warning to the sender if the messages has not been delivered.

/etc/rc.d/init.d/sendmail: The Sendmail Initialization File

This section applies to every kind of Sendmail servers that you may want to run in your system. The /etc/rc.d/init.d/sendmail script file is responsible to automatically start and stop the sendmail daemon on your server even if you're not running a Mail Hub Server. Loading the sendmail daemon, as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Step 1
Create the **sendmail** script file (`touch /etc/rc.d/init.d/sendmail`) and add the following lines:

```
#!/bin/sh
#
# sendmail      This shell script takes care of starting and stopping
#               sendmail.
#
# chkconfig: 2345 80 30
# description: Sendmail is a Mail Transport Agent, which is the program \
#              that moves mail from one machine to another.
# processname: sendmail
# config: /etc/sendmail.cf
# pidfile: /var/run/sendmail.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source sendmail configureation.
if [ -f /etc/sysconfig/sendmail ] ; then
        . /etc/sysconfig/sendmail
else
        DAEMON=no
        QUEUE=1h
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/sbin/sendmail ] || exit 0

RETVAL=0

start() {
        # Start daemons.

        echo -n "Starting Sendmail: "
        /usr/bin/newaliases > /dev/null 2>&1
        for i in virtusertable access domaintable mailertable ; do
            if [ -f /etc/mail/$i ] ; then
                makemap hash /etc/mail/$i < /etc/mail/$i
            fi
        done
        daemon /usr/sbin/sendmail $([ "$DAEMON" = yes ] && echo -bd) \
                                  $([ -n "$QUEUE" ] && echo -q$QUEUE)
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/sendmail
        return $RETVAL
}

stop() {
        # Stop daemons.
        echo -n "Shutting down Sendmail: "
        killproc sendmail
        RETVAL=$?
        echo
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/sendmail
```

```
        return $RETVAL
}

# See how we were called.
case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart|reload)
        stop
        start
        RETVAL=$?
        ;;
  condrestart)
        if [ -f /var/lock/subsys/sendmail ]; then
            stop
            start
            RETVAL=$?
        fi
        ;;
  status)
        status sendmail
        RETVAL=$?
        ;;
  *)
        echo "Usage: sendmail {start|stop|restart|condrestart|status}"
        exit 1
esac

exit $RETVAL
```

Step 2
Once the `sendmail` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/rc.d/init.d/sendmail
  [root@deep /]# chown 0.0 /etc/rc.d/init.d/sendmail
  ```

- To create the symbolic `rc.d` links for `Sendmail`, use the following commands:
  ```
  [root@deep /]# chkconfig --add sendmail
  [root@deep /]# chkconfig --level 2345 sendmail on
  ```

- To start `Sendmail` software manually, use the following command:
  ```
  [root@deep /]# /etc/rc.d/init.d/sendmail start
  Starting Sendmail:               [OK]
  ```

NOTE: All software we describe in this book has a specific directory and subdirectory in the tar compressed archive named `floppy-2.0.tgz` containing configuration files for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files manually or cut and paste them to create or change your configuration files. Whether you decide to copy manually or get the files made for your convenience from the archive compressed files, it will be to your responsibility to modify them to adjust for your needs, and place the files related to this software to the appropriate places on your server. The server configuration file archive to download is located at the following Internet address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz.

Step 3
Once compilation, optimization, installation, and configuration by the use of the `m4` macro of the Mail Server type you want to run have been finished, we can free up some disk space by deleting the program tar archives and the related source directory since they are no longer needed.

- To delete the programs and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf sendmail-version/
[root@deep tmp]# rm -f sendmail-version.tar.gz
```

The `rm` commands as used above will remove the source files we have used to compile and install `Sendmail`. It will also remove the `Sendmail` compressed archive from the `/var/tmp/` directory.

Running `Sendmail` with `SSL` support

This section applies only if you want to run `Sendmail` through `SSL` connection. `Sendmail` (version 8.11 and above) support `SMTP STARTTLS` or if you prefer `SSL`. To enable the encryption feature of `Sendmail` we need to recompile it and add `STARTTLS` support in the default local configuration file called (`site.config.m4`). This also implies that we need to install an external program named `sfio`, which is required by `Sendmail` to run with `SSL`. Of course we assume that `OpenSSL` is already installed in your server.

To begin our implementation of `SSL` into `Sendmail` we will first install the `sfio` program, recompile `Sendmail` by adding some new options to the '`site.config.m4`' macro file to recognize the add of `SSL` support, then create the necessary certificate keys and finally add the required `SSL` parameters to the '`sendmail.mc`' macro file before creating its '`sendmail.cf`' configuration file. Running `Sendmail` with `SSL` support is no easy task. Before we embark on this, we need to first decide whether it is beneficial for you to do so. Some pros and cons are, but most certainly not limited to, the following:

Pros:

- ✓ Client and server of a `SMTP` connection can be identified.
- ✓ The transmission of e-mail between a client and server utilizing `STARTTLS` cannot be read and retranslated into plaintext provided a sufficiently secure cipher suite has been negotiated.
- ✓ The plaintext of e-mail between a client and server utilizing `STARTTLS` cannot be modified by someone, provided a sufficiently secure cipher suite has been negotiated.

Cons:

- ✓ It does not provide end-to-end encryption, since a user can doesn't usually control the whole transmission. This is in contrast to the use of TLS for http: here the user's client (a WWW browser) connects directly to the server that provides the data. E-mail can be transferred via multiple hops of which the sender can control at most the first.

- ✓ It does not provide message authentication, unless the e-mail has been sent directly from the client's (STARTTLS-capable) MUA to the recipients MTA that must record the client's certificate. Even then the message might be faked during local delivery.

Part 1: Compiling, Optimizing & Installing sfio

This section applies only if you want to run Sendmail through an SSL connection. Sfio is a portable library for managing I/O streams. It provides functionality similar to that of Stdio, the ANSI C Standard I/O library, but via a new interface that is more powerful, robust and efficient. Please note that the Sendmail organization don't recommend to use sfio2000 but sfio1999. For some (as for now unknown) reason, the version of sfio2000 doesn't work with Sendmail and it is for this reason that we will download and install sfio1999.

Step 1
Once you get the programs from the sfio website (http://www.research.att.com/sw/tools/sfio/) or directly from http://www.research.att.com/tmp/reuse/pkgBAAa2jzfD/sfio_1999.src.unix.cpio, you must copy it to the /var/tmp directory of your Linux system and change to this location before creating an installation directory for the package. After that, move the package to the installation directory to expand the archive then perform the rest of the required steps to compile, optimize and install it.

```
[root@deep /]# cp sfio_1999.src.unix.cpio /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# mkdir sfio
[root@deep tmp]# cp sfio_1999.src.unix.cpio sfio
[root@deep tmp]# cd sfio
[root@deep sfio]# cpio -i --make-directories < sfio_1999.src.unix.cpio
1258 blocks
[root@deep sfio]# mkdir include/sfio
[root@deep sfio]# PATH=/sbin:/bin:/usr/bin:/usr/sbin:/var/tmp/sfio/bin
[root@deep sfio]# cd src/lib/sfio/
```

Hey! We see here a new command called 'cpio'. Yes, cpio is a program that copies files into or out of a cpio or tar archive. The sfio package we have download comes in a cpio archive format and the way to uncompress it, is by using the cpio command (for more information about cpio read the manual page man cpio(1)).

Note the PATH command, the sfio build instruction say that we should change our shell PATH variable to include "**.**/bin" so that the program "iffe" which resides under "**.**/bin" will be available in our search path. Iffe (**IF F**eatures **E**xist) is a language interpreter to execute specifications that define configuration parameters for the sfio program. By redefining our PATH environment as shown above, the /var/tmp/sfio/bin directory which handle iffe will be included in our environment variable. Once our PATH as been set, we move to the source code directory of the program.

Step 2
There is a small bug in the `sfio-1999`, which manifests itself when `Sendmail` is delivering to the file (`.forward -> /var/spoll/mail/user`) and the user `quota` is exceeded. The problem is in `sfio/src/lib/sfio/sfputr.c (puts)`, it doesn't check for an error `from _sfflsbuf()` (in macro `SFWPEEK()`) - if an error occurs, it starts endless loop. We must fix it now.

- Edit the **`sfputr.c`** file (`vi +27 sfputr.c`) and modify the part:

```
for(w = 0; (*s || rc >= 0); )
        {       SFWPEEK(f,ps,p);

                if(p == 0 || (f->flags&SF_WHOLE) )
                {       n = strlen(s);
                        if(p >= (n + (rc < 0 ? 0 : 1)) )
```

To read:

```
for(w = 0; (*s || rc >= 0); )
        {       SFWPEEK(f,ps,p);

                if(p == -1) return -1;
                if(p == 0 || (f->flags&SF_WHOLE) )
                {       n = strlen(s);
                        if(p >= (n + (rc < 0 ? 0 : 1)) )
```

Step 3
Finally, before going into the compilation of the program, we'll edit the `makefile` and `Makefile` files to change the default compiler flags to fit our own CPU architecture for better performance. We will also change the `sfio` include file `stdio.h` to install it into a subdirectory called `sfio` as `Sendmail` recommends.

- Edit the **`makefile`** file (`vi makefile`) and change the line:

```
INCDIR= ../../../include
```

To read:

```
INCDIR= ../../../include/sfio
```

- Edit the **`makefile`** file (`vi makefile`) and change the line:

```
CCMODE= -O
```

To read:

```
CCMODE= -O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer
```

- Edit the **`makefile`** file (`vi makefile`) and change the line:

```
CC=     cc
```

To read:

```
CC=     gcc
```

- Edit the **Makefile** file (`vi Makefile`) and change the line:

  ```
  CCFLAGS = -O
  ```

  To read:

  ```
  CCFLAGS = -O3 -march=i686 -mcpu=i686 –funroll-loops –fomit-frame-pointer
  ```

Step 4
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files were placed where and finally install `sfio` in the system. Also we will run the test utility program "`runtest`" that comes with the software to be sure that everythings right.

```
[root@deep sfio]# make
[root@deep sfio]# cd Sfio_t/
[root@deep Sfio_t]# ./runtest
[root@deep Sfio_t]# cd
[root@deep /root]# find /* > sfio1
[root@deep /root]# cd /var/tmp/sfio/
[root@deep sfio]# mv include/sfio /usr/include/
[root@deep sfio]# mv lib/* /usr/lib
[root@deep sfio]# cd
[root@deep /root]# find /* > sfio2
[root@deep /root]# diff sfio1 sfio2 > Sfio-Installed
```

WARNING: You may receive a couple error messages during compilation of the program. I don't know why these errors occurs but forget them and compile again with the `make` command from where the error appears.

Step 5
Once compilation, optimization, and installation of the software have been finished, we can free up some disk space by deleting the program `cpio` archive and the related source directory since they are no longer needed.

- To delete the program and its related source directory, use the following commands:

  ```
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# rm -rf sfio/
  [root@deep tmp]# rm -f sfio_1999.src.unix.cpio
  ```

Part 2: Compiling `Sendmail` to support `SSL`

This section applies only if you want to run `Sendmail` through `SSL` connection. Once we have finished installing `sfio` it is time to compile `Sendmail` by adding some new options to its '`site.config.m4`' macro file to recognize the add of `SSL` support. Configuration of the '`site.config.m4`' macro file is explained previously in this chapter, at step 6 under "Compiling - Optimizing & Installing `Sendmail`", but you must complete steps 1 through 5 before going onto step 6 which with the addition of `SSL` support become the one explained next.

Step 1
The only difference between compiling Sendmail without SSL support as described in the beginning of this chapter and compiling the software with SSL support resides in the 'site.config.m4' macro file of the program. Inside this file we add some new parameters to enable SSL support for program.

Below I'll show you the new 'site.config.m4' macro file to create instead of the one we used for Sendmail without SSL support. If you need more information about this macro file, then refer to the previous step 6 of this chapter under the section called "Compiling - Optimizing & Installing Sendmail".

- Create the **site.config.m4** file (touch devtools/Site/site.config.m4), and add the following lines inside the file to enable SSL support with Sendmail.

```
define(`confMAPDEF', `-DMAP_REGEX')
define(`confENVDEF', `-DPICKY_QF_NAME_CHECK -DXDEBUG=0')
define(`confSTDIO_TYPE', `portable')
define(`confINCDIRS', `-I/usr/include/sfio')
APPENDDEF(`confENVDEF', `-DSFIO')
APPENDDEF(`confLIBS', `-lsfio')
APPENDDEF(`conf_sendmail_ENVDEF', `-DSTARTTLS')
APPENDDEF(`conf_sendmail_LIBS', `-lssl -lcrypto')
define(`confCC', `gcc')
define(`confOPTIMIZE', `-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-
frame-pointer')
define(`confNO_HELPFILE_INSTALL')
```

Related to the default local configuration file 'site.config.m4' that we use earlier in this chapter to install Sendmail, we have added the option "-DHASURANDOMDEV" to specify to use the /dev/urandom device that our system provides, "-DSFIO" to use sfio if available instead of the default UNIX stdio, and other macro options related to STARTTLS as shown above. If you need more information about all new options we have added to the 'site.config.m4' macro file of Sendmail refer to your Sendmail documentations available in the source file of the program.

Step 2
From this point you can return to step 7 of this chapter under the section called "Compiling - Optimizing & Installing Sendmail" and install the software. Once the program has been installed you must return here and see how to create the certificate keys, which will be used by the new SSL feature of Sendmail.

To summarize, follow step 1 through step 5 under the section called "Compiling - Optimizing & Installing Sendmail", then replace step 6 of the same section for the one explained above "Part 2: Compiling Sendmail to support SSL" and return to step 7 to follow the instructions about installing Sendmail then come back here.

Part 3: Creating the necessary Sendmail certificates keys

This section applies only if you want to run Sendmail through an SSL connection. This part is one of the most interesting. Before configuring the 'sendmail.mc' macro configuration file of Sendmail, it is important to create the appropriate certificate keys since during the configuration of the 'sendmail.mc' macro file we will add some new macro options which will indicate to Sendmail where to find them.

Below I'll show you how to create a self-signed certificate with your own CA for Sendmail. The principle is exactly the same as for creating a self-signed certificate for a Web Server (as described under chapter related to OpenSSL). I'll assume that your own CA has been already created, if this is not the case refer to OpenSSL chapter for further information.

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the Sendmail Mail Hub Server for which you want to request a certificate. When your outgoing mail server address is boreas.openna.com then the FQDN of your Mail Hub Server is boreas.openna.com.

Step 2
Make a new certificate for Sendmail. This certificate become our private key and doesn't need to be encrypted. This is required for an unattended startup of Sendmail. Otherwise you will have to enter the pass phrase each time Sendmail is started as server or client. To generate an unencrypted certificate we use the '-nodes' option as shown below.

With the '-nodes' option, the private key is not protected by a password, so the server can start without external intervention. Without this option, you must provide the password every time the server is started.

- To create a private key certificate without a pass phrase, use the following command:

```
[root@deep /]# cd /usr/share/ssl
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out smtp.key 1024
22383 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.++++++
.................................++++++
e is 65537 (0x10001)
```

Step 3
Once the private key has been made, we must generate a Certificate Signing Request (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. If you prefer to have your Certificate Signing Request (CSR) signed by a commercial Certifying Authority (CA) like Thawte or Verisign you need to post the CSR file that will be generated below into a web form, pay for the signing, and await the signed Certificate.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key smtp.key -out smtp.csr
Using configuration from /usr/share/ssl/openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA.com SMTP Mail Server]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [boreas.openna.com]:
Email Address [noc@openna.com]:
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

WARNING: Be sure that you've entered the FQDN (Fully Qualified Domain Name) of the Outgoing Mail Hub Server when OpenSSL prompts you for the "**Common Name**".

Step 4
This step is needed only if you want to sign as your own CA the csr certificate key. Now we must sign the new certificate with our own certificate authority that we have already created for generation of the Web Server certificate under the OpenSSL chapter (ca.crt). If the self signed CA certificate doesn't exist, then refer to the chapter related to OpenSSL for more information about how to create it.

- To sign with our own CA, the csr certificate, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign.sh smtp.csr
CA signing: smtp.csr -> smtp.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'Open Network Architecture'
commonName            :PRINTABLE:'boreas.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Dec 21 11:36:12 2001 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: smtp.crt <-> CA cert
smtp.crt: OK
```

WARNING: If you receive an error message saying that the csr certificate that you are trying to sign already exists, it is because the information you have entered during the generation of the certificate key is the same as another which you have already created. In this case you must at least, change one bit of information in the new certificate key you want to create before signing the certificate with your own CA.

Step 5
After, we must place the certificates files (smtp.key and smtp.crt) to the appropriate directories for Sendmail to be able to find them when it will start its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv smtp.key private/
[root@deep ssl]# mv smtp.crt certs/
[root@deep ssl]# chmod 400 private/smtp.key
[root@deep ssl]# chmod 400 certs/smtp.crt
[root@deep ssl]# rm -f smtp.csr
```

First we move the smtp.key file to the private directory and the smtp.crt file to the certs directory. After that we change the mode of the both certificates to be only readable by the super-user 'root' for security reason (if the mode of the certificates are not 0400, then Sendmail will refuse to start with SSL support enable). Finally we remove the smtp.csr file from our system since it is no longer needed.

Part 4: Adding the required SSL parameters to the 'sendmail.mc' macro file

This section applies only if you want to run Sendmail through SSL connection. Finally once Sendmail certificates have been created and moved to the appropriate location, we must create a new 'sendmail.mc' macro configuration file which, will contain macro options to indicate where certificates can be found.

Step 1
Below I'll show you the new 'sendmail.mc' macro configuration file to create instead of the one we use for Sendmail without SSL support. If you need more information about this macro configuration file, then refer to the section called "/etc/mail/sendmail.mc: The Sendmail Macro Configuration File" in this chapter.

- Create the **sendmail.mc** file (touch /etc/mail/sendmail.mc), and add the following lines inside the file to enable SSL support with Sendmail.

```
VERSIONID(`linux setup for LINUX OpenNA Boreas')dnl
OSTYPE(`linux')dnl
DOMAIN(`generic')dnl
define(`confTRY_NULL_MX_LIST',true)dnl
define(`PROCMAIL_MAILER_PATH',`/usr/bin/procmail')dnl
define(`confPRIVACY_FLAGS',
`authwarnings,goaway,restrictmailq,restrictqrun')dnl
define(`confSAFE_FILE_ENV',`/home')dnl
define(`confCACERT_PATH', `/usr/share/ssl/certs/ca.crt')dnl
define(`confCACERT', `/usr/share/ssl/certs/ca.crt')dnl
define(`confSERVER_CERT', `/usr/share/ssl/certs/smtp.crt')dnl
define(`confSERVER_KEY', `/usr/share/ssl/private/smtp.key')dnl
define(`confCLIENT_CERT', `/usr/share/ssl/certs/smtp.crt')dnl
define(`confCLIENT_KEY', `/usr/share/ssl/private/smtp.key')dnl
FEATURE(`smrsh',`/usr/sbin/smrsh')dnl
FEATURE(`mailertable',`hash -o /etc/mail/mailertable')dnl
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable')dnl
FEATURE(`redirect')dnl
FEATURE(`always_add_domain')dnl
FEATURE(`relay_hosts_only')dnl
FEATURE(`use_cw_file')dnl
FEATURE(`local_procmail')dnl
FEATURE(`access_db')dnl
FEATURE(`blacklist_recipients')dnl
FEATURE(`dnsbl')dnl
```

```
MAILER(`local')dnl
MAILER(`smtp')dnl
MAILER(`procmail')dnl
```

Step 2
Now that our macro configuration file "sendmail.mc" is configured and created to correspond to our specific needs, we can build the Sendmail configuration file "sendmail.cf" from these statements.

- To build the sendmail.cf configuration file, use the following commands:

```
[root@deep /]# cd /etc/mail/
[root@deep mail]# m4 /var/tmp/sendmail-8.11.4/cf/m4/cf.m4 sendmail.mc >
/etc/mail/sendmail.cf
```

NOTE: Here, the /var/tmp/sendmail-8.11.4/cf/m4/cf.m4 tells m4 program where to look for its default configuration file information. Please note that the Sendmail version may change and in this case don't forget to update the above command line to reflect the change. Also, your Sendmail source directory, from where you have compiled and installed the program, must be present on your system for the above command to find the required cf.m4 macro file for the generation of your new sendmail.cf file.

Step 3
Finally, we must set the mode permission of this file to be (0600/-rw-------) and owned by the super-user 'root' for security reason.

- To change the mode permission and ownership of the sendmail.cf file, use the following commands:

```
[root@deep mail]# chmod 600 sendmail.cf
[root@deep mail]# chown 0.0 sendmail.cf
```

Step 4
From this point you can return to the section called "Configuring Sendmail" and perform the rest of the steps necessary to configure the other required Sendmail configuration files. Finally come back to this section and read below the other security measures you may need to implement to improve security under Sendmail. Your software is now installed and configured to use the SSL encryption feature. Congratulations!

Securing Sendmail

This section deals specifically with actions we can take to improve security under Sendmail. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

The Sendmail restricted shell "smrsh"

The smrsh program is intended as a replacement for /bin/sh in the program mailer definition of Sendmail. It's a restricted shell utility that provides the ability to specify, through the /etc/smrsh directory, an explicit list of executable programs available to Sendmail.

To be more accurate, even if a “bad guy” can get Sendmail to run a program without going through an aliases or forward file, smrsh limits the set of programs that he or she can execute. When used in conjunction with Sendmail, smrsh effectively limits Sendmail's scope of program execution to only those programs specified in smrsh's directory. If you have followed what we did above, smrsh program is already compiled and installed on your computer under /usr/sbin/smrsh.

Step 1
The first thing we need to do is to determine the list of commands that “smrsh” should allow Sendmail to run.

By default we include, but are not limited to:

```
“/bin/mail”
“/usr/bin/procmail” (if you have it installed on your system)
```

WARNING: You should NOT include interpreter programs such as sh(1), csh(1), perl(1), uudecode(1) or the stream editor sed(1) in your list of acceptable commands.

Step 2
You will next need to populate the /etc/smrsh directory with the programs that are allowable for Sendmail to execute. To prevent duplicate programs, and do a nice job, it is better to establish links to the allowable programs from /etc/smrsh rather than copy programs to this directory.

- To allow the **mail** program /bin/mail, use the following commands:
```
[root@deep /]# cd /etc/smrsh
[root@deep smrsh]# ln -s /bin/mail mail
```

- To allow the **procmail** program /usr/bin/procmail, use the following commands:
```
[root@deep /]# cd /etc/smrsh
[root@deep smrsh]# ln -s /usr/bin/procmail procmail
```

This will allow the **mail** and **procmail** programs to be run from a user's “.forward” file or an “aliases” file which uses the "program" syntax.

NOTE: Procmail is required only in a Mail Hub Server and not in a Local Client Mail Server. If you’re configured your system like a Mail Hub Server then make the link with procmail as explained above, if you’re configured your system as a Local Client Server then skip the procmail step above.

Step 3
We can now configure Sendmail to use the restricted shell. The program mailer is defined by a single line in the Sendmail configuration file, /etc/mail/sendmail.cf. You must modify this single line "Mprog" definition in the "sendmail.cf" file, by replacing the /bin/sh specification with /usr/sbin/smrsh.

- Edit the **sendmail.cf** file (vi /etc/mail/sendmail.cf) and change the line:

```
Mprog,  P=/bin/sh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/, T=X-Unix, A=sh
-c $u
```

To read:

```
Mprog,  P=/usr/sbin/smrsh, F=lsDFMoqeu9, S=10/30, R=20/40, D=$z:/, T=X-
Unix, A=sh -c $u
```

- Now re-start the Sendmail process manually with the following command:

```
[root@deep /]# /etc/rc.d/init.d/sendmail restart
Shutting down sendmail:          [OK]
Starting sendmail:               [OK]
```

NOTE: In our "sendmail.mc" configuration file for the Mail Hub Server above, we have already configured this line "Mprog" to use the restricted shell /usr/sbin/smrsh with the m4 macro "FEATURE(`smrsh',`/usr/sbin/smrsh')", so don't be surprised if the /usr/sbin/smrsh specification is already set in your /etc/mail/sendmail.cf file for the Mail Hub relay. Instead, use the technique shown above for other /etc/mail/sendmail.cf files in your network like the one for the nullclient ("local or neighbor client and servers") that use the "null.mc" macro configuration file to generate the /etc/mail/sendmail.cf file.

The /etc/mail/aliases file

A poorly or carelessly administered "aliases" file can easily be used to gain privileged status. For example, many vendors ship systems with a "**decode**" alias in the /etc/mail/aliases file. The intention is to provide an easy way for users to transfer binary files using mail. At the sending site the user converts the binary to ASCII with "**uuencode**", then mails the result to the "**decode**" alias at the receiving site. That alias pipes the mail message through the /usr/bin/uuencode program, which converts the ASCII back into the original binary file.

Remove the "**decode**" alias line from your /etc/mail/aliases file. Similarly, every alias that executes a program that you did not place there yourself and checked completely should be questioned and probably removed.

- Edit the **aliases** file (vi /etc/mail/aliases) and remove the following lines:

```
# Basic system aliases -- these MUST be present.
MAILER-DAEMON:  postmaster
postmaster:  root

# General redirections for pseudo accounts.
bin:         root
daemon:      root
games:       root  ← remove this line.
ingres:      root  ← remove this line.
```

```
nobody:       root
system:       root  ← remove this line.
toor:         root  ← remove this line.
uucp:         root  ← remove this line.

# Well-known aliases.
manager:      root  ← remove this line.
dumper:       root  ← remove this line.
operator:     root  ← remove this line.

# trap decode to catch security attacks
decode:       root  ← remove this line.

# Person who should get root's mail
#root:        marc
```

- For the changes to take effect you will need to run:
  `[root@deep /]# /usr/bin/newaliases`

> **NOTE:** Don't forget to rebuild your `aliases` file with the `newaliases` command of `Sendmail` for the changes to take effect.

The `SMTP` greeting message

When `Sendmail` accepts an incoming `SMTP` connection it sends a greeting message to the other host. This message identifies the local machine and is the first thing it sends to say it is ready.

- Edit the **`sendmail.cf`** file (`vi /etc/mail/sendmail.cf`) and change the line:

  ```
  O SmtpGreetingMessage=$j Sendmail $v/$Z; $b
  ```

  To read:

  ```
  O SmtpGreetingMessage=$j
  ```

- Now re-start the `Sendmail` process manually for the change to take effect:

  ```
  [root@deep /]# /etc/rc.d/init.d/sendmail restart
  Shutting down sendmail:            [OK]
  Starting sendmail:                 [OK]
  ```

This change doesn't actually affect anything, but was recommended by folks in the `news.admin.net-abuse.email` newsgroup as a legal precaution. It modifies the banner, which `Sendmail` displays upon receiving a connection.

Change all the default `Sendmail` files mode into the `/etc/mail` directory

For the paranoids, we can change the default mode of all `Sendmail` files under the `/etc/mail` directory to be readable and writable only by the super-user '`root`'. There are no reasons to let everyone read access to these files.

- To change the mode of all files under `/etc/mail` directory, use the following command:
  `[root@deep /]# chmod 600 /etc/mail/*`

The undisclosed recipients mail message

One of the biggest problems in an ISP environment is the undisclosed recipients mail message that the users send to a lot of people. It's like spamming, but in a minor grade. There is a feature to stop this by changing the `sendmail.cf` file the default value of the `MaxRecipientsPerMessage` macro option.

By default, `Sendmail` limits the number of recipients that a mail can have to 100.
A good value for start testing it is 30.

- Edit the **`sendmail.cf`** file (`vi /etc/mail/sendmail.cf`) and change the line:

```
# maximum number of recipients per SMTP envelope
#O MaxRecipientsPerMessage=100
```

To read:

```
# maximum number of recipients per SMTP envelope
#O MaxRecipientsPerMessage=30
```

Set the immutable bit on important `Sendmail` files

Important `Sendmail` files can have their immutable bit set for better security with the "`chattr`" command of Linux. A file with the "`+i`" attribute cannot be modified, deleted or renamed; No link can be created to this file, and no data can be written to the file. Only the super-user can set or clear this attribute.

- Set the immutable bit on the "`sendmail.cf`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/sendmail.cf`**
- Set the immutable bit on the "`local-host-names`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/local-host-names`**
- Set the immutable bit on the "`relay-domains`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/relay-domains`**
- Set the immutable bit on the "`aliases`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/aliases`**
- Set the immutable bit on the "`access`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/access`**
- Set the immutable bit on the "`virtusertable`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/virtusertable`**
- Set the immutable bit on the "`domaintable`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/domaintable`**
- Set the immutable bit on the "`mailertable`" file:
  `[root@deep /]#` **`chattr +i /etc/mail/mailertable`**

Further documentation

For more details about Sendmail program, there are several manual pages you can read:

```
$ man aliases (5)          - aliases file for Sendmail
$ man makemap (8)          - create database maps for Sendmail
$ man sendmail (8)         - an electronic mail transport agent
$ man mailq (1)            - print the mail queue
$ man newaliases (1)       - rebuild the data base for the mail aliases file
$ man mailstats (8)        - display mail statistics
$ man praliases (8)        - display system mail aliases
```

Sendmail Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual page and documentation for more information.

newaliases

The purpose of the "newaliases" program utility of Sendmail is to rebuild and update the random access database for the mail aliases file /etc/mail/aliases. It must be run each time you change the contents of this file in order for the changes to take effect.

- To update the aliases file with the "newaliases" utility, use the following command:
  ```
  [root@deep /]# /usr/bin/newaliases
  ```

makemap

The purpose of the "makemap" program utility is to create the database-keyed maps in Sendmail. The "makemap" command must be used only when you need to create a new database for file like aliases, access, or domaintable, mailertable, and virtusertable.

- To run makemap to create a new database for access, use the following command:
  ```
  [root@deep /]# makemap hash /etc/mail/access.db < /etc/mail/access
  ```

Where <hash> is the database format, makemap can handle up to three different database formats; they may be "hash", "btree" or "dbm". The </etc/mail/access.db> is the location and the name of the new database that will be created. The </etc/mail/access> is the location of the file from where makemap will read from the standard input file. In our example, we have created a new "access.db" file with the makemap command above. To create a database for other files like aliases, domaintable, mailertable, and virtusertable, you must indicate the location and name of the corresponding file in the "makemap" command.

mailq

The purpose of the "mailq" program utility is to print a summary of the mail messages queued for future delivery.

- To print a summary of the mail messages queued, use the following command:
  ```
  [root@deep /]# mailq
  Mail queue is empty
  ```

- To process all messages in the queue manualy, use the following command:
  ```
  [root@deep /]# sendmail -q
  ```

NOTE: If you want to have more useful information about the queued message, you can try a command like:

```
[root@deep /]# sendmail -qIqueuid -v
```

where you replace "queuid" by the actual identifier for a queued message.

Sendmail Users Tools

The commands listed below are some that we use often, but many more exist. Check the manual page and documentation for more information.

mailstats

The "mailstats" program utility is a statistics-printing program and its purpose is to display the current mail statistics.

- To displays the current mail statistics, use the following command:

```
[root@deep /]# mailstats
Statistics from Wed Nov 29 09:00:29 2000
 M   msgsfr  bytes_from   msgsto    bytes_to  msgsrej msgsdis  Mailer
 3        1          1K        0          0K        0       0  local
 8        0          0K        1          1K        0       0  relay
=============================================================
 T        1          1K        1          1K        0       0
 C        1                    1                    0
```

praliases

The "praliases" program utility is a program to print the DBM or NEWDB version of the aliases file and its purpose is to display one per line, in no particular order, the contents of the current system mail aliases.

- To display the current system aliases, use the following command:

```
[root@deep /]# praliases
postmaster:root
daemon:root
root:admin
@:@
mailer-daemon:postmaster
bin:root
nobody:root
webadmin:admin
www:root
```

List of installed `Sendmail` files on your system for Central Mail Hub

> /etc/mail
> /etc/mail/statistics
> /etc/mail/sendmail.cf
> /etc/mail/sendmail.mc
> /etc/mail/access
> /etc/mail/access.db
> /etc/mail/aliases
> /etc/mail/aliases.db
> /etc/mail/virtusertable
> /etc/mail/virtusertable.db
> /etc/mail/domaintable
> /etc/mail/domaintable.db
> /etc/mail/mailertable
> /etc/mail/mailertable.db
> /etc/mail/local-host-names
> /etc/sysconfig/Sendmail
> /etc/smrsh
> /usr/bin/newaliases
> /usr/bin/mailq
> /usr/bin/hoststat
> /usr/bin/purgestat
> /usr/lib/sendmail
> /usr/sbin/sendmail
> /usr/sbin/mailstats
> /usr/sbin/smrsh
> /usr/sbin/makemap
> /usr/sbin/praliases
> /usr/share/man/man1/mailq.1
> /usr/share/man/man1/newaliases.1
> /usr/share/man/man5/aliases.5
> /usr/share/man/man8/smrsh.8
> /usr/share/man/man8/sendmail.8
> /usr/share/man/man8/mailstats.8
> /usr/share/man/man8/makemap.8
> /usr/share/man/man8/praliases.8
> /var/spool/mqueue

List of installed `Sendmail` files on your system for local server or client

> /etc/mail
> /etc/mail/statistics
> /etc/mail/local-host-names
> /etc/smrsh
> /usr/bin/mailq
> /usr/bin/hoststat
> /usr/bin/purgestat
> /usr/lib/sendmail
> /usr/sbin/sendmail
> /usr/sbin/mailstats
> /usr/sbin/smrsh
> /usr/share/man/man1/mailq.1
> /usr/share/man/man8/sendmail.8
> /usr/share/man/man8/mailstats.8
> /usr/share/man/man8/smrsh.8
> /var/spool/mqueue

List of installed `sfio` files on your system

> /usr/include/sfio
> /usr/include/sfio/sfio.h
> /usr/include/sfio/ast_common.h
> /usr/include/sfio/sfio_t.h
> /usr/include/sfio/stdio.h
> /usr/lib/libsfio.a
> /usr/lib/libstdio.a

21 Mail Transfer Agent - `qmail`

In this Chapter

Linux qmail Mail Transfer Agent Server

Abstract

If you decide to use qmail successfully as a Mail Server, you must be aware of how it works. It is completely different to Sendmail, it has pretty much everything built into a single binary. The qmail system is built using the philosophy of having many small utilities that do one thing, and then combining these utilities to make something useful happen. qmail delivery takes place using a number of separate programs that communicate with each other in well defined ways.

Finally and before going into qmail deeper, it's important to note that qmail runs through a program named tcpserver which functions in the same manner as Xinetd, but is supposed to be faster. Therefore, to install the base qmail features in your system, you'll have to play with at least two different packages related to it, which are named respectively ucspi-tcp and checkpassord. Personally, I think that there are too many add-ons to do with qmail to be able to run it. In the other hand, if we look for some surveys, we'll find that Hotmail with thirty million users has been using qmail for outgoing mail since 1997. (Reportedly, after Microsoft purchased Hotmail, it tried to move Hotmail to Microsoft Exchange under Windows NT. Exchange crashed.)

According to the creators note:
qmail is a secure, reliable, efficient, simple message transfer agent. It is meant as a replacement for the entire sendmail-binmail system on typical Internet-connected UNIX hosts. Security isn't just a goal, but an absolute requirement. Mail delivery is critical for users; it cannot be turned off, so it must be completely secure. (This is why I started writing qmail: I was sick of the security holes in sendmail and other MTAs.).

qmail supports host and user masquerading, full host hiding, virtual domains, null clients, list-owner rewriting, relay control, double-bounce recording, arbitrary RFC 822 address lists, cross-host mailing list loop detection, per-recipient checkpointing, downed host backoffs, independent message retry schedules, etc. In short, it's up to speed on modern MTA features. qmail also includes a drop-in "sendmail" wrapper so that it will be used transparently by your current UAs.

As with the previous Sendmail set up, we'll show you two different configurations that you can use for qmail; one for a Central Mail Hub Relay, and another for a null client, which can be used for any server that doesn't run as a Mail Hub Server. Contrary to the Sendmail null client configuration, you'll see here that with qmail configuring a null client is far more easier and doesn't require you to play around with many different macros or files.

Finally, I'd like to advise you that qmail is not supported by the majority of external mailing list applications or programs like mailman, logcheck, tripwire, etc. It can be very difficult to make it work with this kind of program and trying to find help on the qmail mailing list can also be very difficult, since support is not as you would expect it to be, like with Sendmail. A lot of serious questions are asked without any answers and only stupid question are answer by the mailing list users (I'm sorry but it is true). Therefore and before going into compilation and installation of this software I recommend you think about your decision.

Recommended RPM packages to be installed for a `Mail` Server

A minimal configuration provides the basic set of packages required by the Linux operating system. A minimal configuration is a perfect starting point for building a secure operating system. Below is the list of all recommended RPM packages required to run your Linux server as a Mail Server (`SMTP`) running on `qmail` software properly.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install `qmail` by RPM package. Therefore, `qmail` RPM package is already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

basesystem	e2fsprogs	iptables	openssh-server	slang
bash	ed	kernel	openssl	slocate
bdflush	file	less	pam	sysklogd
bind	filesystem	libstdc++	passwd	syslinux
bzip2	fileutils	libtermcap	popt	SysVinit
chkconfig	findutils	lilo	procps	tar
console-tools	gawk	logrotate	psmisc	termcap
cpio	gdbm	losetup	pwdb	textutils
cracklib	gettext	MAKEDEV	**qmail**	tmpwatch
cracklib-dicts	glib	man	**quota**	utempter
crontabs	glibc	mingetty	readline	util-linux
db1	glibc-common	mktemp	rootfiles	vim-common
db2	grep	mount	rpm	vim-minimal
db3	groff	ncurses	sed	vixie-cron
dev	gzip	net-tools	setup	words
devfsd	info	newt	sh-utils	which
diffutils	initscripts	openssh	shadow-utils	zlib

Tested and fully functional on OpenNA.com.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest qmail version number is 1.03
Latest ucspi-tcp version number is 0.88
Latest checkpassword version number is 0.90

Packages

The following is based on information as listed by qmail as of 2001/03/25. Please regularly check at www.qmail.org for the latest status.

Source code is available from:
qmail Homepage: http://www.qmail.org/
qmail FTP Site: 131.193.178.181
You must be sure to download: qmail-1.03.tar.gz

ucspi-tcp Homepage: http://cr.yp.to/ucspi-tcp/install.html
You must be sure to download: ucspi-tcp-0.88.tar.gz

checkpassword Homepage: http://cr.yp.to/checkpwd/install.html
You must be sure to download: checkpassword-0.90.tar.gz

Prerequisites

qmail requires that the listed software below be already installed on your system to be able to compile as a full Central Mail Hub Server successfully. If this is not the case, you must install them from the source archive files. Please make sure you have all of these programs installed on your box before you proceed with this chapter. If you're intended to install qmail as a standalone null client Mail Server, you don't need to install those programs.

- ✓ ucspi-tcp is needed by qmail and should be already installed on your system.
- ✓ checkpassword is needed by qmail and should be already installed on your system.

Verifying & installing all the prerequisites to run qmail

As I've mentioned before, qmail use a modular design to build everything into a single binary. This means, for example, that its binary program, which is responsible for sending mail, is separate from its program that is responsible for receiving mails, and so on. In order to perform some other useful actions you need the utilities supplied in the ucspi-tcp and checkpassord packages.

The ucspi-tcp package includes a high-speed inetd replacement for the SMTP server, and a generic tool to reject mail from RBL-listed sites. ucspi-tcp is required by qmail to be able to run its smtpd program on port 25, to receive mail via SMTP. Without this package, you cannot receive mail on the machine where qmail is installed, but just send mail. It's also required by the qmail-popd3 and qmail-popup programs of qmail to be able to read your mail from an external computer.

The checkpassword program is also required by qmail if you need to run qmail-pop3d, which is already included in the qmail package and responsible to distribute mail via POP through qmail-popup, which reads a POP username and password.

As you can see, with qmail you don't need to install external programs, like IMAP/POP, to be able to read mail from another computer. qmail includes its own secure programs, qmail-pop3d and qmail-popup, for this purpose.

Therefore, what is the relation between qmail and checkpassword? checkpassword provides a simple, uniform password-checking interface to all root applications and it is suitable for use by applications such as pop3d.

Compiling, Optimizing & Installing ucspi-tcp

This section applies only if you chose to install and use qmail as a Central Mail Hub Server in your system. The ucspi-tcp package includes many small utilities to run with qmail. One of the most important, and the one we need here, is named tcpserver and works in the same way as Xinetd or inetd works, but the difference is that it's really much faster.

The tcpserver program accepts incoming TCP connections and waits for connections from TCP clients. Without it, you cannot access you pop3 account and read your mail on the qmail Mail Server.

Step 1
Once you get the program from the qmail website you must copy it to the /var/tmp directory of your Linux system and change to this location before expanding the archive. After that, move into the newly created ucspi-tcp directory and perform the following steps to compile and optimize it.

```
[root@deep /]# cp ucspi-tcp-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf ucspi-tcp-version.tar.gz
[root@deep tmp]# cd ucspi-tcp-0.88
```

Step 2
Now, it's important to edit the conf-home file and change the default location where ucspi-tcp programs will be installed to fit our operating system environment.

- Edit the **conf-home** file (vi conf-home) and change the line:

  ```
  /usr/local
  ```

  To read:

  ```
  /usr
  ```

Step 3
Finally, before going into the compilation of the program, we'll edit the `conf-cc` file and change the default compiler flags to fit our own CPU architecture for better performance.

- Edit the **`conf-cc`** file (`vi conf-cc`) and change the line:

  ```
  gcc -O2
  ```

  To read:

  ```
  gcc -O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer
  ```

WARNING: Please don't forget to adjust the above optimization `FLAGS` to reflect your own system and CPU architecture.

Step 4
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `ucspi-tcp` in the system.

```
[root@deep ucspi-tcp-0.88]# make
[root@deep ucspi-tcp-0.88]# cd
[root@deep /root]# find /* > ucspitcp1
[root@deep /root]# cd /var/tmp/ucspi-tcp-0.88/
[root@deep ucspi-tcp-0.88]# make setup check
[root@deep ucspi-tcp-0.88]# cd
[root@deep /root]# find /* > ucspitcp2
[root@deep /root]# diff ucspitcp1 ucspitcp2 > Ucspitcp-Installed
```

NOTE: Executing the `find` command under the `/root` directory is only needed to keep trace of what files the program will install into the system and where. It's a good practice to keep log of installed files in the system in case of future upgrade or bug fixes.

Compiling, Optimizing & Installing `checkpassword`

This section applies only if you chose to install and use `qmail` as a Central Mail Hub Server in your system. As described on the `qmail` website, `qmail-popup` and `qmail-pop3d` are glued together by a program called `checkpassword`. It's run by `qmail-popup`, it reads the username and password handed to the `POP3` daemon, then looks them up in `/etc/passwd`, verifies them, switches to the `username/home` directory, and then it runs `pop3d`.

Step 1
Once you get the program from the `qmail` website you must copy it to the `/var/tmp` directory of your Linux system and change to this location before expanding the archive. After that, move into the newly created `checkpassword` directory and perform the following steps to compile and optimize it.

```
[root@deep /]# cp checkpassword-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf checkpassword-version.tar.gz
[root@deep tmp]# cd checkpassword-0.90
```

Step 2
Before going into compilation of the program, we'll edit the `conf-cc` file and change the default compiler flags to fit our own CPU architecture for better performance.

- Edit the **`conf-cc`** file (`vi conf-cc`) and change the line:

  ```
  cc -O2
  ```

  To read:

  ```
  gcc -O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer
  ```

> **WARNING:** Please don't forget to adjust the above optimization `FLAGS` to reflect your own system and CPU architecture.

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `checkpassword` in the system.

```
[root@deep checkpassword-0.90]# make
[root@deep checkpassword-0.90]# cd
[root@deep /root]# find /* > checkpass1
[root@deep /root]# cd /var/tmp/checkpassword-0.90/
[root@deep checkpassword-0.90]# make setup check
[root@deep checkpassword-0.90]# cd
[root@deep /root]# find /* > checkpass2
[root@deep /root]# diff checkpass1 checkpass2 > CheckPass-Installed
```

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `qmail`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  `[root@deep /root]#  find /* > qmail1`

- And the following one after you install the software:
  `[root@deep /root]#  find /* > qmail2`

- Then use the following command to get a list of what changed:
  `[root@deep /root]#  diff qmail1 qmail2 > Qmail-Installed`

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling, Optimizing & Installing qmail

Now you are ready to go to the installation steps of the qmail program. Below are the required steps that you must make to compile the qmail software before installing it into your Linux system.

As you'll see later, qmail has no pre-compilation configuration like Sendmail, which required a big decision list of what to compile in the software. Instead qmail automatically adapts itself to your UNIX variant and allows a quick installation. On the other hand, due to its quick installation feature, it doesn't let us install different parts of the software where we want them to go and this is why we must do a bit of tweaking to make it fit our system environment.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp qmail-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf qmail-version.tar.gz
```

Step 2
In order to check that the version of qmail, which you are going to install, is an original and unmodified one, use the commands described below and check the supplied signature.

- To verify the MD5 checksum of qmail, use the following command:

```
[root@deep tmp]# md5sum qmail-version.tar.gz
```

This should yield an output similar to this:

```
622f65f982e380dbe86e6574f3abcb7c  qmail-1.03.tar.gz
```

Now check that this checksum is exactly the same as the one published on the qmail website at the following URL: http://cr.yp.to/qmail/dist.html

Step 3
After that, move into the newly created qmail directory and create the qmail home directory manually. The qmail home directory is where everything related to qmail software are handle.

- To move into the newly created qmail archive directory, use the following command:

```
[root@deep tmp]# cd qmail-1.03/
```

- To create the qmail home directory, use the following command:

```
[root@deep qmail-1.03]# mkdir /var/qmail
```

Step 4
Once the qmail home directory has been created, we must set up the qmail groups and the qmail users accounts before compiling the program. It's important to note that no qmail users or groups have a shell account on the system; this is an important security point to consider.

During the creation of all the required qmail accounts as shown below, we'll redirect all qmail users and groups account to a /bin/false shell. Once again this is an important security measure to take.

- To create all the required qmail users and groups, use the following commands:

```
[root@deep qmail-1.03]# groupadd -f -g81 nofiles
[root@deep qmail-1.03]# groupadd -f -g82 qmail
[root@deep qmail-1.03]# useradd -g nofiles -d /var/qmail/alias -u 82 -s
/bin/false alias 2>/dev/null || :
[root@deep qmail-1.03]# useradd -g nofiles -d /var/qmail -u 81 -s
/bin/false qmaild 2>/dev/null || :
[root@deep qmail-1.03]# useradd -g nofiles -d /var/qmail –u 86 –s
/bin/false qmaill 2>/dev/null || :
[root@deep qmail-1.03]# useradd -g nofiles -d /var/qmail –u 87 –s
/bin/false qmailp 2>/dev/null || :
[root@deep qmail-1.03]# useradd -g qmail -d /var/qmail –u 83 –s
/bin/false qmailq 2>/dev/null || :
[root@deep qmail-1.03]# useradd -g qmail -d /var/qmail –u 84 –s
/bin/false qmailr 2>/dev/null || :
[root@deep qmail-1.03]# useradd -g qmail -d /var/qmail –u 85 –s
/bin/false qmails 2>/dev/null || :
```

The above commands will create all the required qmail groups and users accounts necessary for the program to run properly and in a secure manner.

Step 5
Before going into the compilation of the program, we'll edit the conf-cc file and change the default compiler flags to fit our own CPU architecture for better performance.

- Edit the **conf-cc** file (vi conf-cc) and change the line:

  ```
  cc -O2
  ```

  To read:

  ```
  gcc -O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer
  ```

WARNING: Please don't forget to adjust the above optimization FLAGS to reflect your own system and CPU architecture.

Step 6
Now, we must make a list of files on the system before installing the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally compile the programs and create the qmail directory tree in the server.

```
[root@deep qmail-1.03]# cd
[root@deep /root]# find /* > qmail1
[root@deep /root]# cd /var/tmp/qmail-1.03/
[root@deep qmail-1.03]# make setup check
[root@deep qmail-1.03]# strip /var/qmail/bin/*
[root@deep qmail-1.03]# ln -s /var/qmail/bin/sendmail /usr/lib/sendmail
[root@deep qmail-1.03]# ln -s /var/qmail/bin/sendmail /usr/sbin/sendmail
[root@deep qmail-1.03]# mv -f /var/qmail/bin/maildir2mbox /usr/bin
[root@deep qmail-1.03]# mv -f /var/qmail/bin/maildirmake /usr/bin
[root@deep qmail-1.03]# mv -f /var/qmail/bin/maildirwatch /usr/bin
[root@deep qmail-1.03]# mv -f /var/qmail/bin/qmail-qread /usr/bin
[root@deep qmail-1.03]# mv -f /var/qmail/bin/qmail-qstat /usr/bin
[root@deep qmail-1.03]# chmod 444 /var/qmail/man/man1/*
[root@deep qmail-1.03]# chmod 444 /var/qmail/man/man5/*
[root@deep qmail-1.03]# chmod 444 /var/qmail/man/man7/*
[root@deep qmail-1.03]# chmod 444 /var/qmail/man/man8/*
[root@deep qmail-1.03]# mv -f /var/qmail/man/man1/* /usr/share/man/man1
[root@deep qmail-1.03]# mv -f /var/qmail/man/man5/* /usr/share/man/man5
[root@deep qmail-1.03]# mv -f /var/qmail/man/man7/* /usr/share/man/man7
[root@deep qmail-1.03]# mv -f /var/qmail/man/man8/* /usr/share/man/man8
[root@deep qmail-1.03]# rm -rf /var/qmail/man/
[root@deep qmail-1.03]# rm -rf /var/qmail/doc/
[root@deep qmail-1.03]# maildirmake /etc/skel/Maildir
[root@deep qmail-1.03]# cd
[root@deep /root]# find /* > qmail2
[root@deep /root]# diff qmail1 qmail2 > qmail-Installed
```

The first two **ls -s** commands are used to make qmail's "sendmail" wrapper available to MUAs. This is important to stay compatible with an external program you may install in the future. Many programs assume by default that you have Sendmail installed in your system. Therefore to eliminate possible problems and compatibility issues, we'll create symbolic links from qmail to Sendmail. To summarize we'll make a link under /usr/lib and /usr/sbin where sendmail binary usually lives to the qmail program. This way any program that look for Sendmail in these directories will be automatically redirected to the qmail program for execution.

The **mv -f** commands are used to put all manual pages related to qmail under our default manual pages directories for Linux. Once all the qmail manual pages have been placed under /usr/share/man directories, we remove the old /var/qmail/man directories which are not needed now.

The **rm -f** command is also used to remove the /var/qmail/doc how-to pages directory, where all the documentation related to qmail lives after a successful installation of the program. These documentation files are the same as the ones you have surely read during the installation of qmail. If this is the case, you can remove them to make space and clean up the qmail directory.

The **maildirmake** command is used to created a skeleton of maildir for incoming mail under the /etc/skel directory which is used by Linux during creation of new users account on the system. By creating a Maildir directory under this location (/etc/skel), we solve the problem of creating manually a new maildir directory under each new added user in the system (see further down in this chapter for more information about the Maildir feature of qmail).

Step 7
You MUST tell qmail your hostname. To do it, use the config script of qmail, which looks up your host name in DNS. This config script will also look up your local IP addresses in DNS to decide which hosts to it should accept mail for.

```
[root@deep /root]# cd /var/tmp/qmail-1.03/
[root@deep qmail-1.03]# ./config
Your hostname is boreas
Your host's fully qualified name in DNS is boreas.openna.com.
Putting boreas.openna.com into control/me...
Putting openna.com into control/defaultdomain...
Putting openna.com into control/plusdomain...

Checking local IP addresses:
127.0.0.1: Adding localhost to control/locals...
207.35.78.4: Adding boreas.openna.com to control/locals...

If there are any other domain names that point to you,
you will have to add them to /var/qmail/control/locals.
You don't have to worry about aliases, i.e., domains with CNAME records.

Copying /var/qmail/control/locals to /var/qmail/control/rcpthosts...
Now qmail will refuse to accept SMTP messages except to those hosts.
Make sure to change rcpthosts if you add hosts to locals or
virtualdomains!
```

NOTE: If you receive an error message like:

```
Your hostname is boreas.
hard error
Sorry, I couldn't find your host's canonical name in DNS.
You will have to set up control/me yourself.
```

You'll have to run the config-fast script located in the same source directory as follow:

```
./config-fast boreas.openna.com
```

Here I assume that your domain is openna.com and the hostname of your computer is boreas.

Step 8
Now it's time to add the minimum required aliases for qmail to run properly on your system. You should set up at least aliases for Postmaster, Mailer-Daemon, and root. For security reasosn the super-user 'root' never receives mail with qmail. Because many programs on our server need to send system messages to 'root', we can create an alias to another user locally or remotely. Finally an important note is the fact that qmail uses files for every alias. This is one of the major ways that qmail differs from sendmail. Therefore don't forget to create an ".qmail" aliases file for every users on the system.

```
[root@deep qmail-1.03]# cd ~alias
[root@deep alias]# touch .qmail-postmaster
[root@deep alias]# touch .qmail-mailer-daemon
[root@deep alias]# touch .qmail-root
[root@deep alias]# chmod 644 ~alias/.qmail-*
```

- To create an alias for the super-user 'root' use command like:
  `[root@deep alias]# echo noc@openna.com > .qmail-root`

Here I instruct qmail to send all message intended to the super-user 'root' to a remote non-privileged user account named noc at openna.com. You can also instruct qmail to send all message reserved for 'root' to a local user by specifying just the name of an existing local user like 'gmourani'. Finally, this method is applicable for any other aliases files with qmail and I recommend you, at the minimum, create an alias for the users 'postmaster' and 'mailer-daemon' too. In this way all possible messages intended for these users will be forwarded to the alias user.

- To create an alias for users 'postmaster' and 'mailer-daemon' use commands like:
```
[root@deep alias]# echo noc@openna.com > .qmail-postmaster
[root@deep alias]# echo noc@openna.com > .qmail-mailer-daemon
```

NOTE: qmail doesn't have any built-in support for Sendmail /etc/aliases. If you have a big /etc/aliases and you'd like to keep it, install the fastforward package, which is available separately from the qmail website. This package "fastforward" is faster and more secure than the default Sendmail aliases feature. As a security precaution, qmail refuses to deliver mail to users who don't own their home directory. In fact, such users aren't even considered users by qmail. As a result, if "postmaster" doesn't own ~postmaster, then postmaster isn't a user, and postmaster@openna.com isn't a valid mailbox.

Step 9
The qmail package, once installed on your system, includes a local delivery agent, called 'qmail-local', which provides user-controlled mailing lists, cross-host alias loop detection, and many other important qmail features like the qmail crashproof Maildir directory for your incoming mail messages. This qmail program (qmail-local) is intended to replace binmail which is the default Unix /bin/mail program used under Linux to delivers mail locally into a central spool directory called /var/spool/mail.

There's one important difference between qmail-local and binmail: qmail-local delivers mail by default into ~user/Mailbox or ~user/Maildir, rather than /var/spool/mail/user. What does this imply?

As explained in the documentation of qmail, there are two basic problems with /var/spool/mail:

- ✓ It's slow. On systems with thousands of users, /var/spool/mail has thousands of entries. A few UNIX systems support fast operations on large directories, but most don't.
- ✓ It's insecure. Writing code that works safely in a world-writable directory is not easy. See, for example, CERT advisory 95:02.

For these reasons and to tighten the security of our configured system, as well as to optimize the qmail Mail Server to perform at its peak, we'll change and configure mail software to look at the qmail ~user/Maildir directly. Maildir is a feature of qmail to replace the old well known Unix Mailbox directory that is less reliable then Maildir.

Usually, you can create this new Maildir directory manually for all existing users in the system, but it is recommended to automate the task for future users by setting up Maildir as the default for everybody, by creating a maildir in the new-user template directory (/etc/skel). Below, we show you both methods:

For all existing users in your system:

- To create a new Maildir for all existing users in the system, use the commands:
```
[root@deep /]# maildirmake $HOME/Maildir
[root@deep /]# echo ./Maildir/ > ~/.qmail
```

Where <$HOME> is the username directory where you want to create this new qmail Maildir directory for all incoming mail messages. The echo command is required only if you want to create an alias file for this user (see your qmail documentation for more information about users alias file).

WARNING: The <echo ./Maildir/ > ~/.qmail> command is not required for the super-user 'root' since we have already create its alias file under /var/qmail/alias directory previously during the installation of qmail.

For all future users in your system:

- Create the **qmail.csh** file (touch /etc/profile.d/qmail.csh) and add the lines:

```
setenv MAIL $HOME/Maildir/
setenv MAILDIR $MAIL
```

- Create the **qmail.sh** file (touch /etc/profile.d/qmail.sh) and add the lines:

```
export MAILDIR=$HOME/Maildir/
export MAILDROP=$HOME/Maildir/
```

- Once the **qmail.csh** and the **qmail.sh** files have been created, we must be sure that their default modes are (0755/-rwxr-xr-x) and owned by the super-user 'root':

```
[root@deep /]# chmod 755 /etc/profile.d/qmail.csh
[root@deep /]# chmod 755 /etc/profile.d/qmail.sh
[root@deep /]# chown 0.0 /etc/profile.d/qmail.csh
[root@deep /]# chown 0.0 /etc/profile.d/qmail.sh
```

WARNING: If you use a special mail software like procmail, elm, pine, or qpopper, you must read the documentation that comes with qmail to know how to configure them to look at ~user/Mailbox or ~user/Maildir directly. We assume qmail-local as the default new mail software local delivery agent in this example because it's enough for the job.

Since we have configured qmail to use qmail-local for local deliveries, it's important to note that your mailbox will be moved to ~you/Mailbox or ~you/Maildir if you have decided to switch to the new Maildir feature of qmail.

Step 10
One last step to do with the new `Maildir` feature of `qmail` is to set up it as the default delivery by creating a file named `dot-qmail` under `/etc` directory. The `qmail` script initialization file reads this file each time you restart the mail server.

- Create the **`dot-qmail`** file (`touch /etc/dot-qmail`) and add the lines:

```
./Maildir/
|qbiff
```

Step 11
Since we have decided to use, and to give to, the local delivery agent of `qmail` (`qmail-local`) the task of delivering mail locally, we can remove the default Unix `mailx` package from our system and the `/var/mail` and `/var/spool/mail` directories too.

- To remove `mailx` from your system, use the following command:
  `[root@deep /]#` **`rpm -e mailx`**

- To remove `/var/mail` and `/var/spool/mail` from your system, use the commands:
  `[root@deep /]#` **`rm -rf /var/mail`**
  `[root@deep /]#` **`rm -rf /var/spool/mail`**

Also, we can remove `procmail` if this is not already done since `qmail` doesn't need it to function properly.

- To remove `procmail` from your system, use the following command:
  `[root@deep /]#` **`rpm -e procmail`**

WARNING: If you have scripts that use `/bin/mail` to send out status reports without using specific command line interface of `mailx`, you can just create a link after uninstall of `mailx` package to `/var/qmail/bin/qmail-inject` and don't worry about incompatibility. In other case, you must retain the `mailx` package on your system. To create a link to `qmail-inject` program use:

```
ln –s /var/qmail/bin/qmail-inject /bin/mail
```

Step 13
Once the compilation, optimization, and installation of the Mail Server has finished, we can free up some disk space by deleting the program tar archives and the related source directories since they are no longer needed.

- To delete the programs and its related source directories, use the following commands:
  `[root@deep /]#` **`cd /var/tmp/`**
  `[root@deep tmp]#` **`rm -rf qmail-version/`**
  `[root@deep tmp]#` **`rm -f qmail-version.tar.gz`**
  `[root@deep tmp]#` **`rm -rf ucspi-tcp-version/`**
  `[root@deep tmp]#` **`rm -f ucspi-tcp-version.tar.gz`**
  `[root@deep tmp]#` **`rm -rf checkpassword-version/`**
  `[root@deep tmp]#` **`rm -f checkpassword-version.tar.gz`**

The rm commands as used above will remove the source files we have used to compile and install qmail, ucspi-tcp and checkpassword. It will also remove the qmail, ucspi-tcp and checkpassword compressed archives from the /var/tmp directory.

Configuring qmail

After qmail has been built and installed successfully in your system, your next step is to create, configure and customize all the options and parameters in your different qmail configuration files (if necessary). Depending of the kind of Mail Server you want to run, there are different configuration files to set up, these files are:

For running qmail as a Central Mail Hub Server:

- /var/qmail/control/me (The qmail hostname Files)
- /var/qmail/control/locals (The qmail local File)
- /var/qmail/control/rcpthosts (The qmail rcpthost File)
- /etc/tcp.smtp (The qmail tcp.smtp File)
- /etc/qmqp.tcp (The qmail qmqp.tcp File)
- /var/qmail/control/defaultdomain (The qmail defaultdomain File)
- /var/qmail/control/plusdomain (The qmail plusdomain File)
- /etc/rc.d/init.d/qmail (The qmail Mail Hub Initialization File)

For running qmail as a Standalone Mail Server:

- /var/qmail/control/me (The qmail hostname Files)
- /var/qmail/control/locals (The qmail local File)
- /var/qmail/control/rcpthosts (The qmail rcpthost File)
- /var/qmail/control/defaultdomain (The qmail defaultdomain File)
- /var/qmail/control/plusdomain (The qmail plusdomain File)
- /etc/rc.d/init.d/qmail (The qmail null client Initialization File)

/var/qmail/control/me: The qmail Hostname Configuration File

All files under /var/qmail/control directory are configuration files for the qmail system. qmail can run with just one control file named 'me' which contains the fully-qualified name of the current host. This file 'me' is used as the default for other hostname-related control files. Usually you don't have to change this file 'me' since it's already contains your fully qualified domain name for qmail to work, otherwise if it doesn't exist, create it and add your fully qualified domain name (my.domain.com) inside it.

/var/qmail/control/locals: The qmail locals Configuration File

The qmail configuration file locals can be used to handle a list of domain names that the current host receives mail for, one per line. qmail will know through the content of this file which addresses it should deliver locally.

This file becomes important when you configure qmail as a Central Mail Hub Server. If you want to configure your qmail software to run as a standalone Mail Server, you will need to remove the default value in this file, which is "localhost". See later in this chapter for more information about running qmail as a standalone Mail Server.

- Edit the **locals** file (vi /var/qmail/control/locals) and add:

```
localhost
boreas.openna.com
boreas.customer.com
```

Where <boreas.openna.com> in this example is our Mail Hub Server, and <boreas.customer.com> is one of our customer for which we decide to receive and send mail on the Mail Hub server. Note that the hostname is always 'boreas' since this is the hostname where our Central Mail Hub Server lives.

/var/qmail/control/rcpthosts: The qmail rcpthosts File

This file 'rcpthosts' specifies which domains are allowed to use the qmail Mail Server. If a domain is not listed in rcpthosts, then qmail-smtpd will reject any envelope recipient address. To summarize, qmail will know through the content of this file which messages it should accept from remote systems.

By default with qmail, relaying is turned off and you must populate the rcpthosts file with the fully qualified domain of authorized hosts. As for Sendmail local-host-names file, one use for such a file might be to declare a list of hosts in your network for which the local host is acting as the MX recipient.

If you want to configure your qmail software to run as a standalone Mail Server, you don't need to change the default values in this file, which are again your FQDN "boreas.openna.com" and "localhost". See later in this chapter for more information about running qmail as a standalone Mail.

- Edit the **rcpthosts** file (vi /var/qmail/control/rcpthosts) and add:

```
localhost
openna.com
customer.com
```

Where <openna.com> represents in this example our Mail Hub Server. The <customer.com> parameter means to allow every hostnames under the domain <customer.com> to use the Mail Hub Server (boreas.openna.com) to send and receive mail.

/etc/tcp.smtp: The qmail tcp.smtp File

This section applies only if you chose to install and use qmail as a Central Mail Hub Server in your system. This file 'tcp.smtp' allow selected clients to send outgoing messages through the qmail SMTP server. Without it, only the localhost where qmail is running will be able to send outgoing messages. If you want to configure your qmail software to run as a standalone Mail Server, you don't need to create and have this file.

Step 1
As we now know, relaying is disabled by default with our configuration and to allow relaying from selected and authorized users, we will have to create a file named `tcp.smtp` under the `/etc` directory. This file will contain only the IP addresses for which we want to authorize relaying.

- Create the **`tcp.smtp`** file (`touch /etc/tcp.smtp`) and add for example:

```
207.35.78.3:allow,RELAYCLIENT=""
192.168.1.:allow,RELAYCLIENT=""
```

Where `<207.35.78.3>` and `<192.168.1.>` means to allow IP address client 207.35.78.3 and all hostnames under the private IP addresses range 192.168.1. to use `qmail` Mail Hub Server to relay mail.

Step 2
Now we must run the `tcprules` utility of `qmail`, which compiles rules for `tcpserver` and creates the appropriate database file related to information in the `tcp.smtp` file.

- To create the `tcp.smtp` database file, use the following command:

```
[root@deep /]# tcprules /etc/tcp.smtp.cdb /etc/tcp.smtp.tmp <
/etc/tcp.smtp
[root@deep /]# chmod 644 /etc/tcp.smtp*
```

NOTE: If you make any changes to `/etc/tcp.smtp` file, you must run the above `tcprules` command again.

`/etc/qmqp.tcp`: The `qmail` `qmqp.tcp` File

This section applies only if you chose to install and use `qmail` as a Central Mail Hub Server in your system. This file '`qmqp.tcp`' allow fast queuing of outgoing mail from authorized client hosts through `QMQP` that provides a centralized mail queue within a cluster of hosts.

One central server runs a message transfer agent. The other hosts (Standalone Mail Server) do not have their own mail queues (see later in this chapter the section 'Running `qmail` as an extremely secure standalone client (mini-qmail)'); they give each new message to the central server through `QMQP`. If you want to configure your `qmail` software to run as a standalone Mail Server, you don't need to create or have this file.

Step 1
The first step is to create the `qmqp.tcp` file in `tcprules` format to allow queuing from the authorized hosts. This file will contain any IP addresses for which we want to authorize queuing.

- Create the **`qmqp.tcp`** file (`touch /etc/qmqp.tcp`) and add:

```
207.35.78.:allow
:deny
```

Where `<207.35.78.:allow>` means to allow all hostnames under the IP address range 207.35.78. to use `qmail` Mail Hub Server for queuing. The `<:deny>` parameter makes sure to deny connections from unauthorized hosts.

Step 2
Now we must run the tcprules utility of qmail to convert /etc/qmqp.tcp to /etc/qmqp.cdb format.

- To create the qmqp.cdb format, use the following command:

```
[root@deep /]# tcprules /etc/qmqp.cdb /etc/qmqp.tmp < /etc/qmqp.tcp
[root@deep /]# chmod 644 /etc/qmqp.*
```

NOTE: If you make any changes to the /etc/qmqp.tcp file, you must run the above tcprules command again.

Step 3
After that, edit the /etc/services file and add an entry for qmail-qmqpd on port 628. This port number doesn't exist for qmail and we must add it to the list.

- Edit the **services** file (vi /etc/services), and add the line:

```
qmail-qmqpd     628/tcp         QMQP: Quick Mail Queueing Protocol
```

Step 4
Finally, it's important to allow traffic through port 628 into our firewall script file for the qmail-qmqpd daemon to work properly in the system.

- Edit the **iptables** script file (vi /etc/rc.d/init.d/iptables), and add/check the following lines to allow qmail-qmqpd packets to traverse the network:

```
# QMQP server (628)
# ----------------

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR --source-port $UNPRIVPORTS \
         -d $PRIVATENET --destination-port 628 -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
         -s $PRIVATENET --source-port 628 \
         -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT
```

Where EXTERNAL_INTERFACE="eth0" # Internet connected interface
Where IPADDR="207.35.78.4" # Your IP address for eth0
Where PRIVATENET="207.35.78.0/32" # IP ranges assigned by your ISP
Where UNPRIVPORTS="1024:65535" # Unprivileged port range

WARNING: Note that QMQP is not a public service. Servers should not accept QMQP connections from unauthorized IP addresses.

/var/qmail/control/defaultdomain: The qmail defaultdomain File

The defaultdomain file is used by qmail to add the domain name listed in the file (defaultdomain) to any host name without dots. Usually you don't have to change the default information (i.e. openna.com) listed in this file (defaultdomain).

/var/qmail/control/plusdomain: The qmail plusdomain File

The plusdomain file is used by qmail to add the domain name listed in the file (plusdomain) to any host name that ends with a plus sign. Usually you don't have to change the default information (i.e. openna.com) listed in this file (plusdomain).

/etc/rc.d/init.d/qmail: The qmail Initialization for File Mail Hub

This section applies only if you chose to install and use qmail as a Central Mail Hub Server in your system. The /etc/rc.d/init.d/qmail script file is responsible to automatically start and stop all the required qmail daemons on your server.

Step 1
Create the **qmail** script file (touch /etc/rc.d/init.d/qmail) and add the following lines:

```
#! /bin/sh
#
# qmail        This starts and stops qmail.
#
# chkconfig: 2345 80 30
# description:  qmail is a small, fast, secure replacement \
#               for the sendmail package, which is the \
#               program that actually receives, routes, \
#               and delivers electronic mail. \
#
# config: /etc/sysconfig/network

PATH=/sbin:/bin:/usr/bin:/usr/sbin

# Source function library.
. /etc/init.d/functions

# Get config.
test -f /etc/sysconfig/network && . /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "yes" ] || exit 0

[ -f /var/qmail/bin/qmail-send ] || exit 1

RETVAL=0

start(){
    echo -n "Starting qmail: "
    exec env - PATH="/var/qmail/bin:$PATH" \
        qmail-start "`cat /etc/dot-qmail`" splogger qmail &
    tcpserver -p -v -c 400 -x /etc/tcp.smtp.cdb -u 81 -g 81 0 smtp \
         rblsmtpd -rrelays.orbs.org -rrbl.maps.vix.com \
         /var/qmail/bin/qmail-smtpd 2>&1 | \
         /var/qmail/bin/splogger smtpd 3 &
    tcpserver 0 110 /var/qmail/bin/qmail-popup \
```

```
        `hostname` /bin/checkpassword \
        /var/qmail/bin/qmail-pop3d Maildir &
    tcpserver -x /etc/qmqp.cdb -u 81 -g 81 0 628 qmail-qmqpd &
    RETVAL=$?
    echo
    touch /var/lock/subsys/qmail
    return $RETVAL
}

stop(){
    echo -n "Stopping qmail: "
    killproc qmail-send
    killproc tcpserver
    RETVAL=$?
    echo
    rm -f /var/lock/subsys/qmail
    return $RETVAL

}

restart(){
    stop
    start
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status qmail-send
        ;;
    restart)
        restart
        ;;
    reload)
        reload
        ;;
    *)
        echo "Usage: qmail {start|stop|status|restart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

WARNING: Many security and optimization features of `qmail` need to be added to the initialization script file with the `tcpserver`. For this reason I decided to explain each one here to simplify interpretation.

Adding the lines "`2>&1 | /var/qmail/bin/splogger smtpd 3 &`" at the end of the "`tcpserver -v -c 400 -x /etc/tcp.smtp.cdb -u 81 -g 81 0 smtp /var/qmail/bin/qmail-smtpd`" line in the initialization script of `qmail` will keep track of who's connecting and for how long.

By default, `tcpserver` allows at most 40 simultaneous `qmail-smtpd` processes. To raise this limit, we add the command "`-c 400`" to the above script file to set it to 400 simultaneous processes.

Adding the `-p` option to your startup script file will reject `SMTP` connections at the network level from hosts with bad DNS. This is one way to cut down on e-mail from hosts that have misconfigured their DNS, and therefore are thought by some to be more likely to be spam-friendly.

In the `ucspi-tcp` package there is a program named '`rblsmtpd`' that can be used with `qmail SMTP` daemon to reject known spammers. Adding a parameter like '`rblsmtpd -rrelays.orbs.org -rrbl.maps.vix.com`' to your startup script file will use the ORBS database in addition to the RBL to reject know spammers. Note that ORBS and RBL are 'Real-time Third-Party Blocking Solutions' see at http://www.orbs.org/ and http://rblcheck.sourceforge.net/ for more information.

Step 2
Once the `qmail` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/qmail
[root@deep /]# chown 0.0 /etc/rc.d/init.d/qmail
```

- To create the symbolic `rc.d` links for `qmail`, use the following commands:
```
[root@deep /]# chkconfig --add qmail
[root@deep /]# chkconfig --level 2345 qmail on
```

- To start `qmail` software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/qmail start
Starting qmail:               [OK]
```

/etc/rc.d/init.d/qmail: The qmail Initialization File for null client

This section applies only if you chose to install and use qmail as a Standalone Mail Server (null client) in your system. The /etc/rc.d/init.d/qmail script file is responsible to automatically start and stop all the qmail daemons on your server.

Step 1
Create the **qmail** script file (touch /etc/rc.d/init.d/qmail) and add the following lines:

```
#! /bin/sh
#
# qmail        This starts and stops qmail.
#
# chkconfig: 2345 80 30
# description:  qmail is a small, fast, secure replacement \
#               for the sendmail package, which is the \
#               program that actually receives, routes, \
#               and delivers electronic mail. \
#
# config: /etc/sysconfig/network

PATH=/sbin:/bin:/usr/bin:/usr/sbin

# Source function library.
. /etc/init.d/functions

# Get config.
test -f /etc/sysconfig/network && . /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "yes" ] || exit 0

[ -f /var/qmail/bin/qmail-send ] || exit 1

RETVAL=0

start(){
    echo -n "Starting qmail: "
    exec env - PATH="/var/qmail/bin:$PATH" \
        qmail-start "`cat /etc/dot-qmail`" splogger qmail &
    RETVAL=$?
    echo
    touch /var/lock/subsys/qmail
    return $RETVAL
}

stop(){
    echo -n "Stopping qmail: "
    killproc qmail-send
    RETVAL=$?
    echo
    rm -f /var/lock/subsys/qmail
    return $RETVAL

}

restart(){
    stop
    start
}

# See how we were called.
```

```
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status qmail-send
        ;;
    restart)
        restart
        ;;
    reload)
        reload
        ;;
    *)
        echo "Usage: qmail {start|stop|status|restart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

Step 2
Once the `qmail` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/qmail
[root@deep /]# chown 0.0 /etc/rc.d/init.d/qmail
```

- To create the symbolic `rc.d` links for `qmail`, use the following commands:
```
[root@deep /]# chkconfig --add qmail
[root@deep /]# chkconfig --level 2345 qmail on
```

- To start `qmail` software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/qmail start
Starting qmail:              [OK]
```

NOTE: All software we describe in this book has a specific directory and subdirectory in the tar compressed archive named `floppy-2.0.tgz` containing configuration files for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files manually or cut and paste them to create or change your configuration files. Whether you decide to copy manually or get the files made for your convenience from the archive compressed files, it will be to your responsibility to modify them to adjust for your needs, and place the files related to this software to the appropriate places on your server. The server configuration file archive to download is located at the following Internet address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz.

Running qmail as a standalone null client

This section applies only if you chose to install and use qmail as a Standalone Mail Server in your system. As for Sendmail null client setup (null.mc), we can configure qmail local clients machines to never receive mail directly from the outside world, but to relay (send) all their mail through a centralized mail service known as a Mail Hub Server. Contrary to the Sendmail setup, which requires a special macro file named null.mc, qmail in its default install can easily be configured to run as a standalone mail server. Configuring qmail to run into this null client configuration mode will work with any kind of Central Mail Hub Server in the other side.

Step 1
Here we need to set up the null client of qmail to send all local mail to the Central Mail Hub (i.e. boreas.openna.com). To do it we need to create a new file named **smtproutes** under /var/qmail/control directory and add inside this file the FQDN or domain name of the remote Mail Hub which handle all mail for our null client Mail Server. The "**:**" mean to transfer all outgoing mail through "openna.com" domain name.

- To create the smtproutes file, use the following commands:

```
[root@deep /]# echo :smtp.openna.com > /var/qmail/control/smtproutes
[root@deep /]# chmod 644 /var/qmail/control/smtproutes
```

In the above example, <:smtp.openna.com> is the domain name of our Central Mail Hub Server where we want to send all outgoing mail messages.

Step 2
Now it's important to stop our local null client Mail Server delivering mail locally. This is important since we want to forward all local mail to the Mail Hub Server. The solution is to remove the "localhost" entry into the /var/qmail/control/locals file on null client server.

- Edit the **locals** file (vi /var/qmail/control/locals), and remove the line:

```
localhost
```

WARNING: It's important to be sure that the MX record is set up properly in your DNS (**D**omain **N**ame **S**erver) server before you do this. Also be sure that ucspi-tcp and checkpassword packages are not installed. A qmail null client doesn't need those programs.

Step 3
Finally, it's important to restart qmail null client Mail Server for the changes to take effect.

- To restart qmail, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/qmail restart
Stopping qmail:          [OK]
Starting qmail:          [OK]
```

NOTE: Don't forget to use the null client initialization script file of qmail and not the regular one for this setup. This is important since qmail-smtpd and qmail-popd3 must be turned off in this configuration. We don't need to have those daemons running and open in background.

Running qmail with SSL support

There is a patch, which implements RFC2487 in qmail. This means you can get SSL or TLS encrypted and authenticated SMTP between the MTAs and between MTA and an MUA like Netscape or MS Outlook. The code is considered **experimental** and can be found at: http://www.esat.kuleuven.ac.be/~vermeule/qmail/tls.patch

Securing qmail

This section deals especially with actions we can make to improve and tighten security under qmail. Related to its highly secure nature, there is not lot of things we can do here if it's not to improve security for its null client part.

Running qmail as an extremely secure standalone client (mini-qmail)

This section applies only if you chose to install and use qmail as a Standalone Mail Server in your system. We have already and successfully configured qmail to run as a standalone Mail Server previously. Here we'll show you how to run it in a much more secure manner. The difference with the previous null client configuration of qmail is the fact that in this set up, the Central Mail Hub Server at the other side must be a qmail server. This kind of configuration is known as a mini-qmail installation. A mini-qmail installation doesn't have a mail queue; instead it gives each new message to a central server through QMQP (see further up under the section '/etc/qmqp.tcp: The qmail qmqp.tcp File' for more information about QMQP).

Step 1
With qmail running as a standalone mail server under a mini-qmail configuration, you don't need /var/qmail/alias. A mini-qmail installation doesn't do any local delivery.

- To remove the /var/qmail/alias directory, use the following command:
```
[root@deep /]# rm -rf /var/qmail/alias
```

Step 2
A null client don't need qmail entries in /etc/group and /etc/passwd. A mini-qmail runs with the same privileges as the user sending mail; it doesn't have any of its own files.

- Remove all qmail users and groups from your system with the following commands:
```
[root@deep /]# userdel alias
[root@deep /]# userdel qmaild
[root@deep /]# userdel qmaill
[root@deep /]# userdel qmailp
[root@deep /]# userdel qmailq
[root@deep /]# userdel qmailr
[root@deep /]# userdel qmails
[root@deep /]# groupdel qmail
[root@deep /]# groupdel nofiles
```

Step 3
A mini-qmail installation doesn’t need to start anything from the boot scripts. A null client doesn't have a queue, so it doesn't need a long-running queue manager and doesn't receive incoming mail.

- Deactive the `/etc/rc.d/init.d/qmail` initialization file with the following commands:

```
[root@deep /]# chkconfig --level 2345 qmail off
[root@deep /]# chkconfig --del qmail
[root@deep /]# rm -f /etc/rc.d/init.d/qmail
```

Step 4
Since we run a highyl secure and fast null client, there are many `qmail` binaries that we can remove from the `/var/qmail/bin` directory of the system.

- Remove all non needed `qmail` binaries from the system with the following commands:

```
[root@deep /]# rm -f /var/qmail/bin/bouncesaying
[root@deep /]# rm -f /var/qmail/bin/condredirect
[root@deep /]# rm -f /var/qmail/bin/except
[root@deep /]# rm -f /var/qmail/bin/preline
[root@deep /]# rm -f /var/qmail/bin/qbiff
[root@deep /]# rm -f /var/qmail/bin/qmail-clean
[root@deep /]# rm -f /var/qmail/bin/qmail-getpw
[root@deep /]# rm -f /var/qmail/bin/qmail-local
[root@deep /]# rm -f /var/qmail/bin/qmail-lspawn
[root@deep /]# rm -f /var/qmail/bin/qmail-newmrh
[root@deep /]# rm -f /var/qmail/bin/qmail-newu
[root@deep /]# rm -f /var/qmail/bin/qmail-pop3d
[root@deep /]# rm -f /var/qmail/bin/qmail-popup
[root@deep /]# rm -f /var/qmail/bin/qmail-pw2u
[root@deep /]# rm -f /var/qmail/bin/qmail-qmqpd
[root@deep /]# rm -f /var/qmail/bin/qmail-queue
[root@deep /]# rm -f /var/qmail/bin/qmail-remote
[root@deep /]# rm -f /var/qmail/bin/qmail-rspawn
[root@deep /]# rm -f /var/qmail/bin/qmail-qmtpd
[root@deep /]# rm -f /var/qmail/bin/qmail-send
[root@deep /]# rm -f /var/qmail/bin/qmail-smtpd
[root@deep /]# rm -f /var/qmail/bin/qmail-start
[root@deep /]# rm -f /var/qmail/bin/qmail-tcpok
[root@deep /]# rm -f /var/qmail/bin/qmail-tcpto
[root@deep /]# rm -f /var/qmail/bin/qreceipt
[root@deep /]# rm -f /var/qmail/bin/qsmhook
[root@deep /]# rm -f /var/qmail/bin/splogger
[root@deep /]# rm -f /var/qmail/bin/tcp-env
```

NOTE: Be sure that `ucspi-tcp` and `checkpassword` packages are not installed. A mini-qmail configuration doesn’t need these programs.

Step 5
One of the last steps to do is to create a symbolic link to `qmail-qmqpc` from `/var/qmail/bin/qmail-queue`. The `qmail-qmqpc` offers the same interface as `qmail-queue`, but it gives the message to a `QMQP` server instead of storing it locally.

- To create the symbolic link, use the following command:

```
[root@deep /]# cd /var/qmail/bin
[root@deep /]# ln -s qmail-qmqpc /var/qmail/bin/qmail-queue
```

Step 6
Now, it's important to create a list of IP addresses of QMQP servers, one per line, in /var/qmail/control/qmqpservers. The qmail-qmqpc utility will try each address in turn until it establishes a QMQP connection or runs out of addresses.

- To create a list of IP addresses of QMQP servers, use the following command:
  [root@deep /]# **touch /var/qmail/control/qmqpservers**

- Edit the **qmqpservers** file (vi /var/qmail/control/qmqpservers), and add the FQDN of your Mail Hub Server in the list:

  boreas.openna.com

NOTE: In this example, we assume that you have only one Central Mail Hub Server located at boreas.openna.com. if you handle more than one Mail Hub Server, then don't forget to add its FQDN (Fully Qualified Domain Name) in the list (one per line).

Step 7
After that, you need a copy of /var/qmail/control/me, /var/qmail/control/defaultdomain, and /var/qmail/control/plusdomain files from your qmail Central Mail Hub Server, so that the qmail-inject program uses appropriate host names in outgoing mail. You must copy these qmail files from the remote Central Mail Hub Server to your /var/qmail/control directory on the local null client mini-qmail Mail Server.

WARNING: It's important that the remote Mail Hub is a qmail Mail Hub Server and not a Sendmail Mail Hub Server or you will not be able to get those required files for the local null client mini-qmail Mail server. Remember that Sendmail don't use the same file that qmail use.

Step 8
Finally, we must create a new file named **idhost** under /var/qmail/control directory on the mini-qmail Mail server which will contain its host's name, so that qmail-inject program generates Message-ID without any risk of collision.

- Create the **idhost** file (touch /var/qmail/control/idhost), and add:

  cronus.openna.com

Where <cronus.openna.com> is a fully-qualified name within the domain. Therefore, don't forget to put in this file your own current host's name of your mini-qmail Mail server.

Further documentation

For more details about `qmail` program, there are several manual pages you can read. I highly recommend you to take the time and run through them. By doing this, you'll be more comfortable with the way `qmail` work.

`$ man bouncesaying (1)`	bounce each incoming message
`$ man condredirect (1)`	redirect mail to another address
`$ man except (1)`	reverse the exit code of a program
`$ man forward (1)`	forward new mail to one or more addresses
`$ man maildir2mbox (1)`	move mail from a maildir to an mbox
`$ man maildirmake (1)`	create a maildir for incoming mail
`$ man maildirwatch (1)`	look for new mail in a maildir
`$ man mailsubj (1)`	send a mail message with a subject line
`$ man preline (1)`	prepend lines to message
`$ man qbiff (1)`	announce new mail the moment it arrives
`$ man qreceipt (1)`	respond to delivery notice requests
`$ man tcp-env (1)`	set up TCP-related environment variables
`$ man addresses (5)`	formats for Internet mail addresses
`$ man mbox (5)`	file containing mail messages
`$ man dot-qmail (5)`	control the delivery of mail messages
`$ man envelopes (5)`	sender/recipient lists attached to messages
`$ man maildir (5)`	directory for incoming mail messages
`$ man qmail-control (5)`	qmail configuration files
`$ man qmail-header (5)`	format of a mail message
`$ man qmail-log (5)`	the qmail activity record
`$ man qmail-users (5)`	assign mail addresses to users
`$ man tcp-environ (5)`	TCP-related environment variables
`$ man forgeries (7)`	how easy it is to forge mail
`$ man qmail (7)`	overview of qmail documentation
`$ man qmail-limits (7)`	artificial limits in the qmail system
`$ man qmail-newu (8)`	prepare address assignments for qmail-lspawn
`$ man qmail-command (8)`	user-specified mail delivery program
`$ man qmail-getpw (8)`	give addresses to users
`$ man qmail-inject (8)`	preprocess and send a mail message
`$ man qmail-local (8)`	deliver or forward a mail message
`$ man qmail-lspawn (8)`	schedule local deliveries
`$ man qmail-newmrh (8)`	prepare morercpthosts for qmail-smtpd
`$ man qmail-pop3d (8)`	distribute mail via POP
`$ man qmail-popup (8)`	read a POP username and password
`$ man qmail-pw2u (8)`	build address assignments from a passwd file
`$ man qmail-qmqpc (8)`	queue a mail message via QMQP
`$ man qmail-qmqpd (8)`	receive mail via QMQP
`$ man qmail-qmtpd (8)`	receive mail via QMTP
`$ man qmail-send (8)`	deliver mail messages from the queue
`$ man qmail-qread (8)`	list outgoing messages and recipients
`$ man qmail-qstat (8)`	summarize status of mail queue
`$ man qmail-queue (8)`	queue a mail message for delivery
`$ man qmail-remote (8)`	send mail via SMTP
`$ man qmail-rspawn (8)`	schedule remote deliveries
`$ man qmail-showctl (8)`	analyze the qmail configuration files
`$ man qmail-smtpd (8)`	receive mail via SMTP
`$ man qmail-start (8)`	turn on mail delivery
`$ man qmail-tcpok (8)`	clear TCP timeout table
`$ man qmail-tcpto (8)`	print TCP timeout table
`$ man splogger (8)`	make entries in syslog

qmail Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages and documentation of qmail for more information.

qmail-showctl

This command utility allows you to analyze your existing qmail configuration files on the system and explains the current qmail configuration. It can be useful when you want to verify if modifications made to your configuration files have been updated by the system.

- To run qmail-showctl, use the following command:

```
[root@deep /]# /var/qmail/bin/qmail-showctl
qmail home directory: /var/qmail.
user-ext delimiter: -.
paternalism (in decimal): 2.
silent concurrency limit: 120.
subdirectory split: 23.
user ids: 82, 81, 86, 0, 87, 83, 84, 85.
group ids: 81, 82.

badmailfrom: (Default.) Any MAIL FROM is allowed.
bouncefrom: (Default.) Bounce user name is MAILER-DAEMON.
bouncehost: (Default.) Bounce host name is boreas.openna.com.
concurrencylocal: (Default.) Local concurrency is 10.
concurrencyremote: (Default.) Remote concurrency is 20.
databytes: (Default.) SMTP DATA limit is 0 bytes.
defaultdomain: Default domain name is openna.com.
defaulthost: (Default.) Default host name is boreas.openna.com.
doublebouncehost: (Default.) 2B recipient host: boreas.openna.com.
doublebounceto: (Default.) 2B recipient user: postmaster.
envnoathost: (Default.) Presumed domain name is boreas.openna.com.
helohost: (Default.) SMTP client HELO host name is boreas.openna.com.
idhost: (Default.) Message-ID host name is boreas.openna.com.
localiphost: (Default.) Local IP address becomes boreas.openna.com.

locals:
Messages for localhost are delivered locally.
Messages for boreas.openna.com are delivered locally.
me: My name is boreas.openna.com.
percenthack: (Default.) The percent hack is not allowed.
plusdomain: Plus domain name is openna.com.
qmqpservers: (Default.) No QMQP servers.
queuelifetime: (Default.) Message lifetime in the queue is 604800 sec.

rcpthosts:
SMTP clients may send messages to recipients at localhost.
SMTP clients may send messages to recipients at boreas.openna.com.
SMTP clients may send messages to recipients at .openna.com.
SMTP clients may send messages to recipients at .customer.com.

morercpthosts: (Default.) No effect.
morercpthosts.cdb: (Default.) No effect.
smtpgreeting: (Default.) SMTP greeting: 220 boreas.openna.com.
smtproutes: (Default.) No artificial SMTP routes.
timeoutconnect: (Default.) SMTP client connection timeout is 60 seconds.
timeoutremote: (Default.) SMTP client data timeout is 1200 seconds.
timeoutsmtpd: (Default.) SMTP server data timeout is 1200 seconds.
virtualdomains: (Default.) No virtual domains.
```

qmail-qread
This command utility is used to list outgoing messages and recipients on the system in human-readable format. If you want to see your queue messages in the system, then you must use the `qmail-qread` command. `qmail-qread` scans the queue for messages that haven't been completely delivered yet. If a message has multiple recipients, it's not unusual for some of the recipients to receive the message before others.

- To scans the outgoing queue of messages, use the following command:
```
[root@deep /]# qmail-qread
```

NOTE: If you want to process `qmail` queues manually, you can send an `ALRM` signal to `qmail-send` daemon to have it run through everything in the queue immediately.
E.g., "`killall -ALRM qmail-send`"

qmail-qstat
The `qmail-qstat` command gives a human-readable breakdown of the number of messages at various stages in the mail queue. To summarize, it summarizes the status of your mail queue.

- To see the status of your mail queue, use the following command:
```
[root@deep /]# qmail-qstat
messages in queue: 0
messages in queue but not yet preprocessed: 0
```

`qmail` Users Tools
The commands listed below are some that we use often, but many more exist. Check the manual page and documentation of `qmail` for more information.

maildirwatch
The "`maildirwatch`" program is used to look for new users mail in a maildir inside terminal screem. This is the program we use to replace the `mailx` package we have uninstalled previously durring installation of `qmail`. Recall that the `maildirwatch` tool is more reliable, fast and secure then `mailx`.

NOTE: If you receive an error message like: `maildirwatch: fatal: MAILDIR not set`

It is because you have forgotten to "give it" the `MAILDIR` variable, for instance:
`export MAILDIR=$HOME/Maildir`

List of installed `qmail` files on your system

> /etc/skel/Maildir
> /etc/skel/Maildir/tmp
> /etc/skel/Maildir/new
> /etc/skel/Maildir/cur
> /etc/tcp.smtp
> /etc/tcp.smtp.cdb
> /etc/qmqp.tcp
> /etc/qmqp.cdb
> /etc/dot-qmail
> /usr/bin/maildir2mbox
> /usr/bin/mailq
> /usr/bin/maildirmake
> /usr/bin/maildirwatch
> /usr/bin/qmail-qread
> /usr/bin/qmail-qstat
> /usr/lib/sendmail
> /usr/share/man/man1/bouncesaying.1
> /usr/share/man/man1/condredirect.1
> /usr/share/man/man1/except.1
> /usr/share/man/man1/forward.1
> /usr/share/man/man1/maildir2mbox.1
> /usr/share/man/man1/maildirmake.1
> /usr/share/man/man1/maildirwatch.1
> /usr/share/man/man1/mailsubj.1
> /usr/share/man/man1/preline.1
> /usr/share/man/man1/qbiff.1
> /usr/share/man/man1/qreceipt.1
> /usr/share/man/man1/tcp-env.1
> /usr/share/man/man5/addresses.5
> /usr/share/man/man5/mbox.5
> /usr/share/man/man5/dot-qmail.5
> /usr/share/man/man5/envelopes.5
> /usr/share/man/man5/maildir.5
> /usr/share/man/man5/qmail-control.5
> /usr/share/man/man5/qmail-header.5
> /usr/share/man/man5/qmail-log.5
> /usr/share/man/man5/qmail-users.5
> /usr/share/man/man5/tcp-environ.5
> /usr/share/man/man7/forgeries.7
> /usr/share/man/man7/qmail.7
> /usr/share/man/man7/qmail-limits.7
> /usr/share/man/man8/qmail-newu.8
> /usr/share/man/man8/qmail-command.8
> /usr/share/man/man8/qmail-getpw.8
> /usr/share/man/man8/qmail-inject.8
> /usr/share/man/man8/qmail-local.8
> /usr/share/man/man8/qmail-lspawn.8
> /usr/share/man/man8/qmail-newmrh.8
> /usr/share/man/man8/qmail-pop3d.8
> /usr/share/man/man8/qmail-popup.8
> /usr/share/man/man8/qmail-pw2u.8
> /usr/share/man/man8/qmail-qmqpc.8
> /usr/share/man/man8/qmail-qmqpd.8
> /usr/share/man/man8/qmail-qmtpd.8
> /usr/share/man/man8/qmail-send.8
> /usr/share/man/man8/qmail-qread.8
> /usr/share/man/man8/qmail-qstat.8
> /usr/share/man/man8/qmail-queue.8
> /usr/share/man/man8/qmail-remote.8
> /usr/share/man/man8/qmail-rspawn.8
> /usr/share/man/man8/qmail-showctl.8
> /usr/share/man/man8/qmail-smtpd.8
> /usr/share/man/man8/qmail-start.8
> /usr/share/man/man8/qmail-tcpok.8
> /usr/share/man/man8/qmail-tcpto.8
> /usr/share/man/man8/splogger.8
> /var/qmail

List of installed `ucspi-tcp` files on your system

> /usr/bin/tcpserver
> /usr/bin/tcprules
> /usr/bin/tcprulescheck
> /usr/bin/argv0
> /usr/bin/recordio
> /usr/bin/who@
> /usr/bin/tcpclient
> /usr/bin/date@
> /usr/bin/finger@
> /usr/bin/http@
> /usr/bin/tcpcat
> /usr/bin/mconnect
> /usr/bin/mconnect-io
> /usr/bin/addcr
> /usr/bin/delcr
> /usr/bin/fixcrio
> /usr/bin/rblsmtpd

List of installed `checkpassword` files on your system

> /bin/checkpassword

Part IX Internet Message Access Protocol Related Reference

In this Part

Internet Message Access Protocol - `UW IMAP`

An Internet Message Access Protocol server provides access to personal mail and system-wide bulletin boards. It is the software that runs in the background and allows users, which use a **M**ail **U**ser **A**gent (`MUA`) program like `Netscape Messenger` or `MS Outlook` to transparently access and read mail on the server. It is important to note that an Internet Message Access Protocol server is not required on all servers but only on a mail server that runs as a Central Mail Hub Server. It is not every **M**ail **T**ransfer **A**gent (`MTA`) which can run with `UW IMAP`, this is especially true for `qmail`. If you have installed `Sendmail` as a Mail Hub Server, then you must install an Internet Message Access Protocol server like `UW IMAP` to let users access and read mail on the `Sendmail` Central Mail Hub Server. In the other hand, if you have installed `qmail` as your Central Mail Hub Server, then you can skip this part of the book and continuing your reading.

22 Internet Message Access Protocol - UW IMAP

In this Chapter

Linux UW IMAP Servers

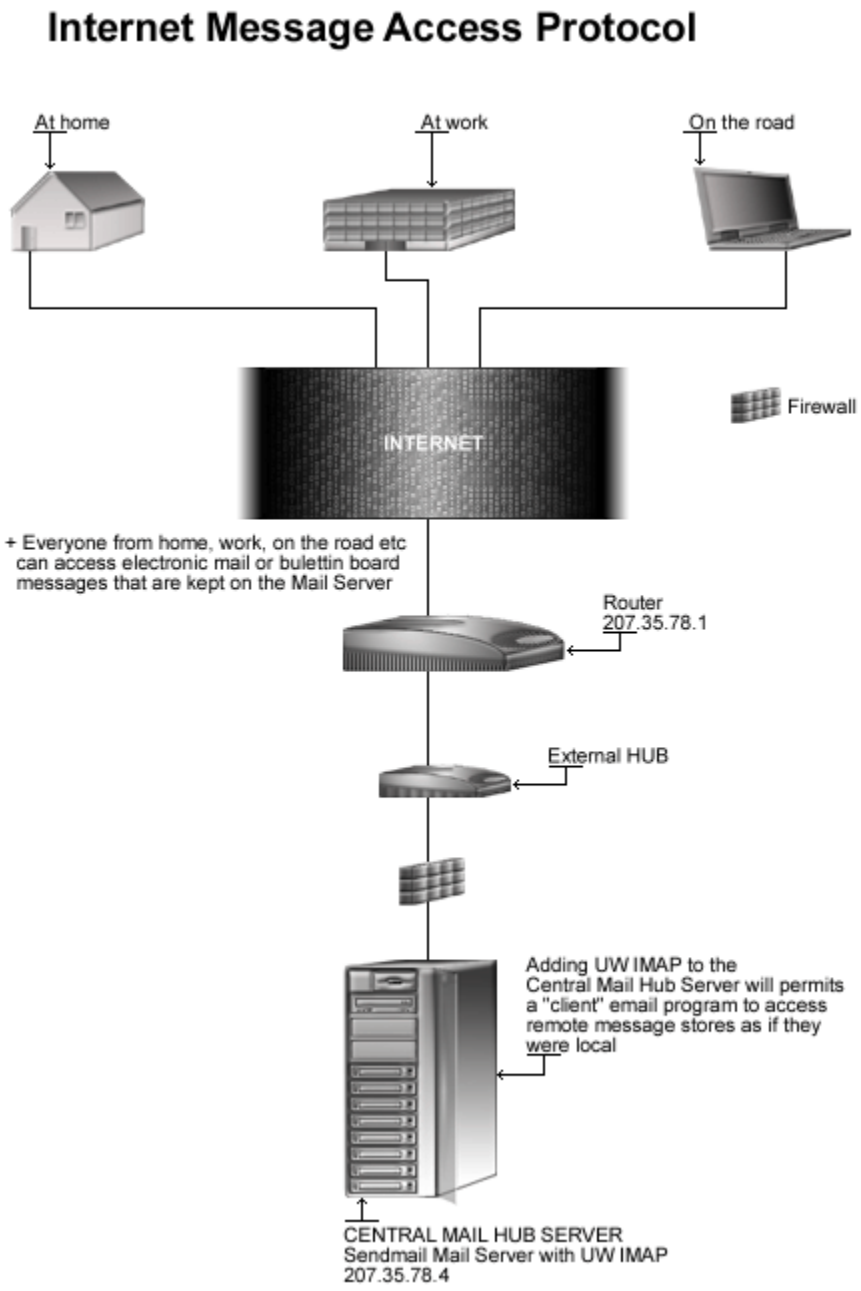
Abstract

imap-2001 is a major release, it now supports SSL client functionality for IMAP, POP3, SMTP, and NNTP; With this new release of the UW IMAP software you don't need any separate SSL modules anymore, that's why I recommend it. If you have configured Sendmail as a Central Mail Hub Server, you must install UW IMAP software or you'll not be able to use the advantage of your Linux Mail Server, since Sendmail is just software that sends mail from one machine to another, and nothing else. A mail server is a server that is running one or more of the following: an IMAP server, a POP3 server, a POP2 server, or an SMTP server. An example of SMTP server is Sendmail that must be already installed on your Linux server as a Central Mail Hub before continuing with this part of the book. For now, we are going to cover installing IMAP4, POP3, and POP2, which all come in a single package.

With UW IMAP software, a remote "client" email program can access message stored on the Linux mail server as if they were local. For example, email received and stored on an IMAP server for a user can be manipulated from his/her computer at home, office, etc, without the need to transfer messages or files back and forth between these computers.

POP stands for "**P**ost **O**ffice **P**rotocol" and simply allows you to list messages, retrieve them, and delete them. IMAP that stands for (**I**nternet **M**essage **A**ccess **P**rotocol) is POP on steroids. It allows you to easily maintain multiple accounts, have multiple people access one account, leave mail on the server, just download the headers, or bodies, no attachments, and so on. IMAP is ideal for anyone on the go, or with serious email needs. The default POP and IMAP servers that most distributions ship fulfill most needs and with the addition of SSL capability UW IMAP become now a very powerful, strong and secure program.

Disclaimer


Internet Message Access Protocol
At home
At work
On the road
INTERNET
Firewall
+ Everyone from home, work, on the road etc can access electronic mail or bulettin board messages that are kept on the Mail Server
Router
207.35.78.1
External HUB
Adding UW IMAP to the Central Mail Hub Server will permits a "client" email program to access remote message stores as if they were local
CENTRAL MAIL HUB SERVER
Sendmail Mail Server with UW IMAP
207.35.78.4

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest UW IMAP version number is 2001

Packages

The following is based on information as listed by UW IMAP as of 2001/03/25. Please regularly check at www.washington.edu/imap/ for the latest status.

Source code is available from:
UW IMAP Homepage: http://www.washington.edu/imap/
UW IMAP FTP Site: 140.142.3.227, 140.142.4.227
You must be sure to download: imap-2001.BETA.tar.Z

Prerequisites

UW IMAP requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install them from source archive files. Please make sure you have all of these programs installed on your machine before you proceed with this chapter.

- ✓ OpenSSL, which enables support for SSL functionality, must already be installed on your system to be able to use the IMAP & POP SSL features.
- ✓ Xinetd must already be installed on your system to be able to control, start, and stop the IMAP & POP servers.
- ✓ Sendmail should be already installed on your system to be able to use UW IMAP.

NOTE: For more information on the required software, see the related chapters in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install UW IMAP, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > IMAP1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > IMAP2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff IMAP1 IMAP2 > IMAP-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing UW IMAP

Below are the required steps that you must make to compile and optimize the `UW IMAP` software before installing it into your Linux system. There are some files to modify by specifying the installation paths, compilation and optimizations flags for the Linux system. We must hack those files to be compliant with our Linux file system structure and install/optimize `UW IMAP` under our PATH Environment variable.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp imap-version.tar.Z /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf imap-version.tar.Z
```

Step 2
After that, move into the newly created `UW IMAP` directory and perform the following steps before compiling and optimizing it.

- To move into the newly created `UW IMAP` directory use the following command:

```
[root@deep tmp]# cd imap-2001/
```

Step 3
It is important to set our optimization flags for the compilation of `UW IMAP` software on the server and change some default installation path to reflect our environment under Linux.

- Edit the **`Makefile`** file (`vi +425 src/osdep/unix/Makefile`) and change the line:

```
BASECFLAGS="-g -fno-omit-frame-pointer -O6" \
```

To read:

```
BASECFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fno-omit-frame-
pointer" \
```

NOTE: You will see many identical or similar lines related to different operating systems in this file. The one, which interests us here, is named "`lnp`" for Linux Pluggable Authentication modules. It is under this section that we must change the above line. This is important since from release to release this line might change with the addition of new code.

- Edit the **Makefile** file (`vi +100 src/osdep/unix/Makefile`) and change the line:

  ```
  CC=cc
  ```

  To read:

  ```
  CC=gcc
  ```

NOTE: Pay special attention to the compile `BASECFLAGS` line above. We optimize `UW IMAP` for an i686 CPU architecture with the parameter "`-march=i686` and `-mcpu=i686`". Please don't forget to adjust the above optimization `FLAGS` to reflect your own system and CPU architecture.

- Edit the **Makefile** file (`vi +72 src/osdep/unix/Makefile`) and change the lines:

  ```
  ACTIVEFILE=/usr/lib/news/active
  ```

  To read:

  ```
  ACTIVEFILE=/var/lib/news/active
  ```

  ```
  SPOOLDIR=/usr/spool
  ```

  To read:

  ```
  SPOOLDIR=/var/spool
  ```

  ```
  RSHPATH=/usr/ucb/rsh
  ```

  To read:

  ```
  RSHPATH=/usr/bin/rsh
  ```

  ```
  LOCKPGM=/etc/mlock
  ```

  To read:

  ```
  #LOCKPGM=/etc/mlock
  ```

NOTE: The "`ACTIVEFILE=`" line specifies the path of the "`active`" directory for `UW IMAP`, the "`SPOOLDIR=`" is where reside the "`spool`" directory of Linux `UW IMAP`, and the "`RSHPATH=`" specify the path for "`rsh`" directory on our system. It's important to note that we don't use `rsh` services on our server, but even so, we specify the right directory to "`rsh`".

Step 4
This section applies only if you want to run IMAP & POP servers through SSL connection. The default installation of UW IMAP assumes that OpenSSL, which is required for IMAP/POP with SSL, support has been built under /usr/local/ssl directory, but because we have a non-standard installation (OpenSSL is under /usr/share/ssl, /usr/lib and /usr/include/openssl directories), we must modify the src/osdep/unix/Makefile file to point to the appropriate locations.

- Edit the **Makefile** file (vi +31 src/osdep/unix/Makefile) and change the lines:

  SSLDIR=/usr/local/ssl

  To read:

  SSLDIR=/usr/share/ssl

  SSLINCLUDE=$(SSLDIR)/include

  To read:

  SSLINCLUDE=$(SSLDIR)/../../include

  SSLLIB=$(SSLDIR)/lib

  To read:

  SSLLIB=$(SSLDIR)/../../lib

Step 5
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install UW IMAP in the server:

```
[root@deep imap-2001]# make lnp SSLTYPE=unix
[root@deep imap-2001]# cd
[root@deep /root]# find /* > IMAP1
[root@deep /root]# cd /var/tmp/imap-2001/
[root@deep imap-2001]# install –m444 ./src/ipopd/ipopd.8c
/usr/share/man/man8/ipopd.8c
[root@deep imap-2001]# install –m444 ./src/imapd/imapd.8c
/usr/share/man/man8/imapd.8c
[root@deep imap-2001]# install -s -m755 ./ipopd/ipop3d /usr/sbin/
[root@deep imap-2001]# install -s -m755 ./imapd/imapd /usr/sbin/
[root@deep imap-2001]# install -m644 ./c-client/c-client.a /usr/lib
[root@deep imap-2001]# ln -s /usr/lib/c-client.a /usr/lib/libc-client.a
[root@deep imap-2001]# mkdir -p /usr/include/imap
[root@deep imap-2001]# install -m644 ./c-client/*.h /usr/include/imap/
[root@deep imap-2001]# install -m644 ./src/osdep/tops-20/shortsym.h
/usr/include/imap/
[root@deep imap-2001]# cd
[root@deep /root]# find /* > IMAP2
[root@deep /root]# diff IMAP1 IMAP2 > IMAP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations. Note that the **make lnp** command above will configure your Linux system with **P**luggable **A**uthentication **M**odules (PAM) capability for better password security.

The '**SSLTYPE=unix**' parameter will build UW IMAP with SSL capability enabled. If you don't want to include SSL support with UW IMAP, then all you have to do is to omit the '**SSLTYPE=unix**' parameter in your compile line above, but be aware that you can always run UW IMAP without SSL support even if you have included the '**SSLTYPE=unix**' parameter in your compilation to enable SSL support into the software.

The **mkdir** command will create a new directory named "imap" under /usr/include. This new directory "imap" will keep all header development files related to the imapd program "c-client/*.h", and "shortsym.h" files. The **ln -s** command would create a symbolic link from "c-client.a" file to "libc-client.a" which may be required by some third party programs that you might install in the future.

NOTE: For security reasons, if you only use imapd service, remove the ipop2d and ipop3d binaries from your system. The same applies for ipop2d or ipop3d; if you only use ipop2d or ipop3d service then remove the imapd binary from your server. If you intend to use imapd, ipop2d and ipop3d services all together then keep all binaries.

Step 6
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete UW IMAP and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf imap-version/
[root@deep tmp]# rm -f imap-version.tar.Z
```

The rm command as used above will remove all the source files we have used to compile and install UW IMAP. It will also remove the UW IMAP compressed archive from the /var/tmp directory.

Configuring UW IMAP

After UW IMAP has been built and installed successfully in your system, your next step is to configure and customize UW IMAP configuration files. Those files are:

- ✓ /etc/pam.d/imap (The IMAP PAM Support Configuration File)
- ✓ /etc/pam.d/pop (The POP PAM Support Configuration File)

/etc/pam.d/imap: The IMAP PAM Support Configuration File

During compilation of UW IMAP, we have compiled the software to use **P**luggable **A**uthentication **M**odules (PAM) capability with the '**make lnp**' command. Now, we must configure the software to use PAM password authentication support or nothing will work. Do to that, you must create the /etc/pam.d/imap file. This PAM file is required only if you intended to provide IMAP service in your system.

- Create the **imap** file (touch /etc/pam.d/imap) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
```

/etc/pam.d/pop: The POP PAM Support Configuration File

As for the IMAP PAM file above, if you intended use POP instead of IMAP service, you must configure the software to use PAM password authentication support or nothing will work. Do to that, create the /etc/pam.d/pop file. This PAM file is required only if you intended to provide POP service in your system. If you want to provide IMAP and POP support, then you must create and use the both files (/etc/pam.d/imap and /etc/pam.d/pop).

- Create the **pop** file (touch /etc/pam.d/pop) and add the following lines:

```
#%PAM-1.0
auth       required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_stack.so service=system-auth
```

Enable IMAP or POP services via Xinetd

Xinetd is the successor of inetd and tcp_wrappers, it is more secure, powerful and faster, therefore I recommend you use it instead of inetd and tcp_wrappers to control IMAP & POP servers. The super server Xinetd take care of starting and stopping IMAP or POP servers. Upon execution, Xinetd reads its configuration information from a configuration file which, by default, is /etc/xinetd.conf.

Below we show you four different examples, which can be used to start IMAP or POP services depending of your needs.

Example 1
Here is the sample /etc/xinetd.conf entry for IMAP service (imap):

- Edit the **xinetd.conf** file (vi /etc/xinetd.conf) and enter your requirements under the services sections of this file. Below is the required configuration lines we recommend you add to enable the imap service:

```
# description: The IMAP service allows remote users to access their \
#               mail using an IMAP client such as Mutt, Pine, \
#               fetchmail, or Netscape Communicator.
service imap
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/imapd
        only_from       = 0.0.0.0/0 localhost
        log_on_success  += DURATION USERID
        log_on_failure  += USERID
        nice            = -2
}
```

Example 2
This section applies only if you want to run IMAP server through SSL connection. Here is the sample /etc/xinetd.conf entry for IMAP service with SSL support (imaps):

- Edit the **xinetd.conf** file (vi /etc/xinetd.conf) and enter your requirements under the services sections of this file. Below is the required configuration lines we recommend you add to enable the imaps service:

```
# description: The IMAP service allows remote users to access their \
#               mail using an IMAP client with SSL support such as \
#               Netscape Communicator or fetchmail.
service imaps
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/imapd
        only_from       = 0.0.0.0/0 localhost
        log_on_success  += DURATION USERID
        log_on_failure  += USERID
        nice            = -2
}
```

Example 3
Here is the sample /etc/xinetd.conf entry for POP3 service (pop3):

- Edit the **xinetd.conf** file (vi /etc/xinetd.conf) and enter your requirements under the services sections of this file. Below is the required configuration lines we recommend you add to enable the pop3 service:

```
# description: The POP3 service allows remote users to access their \
#               mail using an POP3 client such as Netscape Communicator,\
#               mutt, or fetchmail.
service pop3
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/ipop3d
        only_from       = 0.0.0.0/0 localhost
        log_on_success  += USERID
        log_on_failure  += USERID
        nice            = -2
}
```

Example 4
This section applies only if you want to run POP3 server through SSL connection. Here is the sample /etc/xinetd.conf entry for POP3 service with SSL support (pop3s):

- Edit the **xinetd.conf** file (vi /etc/xinetd.conf) and enter your requirements under the services sections of this file. Below is the required configuration lines we recommend you add to enable the pop3s service:

```
# description: The POP3S service allows remote users to access their \
#               mail using an POP3 client with SSL support such as
fetchmail.
service pop3s
{
        socket_type     = stream
        wait            = no
        user            = root
        server          = /usr/sbin/ipop3d
        only_from       = 0.0.0.0/0 localhost
        log_on_success  += USERID
        log_on_failure  += USERID
        nice            = -2
}
```

NOTE: To my knowledge, the only POP3 client which supports POP3 with SSL technology is fetchmail; therefore don't try to use Netscape or Outlook to read your mail through pop3s. Instead use imaps.

Don't forget to update your `xinetd.conf` file for the changes to take effect by restarting the `Xinetd` daemon program.

- To update you `xinetd.conf` file, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/xinetd restart
Stopping xinetd:               [OK]
Starting xinetd:               [OK]
```

WARNING: All the above `Xinetd` configurations assume that the `default section` of your `xinetd` configuration file is configured as follow to enable `pop3s`, `pop3`, `imaps`, and `imap` services:

```
defaults
{
        instances       = 60
        log_type        = SYSLOG authpriv
        log_on_success  = HOST PID
        log_on_failure  = HOST RECORD
        only_from       =
        per_source      = 5
        enabled         = pop3s pop3 imaps imap
}
```

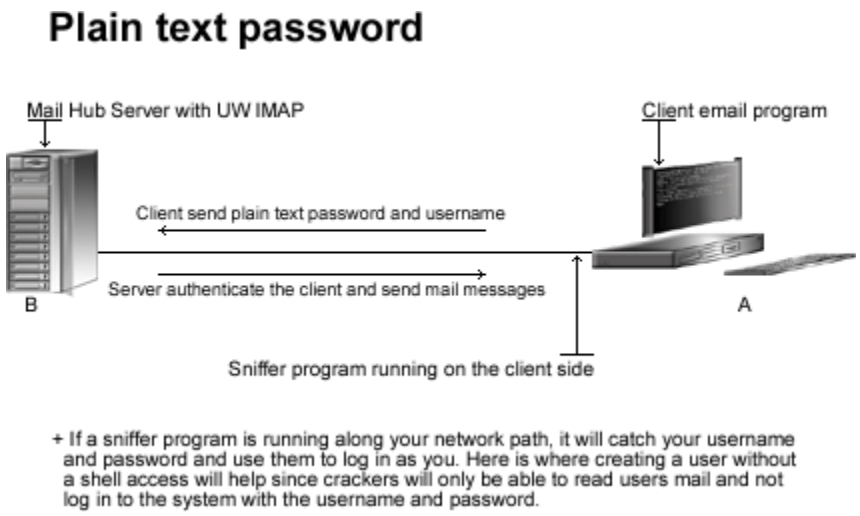
If you don't want to enable `pop3s`, `imaps` or `imap` then remove them from the line. The same applies for other the `IMAP/POP` services as shown above. For more information about `Xinetd`, please read the appropriate `Xinetd` chapter in this book.

Securing `UW IMAP`

This section deals with actions we can make to improve and tighten security under `UW IMAP`. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Do you really need `UW IMAP` server and its services?

Be aware that `IMAP/POP` programs use plain text passwords by default. Anyone running a sniffer program along your network path can grab the username/password and use them to log in as you. It is not because you use an `IMAP/POP` Mail User Agent **reader** (`MUA`) like `Netscape` on your LINUX system that you need to run `UW IMAP` server locally. Check your configuration, and if you use a remote/external `IMAP/POP` server then uninstall `UW IMAP` on your system.



The right way to create mail users on the Mail Server

It is not because you have to set up and add a new user to the Mail Server that this user needs to have a shell account on the system. Shell accounts are precious and must be given out only and only if it is necessary. If you only want to allow mail user to get, read and send mails (usually this is what all of us are looking for), then all you have to do is to create a new account for this user without shell access. Creating a mail user account without shell access to the system will eliminate many risks related to the fact that crackers can use mail user account to access the server.

From here, we can explain one reason for which having a dedicated machine that runs a Mail Server is important. If you have a server dedicated for electronic mail, then the only legitimate user allowed to have login shell access by default to the system will be the super-user 'root'. Imagine, it this way, you can run for example 1000 mail users and even if one of them are compromised, there is no problem since access to the system can be done only by our super-user 'root'.

Step 1

The principle of creating a user without a login shell account is the same as for creating an `FTP` user without a shell account. This procedure can be applied for any other services for which you want a user without shell access to the system.

- Use the following command to create users in the `/etc/passwd` file. This step must be done for each additional new mail user you allow to access your `Mail` server.

```
[root@deep /]# useradd -s /bin/false gmourani 2>/dev/null || :
[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The `useradd` command will add the new mail user named `gmourani` to the Linux Mail Server. The '`-s`' option specifies the name of the user's login shell, in our case we choose `/bin/false` and redirect it to `/dev/null`. Finally, the `passwd` command will set the password for this user `gmourani`.

Step 2
Now, edit the **shells** file (vi /etc/shells) and add a non-existent shell name like "/bin/false", which is the one we used in the passwd command above.

```
[root@deep /]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

NOTE: You only have to make the 'Step2' one time. If the shell called "/bin/false" already exist in the /etc/shells file, then you don't have to add it again. Yes I know, but I prefer to be clear here.

Running UW IMAP with SSL support

This section applies only if you want to run UW IMAP through SSL connection. If you are an ISP with many regular users, this may not be the case for you, but if you are a company that provides for your particular limited users a mail service, this can be good for you. We know now that IMAP/POP programs use plain text passwords by default. The solution to prevent someone using a sniffer program to grab the username/password of your mail users is to use the new SSL capability of UW IMAP to encrypt the client sessions.

We have already configured UW IMAP during compilation to enable its SSL support with the use of the special parameter '**SSLTYPE=unix**', therefore UW IMAP is SSL compatible even if you decide to not use its SSL functionality at this time. Now, all we have to do is to set up the certificates. Below I'll show you how to set up a self signed certificate to use with UW IMAP, the principle is the same as for creating a certificate for a Web Server (refer to OpenSSL chapter if you have problem creating the certificates).

Step 1
First you have to know the Fully Qualified Domain Name (FQDN) of the Mail Hub Server for which you want to request a certificate. When your incoming mail server address is boreas.openna.com then the FQDN of your Mail Hub Server is boreas.openna.com.

Step 2
Create a self-signed certificate (x509 structure) without a pass-phrase. The req command creates a self-signed certificate when the -x509 switch is used. For certificates signed by commercial Certifying Authority (CA) like Thawte refer to the OpenSSL chapter for the required procedures to follow.

- To create a self-signed certificate, use the following command:

```
[root@deep ssl]# cd /usr/share/ssl
[root@deep ssl]# openssl req -new -x509 -nodes -days 365 -out tmp.pem
Using configuration from /usr/share/ssl/openssl.cnf
Generating a 1024 bit RSA private key
.............................................++++++
...........++++++
writing new private key to 'privkey.pem'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [Open Network Architecture]:
Organizational Unit Name (eg, section) []:
Common Name (eg, YOUR name) [boreas.openna.com]:
Email Address [noc@openna.com]:
```

WARNING: Pay special attention to the '`-nodes`' option we have used, in the above command, to create the self-signed certificate. The option '`-nodes`' creates a certificate without a protected pass-phrase, it is very important to create a certificate without a pass-phrase because `IMAP/POP` server cannot ask you to enter a password before starting its daemon. Also, be sure that you've entered the FQDN (Fully Qualified Domain Name) of the Mail Hub Server when `OpenSSL` prompts you for the "**`CommonName`**".

Step 3
Once the self-signed certificate has been created, we must be sure that the future **`imapd.pem`** file will has both a RSA PRIVATE KEY and a CERTIFICATE section.

- To include the CERTIFICATE section to RSA PRIVATE KEY, use the command:
  `[root@deep ssl]#` **`cat tmp.pem >> privkey.pem`**

The above command will include the CERTIFICATE file named '`tmp.pem`' to the RSA PRIVATE KEY named '`privkey.pem`'.

Step 4
After, we must place the certificate file to its appropriate directory and rename it `imapd.pem` for `IMAP/POP` server to recognize it. If you rename the certificate something other than '`imapd.pem`' be aware that the `UW IMAP` will not recognize it.

- To place the file into its appropriate directory, use the following command:
  `[root@deep ssl]#` **`mv privkey.pem certs/imapd.pem`**
  `[root@deep ssl]#` **`chmod 400 certs/imapd.pem`**
  `[root@deep ssl]#` **`rm -f tmp.pem`**

First we move the `privkey` file which contain both RSA PRIVATE KEY and CERTIFICATE section to the `certs` directory and rename it `imapd.pem` for `UW IMAP` to use it. After that we remove the `tmp.pem` file from our system since it is no longer needed.

WARNING: `Netscape` and `Outlook` support only `imapd` through `SSL`, and `pop3d` with `SSL` work only with `fetchmail`. If you intended to use `Netscape` or `Outlook` to read your mail through `SSL`, then use `imapd` and not `pop3d`. Also don't forget to configure `imaps` into the `xinetd.conf` configuration file to enable `imapd` with `SSL` support on your system.

Step 5
Now, it is important to verify if the new `imapd.pem` certificate file works before connecting with client `MUA` program like `Netscape` to read mail through `SSL`. Please make sure that the `Xinetd` daemon with the `imaps` value enabled is already running before proceding with the test.

- To test your new `IMAP` certificate, use the following command:

```
[root@deep ssl]# openssl
OpenSSL> s_client -host boreas.openna.com -port 993
CONNECTED(00000003)
depth=0 /C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=boreas.openna.com/Email=noc@openna.com
verify error:num=18:self signed certificate
verify return:1
depth=0 /C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=boreas.openna.com/Email=noc@openna.com
verify return:1
---
Certificate chain
 0 s:/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=boreas.openna.com/Email=noc@openna.com
   i:/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=boreas.openna.com/Email=noc@openna.com
---
Server certificate
-----BEGIN CERTIFICATE-----
MIIDlTCCAv6gAwIBAgIBADANBgkqhkiG9w0BAQQFADCBlDELMAkGA1UEBhMCQ0Ex
DzANBgNVBAgTBlF1ZWJlYzERMA8GA1UEBxMITW9udHJlYWwxIjAgBgNVBAoTGU9w
ZW4gTmV0d29yayBBcmNoaXRlY3R1cmUxHjAcBgNVBAMTFXVsbHlzZS5tdHRjb25z
ZWlsLmNvbTEdMBsGCSqGSIb3DQEJARYObm9jQG9wZW5uYS5jb20wHhcNMDAxMjE2
MDQ1NjI2WhcNMDIwNzE3MTU1OTU0WjCBlDELMAkGA1UEBhMCQ0ExDzANBgNVBAgT
BlF1ZWJlYzERMA8GA1UEBxMITW9udHJlYWwxIjAgBgNVBAoTGU9wZW4gTmV0d29y
ayBBcmNoaXRlY3R1cmUxHjAcBgNVBAMTFXVsbHlzZS5tdHRjb25zZWlsLmNvbTEd
MBsGCSqGSIb3DQEJARYObm9jQG9wZW5uYS5jb20wgZ8wDQYJKoZIhvcNAQEBBQAD
gY0AMIGJAoGBAM7HC7h/Vxi3ox5nECmd3odhJwGZFdq4tOvbMkknn3F7HAsEpcpJ
OddtZtHNhN3rDn1vYLzuWc0flmG/ry3G5grshsd8JFHp024kRjsdOZSWjoAcT+UE
hD/jF0Wg8L5nRlOuD1RiU9eGqMma7vG80QKGvq/4y5bKUfLYEdHbCTEnAgMBAAGj
gfQwgfEwHQYDVR0OBBYEFLSZEXinVoRgQjKe8pZt6NWWTOFPMIHBBgNVHSMEgbkw
gbaAFLSZEXinVoRgQjKe8pZt6NWWTOFPoYGapIGXMIGUMQswCQYDVQQGEwJDQTEP
MA0GA1UECBMGUXVlYmVjMREwDwYDVQQHEwhNb250cmVhbDEiMCAGA1UEChMZT3Bl
biBOZXR3b3JrIEFyY2hpdGVjdHVyZTEeMBwGA1UEAxMVdWxseXNlLm10dGNvbnNl
aWwuY29tMR0wGwYJKoZIhvcNAQkBFg5ub2NAb3Blbm5hLmNvbYIBADAMBgNVHRME
BTADAQH/MA0GCSqGSIb3DQEBBAUAA4GBAAJC7BzgXPJ2PezOH1R8I9a/xdW36mpp
6YB08P6p1a3oO5NAauf9KW+1bUd7UAM6c61Jyj2g8oL4v9ukx27Z9r2nE4Y4Jubs
HQ1VuZ9zpqbHINcMRlugCUWSqKdTcYoQNL+EXnPefs6+JjCmEiaTMEmn2Ggm7yE3
ef+0J3LXhrzr
-----END CERTIFICATE-----
subject=/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=boreas.openna.com/Email=noc@openna.com
issuer=/C=CA/ST=Quebec/L=Montreal/O=Open Network
Architecture/CN=boreas.openna.com/Email=noc@openna.com
---
No client certificate CA names sent
---
SSL handshake has read 1075 bytes and written 314 bytes
---
New, TLSv1/SSLv3, Cipher is DES-CBC3-SHA
Server public key is 1024 bit
SSL-Session:
    Protocol  : TLSv1
    Cipher    : DES-CBC3-SHA
```

```
    Session-ID:
FB1C9CCF4F540CECEF138625549C0391CAC1BBC84A5FDBC37F6AFC4616D785EA
    Session-ID-ctx:
    Master-Key:
AC9E7F536E5E5C7F3CDE76C9590F95894E5BAE3A0EF2A466867D5A7BD57B44327CAE455D4
EBAFFFE10A6C3B2451A7866
    Key-Arg   : None
    Start Time: 976954222
    Timeout   : 300 (sec)
    Verify return code: 0 (ok)
---
* OK [CAPABILITY IMAP4 IMAP4REV1 STARTTLS LOGIN-REFERRALS AUTH=PLAIN
AUTH=LOGIN] ullyse.mttconseil.com IMAP4rev1 2000.284 at Sat, 16 Dec 2000
03:10:22 -0500 (EST)
```

If the results look like the one above, then communications from the Mail Hub Server to the client machine are encrypted for `imapd` with the `SSL` protocol. Congratulations!

Step 6
Recall that by default all connections from a external client to the `imap` secure server are allowed via port 143 (the default `imap` port) only, therefore it is important to allow traffic through the `imap` port `993` into our firewall script file for the **I**nternet **M**essage **A**ccess **P**rotocol to accept external connections.

- Edit the **`iptables`** script file (`vi /etc/rc.d/init.d/iptables`), and add/check the following lines to allow `imaps` packets to traverse the network:

```
    # IMAP server over SSL (993)
    # --------------------------
    iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp  \
             --source-port $UNPRIVPORTS \
             -d $IPADDR --destination-port 993 -j ACCEPT

    iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp ! --syn \
             -s $IPADDR --source-port 993 \
             --destination-port $UNPRIVPORTS -j ACCEPT
```

Where `EXTERNAL_INTERFACE="eth0"` # Internet connected interface
Where `IPADDR="207.35.78.4"` # Your IP address for eth0
Where `UNPRIVPORTS="1024:"` # Unprivileged port range

Step 7
Finally, if you have installed `PortSentry` on your server, it is important to add the `imaps` port 993 to the list of allowed ports into the `PortSentry` configuration file called "`portsentry.conf`" or any future connections to this port will be blocked by the program.

- Edit the **`portsentry.conf`** file (`vi /etc/portsentry/portsentry.conf`), and add the port number 993 to instruct `PortSentry` to ignore this port:

```
# Use these if you just want to be aware:
TCP_PORTS="1,11,15,79,111,119,143,993,540,635,1080,1524,2000,5742,6667,12345,12
346,20034,31337,32771,32772,32773,32774,40421,49724,54320"
```

Further documentation

For more details, there are some `UW IMAP` manual pages that you can read:

```
$ man imapd (8C)          - Internet Message Access Protocol server
$ man ipopd (8C)          - Post Office Protocol server
```

List of installed `UW IMAP` files on your system

> /etc/pam.d/imap
> /etc/pam.d/pop
> /usr/include/imap
> /usr/include/imap/c-client.h
> /usr/include/imap/dummy.h
> /usr/include/imap/env.h
> /usr/include/imap/env_unix.h
> /usr/include/imap/fdstring.h
> /usr/include/imap/flstring.h
> /usr/include/imap/fs.h
> /usr/include/imap/ftl.h
> /usr/include/imap/imap4r1.h
> /usr/include/imap/linkage.h
> /usr/include/imap/lockfix.h
> /usr/include/imap/mail.h
> /usr/include/imap/mbox.h
> /usr/include/imap/mbx.h
> /usr/include/imap/mh.h
> /usr/include/imap/misc.h
> /usr/include/imap/mmdf.h
> /usr/include/imap/mtx.h
> /usr/include/imap/mx.h
> /usr/include/imap/netmsg.h
> /usr/include/imap/news.h
> /usr/include/imap/newsrc.h
> /usr/include/imap/nl.h
> /usr/include/imap/nntp.h
> /usr/include/imap/os_a32.h
> /usr/include/imap/os_a41.h
> /usr/include/imap/os_aix.h
> /usr/include/imap/os_aos.h
> /usr/include/imap/os_art.h
> /usr/include/imap/os_asv.h
> /usr/include/imap/os_aux.h
> /usr/include/imap/os_bsd.h
> /usr/include/imap/os_bsi.h
> /usr/include/imap/os_cvx.h
> /usr/include/imap/osdep.h
> /usr/include/imap/os_d-g.h
> /usr/include/imap/os_do4.h
> /usr/include/imap/os_drs.h
> /usr/include/imap/os_dyn.h
> /usr/include/imap/os_hpp.h
> /usr/include/imap/os_isc.h
> /usr/include/imap/os_lnx.h
> /usr/include/imap/os_lyn.h
> /usr/include/imap/os_mct.h
> /usr/include/imap/os_mnt.h
> /usr/include/imap/os_nxt.h
> /usr/include/imap/os_os4.h
> /usr/include/imap/os_osf.h
> /usr/include/imap/os_osx.h
> /usr/include/imap/os_ptx.h
> /usr/include/imap/os_pyr.h
> /usr/include/imap/os_qnx.h
> /usr/include/imap/os_s40.h
> /usr/include/imap/os_sc5.h
> /usr/include/imap/os_sco.h
> /usr/include/imap/os_sgi.h
> /usr/include/imap/os_shp.h
> /usr/include/imap/os_slx.h
> /usr/include/imap/os_sol.h
> /usr/include/imap/os_sos.h
> /usr/include/imap/os_sun.h
> /usr/include/imap/os_sv2.h
> /usr/include/imap/os_sv4.h
> /usr/include/imap/os_ult.h
> /usr/include/imap/os_vu2.h
> /usr/include/imap/phile.h
> /usr/include/imap/pop3.h
> /usr/include/imap/pseudo.h
> /usr/include/imap/rfc822.h
> /usr/include/imap/smtp.h
> /usr/include/imap/tcp.h
> /usr/include/imap/tcp_unix.h
> /usr/include/imap/tenex.h
> /usr/include/imap/unix.h
> /usr/include/imap/utf8.h
> /usr/include/imap/shortsym.h
> /usr/lib/c-client.a
> /usr/lib/libc-client.a
> /usr/sbin/ipop3d
> /usr/sbin/imapd
> /usr/share/man/man8/ipopd.8c
> /usr/share/man/man8/imapd.8c

I've put a break here to summarize what we have been doing since the beginning of the book, and hope that you have found it interesting.

1) First, we have installed Linux by removing all unneeded programs to have a clean and secure server. Recall that the beginning of a secure server is one where all unneeded services and programs are uninstalled.

2) After that, we have tightened the security of our configured system by using the default tools of Linux without the need of external programs.

3) We have optimized our system to perform at its peak by using specific compiler flags and by replacing the default Linux libraries files by ones, which has been optimized for our processor.

4) We have recompiled the Linux kernel to best fit our system and to get the most in kernel security and optimization.

5) We have tuned and secured our `TCP/IP` networking.

6) We have installed a firewall, which respond closely to our networking architecture and services we want to enable in a manner to build a fortress around our server.

7) We have installed the entire minimum recommended security tools on the server to keep communications the as secure as possible and to prevent possible attacks, holes, etc that will certainly come to our network.

8) We have installed `ISC BIND & DNS` related to the configuration we want for the server. Recall that `ISC BIND & DNS` is very important and must be installed in every kind of server, since many services described in this book rely on it to work properly.

9) Finally, we have installed a mail server related to the configuration and the tasks of the server we want to install. Once again, don't forget that on all kinds of machines that run a Unix operating system it's necessary and **NOT** optional to have a mail server.

From now, every chapter described later in this book are optional and depend on what you want to do on your server. (E.g., What kind of tasks will your server perform, and for which part of your network Intranet/Internet?) For all kinds of servers and whatever you decide to install, a Web, `FTP`, `SQL`, Backup, File Sharing Servers, etc, it is absolutely vital to apply all of the information and tutorials shown here, on all of your Linux machines.

Everything that you have read in this book up to here are the minimum amount of actions to make in all your Linux systems you hope to put online, if you want a secure, optimized and functional Linux server. After that, any specific service you install will make this machine become a Web Server, Mail, etc depending of the kind of service installed.

Finally, don't forget that security and optimization doesn't stop here because even if you have secured your system to the best, any additional services you may install and enable will bring a new security risk and it is for this reason that they must be configured and installed in the most secure manner available. This is why all chapters related to a specific service are explained from Part X through the end of this book.

Part X Database Server Related Reference

In this Part

Database Server - MySQL
Database Server - PostgreSQL
Database Server - OpenLDAP

Once you decide to go into serious business, you'll inevitably find that you need a database to store/retrieve information. One of the primary reasons for the invention of computer is to store, retrieve and process information and do all this very quickly. The most popular database systems are based on the International Standard Organization (ISO) SQL specifications which are also based on ANSI SQL (American) standards.

This part of the book will deal with software other than the one's which the Linux distribution, may or may not provide as a part of its core distribution. In some cases it may be provided as an extra but may also come as a pre-compiled binary, which may not exactly suit your purpose. Hence we have, in most cases, used source packages, usually packed as tar gzipped -*.tar.gz. This gives us the maximum amount of choice to tweak, secure, optimize and delete the options within this software.

23 Database Server - MySQL

In this Chapter

Linux MySQL Database Server

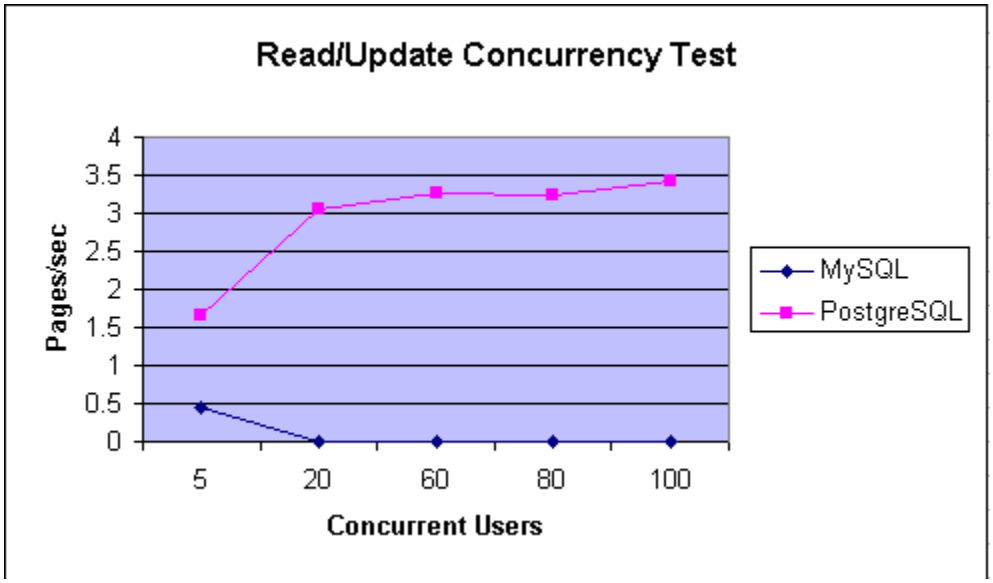
Abstract

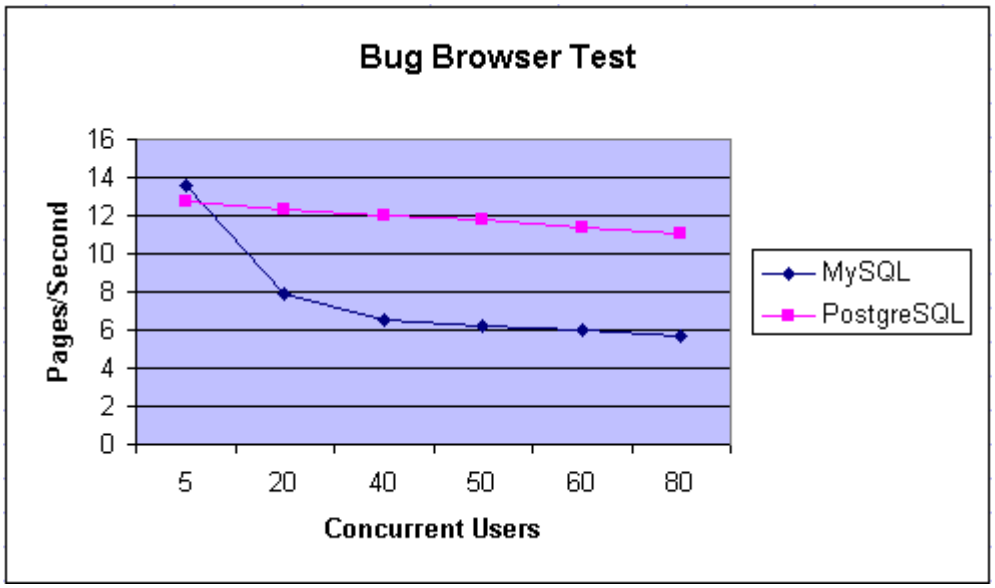
Once you begin to serve, and supply services to your customers, you'll inevitably find that you need to keep information about them in an archive which has to be accessible and modifiable at any time you want it. These tasks can be accomplished with the use of a database. There are many databases are available on Linux; choosing one can be complicated, as it must be able to support a number of programming languages, standards and features. PostgreSQL is a sophisticated Object-Relational DBMS and supports almost all SQL constructs, which may respond to complicated and complex database needs.

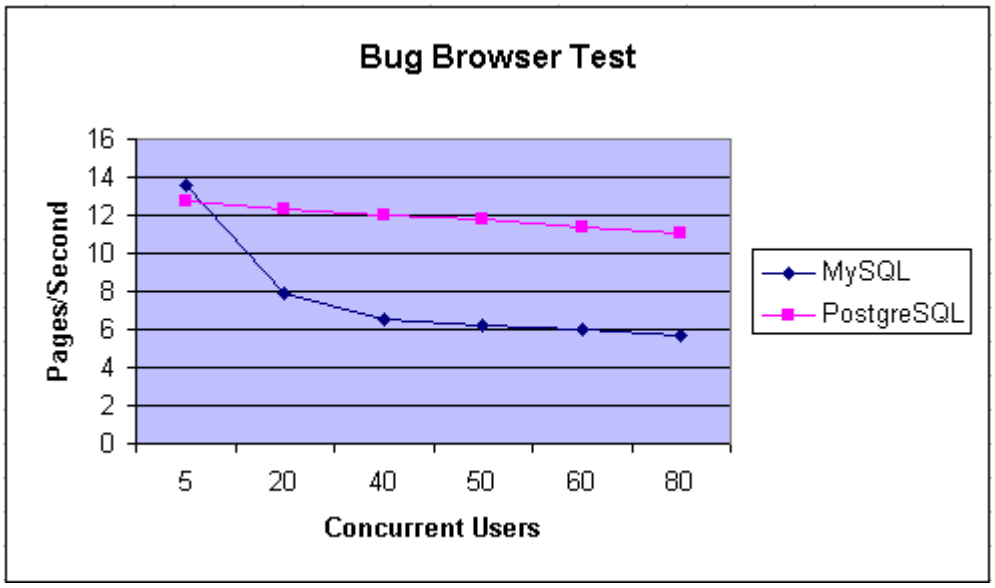
In real use, and especially for Web server connectivity with SQL databases, the need for this kind of complex arrangement is not always true and may penalize performance. For this reason some companies decide to create an SQL server which responds to these requirements. MySQL is a small SQL database built with the most essential SQL constructs only and increases performance by eliminating functions.

As explained in the MySQL web site:
MySQL is a true multi-user, multi-threaded SQL database server. SQL (**S**tructured **Q**uery **L**anguage) is the most popular and standardized database language in the world. MySQL is a client/server implementation that consists of a server daemon "mysqld" and many different client programs and libraries. The main goals of MySQL are speed, robustness and ease of use. MySQL was originally developed for the need of SQL server that could handle very large databases an order of magnitude faster than what any database vendor could offer on inexpensive hardware.

Many of us use MySQL as a database for an application server and presume that it is the fastest SQL server available today. I don't think the quite same, MySQL is very fast, but with the fast development of open source, situations change quickly and this might be taken on with the new release of PostgreSQL.







Open Source Databases: As The Tables Turn. The graphs comes from the www.phpbuilder.com website from an article of Tim Perdue. http://www.phpbuilder.com/columns/tim20001112.php3

Yes, contrary about what we may think, `PostgreSQL` is faster than `MySQL` in many areas. But to be honest `MySQL` is easier to use and link with external applications than `PostgreSQL`. Also it is widely used by many third party programs and from the point of view of compatibility and integration, this is very important. Any way, it is yours to decide which one of these two beautiful databases best suit your needs.

Since many readers asked that `MySQL` was documented in the next release of the book, here we go.

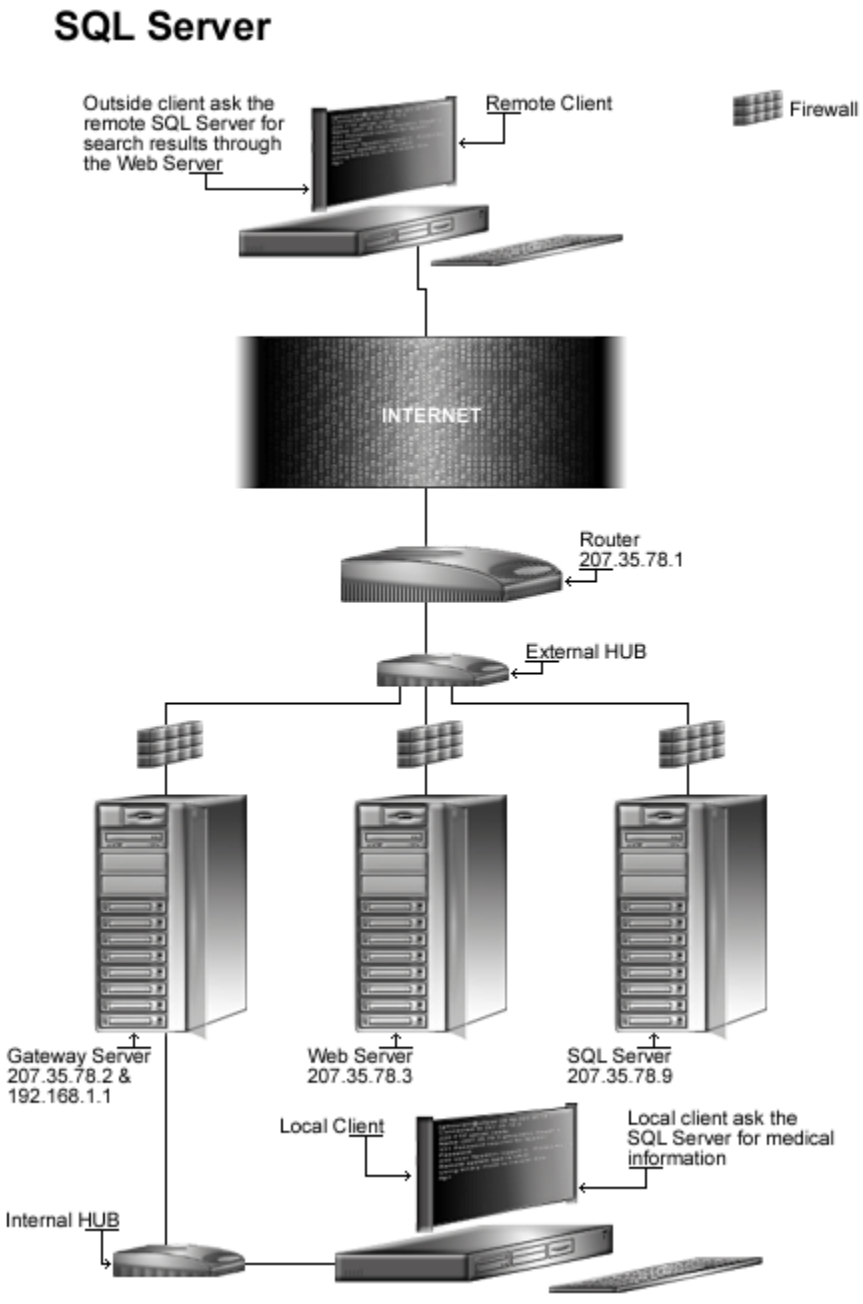
Recommended RPM packages to be installed for a SQL Server

A minimum configuration provides the basic set of packages required by the Linux operating system. A minimum configuration is a perfect starting point for building a secure operating system. Below is the list of all recommended RPM packages required to run your Linux server as a database Server (SQL) running MySQL software properly.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install MySQL by using it's RPM package. Therefore, the mysql and mysql-server RPM packages are already included in the list below. Not all security tools are installed, it is up to you to install them as you need by RPM packages since compiler packages are not installed and included in the list.

basesystem	ed	less	openssl	sysklogd
bash	file	libstdc++	pam	syslinux
bdflush	filesystem	libtermcap	passwd	SysVinit
bind	fileutils	lilo	**perl**	tar
bzip2	findutils	logrotate	popt	termcap
chkconfig	gawk	losetup	procps	textutils
console-tools	gdbm	MAKEDEV	psmisc	tmpwatch
cpio	gettext	man	pwdb	utempter
cracklib	glib	mingetty	qmail	util-linux
cracklib-dicts	glibc	mktemp	readline	vim-common
crontabs	glibc-common	mount	rootfiles	vim-minimal
db1	grep	**mysql**	rpm	vixie-cron
db2	groff	**mysql-server**	sed	words
db3	gzip	ncurses	setup	which
dev	info	net-tools	sh-utils	zlib
devfsd	initscripts	newt	shadow-utils	
diffutils	iptables	openssh	slang	
e2fsprogs	kernel	openssh-server	slocate	

Tested and fully functional on OpenNA.com.



This schema shows you some possible uses of `SQL` Servers.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest MySQL version number is 3.23.38

Packages

The following are based on information as listed by MySQL as of 2001/06/02. Please regularly check at www.mysql.com/ for the latest status.

Source code is available from:
MySQL Homepage: http://www.mysql.com/
MySQL FTP Site: 64.28.67.70
You must be sure to download: mysql-3.23.38.tar.gz

Pristine source

If you don’t use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install MySQL, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > MySQL1
```
- And the following one after you install the software:
```
[root@deep /root]#  find /* > MySQL2
```
- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff MySQL1 MySQL2 > MySQL-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing MySQL

Below are the required steps that you must make to compile and optimize the MySQL database software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp mysql-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf mysql-version.tar.gz
```

Step 2
After that, move into the newly created MySQL source directory and perform the following step before compiling it for your specific system.

- To move into the newly created MySQL source directory use the following command:
```
[root@deep tmp]# cd mysql-3.23.38/
```

Step 3
We must create a new user account called "mysql" into the /etc/passwd file to be the owner of the MySQL database files and daemon.

- To create this special MySQL user account, use the following command:
```
[root@deep mysql-3.23.38]# useradd -M -o -r -d /var/lib/mysql -s
/bin/bash -c "MySQL Server" -u 27 mysql >/dev/null 2>&1 || :
```

Step 4
At this stage, it is time to configure and optimize MySQL for our system.

- To configure and optimize MySQL use the following compilation lines:
```
CFLAGS="-static -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
CXXFLAGS="-static -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer -
felide-constructors -fno-exceptions -fno-rtti" \
./configure \
--prefix=/usr \
--libexecdir=/usr/sbin \
--sysconfdir=/etc \
--localstatedir=/var/lib/mysql \
--mandir=/usr/share/man \
--disable-shared \
--with-mysqld-user=mysql \
--with-unix-socket-path=/var/lib/mysql/mysql.sock \
--with-client-ldflags=-all-static \
--with-mysqld-ldflags=-all-static \
--without-debug \
--without-docs \
--without-bench
```

This tells MySQL to set itself up for this particular configuration setup with:

- Disable shared libraries to compile statically linked programs (13% faster on Linux).
- Define the user mysqld daemon shall be run as (never run the MySQL daemon as 'root' user).
- Using Unix sockets rather than TCP/IP to connect to a database gives better performance.
- Disable shared libraries to avoid error messages when using "CXX=gcc" during compile time.
- Build a production version without debugging code will run MySQL 20% faster for most queries.
- Skip building of the MySQL help documentations to save space on the server.
- Skip building of the benchmark tools to save space on the server.

NOTE: Using `CXX=gcc` during compile time when configuring `MySQL` will avoid inclusion of the `libstdc++` library which it is not needed. It will also improve the performance of the database. Also, note the above optimization `FLAGS`; the optimization level "`-O3`" is not specified here since `MySQL` will automatically adjust and add the required optimization level depending of which parts of its program it will compile. We have decided to compile this software statically due to the benchmarks, it will run 13% faster on Linux when linked statically.

Step 5
Now, we must make a list of files on the system before installing the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `MySQL`.

```
[root@deep mysql-3.23.38]# make
[root@deep mysql-3.23.38]# cd
[root@deep /root]# find /* > MySQL1
[root@deep /root]# cd /var/tmp/mysql-3.23.38/
[root@deep mysql-3.23.38]# make install
[root@deep mysql-3.23.38]# install -m 644 include/my_config.h
/usr/include/mysql/
[root@deep mysql-3.23.38]# mkdir -p /var/run/mysqld
[root@deep mysql-3.23.38]# chmod 0755 /var/run/mysqld
[root@deep mysql-3.23.38]# chown mysql.mysql /var/run/mysqld
[root@deep mysql-3.23.38]# rm -f /usr/share/mysql/mysql-*.spec
[root@deep mysql-3.23.38]# rm -f /usr/share/mysql/mysql-log-rotate
[root@deep mysql-3.23.38]# strip /usr/sbin/mysqld
[root@deep mysql-3.23.38]# cd
[root@deep /root]# find /* > MySQL2
[root@deep /root]# diff MySQL1 MySQL2 > MySQL-Installed
```

The **`make`** command compile all source files into executable binaries, and then **`make install`** will install the binaries and any supporting files into the appropriate locations. The **`mkdir -p`** will create a new directory for `MySQL` pid file under the appropriate location as well as setting its mode permissions and ownership. The **`strip`** command will reduce the size of our **`mysqld`** daemon binary by 50%.

Step 6
At this stage, all the files and binaries related to `MySQL` database have been installed onto your computer. It is time to verify if the `mysqld` daemon is linked statically as we want it to be.

- To verify if the `mysqld` daemon is linked statically, use the following command:

```
[root@deep /]# ldd /usr/sbin/mysqld
        not a dynamic executable
```

If the returned result of the command is the same as the one shown above (`not a dynamic executable`), then congratulations! Every library required by the daemon to successfully run on your server has been compiled directly into the `mysqld` binary.

Step 7
Once the configuration, optimization, compilation, and installation of the database software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete MySQL and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf mysql-version/
[root@deep tmp]# rm -f mysql-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install MySQL. It will also remove the MySQL compressed archive from the /var/tmp directory.

Configuring MySQL

After MySQL has been built and installed successfully in your system, your next step is to configure and customize its different configuration files. MySQL has just three configuration files:

- ✓ /etc/my.cnf (The MySQL Configuration File)
- ✓ /etc/logrotate.d/mysqld (The MySQL Log rotation File)
- ✓ /etc/rc.d/init.d/mysqld (The MySQL Initialization File)

/etc/my.cnf: The MySQL Configuration File

The /etc/my.cnf file is used to specify MySQL system configuration information, such as the directory where databases are stored, where mysqld socket live and the user under which the mysqld daemon will run, etc. This file is checked to get the required information each time the database starts its daemon. It is also used to specify optimization parameters for the database, but for the moment you can add the lines shown below, and see later into this chapter under (Optimizing MySQL) for more information about other possible parameters and especially the ones related to optimization that we could add to this file (my.cnf).

- Create the **my.cnf** file (touch /etc/my.cnf) and add the following lines:

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock

[mysql.server]
user=mysql
basedir=/var/lib

[safe_mysqld]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid
```

/etc/logrotate.d/mysqld: The MySQL Log rotation File

The /etc/logrotate.d/mysqld file allows the MySQL database server to automatically rotate its log files at the specified time. Here we'll configure the /etc/logrotate.d/mysqld file to rotate each week its log files automatically.

- Create the **mysqld** file (touch /etc/logrotate.d/mysqld) and add the lines:

```
/var/log/mysqld.log {
    missingok
    create 0640 mysql mysql
    prerotate
        [ -e /var/lock/subsys/mysqld ] && /usr/bin/mysqladmin flush-logs
|| /bin/true
    endscript
    postrotate
        [ -e /var/lock/subsys/mysqld ] && /usr/bin/mysqladmin flush-logs
|| /bin/true
    endscript
}
```

/etc/rc.d/init.d/mysqld: The MySQL Initialization File

The /etc/rc.d/init.d/mysqld script file is responsible for automatically starting and stopping the mysqld daemon on your server. Loading the mysqld daemon, as a standalone daemon, will eliminate load time and will even reduce swapping since non-library code will be shared. The text in bold are the parts of the script initialization file that must be customized and adjusted to satisfy our needs.

Step 1
Create the **mysqld** script file (touch /etc/rc.d/init.d/mysqld) and add the following lines:

```
#!/bin/bash
#
# mysqld        This shell script takes care of starting and stopping
#               the MySQL subsystem (mysqld).
#
# chkconfig: - 78 12
# description:  MySQL database server.
# processname: mysqld
# config: /etc/my.cnf
# pidfile: /var/run/mysqld/mysqld.pid

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Source subsystem configuration.
[ -f /etc/sysconfig/subsys/mysqld ] && . /etc/sysconfig/subsys/mysqld

start(){
        touch /var/log/mysqld.log
        chown mysql.mysql /var/log/mysqld.log
        chmod 0640 /var/log/mysqld.log
        if [ ! -d /var/lib/mysql/mysql ] ; then
            action "Initializing MySQL database" /usr/bin/mysql_install_db
```

```
            ret=$?
            chown -R mysql.mysql /var/lib/mysql
            if [ $ret -ne 0 ] ; then
                return $ret
            fi
        fi
        /usr/bin/safe_mysqld  --defaults-file=/etc/my.cnf >/dev/null 2>&1 &
        ret=$?
        if [ $ret -eq 0 ]; then
            action "Starting MySQL: " /bin/true
        else
            action "Starting MySQL: " /bin/false
        fi
        [ $ret -eq 0 ] && touch /var/lock/subsys/mysqld
        return $ret
}

stop(){
        /usr/bin/mysqladmin -pmypasswd shutdown > /dev/null 2>&1
        ret=$?
        if [ $ret -eq 0 ]; then
            action "Stopping MySQL: " /bin/true
        else
            action "Stopping MySQL: " /bin/false
        fi
        [ $ret -eq 0 ] && rm -f /var/lock/subsys/mysqld
        [ $ret -eq 0 ] && rm -f /var/lib/mysql/mysql.sock
        return $ret
}

restart(){
    stop
    start
}

condrestart(){
    [ -e /var/lock/subsys/mysqld ] && restart || :
}

reload(){
    [ -e /var/lock/subsys/mysqld ] && mysqladmin -pmypasswd reload
}

# See how we were called.
case "$1" in
  start)
    start
    ;;
  stop)
    stop
    ;;
  status)
    status mysqld
    ;;
  reload)
    reload
    ;;
  restart)
    restart
    ;;
  condrestart)
    condrestart
    ;;
```

```
  *)
    echo $"Usage: $0 {start|stop|status|reload|condrestart|restart}"
    exit 1
esac

exit $?
```

WARNING: Pay special attention to the "-p**mypasswd**" in bold into this script file. The "**mypasswd**" represents your MySQL root user password, and must be set with your real MySQL root user password or the SQL server will ask you for the root user password each time you reboot it or reboot your system. Be aware that MySQL root user has nothing in common with the Unix root user, only the name are the same and NOTHING else.

Step 2
Once the mysqld script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each boot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/mysqld
[root@deep /]# chown 0.0 /etc/rc.d/init.d/mysqld
```

- To create the symbolic rc.d links for mysqld, use the following commands:
```
[root@deep /]# chkconfig --add mysqld
[root@deep /]# chkconfig --level 345 mysqld on
```

- To start MySQL software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/mysqld start
Starting MySQL:              [OK]
```

Step 3
Once the SQL server has been started, it's time to assign a password to the super-user of this database. With MySQL server, this user is named, by default 'root', but be aware that MySQL 'root' user has nothing in common with the Unix 'root' user, only the name are the same and NOTHING else.

For security reasons, it's important to assign a password to the MySQL root user, since by default after the installation of the SQL server, the initial root password is empty and allows anyone to connect with this name and do anything to the database.

- To specify a password for the MySQL root user, perform the following actions:
```
[root@deep /]# mysql -u root mysql
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer
```

```
mysql> SET PASSWORD FOR root=PASSWORD('mypasswd');
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

The value **'mypasswd'** as shown above is where you put the password you want to assign to the MySQL root user (this is the only value you must change in the above command). Once the root password has been set you must, in the future, supply this password to be able to connect as root in the SQL database.

NOTE: All software we describe in this book has a specific directory and subdirectory in the tar compressed archive named floppy-2.0.tgz containing configuration files for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files manually or cut and paste them to create or change your configuration files. Whether you decide to copy manually or get the files made for your convenience from the archive compressed files, it will be to your responsibility to modify them to adjust for your needs, and place the files related to this software to the appropriate places on your server. The server configuration file archive to download is located at the following Internet address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz.

Securing MySQL

This section deals especially with actions we can make to improve and tighten security under the MySQL database. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Protect the MySQL communication socket

The unix-domain socket "mysql.sock" which is used to connect to the MySQL database have by default the following permissions (0777/srwxrwxrwx), this means that anyone can delete this socket and if this happens, then no one will be able to connect to your database.

To avoid deletion of the MySQL communication socket under /var/lib/mysql/mysql.sock, you can protect its /var/lib/mysql directory by setting the sticky bit on it.

- To protect and set the sticky bit on directory where "mysql.sock" file reside, use the following command:
  ```
  [root@deep /]# chmod +t /var/lib/mysql
  ```

This command will protect your /var/lib/mysql directory so that files can be deleted only by their owners or the super-user (root).

- To check if the sticky bit is set on this directory, use the following command:
  ```
  [root@deep /]# ls -ld /var/lib/mysql
  drwx-----T    4 mysql    mysql        1024 Mar  4 12:24 /var/lib/mysql
  ```

If the last permission bit is T, then the bit is set. Congratulations!

Delete the anonymous database

When you install MySQL server, the program creates two databases by default. The first database is named "mysql" and it's used to hold all settings of the MySQL server, users, passwords, privileges etc. The second database named "test" is used to make some tests to your SQL database. Any local user can connect, without a password, to this database and do anything.

This database is not needed by the MySQL server to work and can be removed safety.

- To remove the "test" database from your SQL server, use the following command:

```
[root@deep /]$ mysqladmin drop test -p
Enter password:
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'test' database [y/N]
y
Database "test" dropped


[root@deep /]# mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> DELETE FROM db WHERE Db = "test";
Query OK, 1 row affected (0.00 sec)

mysql> DELETE FROM db WHERE Db = "test\_%";
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

Optimizing MySQL

This section deals specifically with actions we can make to improve and tighten performance of MySQL database. Note that we refer to the features available within the base installed program.

Get some fast SCSI hard disk

One of the most important parts of optimizing MySQL server as well as for the majority of all SQL databases, is the speed of your hard disk, the faster it is, and the faster your databases will run. Consider using a SCSI disk with low seek times, like 4.2ms, can make all the difference, much better performance can also be had with RAID technology.

Skip the updating of the last access time

As you should know by now, the **noatime** attribute of Linux eliminates the need by the system to make writes to the file system for files. Mounting the file system where your MySQL databases live with the noatime attribute will avoid some disk seeks and will improve the performance of you SQL server.

If you want to mount the file system of the MySQL database with the noatime attribute, it's important to create and install the MySQL databases in this partition. In our example, we have create this partition early in the chapter 2 of this book named "Linux Installation" and this partition is located on /var/lib.

Step 1
To mount the file system of MySQL databases with the noatime option, you must edit the **fstab** file (vi /etc/fstab) and add into the line that refer to /var/lib file system the noatime option after the defaults option as show below:

- Edit the **fstab** file (vi /etc/fstab), and change the line:

```
LABEL=/var/lib    /var/lib        ext2    defaults        1 2
```

To read:

```
LABEL=/var/lib    /var/lib        ext2    defaults,noatime        1 2
```

NOTE: The line related to /var/lib into your /etc/fstab file could be different from the one above, as this is just an example.

Step 2
Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the Linux system about the modifications.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /var/lib -oremount
```

Each file system that has been modified must be remounted with the command as shown above.

Step 3
After your file system has been remounted, it is important to verify if the modification in the fstab file has been correctly applied to the system.

- You can verify if the modification has been correctly applied with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root     /           ext2 rw 0 0
/proc         /proc       proc rw 0 0
/dev/sda1     /boot       ext2 rw 0 0
/dev/sda10    /cache      ext2 rw 0 0
/dev/sda9     /chroot     ext2 rw 0 0
/dev/sda8     /home       ext2 rw 0 0
/dev/sda13    /tmp        ext2 rw 0 0
/dev/sda7     /usr        ext2 rw 0 0
/dev/sda11    /var        ext2 rw 0 0
/dev/sda12    /var/lib    ext2 rw,noatime 0 0
none /dev/pts devpts rw 0 0
```

This command will show you all file systems in your Linux server with parameters applied to them. If you see something like:

```
/dev/sda12 /var/lib   ext2    rw,noatime 0 0
```

Congratulations!

NOTE: Look under chapter related to Linux Kernel in this book for more information about the `noatime` attribute and other tunable parameters.

Give MySQL more memory to get better performance

There are four options and configurable variables related directly to the speed of `MySQL` database that you might tune during server startup. The **`key_buffer_size`** parameter is one of the most important tunable variables; it represents the size of the buffer used for index blocks by `MySQL` server. The second important variable is **`table_cache`**, which represents the number of open tables for all threads. By increasing this value, you'll increases the number of file descriptors that `mysqld` requires. The two last variables are **`sort_buffer`**, which speedup the ORDER BY or GROUP BY operations of the database and **`record_buffer`**, which improves speed when you do many sequential, scans.

Step 1
Depending of the amount of memory, RAM, you have in your system and according to the `MYSQL` recommendations:

If you have a large amount of memory (>=256M), many tables and want maximum performance with a moderate number of clients, you should use something like this in your `my.cnf` file:

```
set-variable    = key_buffer=64M
set-variable    = table_cache=256
set-variable    = sort_buffer=4M
set-variable    = record_buffer=1M
```

If you have only 128M and only a few tables, but you still do a lot of sorting, you can use something like this in your `my.cnf` file:

```
set-variable    = key_buffer=16M
set-variable    = sort_buffer=1M
```

If you have little memory and lots of connections, use something like this in your `my.cnf` file:

```
set-variable    = key_buffer=512k
set-variable    = sort_buffer=100k
set-variable    = record_buffer=100k
```

or even:

```
set-variable    = key_buffer=512k
set-variable    = sort_buffer=16k
set-variable    = table_cache=32
set-variable    = record_buffer=8k
set-variable    = net_buffer=1K
```

These are just some examples, a complete list of tunable parameters depending of your type of `SQL` server exist under the `/usr/share/mysql` directory and are available for your to learn. In total there are four example files with lot of tunable parameters for huge, large, medium, and small systems and there are called respectively: `my-huge.cnf`, `my-large.cnf`, `my-medium.cnf`, `my-small.cnf`. Please, check them to better fit your optimization requirements.

Step2
Once you know the values you need for your MySQL database server, it's time to set them in your /etc/my.cnf file. Recall that this file is read each time your database server start. In our example as shows below, we will configure the /etc/my.cnf file for a medium system with little memory (32M - 64M) where MySQL plays a important part and systems up to 128M where MySQL is used together with other programs (like a web server). The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit your **my.cnf** file (vi /etc/my.cnf) and put the values that you have chosen.

```
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
skip-locking
set-variable    = key_buffer=16M
set-variable    = max_allowed_packet=1M
set-variable    = table_cache=64
set-variable    = sort_buffer=512K
set-variable    = net_buffer_length=8K
set-variable    = myisam_sort_buffer_size=8M

[mysql.server]
user=mysql
basedir=/var/lib

[safe_mysqld]
err-log=/var/log/mysqld.log
pid-file=/var/run/mysqld/mysqld.pid

[isamchk]
set-variable    = key_buffer=20M
set-variable    = sort_buffer=20M
set-variable    = read_buffer=2M
set-variable    = write_buffer=2M

[myisamchk]
set-variable    = key_buffer=20M
set-variable    = sort_buffer=20M
set-variable    = read_buffer=2M
set-variable    = write_buffer=2M
```

Step 3

- Restart the MySQL database server for the changes to take effect:

```
[root@deep /]# /etc/rc.d/init.d/mysqld restart
Enter password:
Stopping MySQL:                        [OK]
Starting MySQL:                        [OK]
```

Step 4
Now you should verify you new values with the mysqladmin command as show below. One function of this command allows you to see what values a running MySQL server is using.

- To verify the new variables entered in your startup file, use the following command:

```
[root@deep /]# mysqladmin variables -p
Enter password:
+-------------------------+---------------------------------------------+
| Variable_name           | Value                                       |
+-------------------------+---------------------------------------------+
```

```
| ansi_mode               | OFF                                        |
| back_log                | 50                                         |
| basedir                 | /usr/                                      |
| binlog_cache_size       | 32768                                      |
| character_set           | latin1                                     |
| character_sets          | latin1 dec8 dos german1 hp8 koi8_ru latin2 |
| concurrent_insert       | ON                                         |
| connect_timeout         | 5                                          |
| datadir                 | /var/lib/mysql/                            |
| delay_key_write         | ON                                         |
| delayed_insert_limit    | 100                                        |
| delayed_insert_timeout  | 300                                        |
| delayed_queue_size      | 1000                                       |
| flush                   | OFF                                        |
| flush_time              | 0                                          |
| have_bdb                | NO                                         |
| have_gemini             | NO                                         |
| have_innobase           | NO                                         |
| have_isam               | YES                                        |
| have_raid               | NO                                         |
| have_ssl                | NO                                         |
| init_file               |                                            |
| interactive_timeout     | 28800                                      |
| join_buffer_size        | 131072                                     |
| key_buffer_size         | 16773120                                   |
| language                | /usr/share/mysql/english/                  |
| large_files_support     | ON                                         |
| locked_in_memory        | OFF                                        |
| log                     | OFF                                        |
| log_update              | OFF                                        |
| log_bin                 | OFF                                        |
| log_slave_updates       | OFF                                        |
| long_query_time         | 10                                         |
| low_priority_updates    | OFF                                        |
| lower_case_table_names  | 0                                          |
| max_allowed_packet      | 1047552                                    |
| max_binlog_cache_size   | 4294967295                                 |
| max_binlog_size         | 1073741824                                 |
| max_connections         | 100                                        |
| max_connect_errors      | 10                                         |
| max_delayed_threads     | 20                                         |
| max_heap_table_size     | 16777216                                   |
| max_join_size           | 4294967295                                 |
| max_sort_length         | 1024                                       |
| max_tmp_tables          | 32                                         |
| max_write_lock_count    | 4294967295                                 |
| myisam_recover_options  | OFF                                        |
| myisam_sort_buffer_size | 8388608                                    |
| net_buffer_length       | 7168                                       |
| net_read_timeout        | 30                                         |
| net_retry_count         | 10                                         |
| net_write_timeout       | 60                                         |
| open_files_limit        | 0                                          |
| pid_file                | /var/run/mysqld/mysqld.pid                 |
| port                    | 3306                                       |
| protocol_version        | 10                                         |
| record_buffer           | 131072                                     |
| query_buffer_size       | 0                                          |
| safe_show_database      | OFF                                        |
| server_id               | 0                                          |
| skip_locking            | ON                                         |
| skip_networking         | OFF                                        |
| skip_show_database      | OFF                                        |
| slow_launch_time        | 2                                          |
| socket                  | /var/lib/mysql/mysql.sock                  |
| sort_buffer             | 524280                                     |
| table_cache             | 64                                         |
| table_type              | MYISAM                                     |
| thread_cache_size       | 0                                          |
| thread_stack            | 65536                                      |
| timezone                | EST                                        |
```

```
| tmp_table_size          | 1048576                                     |
| tmpdir                  | /tmp/                                       |
| version                 | 3.23.33                                     |
| wait_timeout            | 28800                                       |
--------------------------+---------------------------------------------+
```

From the above table, we can see that the values have been changed successfully with the new parameters.

NOTE: It's important to note that the value **key_buffer** cannot be more than 50% of your total memory. Or your system may start to page and become REALLY slow. So, if you have, for example, 256 M of RAM the value can be a maximum of 128 MB and no more.

MySQL Administrative Tools

The commands listed below are some that we use often in regular use but many more exist and you must check the reference manual for more information.

There are two statements you may use to create new users into the database, the GRANT and INSERT statements. With MySQL you have the possibility to specify, during user creation, what privileges you want to assign to your users. Privileges can be used to set which parts of the database users are allowed to use, administer, control, etc.

The GRANT statement

The first example below is the steps to follow with the GRANT statements command. In this example we'll create two different users one named "sqladmin" with password "mo" and the second named "operator" with no password and limited privileges.

- To define a new user with a password and full privileges in your database with the GRANT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> GRANT ALL PRIVILEGES ON *.* TO sqladmin@localhost
    -> IDENTIFIED BY 'mo' WITH GRANT OPTION;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

The user we have created is named "sqladmin" with password set to "mo". This user has full privileges "ALL PRIVILEGES" over the database like the super-user MySQL root.

- To define a new user with limited privilege and no password set in your database with the GRANT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
```

```
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> GRANT RELOAD,PROCESS ON *.* TO operator@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

This second user is named "operator" and is granted the reload and process administrative privileges only. He doesn't have a password set and can connect from only the localhost without a password.

NOTE: Using the GRANT statement could penalize the performance of the SQL server; it is better to use the INSERT statement, which do the same function.

The INSERT statement

The INSERT statements are the second method to create new users for the database. It's interesting to know this method, since many third party programs use it during user creation. In the example below, we use the same users name as above to show you the difference between the both methods.

- To define a new user with password and full privilege in your database with the INSERT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> INSERT INTO user VALUES('localhost','sqladmin',PASSWORD('mo'),
    -> 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y','Y');
Query OK, 1 row affected (0.02 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

The 14 'Y' you see in this command represents the privileges allowed for this user, with MySQL version 3.23.33 there are 14 privileges you may associate for the user, since the example user "sqladmin" have full control over the database, the 14 privileges are set to YES 'Y'.

- To define a new user with limited privileges and no password set in your database with the INSERT statements, use the following commands:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> INSERT INTO user SET Host='localhost',User='operator',
    -> Reload_priv='Y', Process_priv='Y';
Query OK, 1 row affected (0.00 sec)

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

In this second example we can see that only 2 privileges have been set for the user, the reload and process privileges. Also, this user has no password set and can connect from only the localhost without the need to specify a password.

Of course if you want to specify a password for this user (always recommended), then all you have to do is to include in the `INSERT` command the line "`Password('mypasswd'),`" after the "`User='operator',`" parameter.

The `UPDATE` & `DELETE` statement

These two statements can be used to manage users security access to the database. The first statement allows us to update an existing user password on the `SQL` database and the second statement lets us remove an existing user from the database.

- To update and change a user password from your database, use the following command:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> UPDATE user SET Password=PASSWORD('mypasswd') WHERE user='root';
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0.00 sec)

mysql> \q
Bye
```

In this example, we update and change the password for the super-user called "`root`". The value **`'mypasswd'`** is where you put the new password you want to update (this is the only value you must change in the above command).

- To remove a user password from your database, use the following command:

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> DELETE FROM user WHERE User = "sqladmin";
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

In this example, we remove the row in the user table of the database related to the user "sqladmin" and all privileges and the password associated to it.

The basic commands

Most of you already know how SQL databases and in our case MySQL work, but for some others, this is the first time. Below, I show you the basic commands for managing a database for beginners.

- To create a new database, run the **mysqladmin create dbname** utility program:

```
[root@deep /]$ mysqladmin create addressbook -p
Enter password:
Database "addressbook" created.
```

or with the MySQL terminal monitor program (mysql)

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> CREATE DATABASE addressbook;
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

- To delete a database and all its tables, run the **mysqladmin drop** utility program:

```
[root@deep /]$ mysqladmin drop addressbook -p
Enter password:
Dropping the database is potentially a very bad thing to do.
Any data stored in the database will be destroyed.

Do you really want to drop the 'addressbook' database [y/N]
y
Database "addressbook" dropped
```

  or with the MySQL terminal monitor program (mysql)

```
[root@deep /]$ mysql -u root mysql -p
Enter password:
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 4 to server version: 3.23.33

Type 'help;' or '\h' for help. Type '\c' to clear the buffer

mysql> DROP DATABASE addressbook;
Query OK, 3 rows affected (0.00 sec)

mysql> \q
Bye
```

- To connect to the new database with the MySQL terminal monitor, use the command:

```
mysql> USE addressbook;
Database changed
mysql>
```

- To create a table named contact with the following values, use the command:

```
mysql> CREATE TABLE contact (FirstName VARCHAR(20),
    -> SecondName VARCHAR(20), Address VARCHAR(80),
    -> WorkPhone VARCHAR(25), HomePhone VARCHAR(25),
    -> MobilePhone VARCHAR(25), Fax VARCHAR(25), Website VARCHAR(20),
    -> Mail VARCHAR(30), Title VARCHAR(20), Description VARCHAR(100));
Query OK, 0 rows affected (0.01 sec)

mysql>
```

- To inspect the new table, use the command:

```
mysql> DESCRIBE contact;
+-------------+-------------+--------+--------+----------+----------+
| Field       | Type        | Null   | Key    | Default  |    Extra |
+-------------+-------------+--------+--------+----------+----------+
| FirstName   | varchar(20) | YES    |        | NULL     |          |
| SecondName  | varchar(20) | YES    |        | NULL     |          |
| Address     | varchar(80) | YES    |        | NULL     |          |
| WorkPhone   | varchar(25) | YES    |        | NULL     |          |
| HomePhone   | varchar(25) | YES    |        | NULL     |          |
| MobilePhone | varchar(25) | YES    |        | NULL     |          |
| Fax         | varchar(25) | YES    |        | NULL     |          |
| Website     | varchar(20) | YES    |        | NULL     |          |
| Mail        | varchar(30) | YES    |        | NULL     |          |
| Title       | varchar(20) | YES    |        | NULL     |          |
| Description | varchar(100)| YES    |        | NULL     |          |
+-------------+-------------+--------+--------+----------+----------+
11 rows in set (0.00 sec)

mysql> \q
Bye
```

The LOAD DATA statement

Once your table has been created, you need to populate it. There are two statements you may use to make the job, the LOAD DATA and INSERT statements.

The LOAD DATA statement is useful when you have a lot of data to enter in your database. An easy way to populate it is to create a text file containing a row for each of your contacts, and then load the contents of the file into the table with the LOAD DATA statement.

You could create a text file "contact.txt" containing one record per line, with values separated by tabs, and given in the order in which the columns were listed in the CREATE TABLE statement. For missing values, you can use NULL values. To represent these in your text file, use \N.

```
Suzanne Smith   300, Av Washington  (514) 123 4567  (514) 890 1234  \N  \N
    www.openna.com  Suzanne@openna.com      DBAdmin \N
```

- To load the text file "contact.txt" into the contact table, use this command:

```
mysql> LOAD DATA LOCAL INFILE "contact.txt" INTO TABLE contact;
```

The INSERT statement

The INSERT statement is useful, when you want to add new records one at a time. As for LOAD DATA statement, you supply values for each column, in the order in which the columns were listed in the CREATE TABLE.

- To add a new record using an INSERT statement, use this command:

```
mysql> INSERT INTO contact
    -> VALUES ('Henri','Smith','301, Av Washington','(514) 234 8765',
    -> '(514) 456 3290',NULL,NULL,'www.openna.com','henri@openna.com',
    -> 'WebAdmin',NULL);
Query OK, 1 row affected (0.00 sec)

mysql> \q
Bye
```

- To dump the structure and data from MySQL databases and tables for backup, use the following command:

```
[root@deep /]# mysqldump mysql > mysqldb.sql -p
Enter password:
```

In this example, we dump the whole database named "mysql" into a backup file named "mysqldb.sql", which can be used later to restore the original database.

- To restore the structure and data from MySQL databases and tables from backup, use the following command:

```
[root@deep /]# mysql -u root mysql < mysqldb.sql -p
Enter password:
```

In this example, we restore the original database we backed up earlier named "mysql".

List of installed MySQL files on your system

> /etc/rc.d/init.d/mysqld
> /etc/logrotate.d/mysqld
> /etc/my.cnf
> /usr/bin/mysql
> /usr/bin/mysqladmin
> /usr/bin/mysqlshow
> /usr/bin/mysqldump
> /usr/bin/mysqlimport
> /usr/bin/mysqltest
> /usr/bin/replace
> /usr/bin/comp_err
> /usr/bin/perror
> /usr/bin/resolveip
> /usr/bin/my_print_defaults
> /usr/bin/resolve_stack_dump
> /usr/bin/isamchk
> /usr/bin/isamlog
> /usr/bin/pack_isam
> /usr/bin/myisamchk
> /usr/bin/myisamlog
> /usr/bin/myisampack
> /usr/bin/mysqlbinlog
> /usr/bin/safe_mysqld
> /usr/bin/mysql_install_db
> /usr/bin/msql2mysql
> /usr/bin/mysql_config
> /usr/bin/mysql_fix_privilege_tables
> /usr/bin/mysql_setpermission
> /usr/bin/mysql_zap
> /usr/bin/mysqlaccess
> /usr/bin/mysqlbug
> /usr/bin/mysql_convert_table_format
> /usr/bin/mysql_find_rows
> /usr/bin/mysqlhotcopy
> /usr/bin/mysqldumpslow
> /usr/bin/mysqld_multi
> /usr/lib/mysql
> /usr/lib/mysql/libmysqlclient.la
> /usr/lib/mysql/libmysqlclient.a
> /usr/lib/mysql/libmystrings.a
> /usr/lib/mysql/libdbug.a
> /usr/lib/mysql/libmysys.a
> /usr/lib/mysql/libnisam.a
> /usr/lib/mysql/libmerge.a
> /usr/lib/mysql/libmyisam.a
> /usr/lib/mysql/libmyisammrg.a
> /usr/share/mysql/japanese
> /usr/share/mysql/japanese/errmsg.sys
> /usr/share/mysql/japanese/errmsg.txt
> /usr/share/mysql/korean
> /usr/share/mysql/korean/errmsg.sys
> /usr/share/mysql/korean/errmsg.txt
> /usr/share/mysql/norwegian
> /usr/share/mysql/norwegian/errmsg.sys
> /usr/share/mysql/norwegian/errmsg.txt
> /usr/share/mysql/norwegian-ny
> /usr/share/mysql/norwegian-ny/errmsg.sys
> /usr/share/mysql/norwegian-ny/errmsg.txt
> /usr/share/mysql/polish
> /usr/share/mysql/polish/errmsg.sys
> /usr/share/mysql/polish/errmsg.txt
> /usr/share/mysql/portuguese
> /usr/share/mysql/portuguese/errmsg.sys
> /usr/share/mysql/portuguese/errmsg.txt
> /usr/share/mysql/romanian
> /usr/share/mysql/romanian/errmsg.sys
> /usr/share/mysql/romanian/errmsg.txt
> /usr/share/mysql/russian
> /usr/share/mysql/russian/errmsg.sys
> /usr/share/mysql/russian/errmsg.txt
> /usr/share/mysql/slovak
> /usr/share/mysql/slovak/errmsg.sys
> /usr/share/mysql/slovak/errmsg.txt
> /usr/share/mysql/spanish
> /usr/share/mysql/spanish/errmsg.sys
> /usr/share/mysql/spanish/errmsg.txt
> /usr/share/mysql/swedish
> /usr/share/mysql/swedish/errmsg.sys
> /usr/share/mysql/swedish/errmsg.txt
> /usr/share/mysql/charsets
> /usr/share/mysql/charsets/README
> /usr/share/mysql/charsets/Index
> /usr/share/mysql/charsets/cp1251.conf
> /usr/share/mysql/charsets/cp1257.conf
> /usr/share/mysql/charsets/croat.conf
> /usr/share/mysql/charsets/danish.conf
> /usr/share/mysql/charsets/dec8.conf
> /usr/share/mysql/charsets/dos.conf
> /usr/share/mysql/charsets/estonia.conf
> /usr/share/mysql/charsets/german1.conf
> /usr/share/mysql/charsets/greek.conf
> /usr/share/mysql/charsets/hebrew.conf

> /usr/lib/mysql/libheap.a
> /usr/sbin/mysqld
> /usr/share/man/man1/mysql.1
> /usr/share/man/man1/isamchk.1
> /usr/share/man/man1/isamlog.1
> /usr/share/man/man1/mysql_zap.1
> /usr/share/man/man1/mysqlaccess.1
> /usr/share/man/man1/mysqladmin.1
> /usr/share/man/man1/mysqld.1
> /usr/share/man/man1/mysqld_multi.1
> /usr/share/man/man1/mysqldump.1
> /usr/share/man/man1/mysqlshow.1
> /usr/share/man/man1/perror.1
> /usr/share/man/man1/replace.1
> /usr/share/man/man1/safe_mysqld.1
> /usr/share/mysql
> /usr/share/mysql/mi_test_all
> /usr/share/mysql/mi_test_all.res
> /usr/share/mysql/czech
> /usr/share/mysql/czech/errmsg.sys
> /usr/share/mysql/czech/errmsg.txt
> /usr/share/mysql/danish
> /usr/share/mysql/danish/errmsg.sys
> /usr/share/mysql/danish/errmsg.txt
> /usr/share/mysql/dutch
> /usr/share/mysql/dutch/errmsg.sys
> /usr/share/mysql/dutch/errmsg.txt
> /usr/share/mysql/english
> /usr/share/mysql/english/errmsg.sys
> /usr/share/mysql/english/errmsg.txt
> /usr/share/mysql/estonian
> /usr/share/mysql/estonian/errmsg.sys
> /usr/share/mysql/estonian/errmsg.txt
> /usr/share/mysql/french
> /usr/share/mysql/french/errmsg.sys
> /usr/share/mysql/french/errmsg.txt
> /usr/share/mysql/german
> /usr/share/mysql/german/errmsg.sys
> /usr/share/mysql/german/errmsg.txt
> /usr/share/mysql/greek
> /usr/share/mysql/greek/errmsg.sys
> /usr/share/mysql/greek/errmsg.txt
> /usr/share/mysql/hungarian
> /usr/share/mysql/hungarian/errmsg.sys
> /usr/share/mysql/hungarian/errmsg.txt
> /usr/share/mysql/italian
> /usr/share/mysql/italian/errmsg.sys
> /usr/share/mysql/italian/errmsg.txt
> /usr/share/mysql/charsets/hp8.conf
> /usr/share/mysql/charsets/hungarian.conf
> /usr/share/mysql/charsets/koi8_ru.conf
> /usr/share/mysql/charsets/koi8_ukr.conf
> /usr/share/mysql/charsets/latin1.conf
> /usr/share/mysql/charsets/latin2.conf
> /usr/share/mysql/charsets/latin5.conf
> /usr/share/mysql/charsets/swe7.conf
> /usr/share/mysql/charsets/usa7.conf
> /usr/share/mysql/charsets/win1250.conf
> /usr/share/mysql/charsets/win1251.conf
> /usr/share/mysql/charsets/win1251ukr.conf
> /usr/share/mysql/make_binary_distribution
> /usr/share/mysql/mysql.server
> /usr/share/mysql/my-small.cnf
> /usr/share/mysql/my-medium.cnf
> /usr/share/mysql/my-large.cnf
> /usr/share/mysql/my-huge.cnf
> /usr/share/mysql/binary-configure
> /usr/mysql-test
> /usr/include/mysql
> /usr/include/mysql/dbug.h
> /usr/include/mysql/m_string.h
> /usr/include/mysql/my_sys.h
> /usr/include/mysql/mysql.h
> /usr/include/mysql/mysql_com.h
> /usr/include/mysql/mysqld_error.h
> /usr/include/mysql/my_list.h
> /usr/include/mysql/my_pthread.h
> /usr/include/mysql/my_no_pthread.h
> /usr/include/mysql/raid.h
> /usr/include/mysql/errmsg.h
> /usr/include/mysql/my_global.h
> /usr/include/mysql/my_net.h
> /usr/include/mysql/sslopt-case.h
> /usr/include/mysql/sslopt-longopts.h
> /usr/include/mysql/sslopt-usage.h
> /usr/include/mysql/sslopt-vars.h
> /usr/include/mysql/mysql_version.h
> /usr/include/mysql/m_ctype.h
> /usr/include/mysql/my_config.h
> /usr/include/mysql/readline.h
> /usr/include/mysql/chardefs.h
> /usr/include/mysql/keymaps.h
> /usr/include/mysql/history.h
> /usr/include/mysql/tilde.h
> /var/run/mysqld

24 Database Server - PostgreSQL

In this Chapter

Linux PostgreSQL Database Server

Abstract

PostgreSQL, developed originally in the UC Berkeley Computer Science Department, pioneered many of the object-relational concepts now becoming available in commercial databases. It provides SQL92/SQL3 language support, transaction integrity, and type extensibility.

As explained on the PostgreSQL web site:
PostgreSQL is an object-relational database management system (ORDBMS) based on POSTGRES, Version 4.2, developed at the University of California at Berkeley Computer Science Department. The POSTGRES project, led by Professor Michael Stonebraker, was sponsored by the Defense Advanced Research Projects Agency (DARPA), the Army Research Office (ARO), the National Science Foundation (NSF), and ESL, Inc. It is the most advanced open-source database available anywhere.

Recommended RPM packages to be installed for a SQL Server

A minimal configuration provides the basic set of packages required by the Linux operating system. A minimal configuration is a perfect starting point for building a secure operating system. Below is the list of all recommended RPM packages required to run your Linux server as a database Server (SQL) running PostgreSQL software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install PostgreSQL by RPM package. Therefore, postgresql and postgresql-server RPM packages are already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages since compilers packages are not installed and included in the list.

basesystem	ed	less	passwd	sysklogd
bash	file	libstdc++	**perl**	syslinux
bdflush	filesystem	libtermcap	popt	SysVinit
bind	fileutils	lilo	**postgresql**	tar
bzip2	findutils	logrotate	**postgresql-server**	termcap
chkconfig	gawk	losetup	procps	textutils
console-tools	gdbm	MAKEDEV	psmisc	tmpwatch
cpio	gettext	man	pwdb	utempter
cracklib	glib	mingetty	qmail	util-linux
cracklib-dicts	glibc	mktemp	readline	vim-common
crontabs	glibc-common	mount	rootfiles	vim-minimal
db1	grep	ncurses	rpm	vixie-cron
db2	groff	net-tools	sed	words
db3	gzip	newt	setup	which
dev	info	openssh	sh-utils	zlib
devfsd	initscripts	openssh-server	shadow-utils	
diffutils	iptables	openssl	slang	
e2fsprogs	kernel	pam	slocate	

Tested and fully functional on OpenNA.com.

These installation instructions assume

Commands are Unix-compatible.
The source path `is /var/tmp` (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest `PostgreSQL` version number is `7.1.2`

Packages

The following is based on information as listed by `PostgreSQL` as of 2001/05/18. Please regularly check at www.postgresql.org for the latest status.

Source code is available from:
`PostgreSQL` Homepage: http://www.postgresql.org/
`PostgreSQL` FTP Site: 216.126.84.28
You must be sure to download: `postgresql-7.1.2.tar.gz`

Prerequisites

`PostgreSQL` requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive file. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ To enable and use `SSL` encryption support into the software, `OpenSSL` library should be already installed on your system.

NOTE: For more information on `OpenSSL` software, please see earlier chapters in this book its related chapter.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `PostgreSQL`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  `[root@deep /root]#` **`find /* > PostgreSQL1`**
- And the following one after you install the software:
  `[root@deep /root]#` **`find /* > PostgreSQL2`**
- Then use the following command to get a list of what changed:
  `[root@deep /root]#` **`diff PostgreSQL1 PostgreSQL2 > PostgreSQL-Installed`**

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing PostgreSQL

Below are the required steps that you must make to compile and optimize the PostgreSQL database software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp postgresql-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf postgresql-version.tar.gz
```

Step 2
In order to check that the version of PostgreSQL, which you are going to install, is an original and unmodified one, use the command described below and check the supplied signature.

- To verify the MD5 checksum of PostgreSQL, use the following command:

```
[root@deep tmp]# md5sum postgresql-version.tar.gz
```

This should yield an output similar to this:

```
8e2e4319828a8a38492c3ce06726237c  postgresql-7.1.2.tar.gz
```

Now check that this checksum is exactly the same as the one available into a file called "postgresql-7.1.2.tar.gz.md5" on the PostgreSQL FTP site: 216.126.84.28

Step 3
To avoid security risks, we must create a new user and group account called "postgres" to be the owner of the PostgreSQL database files and daemon.

- To create this special PostgreSQL user/group account, use the following commands:

```
[root@deep tmp]# groupadd -g 26 postgres >/dev/null 2>&1 || :
[root@deep tmp]# useradd -M -n -g postgres -o -r -d /var/lib/pgsql -s
/bin/bash -c "PostgreSQL Server" -u 26 postgres >/dev/null 2>&1 || :
```

Step 4
After that, move into the newly created PostgreSQL source directory and perform the following steps to configure and optimize PostgreSQL for your system.

- To move into the newly created PostgreSQL source directory use the command:

```
[root@deep tmp]# cd postgresql-7.1.2/
```

- To configure and optimize PostgreSQL use the following compilation lines:

```
CFLAGS="-O3 -static -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
CXXFLAGS="-O3 -static -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer -
felide-constructors -fno-exceptions -fno-rtti" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var/lib/pgsql \
--mandir=/usr/share/man  \
--disable-shared \
--enable-syslog \
--with-openssl
```

This tells PostgreSQL to set itself up for this particular configuration setup with:

- Disable shared libraries to compile statically linked programs.
- Enables the PostgreSQL server to use the syslog logging facility.
- Build with OpenSSL for encryption support.

WARNING: There is a performance penalty associated with the use of locale support (--enable-locale), but if you are not in an English-speaking environment you will most likely need this configuration line. This option is not included in our compilation lines above.

Make a special attention to the compilation CXXFLAGS and CFLAGS lines in the above step. We optimize PostgreSQL for an i686 CPU architecture with the parameter "-march=i686 and -mcpu=i686" and compile it statically with the options "-static" and "--disable-shared" for optimum performance of the database server. Please don't forget to adjust this CXXFLAGS and CFLAGS lines to reflect your own system and CPU architecture.

Step 5
Now, we must make a list of all existing files on the system before installing the software, and one afterwards, then compare them using the **diff** utility tool of Linux to find out what files are placed where and finally install PostgreSQL server:

```
[root@deep postgresql-7.1.2]# make all
[root@deep postgresql-7.1.2]# cd
[root@deep /root]# find /* > PostgreSQL1
[root@deep /root]# cd /var/tmp/postgresql-7.1.2/
[root@deep postgresql-7.1.2]# make install
[root@deep postgresql-7.1.2]# rm -rf /usr/doc/
[root@deep postgresql-7.1.2]# mkdir -p /var/lib/pgsql
[root@deep postgresql-7.1.2]# chmod 700 /var/lib/pgsql/
[root@deep postgresql-7.1.2]# chown -R postgres.postgres /var/lib/pgsql/
[root@deep postgresql-7.1.2]# touch /var/log/postgresql
[root@deep postgresql-7.1.2]# chown postgres.postgres /var/log/postgresql
[root@deep postgresql-7.1.2]# chmod 600 /var/log/postgresql
[root@deep postgresql-7.1.2]# strip /usr/bin/postgres
[root@deep postgresql-7.1.2]# strip /usr/bin/ecpg
[root@deep postgresql-7.1.2]# strip /usr/bin/pg_id
[root@deep postgresql-7.1.2]# strip /usr/bin/pgrep
[root@deep postgresql-7.1.2]# strip /usr/bin/pg_dump
[root@deep postgresql-7.1.2]# strip /usr/bin/pg_passwd
[root@deep postgresql-7.1.2]# strip /usr/bin/psql
[root@deep postgresql-7.1.2]# cd
[root@deep /root]# find /* > PostgreSQL2
[root@deep /root]# diff PostgreSQL1 PostgreSQL2 > PostgreSQL-Installed
```

The **make** command compiles all source files into executable binaries, and the **make install** will installs the binaries and any supporting files into the appropriate locations. We use the

command **mkdir -p** to create the directory database of PostgreSQL called "pgsql" under /var/lib.

The **strip** command will discard all symbols from the object files. This means that our binaries files will be smaller in size. This will improve the performance hit to the program since there will be fewer lines to read by the system when it executes the binary.

Step 6
At this stage, all files and binaries related to PostgreSQL server have been installed onto your computer. It is time to verify if the postgres daemon is linked statically as we want it to be.

- To verify if the postgres daemon is linked statically, use the following command:
```
[root@deep /]# ldd /usr/bin/postgres
        not a dynamic executable
```

If the returned result of the command is the same as the one shown above (not a dynamic executable), congratulations! Every library required by the daemon to run successfully on your server has been included directly into the postgres binaries.

Step 7
Once the configuration, optimization, compilation, and installation of the database software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete PostgreSQL and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf postgresql-version/
[root@deep tmp]# rm -f postgresql-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install PostgreSQL. It will also remove the PostgreSQL compressed archive from the /var/tmp directory.

Configuring PostgreSQL

After PostgreSQL has been built and installed successfully in your system, your next step is to configure and customize its different configuration files. Those files are:

- ✓ /var/lib/pgsql/data/postgresql.conf (The PostgreSQL Configuration File)
- ✓ /etc/logrotate.d/postgres (The PostgreSQL Log rotation File)
- ✓ /etc/rc.d/init.d/postgresql (The PostgreSQL Initialization File)

/var/lib/pgsql/data/postgresql.conf: The PostgreSQL Config File

The /var/lib/pgsql/data/postgresql.conf file is used to specify PostgreSQL system configuration information. This file is checked to get the required information each time the database starts its daemon.

- Edit the **postgresql.conf** file (vi /var/lib/pgsql/data/postgresql.conf) and change the following lines:

```
fsync = false
```

```
max_connections = 128
shared_buffers = 256
silent_mode = true
syslog = 2
log_connections = true
log_timestamp = true
ssl = true
tcpip_socket = false
```

This tells `postgresql.conf` file to set itself up for this particular configuration with:

`fsync = false`
This option "`fsync`" if set to "`false`" allows the operating system to do its best in buffering, sorting, and delaying writes, which can make for a considerable performances increase. If you trust your Linux operating system, your hardware and `UPS`, you can disable this option safety otherwise enable it. This is a performance feature.

`max_connections = 128`
This option "`max_connections`" determines how many concurrent connections the database server will allow. There is also a compiled-in hard upper limit on this value, which is typically 1024. We increase the default value of "32" to become 128.

`shared_buffers = 256`
This option "`shared_buffers`" determines the number of shared memory buffers the database server will use. Typically, the integer must be two times (2*) the value of "`max_connections`" parameter, which become in our configuration "`256`" (2*128=256). This is a performance feature.

`silent_mode = true`
This option "`silent_mode`" if set to "`true`" will automatically run `postmaster` in background and any controlling `ttys` will be disassociated, thus no messages are written to `stdout` or `stderr`. Since we use `syslog` program on our Linux system to report error messages, we can safety disable this option.

`syslog = 2`
This option "`syslog`" if set to "`2`" will enable the use of `syslog` for logging and will sends output only to `syslog` on the system.

`log_connections = true`
This option "`log_connections`" if set to "`true`" will prints a line informing about each successful connection to the server log.

`log_timestamp = true`
This option "`log_timestamp`" if set to "`true`" will prefixes each server log message with a timestamp.

`ssl = true`
This option "`ssl`", if set to "`true`", will enable `SSL` connection for this `PostgreSQL` server. See later for more information about `SSL` with `PostgreSQL` and how to use it if you need it.

`tcpip_socket = false`
This option "`tcpip_socket`", if set to "`false`", will accept only local Unix domain socket connections. If you want to allow external connection to your `PostgeSQL` server, then you must change the default value of "`false`" to become "`true`" and see later in this chapter what this implies and how to secure and control external users connection.

`/etc/logrotate.d/postgres`: The `PostgreSQL` Log rotation File

The `/etc/logrotate.d/postgres` file allows the `PostgreSQL` database server to automatically rotate its log files at a specified time. Here we'll configure the `/etc/logrotate.d/postgres` file to rotate automatically each week its log files.

- Create the **`postgres`** file (`touch /etc/logrotate.d/postgres`) and add the lines:

```
/var/log/postgresql {
        notifempty
        missingok
        copytruncate
}
```

`/etc/rc.d/init.d/postgresql`: The `PostgreSQL` Initialization File

The `/etc/rc.d/init.d/postgresql` script file is responsible to automatically starting and stopping the `postmaster` daemon of `PostgreSQL` on your server. Loading the daemon, as a standalone daemon will eliminate load times and will even reduce swapping since non-library code will be shared.

Step 1
Create the **`postgresql`** script file (`touch /etc/rc.d/init.d/postgresql`) and add the following lines:

```
#! /bin/bash
# postgresql    This is the init script for starting up the PostgreSQL
#               server
#
# chkconfig: - 78 12
# description: Starts and stops the PostgreSQL backend daemon that handles \
#              all database requests.
# processname: postmaster
# pidfile: /var/run/postmaster.pid
#
# PGVERSION is:
PGVERSION=7.1.2

# Source function library.
INITD=/etc/rc.d/init.d
. $INITD/functions

# Get function listing for cross-distribution logic.
TYPESET=`typeset -f|grep "declare"`
POSTGRESQL="postgresql"

# Get config.
. /etc/sysconfig/network
```

```
# Check that networking is up.
# Pretty much need it for postmaster.
[ ${NETWORKING} = "no" ] && exit 0

[ -f /usr/bin/postmaster ] || exit 0


start(){
        echo -n $"Checking postgresql installation: "

        # Check for older PGDATA location.
        if [ -f /var/lib/pgsql/PG_VERSION ] && [ -d
/var/lib/pgsql/base/template1 ]
        then
                export PGDATA=/var/lib/pgsql
        else
                export PGDATA=/var/lib/pgsql/data
        fi

        # Check for the PGDATA structure
        if [ -f $PGDATA/PG_VERSION ] && [ -d $PGDATA/base/template1 ]
        then
        # Check version of existing PGDATA

                if [ `cat $PGDATA/PG_VERSION` != '7.1.2' ]
                then
                        SYSDOCDIR="(Your System's documentation directory)"
                        if [ -d /usr/share/doc/postgresql-$PGVERSION ]
                        then
                                SYSDOCDIR=/usr/share/doc
                        fi
                        echo
                        echo  $"An old version of the database format was
found."
                        echo  $"You need to upgrade the data format before
using PostgreSQL."
                        exit 1
                else
                        if echo "$TYPESET"|grep "declare -f success ()"
>/dev/null
                        then
                                success "Checking postgresql installation: "
                        else
                                echo_success
                        fi
                        echo
                fi

        # No existing PGDATA! Initdb it.

        else
              echo  $"no database files found."
              if [ ! -d $PGDATA ]
              then
                      mkdir -p $PGDATA
                      chown postgres.postgres $PGDATA
              fi
              echo -n  $"Initializing database..."
              su -l postgres -c '/usr/bin/initdb -D --pglib=/usr/lib \
              --pgdata=/var/lib/pgsql/data' < /dev/null > /dev/null 2>& 1
                echo_success
                echo
        fi
```

```
        # Check for postmaster already running...
        pid=`pidof postmaster`
        if [ $pid ]
        then
                echo $"Postmaster already running."
        else
                #all systems go -- remove any stale lock files
                rm -f /tmp/.s.PGSQL.* > /dev/null
                echo -n $"Starting postgresql service: "
                su -l postgres -c "/usr/bin/pg_ctl  -D $PGDATA -p
/usr/bin/postmaster start >/dev/null 2>&1" < /dev/null
                sleep 2
                pid=`pidof postmaster`
                if [ $pid ]
                then
                        if echo "$TYPESET"|grep "declare -f success ()"
>/dev/null
                        then
                                success "Starting postgresql service: "
                        else
                                echo_success
                        fi
                        touch /var/lock/subsys/postgresql
                        echo $pid > /var/run/postmaster.pid
                        echo
                else
                        if echo "$TYPESET"|grep "declare -f failure ()"
>/dev/null
                        then
                                failure "Starting postgresql service: "
                        else
                                echo_failure
                        fi
                        echo
                fi
        fi
}

stop(){
        echo -n $"Stopping postgresql service: "
        killproc postmaster
        sleep 2
        rm -f /var/run/postmaster.pid
        rm -f /var/lock/subsys/postgresql
        echo
}

restart(){
        stop
        start
}

condrestart(){
    [ -e /var/lock/subsys/postgresql ] && restart || :
}


# See how we were called.
case "$1" in
  start)
        start
        ;;
```

```
  stop)
        stop
        ;;
  status)
        status postmaster
        ;;
  restart)
        restart
        ;;
  condrestart)
        condrestart
        ;;
  *)
        echo  $"Usage: $0 {start|stop|status|restart|condrestart}"
        exit 1
esac

exit 0
```

Step 2
Once the postgresql script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/postgresql
[root@deep /]# chown 0.0 /etc/rc.d/init.d/postgresql
```

- To create the symbolic rc.d links for postgresql, use the following commands:
```
[root@deep /]# chkconfig --add postgresql
[root@deep /]# chkconfig --level 345 postgresql on
```

- To start PostgreSQL software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/postgresql start
Checking postgresql installation: no database files found.
Initializing database...            [OK]
Starting postgresql service:        [OK]
```

Step 3
Once the SQL server has been started, it's time to verify that it is working. With the PostgreSQL server default installation, the only user capable to connect to the database is the user we have created previously to handle the database files and daemons called "postgres".

- To connect to the PostgreSQL database, perform the following actions:
```
[root@deep /]# psql template1 -U postgres
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit
```

```
template1=# \q
```

As you can see in the above example, we connect to the database named "template1" through the interactive terminal program "psql" which allows you to interactively enter, edit, and execute SQL commands.

Step 4
Finally, if the SQL server is running and working, it's time to assign a password to the super-user of this database. With PostgreSQL server, this super-user is named by default postgres and has no password assigned to it, which means that anyone could connect with this name and do anything to the database.

- To specify a password for the PostgreSQL super-user, perform the following actions:

```
[root@deep /]# psql template1 -U postgres
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# ALTER USER postgres WITH PASSWORD 'mypasswd';
ALTER USER
template1=# \q
```

The value **'mypasswd'** as shown above is where you put the password you want to assign for the PostgreSQL super-user (this is the only value you must change in the above command).

NOTE: All software we describe in this book has a specific directory and subdirectory in the tar compressed archive named floppy-2.0.tgz containing configuration files for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files manually or cut and paste them to create or change your configuration files. Whether you decide to copy manually or get the files made for your convenience from the archive compressed files, it will be to your responsibility to modify them to adjust for your needs, and place the files related to this software to the appropriate places on your server. The server configuration file archive to download is located at the following Internet address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz.

Running PostgreSQL with SSL support

This section applies only if you want to run PostgreSQL through SSL connection. Below I show you how to set up a certificate to use with PostgreSQL. As you can imagine, the principle is the same as for creating a certificate for a Web Server (refer to OpenSSL chapter if you have problem creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the PostgreSQL (SQL) Server for which you want to request a certificate. When you want to access your database Server through sql.mydomain.com then the FQDN of your SQL Server is sql.mydomain.com.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key **not protected with a pass-phrase** for the PostgreSQL Server (it is important to create a RSA private key **without** a pass-phrase since the PostgreSQL Server cannot ask you during start-up to enter the pass-phrase). The command below will generate 1024 bit RSA Private Key and stores it in the file server.key.

- To generate the Key, use the following command:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -rand
random1:random2:random3:random4:random5 -out server.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
```

WARNING: Please backup your **server.key** file. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give a name like sql.mydomain.com when prompted for `Common Name'. Do not enter your personal name here. We are requesting a certificate for a Database SQL Server, so the Common Name has to match the FQDN of your site.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key server.key -out server.csr
Using configuration from /usr/share/ssl/openssl.cnf
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA.com]:
Organizational Unit Name (eg, section) [OpenNA.com SQL Server]:
```

```
Common Name (eg, YOUR name) [sql.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**CommonName**" (i.e. when you generate a CSR for a Database Server which will be later accessed via sql.mydomain.com, enter sql.mydomain.com here).

After generation of your **C**ertificate **S**igning **R**equest (CSR), you could send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed Certificate and store it into an server.crt file. The result is then a real Certificate, which can be used for PostgreSQL.

Step 5
You are not obligated to send your **C**ertificate **S**igning **R**equest (CSR) to a commercial **C**ertifying **A**uthority (CA) for signing. In some cases, and with PostgreSQL Server, you can become your own **C**ertifying **A**uthority (CA) and sign your certificate for yourself. In the step below, I assume that your CA keys pair, which are required for signing certificate by yourself already exist on the server, if this is not the case, please refer to the chapter related to OpenSSL in this book for more information about how to create your CA keys pair and become your own **C**ertifying **A**uthority (CA).

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign.sh server.csr
CA signing: ldap.csr -> server.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA.com'
organizationalUnitName:PRINTABLE:'OpenNA.com SQL Server'
commonName            :PRINTABLE:'sql.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Mar 15 07:15:45 2002 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: server.crt <-> CA cert
server.crt: OK
```

This signs the CSR and results in a server.crt file.

Step 6
Now, we must place the certificates files (server.key and server.crt) in the data directory of PostgreSQL (/var/lib/pgsql/data) and change their default permission modes to be (0400/-r--------), owned by the user called 'postgres' for PostgreSQL to be able to find and use them when it will start its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv server.key /var/lib/pgsql/data/
[root@deep ssl]# mv server.crt /var/lib/pgsql/data/
[root@deep ssl]# chmod 400 /var/lib/pgsql/data/server.key
[root@deep ssl]# chmod 400 /var/lib/pgsql/data/server.crt
[root@deep ssl]# chown postgres.postgres /var/lib/pgsql/data/server.key
[root@deep ssl]# chown postgres.postgres /var/lib/pgsql/data/server.crt
[root@deep ssl]# rm -f server.csr
```

First we move server.key and server.crt files to the data directory of PostgreSQL. After that we change the permission mode and ownership of both certificates to be only readable and owned by the PostgreSQL user called 'postgres' for security reason. Finally we remove the server.csr file from our system since it is no longer needed.

Step 7
To allow SSL-enabled connections with PostgreSQL, we must change/add one parameter into the postgresql.conf file.

- Edit the **postgresql.conf** file (vi /var/lib/pgsql/postgresql.conf), and change the following line:

```
#ssl = false
```

To read:

```
ssl = true
```

Step 8
Finally, we must restart our PostgreSQL server for the changes to take effect.

- To restart PostgreSQL use the following command:

```
[root@deep /]# /etc/rc.d/init.d/postgresql restart
Stopping postgresql service:          [OK]
Initializing database. . .            [OK]
Starting postgresql service:          [OK]
```

Securing PostgreSQL

This section deals with the actions we can make to improve and tighten security with the PostgreSQL database. The interesting point here is that we refer to the features available within the base installed program and not to any additional software.

The PostgreSQL Host-Based Access Control File

PostgreSQL contains a file named pg_hba.conf located under /var/lib/pgsql/data directory. The meaning of this file is to control who can connect to each available database on the server. Once you look into this file, you'll inevitably remark that connections from clients can be made using a so-called **Unix domain sockets** or **Internet domain sockets** (i.e. TCP/IP).

Unix domain sockets is when a connection to the database appears from the locahost and Internet domain sockets, as its name imply, is when a connection to the database comes from the external (e.i the Internet) but by default all connections from a client to the database server are allowed only via the local Unix socket, not via TCP/IP sockets and the backend must be started with the "tcpip_socket" option set to "true" in the postgresql.conf file to allow non-local clients to connect.

Below, I show some examples for the configuration of the Host-Based Access Control File of PostgreSQL for Unix domain sockets and Internet domain sockets.

Unix domain sockets

Connections made using Unix domain sockets are controlled as follows into the pg_hba.conf file:

```
local DBNAME AUTHTYPE
```

Where **DBNAME** specifies the database that this record applies to. The value "all" specifies that it applies to all databases and the value "sameuser" specifies to restrict a user's access to a database with the same user name.

AUTHTYPE specifies the authentication method a user must use to authenticate them selves when connecting to that database. The different important available methods are:

1) **trust** which means that a connection is allowed unconditionally.

2) **reject** which means that a connection is rejected unconditionally.

3) **crypt** which means that the client is asked for a password for the user. This is sent encrypted and compared against the password held in the pg_shadow system catalog table and, if the passwords match, the connection is allowed.

4) **password** which means that the client is asked for a password for the user. This is sent in clear text and compared against the password held in the pg_shadow system catalog table again, if the passwords match, the connection is allowed.

Step 1
Now let's see a working example:

- Edit the **pg_hba.conf** file (vi /var/lib/pgsql/data/pg_hba.conf), and change the following lines at the end of the file:

```
# By default, allow anything over UNIX domain sockets and localhost.
local        all                                          trust
host         all         127.0.0.1     255.255.255.255     trust
```

To read:

```
# By default, allow anything over UNIX domain sockets and localhost
# only if the user's password in pg_shadow is supplied.
local        all                                          crypt
host         all         127.0.0.1     255.255.255.255     crypt
```

In the above example, we allow all users from UNIX domain sockets and the localhost to connect to all databases, if the user's password in the pg_shadow system catalog table is supplied.

Recall that user passwords are optionally assigned when a user is created; therefore verify if your users are passwords assigned to them before setting this option.

Step 2
Once the necessary modifications have been set into the `pg_hba.conf` file, it is time to verify if the access control security has been applied to the database.

- Connect to the database called `template1`, by using the following command:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \q
```

If the system asks you to enter a password, congratulations!

Internet domain sockets

Connections made using Internet domain sockets are controlled as follows into the `pg_hba.conf` file:

```
host DBNAME IP_ADDRESS ADDRESS_MASK AUTHTYPE
```

The format is the same as that of the "**`local`**" record type except that the **`IP_ADDRESS`** and **`ADDRESS_MASK`** are added. **`IP_ADDRESS`** and **`ADDRESS_MASK`** are a standard dotted decimal `IP` address and mask to identify a set of hosts. These hosts are allowed to connect to the database **`DBNAME`** if the values match.

Step 1
Now see, a working example:

- Edit the **`pg_hba.conf`** file (`vi /var/lib/pgsql/data/pg_hba.conf`), and change the following lines at the end of the file:

```
# By default, allow anything over UNIX domain sockets and localhost
# only if the user's password in pg_shadow is supplied.
local        all                                        crypt
host         all         127.0.0.1     255.255.255.255     crypt
```

To read:

```
# By default, allow anything over UNIX domain sockets and localhost
# only if the user's password in pg_shadow is supplied.
local        all                                        crypt
host         all         127.0.0.1     255.255.255.255     crypt
host         all         0.0.0.0       0.0.0.0             reject
host         all         207.35.78.0   255.255.255.0       crypt
```

In the above example, we kept our previous setting which allow all users from UNIX domain sockets and localhost to connect to all databases, if the user's password in the `pg_shadow` system catalog table is supplied. But we have added two new lines, related to the Internet domain sockets, that say deny anyone from everywhere, except from any host with `IP` address `207.35.78.x` to make a connection to all databases, unless the user's password in the `pg_shadow` system catalog table is supplied. Recall that user passwords are optionally assigned when a user is created; therefore verify that your users passwords are assigned to them before setting this option.

NOTE: Note that a "`host`" record will allow regular connections and `SSL` together. If you want to accept only `SSL`-secured connections from this host or hosts, you must change every "`host`" record to become "`hostssl`" in your `pg_hba.conf` file.

Step 2
Remember that by default all connections from a client to the database server are only allowed via the local Unix socket, therefore it is important to allow traffic through the `PostgreSQL` port `5432` into our firewall script file for the database to accept an external connection.

- Edit the **`iptables`** script file (`vi /etc/rc.d/init.d/iptables`), and add/check the following lines to allow `PostgreSQL` packets to traverse the network:

```
# PostgreSQL server (5432)
# ----------------

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR --source-port $UNPRIVPORTS \
         --destination-port 5432 -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
         --source-port 5432 \
         -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT
```

Where `EXTERNAL_INTERFACE="eth0"` # Internet connected interface
Where `IPADDR="207.35.78.9"` # Your IP address for eth0
Where `UNPRIVPORTS="1024:"` # Unprivileged port range

Step 3
Another important fact is that the backend must be started with the "`tcpip_socket`" option set to "`true`" into the `postgresql.conf` file to allow non-local clients to connect.

- Edit the **`postgresql.conf`** file (`vi /var/lib/pgsql/data/postgresql.conf`) and change the following line:

```
fsync = false
max_connections = 128
shared_buffers = 256
silent_mode = true
syslog = 2
log_connections = true
log_timestamp = true
ssl = true
tcpip_socket = false
```

To read:

```
fsync = false
max_connections = 128
shared_buffers = 256
silent_mode = true
syslog = 2
log_connections = true
log_timestamp = true
ssl = true
tcpip_socket = true
```

Step 4
Once the required modifications have been made, it is time to verify if the access control security is applied to the database from the external connection.

- Connect to the database called `template1` from external, by using the command:

```
[root@ullyse /]# psql –h 207.35.78.9 template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \q
```

If the system asks you to enter a password, congratulations!

Optimizing PostgreSQL

This section deals with actions we can make to improve and tighten performance of PostgreSQL database. Take a note that we refer to the features available within the base installed program.

Get some fast SCSI hard disk

One of the most important parts of optimizing PostgreSQL server as well as for the majority of all SQL databases is the speed of your hard disk, the faster it is, the faster your database will run. Consider a SCSI disk with low seek times like 4.2ms, this can make all the difference, even greater performance can be made with RAID technology.

Skip the updating of the last access time

As you're supposed to know now, the **noatime** attribute of Linux eliminates the need by the system to make writes to the file system for files. Mounting the file system where your PostgreSQL databases live with the noatime attribute will avoid some disk seeks and will improve the performance of you SQL server.

If you want to mount the file system of the PostgreSQL database with the noatime attribute, it's important to create and install the PostgreSQL databases in this partition. In our example, we have create this partition early in the chapter 2 of this book named "Linux Installation" and this partition is located on /var/lib.

Step 1
To mount the file system of PostgreSQL databases with the noatime option, you must edit the **fstab** file (vi /etc/fstab) and add into the line that refer to /var/lib file system the noatime option after the defaults option as show below:

- Edit the **fstab** file (vi /etc/fstab), and change the line:

```
LABEL=/var/lib    /var/lib        ext2    defaults        1 2
```

To read:

```
LABEL=/var/lib    /var/lib        ext2    defaults,noatime        1 2
```

NOTE: The line related to /var/lib into your /etc/fstab file could be different from the one I show above, this is just an example.

Step 2
Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the Linux system about the modifications.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /var/lib -oremount
```

Each file system that has been modified must be remounted with the command as shown above. In our example we have modified the /var/lib file system and it is for this reason that we remount this file system with the above command.

Step 3
After you file system has been remounted, it is important to verify that the modification of the fstab file has been correctly applied.

- You can verify if the modification has been correctly applied with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root     /           ext2 rw 0 0
/proc         /proc       proc rw 0 0
/dev/sda1     /boot       ext2 rw 0 0
/dev/sda10    /cache      ext2 rw 0 0
/dev/sda9     /chroot     ext2 rw 0 0
/dev/sda8     /home       ext2 rw 0 0
/dev/sda13    /tmp        ext2 rw 0 0
/dev/sda7     /usr        ext2 rw 0 0
/dev/sda11    /var        ext2 rw 0 0
/dev/sda12    /var/lib    ext2 rw,noatime 0 0
none /dev/pts devpts rw 0 0
```

This command will show you all the file systems on your Linux server and the parameters applied to them. If you see something like:

```
/dev/sda12 /var/lib   ext2    rw,noatime 0 0
```

Congratulations!

NOTE: Look under chapter related to Linux Kernel in this book for more information about the `noatime` attribute and other tunable parameters.

PostgreSQL Administrative Tools

The commands listed below are some that we use often but many more exist and you must check the reference manual for more information.

With PostgreSQL Server, passwords can be managed with the query language commands CREATE USER and ALTER USER, it can also be managed with shell script wrappers around the SQL command called `createuser` and `dropuser`. By default, if no password has been set up, the stored password is NULL and password authentication will always fail for that user.

The CREATE USER query language command

The first example below is the step to follow with the CREATE USER query language command. In this example we'll create one user named "sqladmin" with no password and limited privileges.

- To create a new user in your PostgreSQL server with no password and limited privileges, use the following commands:

```
[root@deep /]# psql template1 -U postgres
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# CREATE USER sqladmin;
CREATE USER
template1=# \q
```

Since we have not specified any additional clauses to the above query language command, the default clauses will be to deny the new added user the ability to create both databases and new users himself.

- To create a new user in your PostgreSQL server with password "mo" and privileges to create databases and new users himself, use the following commands:

```
[root@deep /]# psql template1 -U postgres
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# CREATE USER sqladmin WITH PASSWORD 'mo' CREATEDB CREATEUSER;
CREATE USER
template1=# \q
```

The ALTER USER query language command

The ALTER USER query language command can be used to modify user account information on the database. It is important to note that only a database super-user can change privileges and password expiration with this command. Ordinary users can only change their own password.

- To modifies a user account in your PostgreSQL server, use the following commands:

```
[root@deep /]# psql template1 -U postgres
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# ALTER USER sqladmin WITH PASSWORD 'mi' NOCREATEUSER;
CREATE USER
template1=# \q
```

In the above example, we modify password for the user sqladmin to become "mi" instead of "mo" and deny him the possibility to created new users by himself.

The shell scripts wrapper createuser and dropuser

The shell script wrapper creteuser command is the second method to create new users for the database. It's interesting to know this method too since many third party programs use it during user creation. In the example below, we use the same users name as above to show you the difference between the both methods.

- To create a new user named sqladmin in your PostgreSQL database with no password and privileges to create databases and new users himself, use the commands:

```
[root@deep /]# su postgres
bash-2.04$ createuser
Enter name of user to add: sqladmin
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) y
Password:
CREATE USER
bash-2.04$ exit
exit
```

Here we create a new user with no password set named sqladmin with privileges to create databases and new users himself.

- To create a new user named sqladmin in your PostgreSQL database with password "mo" and privileges to create databases but not new users himself, use the commands:

```
[root@deep /]# su postgres
bash-2.04$ createuser -P
Enter name of user to add: sqladmin
Enter password for user "sqladmin":
Enter it again:
Shall the new user be allowed to create databases? (y/n) y
Shall the new user be allowed to create more new users? (y/n) n
Password:
CREATE USER
bash-2.04$ exit
exit
```

- To remove a user named sqladmin in your PostgreSQL database, use the commands:

```
[root@deep /]# su postgres
bash-2.04$ dropuser
Enter name of user to delete: sqladmin
Password:
DROP USER
bash-2.04$ exit
exit
```

NOTE: By default, users do not have write access to databases they did not create. All files stored within the database are protected from being read by any account other than the postgres super-user account.

The basic commands

Most of you already know how SQL database and in our case PostgreSQL work, but for others, this is the first time. Below, I show you the basic commands for managing a database.

- To create a new database called “StoreOpenNA” with PostgreSQL, use the commands:

```
[root@deep /]# su postgres
bash-2.04$ createdb StoreOpenNA
Password:
CREATE DATABASE
bash-2.04$ exit
exit
```

- To remove a database called “StoreOpenNA” with PostgreSQL, use the commands:

```
[root@deep /]# su postgres
bash-2.04$ dropdb StoreOpenNA
Password:
DROP DATABASE
bash-2.04$ exit
exit
```

- To create a new database called “StoreOpenNA” with the PostgreSQL terminal monitor program (psql), use the following commands:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# CREATE DATABASE StoreOpenNA;
CREATE DATABASE
template1=# \q
```

NOTE: Client connections can be restricted by IP address and/or user name via the "**pg_hba.conf**" file under /var/lib/pgsql/data directory.

Other useful PostgreSQL terminal monitor program (psql) which allows you to interactively enter, edit, and execute SQL commands are:

- To connect to the new database "StoreOpenNA", use the following command:

```
[root@deep /]# psql template1 -U postgres
Password:
Welcome to psql, the PostgreSQL interactive terminal.

Type:  \copyright for distribution terms
       \h for help with SQL commands
       \? for help on internal slash commands
       \g or terminate with semicolon to execute query
       \q to quit

template1=# \c storeopenna
You are now connected to database storeopenna.
storeopenna=# \q
```

- To create a table called "bar" under the database storeopenna, use the command:

```
storeopenna=# CREATE TABLE bar (i int4, c char(16));
CREATE
storeopenna=#
```

- To inspect the new table called "bar", use the following command:

```
storeopenna=# \d bar
           Table "bar"
 Attribute |   Type   | Modifier
-----------+----------+----------
 i         | integer  |
 c         | char(16) |

storeopenna=# \q
```

List of installed PostgreSQL files on your system

> /etc/rc.d/init.d/postgresql
> /etc/logrotate.d/postgres
> /var/log/postgresql
> /usr/bin/postmaster
> /usr/bin/postgres
> /usr/bin/ecpg
> /usr/bin/initdb
> /usr/bin/initlocation
> /usr/bin/ipcclean
> /usr/bin/pg_ctl
> /usr/bin/pg_dump
> /usr/bin/pg_restore
> /usr/bin/pg_dumpall
> /usr/bin/pg_id
> /usr/bin/pg_passwd
> /usr/bin/psql
> /usr/share/man/manl/create_trigger.l
> /usr/share/man/manl/create_type.l
> /usr/share/man/manl/create_user.l
> /usr/share/man/manl/create_view.l
> /usr/share/man/manl/declare.l
> /usr/share/man/manl/delete.l
> /usr/share/man/manl/drop_aggregate.l
> /usr/share/man/manl/drop_database.l
> /usr/share/man/manl/drop_function.l
> /usr/share/man/manl/drop_group.l
> /usr/share/man/manl/drop_index.l
> /usr/share/man/manl/drop_language.l
> /usr/share/man/manl/drop_operator.l
> /usr/share/man/manl/drop_rule.l
> /usr/share/man/manl/drop_sequence.l
> /usr/share/man/manl/drop_table.l

> /usr/bin/createdb
> /usr/bin/dropdb
> /usr/bin/createuser
> /usr/bin/dropuser
> /usr/bin/droplang
> /usr/bin/vacuumdb
> /usr/bin/createlang
> /usr/bin/pg_config
> /usr/lib/libpq.a
> /usr/lib/libecpg.a
> /usr/lib/libpgeasy.a
> /usr/share/man/man1/createdb.1
> /usr/share/man/man1/createlang.1
> /usr/share/man/man1/createuser.1
> /usr/share/man/man1/dropdb.1
> /usr/share/man/man1/droplang.1
> /usr/share/man/man1/dropuser.1
> /usr/share/man/man1/ecpg.1
> /usr/share/man/man1/initdb.1
> /usr/share/man/man1/initlocation.1
> /usr/share/man/man1/ipcclean.1
> /usr/share/man/man1/pgaccess.1
> /usr/share/man/man1/pg_config.1
> /usr/share/man/man1/pg_ctl.1
> /usr/share/man/man1/pg_dump.1
> /usr/share/man/man1/pg_dumpall.1
> /usr/share/man/man1/pg_passwd.1
> /usr/share/man/man1/pg_restore.1
> /usr/share/man/man1/pgtclsh.1
> /usr/share/man/man1/pgtksh.1
> /usr/share/man/man1/postgres.1
> /usr/share/man/man1/postmaster.1
> /usr/share/man/man1/psql.1
> /usr/share/man/man1/vacuumdb.1
> /usr/share/man/manl
> /usr/share/man/manl/abort.l
> /usr/share/man/manl/alter_group.l
> /usr/share/man/manl/alter_table.l
> /usr/share/man/manl/alter_user.l
> /usr/share/man/manl/begin.l
> /usr/share/man/manl/checkpoint.l
> /usr/share/man/manl/close.l
> /usr/share/man/manl/cluster.l
> /usr/share/man/manl/comment.l
> /usr/share/man/manl/commit.l
> /usr/share/man/manl/copy.l
> /usr/share/man/manl/create_aggregate.l
> /usr/share/man/manl/create_constraint_trigger.l
> /usr/share/man/manl/create_database.l
> /usr/share/man/manl/create_function.l
> /usr/share/man/manl/create_group.l
> /usr/share/man/manl/create_index.l
> /usr/share/man/manl/create_language.l
> /usr/share/man/manl/create_operator.l
> /usr/share/man/manl/create_rule.l
> /usr/share/man/manl/create_sequence.l
> /usr/share/man/manl/create_table_as.l
> /usr/share/man/manl/create_table.l
> /usr/share/man/manl/drop_trigger.l
> /usr/share/man/manl/drop_type.l
> /usr/share/man/manl/drop_user.l
> /usr/share/man/manl/drop_view.l
> /usr/share/man/manl/end.l
> /usr/share/man/manl/explain.l
> /usr/share/man/manl/fetch.l
> /usr/share/man/manl/grant.l
> /usr/share/man/manl/insert.l
> /usr/share/man/manl/listen.l
> /usr/share/man/manl/load.l
> /usr/share/man/manl/lock.l
> /usr/share/man/manl/move.l
> /usr/share/man/manl/notify.l
> /usr/share/man/manl/reindex.l
> /usr/share/man/manl/reset.l
> /usr/share/man/manl/revoke.l
> /usr/share/man/manl/rollback.l
> /usr/share/man/manl/select_into.l
> /usr/share/man/manl/select.l
> /usr/share/man/manl/set_constraints.l
> /usr/share/man/manl/set.l
> /usr/share/man/manl/set_transaction.l
> /usr/share/man/manl/show.l
> /usr/share/man/manl/truncate.l
> /usr/share/man/manl/unlisten.l
> /usr/share/man/manl/update.l
> /usr/share/man/manl/vacuum.l
> /usr/share/postgresql
> /usr/share/postgresql/global.bki
> /usr/share/postgresql/global.description
> /usr/share/postgresql/template1.bki
> /usr/share/postgresql/template1.description
> /usr/share/postgresql/pg_hba.conf.sample
> /usr/share/postgresql/pg_ident.conf.sample
> /usr/share/postgresql/postgresql.conf.sample
> /usr/include/postgresql
> /usr/include/postgresql/lib
> /usr/include/postgresql/lib/dllist.h
> /usr/include/postgresql/libpq
> /usr/include/postgresql/libpq/pqcomm.h
> /usr/include/postgresql/libpq/libpq-fs.h
> /usr/include/postgresql/c.h
> /usr/include/postgresql/postgres_ext.h
> /usr/include/postgresql/postgres_fe.h
> /usr/include/postgresql/os.h
> /usr/include/postgresql/config.h
> /usr/include/postgresql/libpq-fe.h
> /usr/include/postgresql/libpq-int.h
> /usr/include/postgresql/pqexpbuffer.h
> /usr/include/postgresql/ecpgerrno.h
> /usr/include/postgresql/ecpglib.h
> /usr/include/postgresql/ecpgtype.h
> /usr/include/postgresql/sqlca.h
> /usr/include/postgresql/sql3types.h
> /usr/include/postgresql/libpgeasy.h
> /var/lib/pgsql
> /var/log/postgresql

25 Database Server - OpenLDAP

In this Chapter

Linux OpenLDAP Server

Abstract

Until now, we have been talking about security and optimization in this book, so why would we talk about OpenLDAP? Well, the OpenLDAP directory server will expand our horizons through its many possibilities. We can use its replication capability to centralize and consolidate different information on one server for all the others in our network.

Imagine having the possibility of adding or disabling a Unix or NT account, setting access to a restricted Web server, and adding a mail address or alias, all with a single operation available as an NIS service, with the added security of SSL encryption, and the speed of object-oriented hierarchies. Another interesting use is to create an authoritative list of employees on one or more LDAP servers that can be accessible from your private network, or over the Internet.

At present OpenLDAP on Linux is typically used to associate names with phone numbers and e-mail addresses, but in the future this will almost certainly change. Directories are designed to support a high volume of queries since the data in the directory doesn't change all that often, therefore we can imagine an interesting use of OpenLDAP for possible Domain Name System alternative.

As explained in the OpenLDAP web site:
LDAP (**L**ightweight **D**irectory **A**ccess **P**rotocol) is an open-standard protocol for accessing information services. The protocol runs over Internet transport protocols, such as TCP, and can be used to access stand-alone directory servers or X.500 directories. X.500 is an international standard for directories full-featured, which is complex and requires lots of computing resources and the full OSI stack. LDAP, in contrast, can run easily on a PC and over TCP/IP protocol.

In our configuration and installation we'll run OpenLDAP as non root-user and in a chrooted environment with TSL/SSL support. You can configure different kinds of backend databases with OpenLDAP. A high-performance, disk-based database named "LDBM"; a database interface to arbitrary UNIX commands or shell scripts named "SHELL"; a simple password file database named "PASSWD", and other like SQL.

The default installation of OpenLDAP assumes an LDBM backend database and this is the one that we'll show you in this chapter. For the other type of backend database, you must add in your configuration lines the required options.

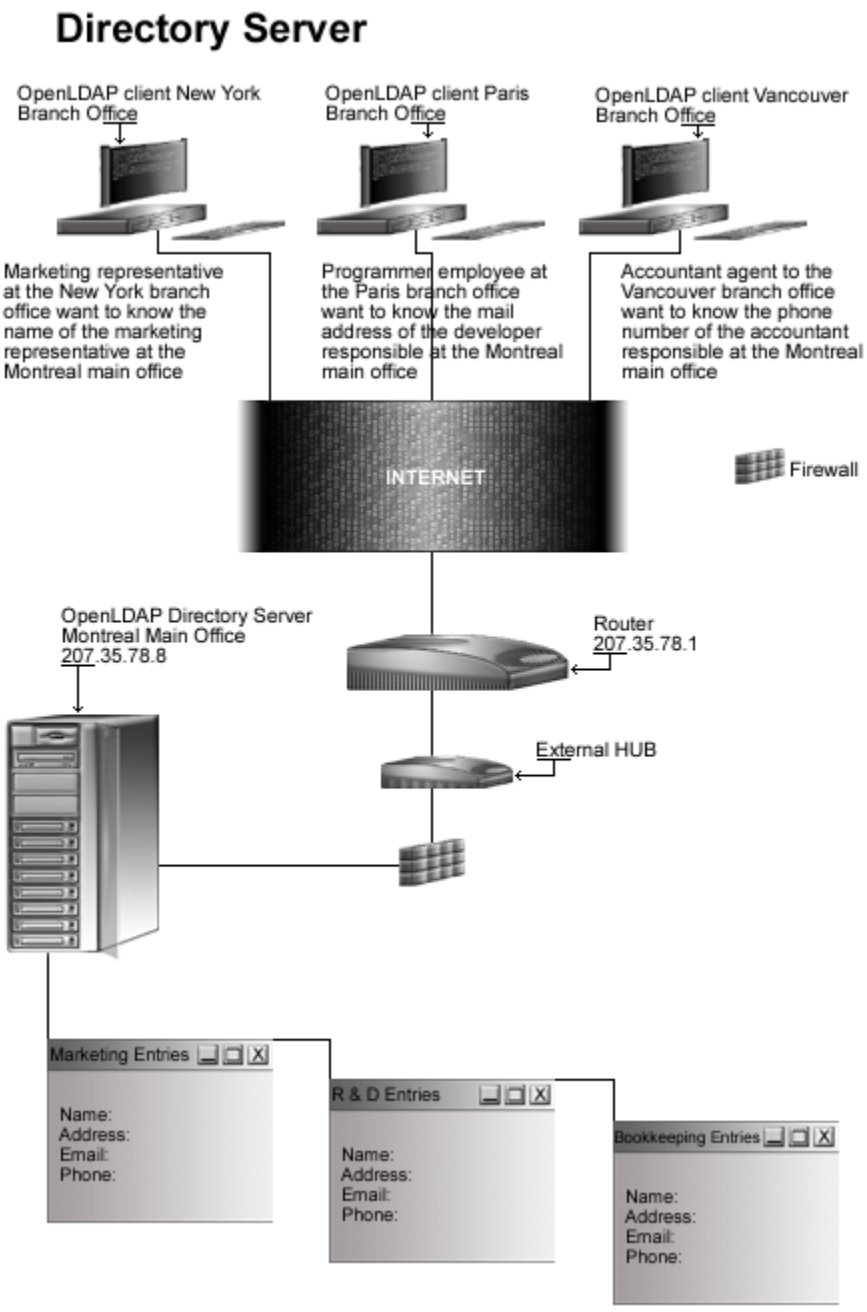
Recommended RPM packages to be installed for a LDAP Server

A minimal configuration provides the basic set of packages required by the Linux operating system. A minimal configuration is a perfect starting point for building s secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a Lightweight Directory Access Protocol (LDAP) server running on OpenLDAP software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install OpenLDAP by RPM package. Therefore, openldap, openldap-servers, and openldap-clients RPM packages are already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

basesystem	ed	less	openssh-server	slocate
bash	file	libstdc++	openssl	sysklogd
bdflush	filesystem	libtermcap	pam	syslinux
bind	fileutils	lilo	passwd	SysVinit
bzip2	findutils	logrotate	**perl**	tar
chkconfig	gawk	losetup	popt	termcap
console-tools	gdbm	MAKEDEV	procps	textutils
cpio	gettext	man	psmisc	tmpwatch
cracklib	glib	mingetty	pwdb	utempter
cracklib-dicts	glibc	mktemp	qmail	util-linux
crontabs	glibc-common	mount	readline	vim-common
db1	grep	ncurses	rootfiles	vim-minimal
db2	groff	net-tools	rpm	vixie-cron
db3	gzip	newt	sed	words
dev	info	**openldap**	setup	which
devfsd	initscripts	**openldap-servers**	sh-utils	zlib
diffutils	iptables	**openldap-clients**	shadow-utils	
e2fsprogs	kernel	openssh	slang	

Tested and fully functional on OpenNA.com.


Directory Server
OpenLDAP client New York Branch Office
OpenLDAP client Paris Branch Office
OpenLDAP client Vancouver Branch Office
Marketing representative at the New York branch office want to know the name of the marketing representative at the Montreal main office
Programmer employee at the Paris branch office want to know the mail address of the developer responsible at the Montreal main office
Accountant agent to the Vancouver branch office want to know the phone number of the accountant responsible at the Montreal main office
INTERNET
Firewall
OpenLDAP Directory Server
Montreal Main Office
207.35.78.8
Router
207.35.78.1
External HUB
Marketing Entries
Name:
Address:
Email:
Phone:
R & D Entries
Name:
Address:
Email:
Phone:
Bookkeeping Entries
Name:
Address:
Email:
Phone:

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest OpenLDAP version number is 2.0.11

Packages

The following are based on information as listed by OpenLDAP as of 2001/05/29. Please regularly check at www.openldap.org for the latest status.

Source code is available from:
OpenLDAP Homepage: http://www.openldap.org/
OpenLDAP FTP Site: 204.152.186.57
You must be sure to download: openldap-2.0.11.tgz

Prerequisites

OpenLDAP requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it from your Linux CD-ROM or source archive file. Please make sure you have this program installed on your machine before you proceed with this chapter.

- ✓ To enable and use TLS/SSL encryption support into the software, OpenSSL library should be already installed on your system.

NOTE: For more information on OpenSSL software, please see earlier chapters in this book its related chapter.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install OpenLDAP, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > OpenLDAP1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > OpenLDAP2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff OpenLDAP1 OpenLDAP2 > OpenLDAP-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing OpenLDAP

Below are the required steps that you must make to compile and optimize the OpenLDAP Lightweight Directory Access Protocol (LDAP) server software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp openldap-version.tgz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf openldap-version.tgz
```

Step 2
In order to check that the version of OpenLDAP, which you are going to install, is an original and unmodified one, use the command described below and check the supplied signature.

- To verify the MD5 checksum of OpenLDAP, use the following command:

```
[root@deep tmp]# md5sum openldap-version.tgz
```

This should yield an output similar to this:

```
e51b06374012b9e7077e1f3e9f65ccd0  openldap-2.0.11.tgz
```

Now check that this checksum is exactly the same as the one available into a file called "openldap-2.0.11.md5" on the OpenLDAP FTP site: 204.152.186.57

Step 3
To avoid security risks, we must create a new user account called "ldap" to be the owner of the OpenLDAP database files and daemon.

- To create this special OpenLDAP user account, use the following command:

```
[root@deep tmp]# useradd -r -d /var/lib/ldap -s /bin/false -c "OpenLDAP
Server" -u 55 ldap >/dev/null 2>&1 || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID.

Step 4
After that, move into the newly created OpenLDAP source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created OpenLDAP source directory use the command:

```
[root@deep tmp]# cd openldap-2.0.11/
```

Step 5
There are some source files to modify before going in configuration and compilation of the program; the changes allow us to fix some problems.

- Edit the **slap.h** file (`vi +15 servers/slapd/slap.h`) and change the lines:

```
#include <sys/types.h>
#include <ac/syslog.h>
#include <ac/regex.h>
#include <ac/socket.h>
#include <ac/time.h>
#include <ac/param.h>
```

To read:

```
#include <sys/types.h>
#include <sys/socket.h>
#include <ac/syslog.h>
#include <ac/regex.h>
#include <ac/socket.h>
#include <ac/time.h>
#include <ac/param.h>
```

- Edit the **openldap.m4** file (`vi +604 build/openldap.m4`) and change the lines:

```
{
        return (void *) (p == NULL);
}
])
```

To read:

```
{
        sleep(30);
        return (void *) (p == NULL);
}
])
```

- Edit the **back-ldbm.h** file (`vi +23 servers/slapd/back-ldbm/back-ldbm.h`) and change the lines:

```
#endif

#define DEFAULT_DB_DIRECTORY    LDAP_RUNDIR LDAP_DIRSEP "openldap-ldbm"
#define DEFAULT_MODE            0600
```

To read:

```
#endif

#define DEFAULT_DB_DIRECTORY   "/var/lib/ldap"
#define DEFAULT_MODE            0600
```

Step 6
Once the required modifications have been made in the related source files of OpenLDAP, it is time configure and optimize it for our system.

- To configure and optimize OpenLDAP use the following compilation lines:

```
CC="gcc" \
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer -
D_REENTRANT -fPIC" \
./configure \
--prefix=/usr \
--libexecdir=/usr/sbin \
--sysconfdir=/etc \
--localstatedir=/var/run \
--mandir=/usr/share/man \
--disable-debug \
--disable-ipv6 \
--enable-crypt \
--with-tls \
--without-threads
```

This tells OpenLDAP to set itself up for this particular configuration setup with:

- Disable debugging support to improve performance.
- Disable IPv6 support.
- Enable crypt(3) passwords support.
- Enable and include TLS/SSL encryption support into the program.
- Disable threads support for OpenLDAP on the system.

NOTE: The default installation of OpenLDAP assumes an LDBM backend database, so if you want to configure another type of backend database, you must specify it during configuration and compilation time. For a SHELL backend database you must add the "--enable-shell" option and for a PASSWD backend database which can be used as replacement for NIS service, you must add the "--enable-passwd" option in your configuration lines.

The compile options we choose here assume that you want to set up an LDBM backend database. For the other type of backend database, you must add in your configuration lines the required options.

Step 7
Now, we must make a list of all existing files on the system before installing the software, and one afterwards, then compare them using the **diff** utility tool of Linux to find out what files are placed where and finally install OpenLDAP Lightweight Directory Access Protocol (LDAP) server.

```
[root@deep openldap-2.0.11]# make depend
[root@deep openldap-2.0.11]# make
[root@deep openldap-2.0.11]# cd tests/
[root@deep tests]# make test
[root@deep tests]# cd
[root@deep /root]# find /* > OpenLDAP1
[root@deep /root]# cd /var/tmp/openldap-2.0.11/
[root@deep openldap-2.0.11]# make install
[root@deep openldap-2.0.11]# install -d -m 700 /var/lib/ldap
```

```
[root@deep openldap-2.0.11]# rm -rf /var/run/openldap-ldbm
[root@deep openldap-2.0.11]# chown -R ldap.ldap /var/lib/ldap/
[root@deep openldap-2.0.11]# rm -f /etc/openldap/*.default
[root@deep openldap-2.0.11]# rm -f /etc/openldap/schema/*.default
[root@deep openldap-2.0.11]# strip /usr/lib/liblber.a
[root@deep openldap-2.0.11]# strip /usr/lib/liblber.so.2.0.5
[root@deep openldap-2.0.11]# strip /usr/lib/libldap.a
[root@deep openldap-2.0.11]# strip /usr/lib/libldap.so.2.0.5
[root@deep openldap-2.0.11]# strip /usr/lib/libldap_r.a
[root@deep openldap-2.0.11]# strip /usr/lib/libldap_r.so.2.0.5
[root@deep openldap-2.0.11]# /sbin/ldconfig
[root@deep openldap-2.0.11]# cd
[root@deep /root]# find /* > OpenLDAP2
[root@deep /root]# diff OpenLDAP1 OpenLDAP2 > OpenLDAP-Installed
```

The **make depend** command will build and make the necessary dependencies of different files, **make** will compiles all source files into executable binaries, and then **make install** will installs the binaries and any supporting files into the appropriate locations.

The **make test** command under the subdirectory `/tests` will do some important tests to verify the functionality of your `OpenLDAP` server before the installation. If any of the tests fail, you'll need to FIXE the problems before continuing the installation.

The **strip** command will discard all symbols from the object files. This means that our library files will be smaller in size and will improve the performance hit to the program since there will be fewer lines to be read by the system when it uses the libraries.

Step 8
Once the configuration, optimization, compilation, and installation of the Lightweight Directory Server software has been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `OpenLDAP` and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf openldap-version/
[root@deep tmp]# rm -f openldap-version.tgz
```

The `rm` command as used above will remove all the source files we have used to compile and install `OpenLDAP`. It will also remove the `OpenLDAP` compressed archive from the `/var/tmp` directory.

Configuring OpenLDAP

After `OpenLDAP` has been built and installed successfully in your system, your next step is to configure and customize its different configuration files. This is an easy task since there are just two files related to `OpenLDAP`:

- ✓ `/etc/openldap/slapd.conf` (The `OpenLDAP` Configuration File)
- ✓ `/etc/rc.d/init.d/ldap` (The `OpenLDAP` Initialization File)

/etc/openldap/slapd.conf: The OpenLDAP Configuration File

The /etc/openldap/slapd.conf file is the main configuration file for the stand-alone slapd daemon and for all of the database back-ends. Options like: permission, password, database type, database location and so on can be configured in this file and will apply to the "slapd" daemon as a whole.

In the example below we configure the slapd.conf file for an LDBM backend database. The text in bold are the parts of the script initialization file that must be customized and adjusted to satisfy our needs.

Step 1
The first thing to do before starting your Lightweight Directory Access Protocol (LDAP) server is to edit the slapd.conf file and change its contents to reflect your environment.

- Edit the **slapd.conf** file (vi /etc/openldap/slapd.conf) and add/adjust the following information:

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/schema/core.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

#######################################################################
# ldbm database definitions
#######################################################################

database        ldbm
readonly        off
suffix          "dc=openna,dc=com"
rootdn          "cn=Manager,dc=openna,dc=com"

# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          secret

# The database directory MUST exist prior to running slapd AND
# should only be accessable by the slapd/tools. Mode 700 recommended.
directory       /var/lib/ldap

# ldbm indexed attribute definitions
index uid pres,eq
index cn,sn,uid pres,eq,approx,sub
index objectClass eq

# ldbm access control definitions
defaultaccess read
access to attr=userpassword
        by self write
        by dn="cn=Manager,dc=openna,dc=com" write
        by * compare
```

This tells `slapd.conf` file to set itself up for this particular configuration with:

`readonly off`
This directive puts the database into "`read-only`" mode. Any attempts to modify the database will return an "unwilling to perform" error. It is useful when you make you directory service available to the publics.

`suffix "dc=openna,dc=com"`
This directive specifies the Distinguished Name (`DN`) of the root of the sub tree you are trying to create. In other words, it indicates what entries are to be held by this database.

`rootdn "cn=Manager,dc=openna,dc=com"`
This directive specifies the Distinguished Name (`DN`) of the entry allowed to do anything on the `LDAP` directory. This `DN` is not subject to access control or administrative limit restrictions for operations on this database. The name entered here can be one that doesn't actually exist in your password file `/etc/passwd`.

`rootpw secret`
This directive specifies the password that can be used to authenticate the super-user entry of the database. This is the password for the `DN` given above that will always work, regardless of whether an entry with the given `DN` exists or has a password. It's important to avoid the use of clear text passwords here and to use a crypto password instead.

`directory /var/lib/ldap`
This directive specifies the directory where the database and associated indexes files of `LDAP` should reside. We must set this to `/var/lib/ldap` because we created this directory earlier in the installation stage specifically to handle the backend database of `LDAP`.

```
index uid pres,eq
index cn,sn,uid pres,eq,approx,sub
index objectClass eq
```
These directives specify the index definitions you want to build and maintain for the given attribute in the database definition. The options we specifies in our `slapd.conf` example file as shown above, cause all indexes to be maintained for the `cn`, `sn`, and `uid` attributes (`index cn,sn,uid`); an equality (`eq`) indexes for the objectclass attribute (`index objectclass eq`). See your user manual for more information on these options.

```
defaultaccess read
access to attr=userpassword
    by self write
    by dn="cn=Manager,dc=openna,dc=com" write
    by * compare
```
The last directives in the `slapd.conf` file relate to access control in `LDAP` directory. The access configuration file directive as shown above is used to control access to `slapd` daemon entries and attributes in the system.

This example applies to all entries in the "`dc=openna,dc=com`" sub tree and mean that read access is granted to everyone <`defaultaccess read`>, and the entry itself can write all attributes, except for *`userpassword`*. The *`userpassword`* attribute is writable only by the specified `cn` entry (`Manager`), and comparable by everybody else. See your user manual for more information on these options.

Step 2
Once you have set your preferences and environment into the `slapd.conf` file, it is important to change its default mode permission and owner to by the user (`ldap`) under which the Lightweight Directory Access Protocol (`LDAP`) server will runs.

- To change the mode permission and owner of this file, use the following commands:
```
[root@deep /]# chmod 600 /etc/openldap/slapd.conf
[root@deep /]# chown ldap.ldap /etc/openldap/slapd.conf
```

`/etc/rc.d/init.d/ldap`: The `OpenLDAP` Initialization File

The `/etc/rc.d/init.d/ldap` script file is responsible to automatically start and stop the `slapd` daemon of `OpenLDAP` on your system. Loading the daemon, as a standalone will eliminate load time and will even reduce swapping since non-library code will be shared.

Step 1
Create the **`ldap`** script file (`touch /etc/rc.d/init.d/ldap`) and add the following lines:

```
#!/bin/bash
#
# ldap  This shell script takes care of starting and stopping
#       ldap servers (slapd and slurpd).
#
# chkconfig: - 39 61
# description: LDAP stands for Lightweight Directory Access Protocol, used \
#              for implementing the industry standard directory services.
# processname: slapd
# config: /etc/openldap/slapd.conf
# pidfile: /var/run/slapd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration and check that networking is up.
if [ -r /etc/sysconfig/network ] ; then
        . /etc/sysconfig/network
        [ ${NETWORKING} = "no" ] && exit 0
fi

slapd=/usr/sbin/slapd
slurpd=/usr/sbin/slurpd
[ -x ${slapd} ] || exit 0
#[ -x ${slurpd} ] || exit 0

RETVAL=0

function start() {
        # Start daemons.
        echo -n $"Starting slapd: "
        if grep -q ^TLS /etc/openldap/slapd.conf ; then
            daemon ${slapd} -u ldap -h '"ldap:/// ldaps:///"'
            RETVAL=$?
        else
            daemon ${slapd} -u ldap
            RETVAL=$?
        fi
        echo
        if [ $RETVAL -eq 0 ]; then
```

```
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                echo -n $"Starting slurpd: "
                daemon ${slurpd}
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
        return $RETVAL
}

function stop() {
        # Stop daemons.
        echo -n $"Stopping slapd: "
        killproc ${slapd}
        RETVAL=$?
        echo
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                echo -n $"Stopping slurpd: "
                killproc ${slurpd}
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/ldap /var/run/slapd.args
        return $RETVAL
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status ${slapd}
        if grep -q "^replogfile" /etc/openldap/slapd.conf ; then
            status ${slurpd}
        fi
        ;;
    restart)
        stop
        start
        ;;
    reload)
        killall -HUP ${slapd}
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
            if grep -q "^replogfile" /etc/openldap/slapd.conf; then
                killall -HUP ${slurpd}
                RETVAL=$?
            fi
        fi
        ;;
    condrestart)
        if [ -f /var/lock/subsys/ldap ] ; then
            stop
            start
        fi
```

```
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|status|condrestart}"
        RETVAL=1
esac

exit $RETVAL
```

Step 2
Once the `ldap` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reasons, and creation of the symbolic links will let the process control initialization of Linux, which is in charge of starting all the normal and authorized processes that need to run at boot time on your system, to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/ldap
[root@deep /]# chown 0.0 /etc/rc.d/init.d/ldap
```

- To create the symbolic `rc.d` links for `ldap`, use the following commands:
```
[root@deep /]# chkconfig --add ldap
[root@deep /]# chkconfig --level 345 ldap on
```

- To start `OpenLDAP` software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/ldap start
Starting slapd:                    [OK]
```

Step 3
Once the Lightweight Directory Access Protocol (`LDAP`) server has been started, it's time to verify if is running and correctly configured.

- To do it, we will run a search against it with its `ldapsearch` command utility:
```
[root@deep /]# ldapsearch -x -b '' -s base '(objectclass=*)'
namingContexts
```

Note the use of single quotes around command parameters to prevent special characters from being interpreted by the shell.

If everything runs as espected, this should return:
```
version: 2

#
# filter: (objectclass=*)
# requesting: namingContexts
#

#
dn:
namingContexts: dc=openna,dc=com

# search result
search: 2
result: 0 Success

# numResponses: 2
```

```
# numEntries: 1
```

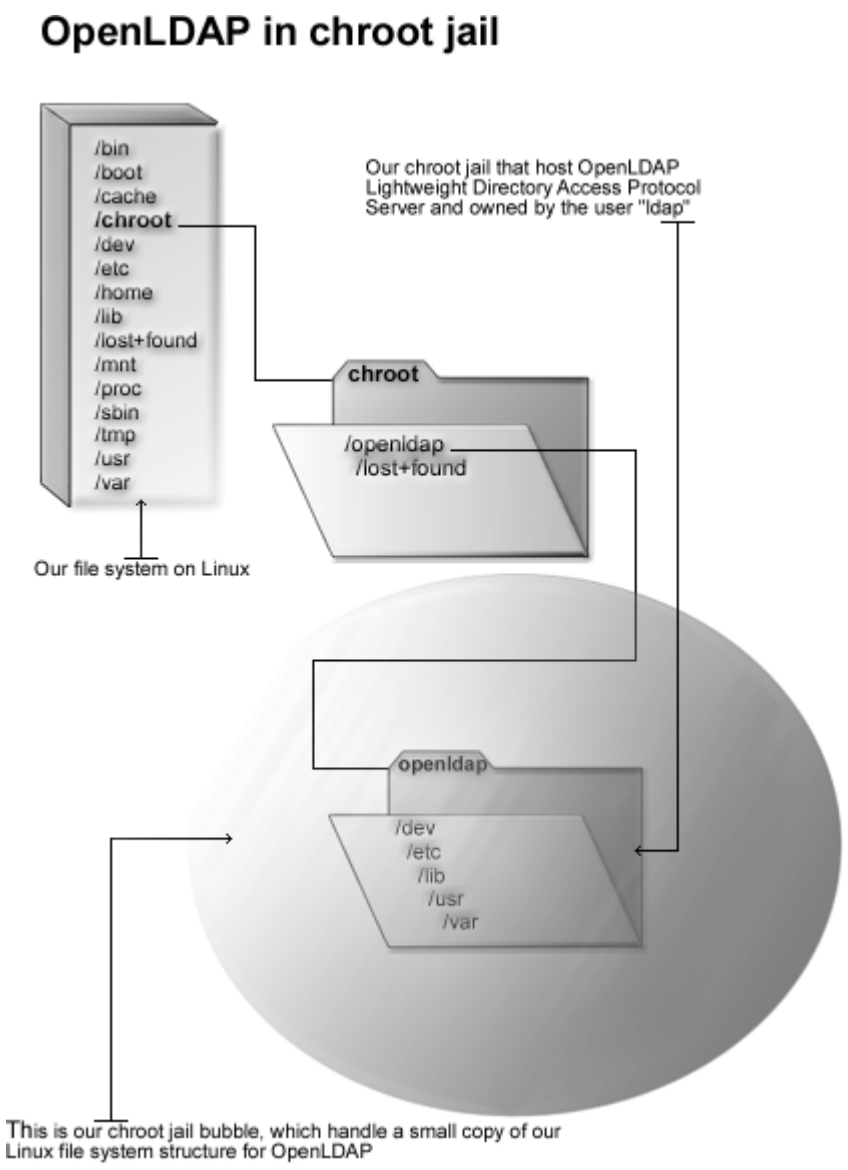
Congratulations!, your Lightweight Directory Access Protocol (LDAP) server is working.

NOTE: All software we describe in this book has a specific directory and subdirectory in the tar compressed archive named `floppy-2.0.tgz` containing configuration files for the specific program. If you get this archive file, you wouldn't be obliged to reproduce the different configuration files manually or cut and paste them to create or change your configuration files. Whether you decide to copy manually or get the files made for your convenience from the archive compressed files, it will be to your responsibility to modify them to adjust for your needs, and place the files related to this software to the appropriate places on your server. The server configuration file archive to download is located at the following Internet address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz.

Running OpenLDAP in a chroot jail

This part focuses on preventing OpenLDAP from being used as a point of break-in to the system hosting it. OpenLDAP by default runs **as a non-root user**, which will limit any damage to what can be done as a normal user with a local shell. Of course, allowing what amounts to an anonymous guest account falls rather short of the security requirements for the OpenLDAP servers, so an additional step can be taken - that is, **running OpenLDAP in a chroot jail**.

The main benefit of a chroot jail is that the jail will limit the portion of the file system the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support OpenLDAP, the programs available into the jail can be extremely limited. Most importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail. By running OpenLDAP in a chroot jail you can improve the security significantly in a Unix environment.


OpenLDAP in chroot jail
/bin
/boot
/cache
/chroot
/dev
/etc
/home
/lib
/lost+found
/mnt
/proc
/sbin
/tmp
/usr
/var
Our chroot jail that host OpenLDAP Lightweight Directory Access Protocol Server and owned by the user "ldap"
chroot
/openldap
/lost+found
Our file system on Linux
openldap
/dev
/etc
/lib
/usr
/var
This is our chroot jail bubble, which handle a small copy of our Linux file system structure for OpenLDAP

Necessary steps to run OpenLDAP in a chroot jail:

What you're essentially doing is creating a skeleton root file system with enough components necessary (directories, libraries, files, etc.) to allow Unix to do a chroot when the OpenLDAP daemon starts.

Step 1

The first step to do for running OpenLDAP in a chroot jail will be to set up the chroot environment, and create the root directory of the jail. We've chosen /chroot/openldap for this purpose because we want to put this on its own separate file system to prevent file system attacks. Early in our Linux installation procedure we created a special partition /chroot for this exact purpose.

```
[root@deep /]# /etc/rc.d/init.d/ldap stop ← Only if OpenLDAP daemon already run.
Stopping slapd:                [OK]

[root@deep /]# mkdir /chroot/openldap
[root@deep /]# mkdir /chroot/openldap/dev
[root@deep /]# mkdir /chroot/openldap/lib
[root@deep /]# mkdir /chroot/openldap/etc
[root@deep /]# mkdir -p /chroot/openldap/usr/share
[root@deep /]# mkdir -p /chroot/openldap/usr/lib
[root@deep /]# mkdir -p /chroot/openldap/usr/sbin
[root@deep /]# mkdir -p /chroot/openldap/var/lib
[root@deep /]# mkdir -p /chroot/openldap/var/run
```

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible.

Step 2

After that, it is important to move the main configuration directory, all configuration files, the database directory and the slapd binary program of the Lightweight Directory Access Protocol (LDAP) server to the chroot jail then create the special devices /dev/null and /dev/urandom which is/are absolutely require by the system to work properly. Note that /dev/urandom is required only if you use TLS/SSL support with OpenLDAP.

```
[root@deep /]# mv /etc/openldap /chroot/openldap/etc/
[root@deep /]# mv /usr/share/openldap /chroot/openldap/usr/share/
[root@deep /]# mv /var/lib/ldap /chroot/openldap/var/lib/
[root@deep /]# mv /usr/sbin/slapd /chroot/openldap/usr/sbin/
[root@deep /]# mknod /chroot/openldap/dev/null c 1 3
[root@deep /]# chmod 666 /chroot/openldap/dev/null
[root@deep /]# mknod /chroot/openldap/dev/urandom c 1 9 ← Only for TLS/SSL.
```

Step 4

This step is required only if you have compiled OpenLDAP with TLS/SSL support. In this case, you must recreate a small copy of the /usr/share/ssl directory with certs and private directories which handles the private and public keys of OpenLDAP to the chroot jail environment.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# mkdir -p /chroot/openldap/usr/share/ssl
[root@deep /]# cd /usr/share/
[root@deep share]# cp -r ssl/certs /chroot/openldap/usr/share/ssl/
[root@deep share]# cp -r ssl/private /chroot/openldap/usr/share/ssl/
```

WARNING: If you have other private and public keys related to other programs and applications into the `certs` and `private` directories, please don't copy them to the jail environment. Only copy the private and public keys related to `OpenLDAP`, which are supposed to be called "`ldap.crt`" and "`ldap.key`" respectively.

Step 5
Now, we must find the shared library dependencies of `slapd` binary and install them into the chroot structure. Use the **`ldd /chroot/openldap/usr/sbin/slapd`** command to find out which libraries are needed. The output (depending on what you've compiled with `OpenLDAP`) will be something similar to:

- To find the shared library dependencies of `slapd`, execute the following command:

```
[root@deep /]# ldd /chroot/openldap/usr/sbin/slapd
libgdbm.so.2 => /usr/lib/libgdbm.so.2 (0x4001b000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x40021000)
libresolv.so.2 => /lib/libresolv.so.2 (0x4004e000)
libc.so.6 => /lib/libc.so.6 (0x40060000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

What we can see here is the fact that depending of what programs have been compiled and included with `OpenLDAP`, the shared library dependencies may differ.

Step 6
Once the required libraries have been identified, copy them to the appropriate location into the chroot jail. In our example these are the shared libraries identified above.

```
[root@deep /]# cp /usr/lib/libgdbm.so.2 /chroot/openldap/usr/lib/
[root@deep /]# cp /lib/libcrypt.so.1 /chroot/openldap/lib/
[root@deep /]# cp /lib/libresolv.so.2 /chroot/openldap/lib/
[root@deep /]# cp /lib/libc.so.6 /chroot/openldap/lib/
[root@deep /]# strip -R .comment /chroot/openldap/usr/lib/*
```

You'll also need the following extra libraries for some network functions, like resolving:

```
[root@deep /]# cp /lib/libnss_compat* /chroot/openldap/lib/
[root@deep /]# cp /lib/libnss_dns* /chroot/openldap/lib/
[root@deep /]# cp /lib/libnss_files* /chroot/openldap/lib/
[root@deep /]# strip -R .comment /chroot/openldap/lib/*
```

NOTE: The "`strip -R .comment`" command will remove all the named section "`.comment`" from the libraries files under the `/usr/lib` and `/lib` directory of the chroot jail and will make them smaller in size to help in performance of them.

Step 7
Now we need to copy the `passwd` and `group` files inside the `/chroot/openldap/etc` directory. Next, we'll remove all entries except for the user that `openldap` runs as in both files (`passwd` and `group`).

```
[root@deep /]# cp /etc/passwd /chroot/openldap/etc/
[root@deep /]# cp /etc/group /chroot/openldap/etc/
```

- Edit the **`passwd`** file under the chroot jail (`vi /chroot/openldap/etc/passwd`) and delete all entries except for the user `openldap` run as (in our configuration, it's "`ldap`"):

```
ldap:x:55:55:OpenLDAP Server:/var/lib/ldap:/bin/false
```

- Edit the **`group`** file under the chroot jail (`vi /chroot/openldap/etc/group`) and delete all entries except the group `openldap` run as (in our configuration it's "`ldap`"):

```
ldap:x:55:
```

Step 8
You will also need **`/etc/resolv.conf`**, **`/etc/nsswitch.conf`**, **`/etc/localtime`**, and **`/etc/hosts`** files in your chroot jail structure.

```
[root@deep /]# cp /etc/resolv.conf /chroot/openldap/etc/
[root@deep /]# cp /etc/nsswitch.conf /chroot/openldap/etc/
[root@deep /]# cp /etc/localtime /chroot/openldap/etc/
[root@deep /]# cp /etc/hosts /chroot/openldap/etc/
```

Step 9
Now we must set some files in the chroot jail directory immutable for better security.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cd /chroot/openldap/etc/
[root@deep etc]# chattr +i passwd
[root@deep etc]# chattr +i group
[root@deep etc]# chattr +i resolv.conf
[root@deep etc]# chattr +i hosts
[root@deep etc]# chattr +i nsswitch.conf
```

WARNING: Don't forget to remove the immutable bit on these files if you have some modifications to bring to them with the command "`chattr -i`".

Step 10
The default `ldap` initialization script file of `OpenLDAP` starts the daemon "`slapd`" and/or "`slurpd`" outside the chroot jail. We must change it now to start `slapd` and or `slurpd` from the chroot jail environment.

Since there are many lines to modify from the original initialization script file of `OpenLDAP` to make it start in the jail environment, I decided to make a new initialization file as shown below. Each line in bold are the one that are different from the original script file. In this way you'll be able to see how I made it.

- Edit the **ldap** script file (`vi /etc/rc.d/init.d/ldap`) and change the following lines:

```
#!/bin/bash
#
# ldap  This shell script takes care of starting and stopping
#       ldap servers (slapd and slurpd) in chroot jail.
#
# chkconfig: - 39 61
# description: LDAP stands for Lightweight Directory Access \
# Protocol, used for implementing the industry standard \
# directory services.
# processname: slapd
# config: /chroot/openldap/etc/openldap/slapd.conf
# pidfile: /var/run/slapd.pid

# Source function library.
. /etc/init.d/functions

# Source networking configuration and check that networking is up.
if [ -r /etc/sysconfig/network ] ; then
        . /etc/sysconfig/network
        [ ${NETWORKING} = "no" ] && exit 0
fi

slapd=/chroot/openldap/usr/sbin/slapd
slurpd=/chroot/openldap/usr/sbin/slurpd
[ -x ${slapd} ] || exit 0
#[ -x ${slurpd} ] || exit 0

RETVAL=0

function start() {
        # Start daemons.
        echo -n $"Starting slapd: "
        if grep -q ^TLS /chroot/openldap/etc/openldap/slapd.conf ; then
          daemon ${slapd} -u ldap -r /chroot/openldap/ -h '"ldap:///
ldaps:///"'
          RETVAL=$?
        else
          daemon ${slapd} -u ldap -r /chroot/openldap/
          RETVAL=$?
        fi
        echo
        if [ $RETVAL -eq 0 ]; then
        if grep -q "^replogfile"
/chroot/openldap/etc/openldap/slapd.conf; then
                echo -n $"Starting slurpd: "
                daemon ${slurpd} -r /chroot/openldap/
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && touch /var/lock/subsys/ldap
        return $RETVAL
}

function stop() {
        # Stop daemons.
        echo -n $"Stopping slapd: "
        killproc ${slapd}
        RETVAL=$?
```

```
        echo
        if [ $RETVAL -eq 0 ]; then
        if grep -q "^replogfile"
/chroot/openldap/etc/openldap/slapd.conf; then
                echo -n $"Stopping slurpd: "
                killproc ${slurpd}
                RETVAL=$?
                echo
            fi
        fi
        [ $RETVAL -eq 0 ] && rm -f /var/lock/subsys/ldap
/var/run/slapd.args
        return $RETVAL
}

# See how we were called.
case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    status)
        status ${slapd}
        if grep -q "^replogfile" /chroot/openldap/etc/openldap/slapd.conf
; then
            status ${slurpd}
        fi
        ;;
    restart)
        stop
        start
        ;;
    reload)
        killall -HUP ${slapd}
        RETVAL=$?
        if [ $RETVAL -eq 0 ]; then
        if grep -q "^replogfile"
/chroot/openldap/etc/openldap/slapd.conf; then
                killall -HUP ${slurpd}
                RETVAL=$?
            fi
        fi
        ;;
    condrestart)
        if [ -f /var/lock/subsys/ldap ] ; then
            stop
            start
        fi
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart|status|condrestart}"
        RETVAL=1
esac

exit $RETVAL
```

Step 11
Finally, we must test the new chrooted jail configuration of our Lightweight Directory Access Protocol (LDAP) server.

- Start the new chrooted jail OpenLDAP with the following command:
```
[root@deep /]# /etc/rc.d/init.d/ldap start
Starting slapd:                            [OK]
```

- If you don't get any errors, do a **ps ax | grep slapd** and see if we're running:
```
[root@deep /]# ps ax | grep slapd
26214 ?        S      0:00 /chroot/openldap/usr/sbin/slapd -u ldap -r
/chroot/openldap
```

If so, lets check to make sure it's chrooted by picking out its process number and doing **ls -la /proc/that_process_number/root/**.

```
[root@deep /]# ls -la /proc/26214/root/
```

If you see something like:
```
dev
etc
lib
usr
var
```

Congratulations! Your Lightweight Directory Access Protocol (LDAP) server in chroot jail is working.

Running OpenLDAP with TLS/SSL support

This section applies only if you want to run OpenLDAP through SSL connection. Finally, the new release of OpenLDAP supports TLS/SSL encryption protocol. This is a very good thing in security area and especially if we remember that in the pass we were complained to hack and play with many poor external program to enable this support into OpenLDAP. Now time is different and as you'll see later in this section, enabling OpenLDAP to support SSL protocol is far easier than before.

Below I show you how to set up a certificate to use with OpenLDAP, the principle is the same as for creating a certificate for a Web Server (refer to OpenSSL chapter if you have problem creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the Lightweight Directory Access Protocol (LDAP) Server for which you want to request a certificate. When you want to access your Lightweight Directory Access Protocol (LDAP) Server through ldap.mydomain.com then the FQDN of your Lightweight Directory Server is ldap.mydomain.com.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key protected with a pass-phrase for your OpenLDAP Server. The command below will generate 1024 bit RSA Private Key and stores it in the file ldap.key. It will ask you for a pass-phrase: use something secure and remember it. Your certificate will be useless without the key. If you don't want to protect your key with a pass-phrase (only if you absolutely trust that server machine, and you make sure the permissions are carefully set so only you can read that key) you can leave out the -des3 option below.

- To generate the Key, use the following command:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -des3 -rand
random1:random2:random3:random4:random5 -out ldap.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

WARNING: Please backup your **ldap.key** file, and remember the pass-phrase you had to enter, at a secure location. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give the name ldap.mydomain.com when prompted for `Common Name'. Do not enter your personal name here. We are requesting a certificate for a Lightweight Directory Access Protocol (LDAP) Server, so the Common Name has to match the FQDN of your website.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key ldap.key -out ldap.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
```

```
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA.com]:
Organizational Unit Name (eg, section) [OpenNA.com LDAP Directory
Server]:
Common Name (eg, YOUR name) [ldap.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**CommonName**" (i.e. when you generate a CSR for a Lightweight Directory Server which will be later accessed via ldap.mydomain.com, enter ldap.mydomain.com here).

After generation of your **C**ertificate **S**igning **R**equest (CSR), you could send this certificate to a commercial **C**ertifying **A**uthority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed Certificate and store it into an ldap.crt file. The result is then a real Certificate, which can be used for OpenLDAP.

Step 5
You are not obligated to send your **C**ertificate **S**igning **R**equest (CSR) to a commercial **C**ertifying **A**uthority (CA) for signing. In some cases and with OpenLDAP Directory Server you can become your own **C**ertifying **A**uthority (CA) and sign your certificate by yourself. In the step below, I assume that your CA keys pair, which are required for signing certificate by yourself, already exist on the server, if this is not the case, please refer to the chapter related to OpenSSL in this book for more information about how to create your CA keys pair and become your own **C**ertifying **A**uthority (CA).

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign.sh ldap.csr
CA signing: ldap.csr -> ldap.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA.com'
organizationalUnitName:PRINTABLE:'OpenNA.com LDAP Directory Server'
commonName            :PRINTABLE:'ldap.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Mar 15 07:15:45 2002 GMT (365 days)
```

```
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: ldap.crt <-> CA cert
ldap.crt: OK
```

This signs the CSR and results in a ldap.crt file.

Step 6
Now, we must place the certificates files (ldap.key and ldap.crt) to the appropriate directories and change their default permission modes to be (0400/-r--------), owned by the user called 'ldap' for OpenLDAP to be able to find and use them when it will start its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv ldap.key private/
[root@deep ssl]# mv ldap.crt certs/
[root@deep ssl]# chmod 400 private/ldap.key
[root@deep ssl]# chmod 400 certs/ldap.crt
[root@deep ssl]# chown ldap.ldap private/ldap.key
[root@deep ssl]# chown ldap.ldap certs/ldap.crt
[root@deep ssl]# rm -f ldap.csr
```

First we move the ldap.key file to the private directory and the ldap.crt file to the certs directory. After that we change the permission mode and ownership of both certificates to be only readable and owned by the OpenLDAP user called 'ldap' for security reasons. Finally we remove the ldap.csr file from our system since it is no longer needed.

Step 7
To allow TLS/SSL-enabled connections with OpenLDAP, we must specify two new options into the slapd.conf file. The text in bold are the parts of the lines that must be customized and adjusted to satisfy your needs.

- Edit the **slapd.conf** file (vi /etc/openldap/slapd.conf), and add the following lines:

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /etc/openldap/schema/core.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

# Enable TLS/SSL connections with OpenLDAP
TLSCertificateFile      /usr/share/ssl/certs/ldap.crt
TLSCertificateKeyFile   /usr/share/ssl/private/ldap.key

#######################################################################
# ldbm database definitions
#######################################################################
```

```
database        ldbm
readonly        off
suffix          "dc=openna,dc=com"
rootdn          "cn=Manager,dc=openna,dc=com"

# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          secret

# The database directory MUST exist prior to running slapd AND
# should only be accessable by the slapd/tools. Mode 700 recommended.
directory       /var/lib/ldap

# ldbm indexed attribute definitions
index uid pres,eq
index cn,sn,uid pres,eq,approx,sub
index objectClass eq

# ldbm access control definitions
defaultaccess read
access to attr=userpassword
        by self write
        by dn="cn=Manager,dc=openna,dc=com" write
        by * compare
```

The `TLSCertificateFile` line specifies the file that contains the `slapd` server certificate, and the `TLSCertificateKeyFile` specifies the file that contains the `slapd` server private key that matches the certificate stored in the `TLSCertificateFile` file.

NOTE: If you are running `OpenLDAP` in chroot jail environment, then the `slapd.conf` file will be located under `/chroot/openldap/etc/openldap` directory and not under `/etc/openldap`.

Step 8
The `OpenLDAP` `TLS/SSL`-enabled connections run by default on port 636. To allow external traffic through this port (`636`), we must add a new rule into our firewall script file for the Lightweight Directory Access Protocol (`LDAP`) server to accept external connections.

- Edit the **`iptables`** script file (`vi /etc/rc.d/init.d/iptables`), and add/check the following lines to allow `OpenLDAP` packets with `TLS/SSL` support to traverse the network:

```
# OpenLDAP TLS server (636)
# ----------------

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR --source-port $UNPRIVPORTS \
         --destination-port 636 -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
         --source-port 636 \
         -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT
```

```
Where EXTERNAL_INTERFACE="eth0"          # Internet connected interface
Where IPADDR="207.35.78.8"               # Your IP address for eth0
Where UNPRIVPORTS="1024:65535"           # Unprivileged port range
```

Step 9
Finally, we must restart our OpenLDAP server and firewall for the changes to take effect.

- To restart OpenLDAP use the following command:
```
[root@deep /]# /etc/rc.d/init.d/ldap restart
Stopping slapd:                    [OK]
Starting slapd:                    [OK]
```

- To restart you firewall use the following command:
```
[root@deep /]# /etc/rc.d/init.d/iptables restart
Shutting Firewalling: done
Starting Firewalling: done
done
```

NOTE: With SSL support acctivated in OpenLDAP, the slapd daemon of the program will ask you during startup to enter the pass phrase of the certificate, therefore don't forget it.

Securing OpenLDAP

This section deals especially with actions we can make to improve and tighten security under OpenLDAP Lightweight Directory Access Protocol (LDAP) server. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Using an encrypted root password

With a default installation of OpenLDAP, clear text passwords for the rootdn are used. Use of strong authentication is encouraged through the use of the slappasswd command utility of the directory server.

Below, I show you how to use an encrypted root password, which is a much better idea than leaving a plain text root password in the slapd.conf file.

Step 1
Our first action will be to use the slappasswd tool of OpenLDAP to generate hashed passwords. The utility will prompt you to enter, twice, the user password that you want it to generate in an encrypted form. The schemes that we must generate is a so called (CRYPT) and we specify it with the "-h" option during hashed password generation.

```
[root@deep /]# /usr/sbin/slappasswd -h {CRYPT}
New password:
Re-enter new password:
{CRYPT}SdmwctNoMkNgQ
```

Here the generated "{CRYPT}SdmwctNoMkNgQ" line is the one that we must copy into the /etc/openldap/slapd.conf file to replace the old clear text password for the rootdn.

Step 2
Once we get the generated hashed password line for our `rootdn`, we must edit the `slapd.conf` file and add it to the `rootpw` line.

- Edit the **`slapd.conf`** file (`vi /etc/openldap/slapd.conf`) and change the line:

  ```
  rootpw          secret
  ```

  To read:

  ```
  rootpw          {CRYPT}SdmwctNoMkNgQ
  ```

NOTE: Use of hashed passwords does not protect passwords during protocol transfer. `TLS` or other eavesdropping protections should be in place before using `LDAP` simple bind. The hashed password values should be protected as if they were clear text passwords.

Immunize important configuration files

The immutable bit can be used to prevent one from accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your `slapd.conf` file has been configured, it's a good idea to immunize it with command like:

```
[root@deep /]# chattr +i /etc/openldap/slapd.conf
```

or:

```
[root@deep /]# chattr +i /chroot/openldap/etc/openldap/slapd.conf
```

if you are running `OpenLDAP` in chroot jail environment.

Optimizing `OpenLDAP`

This section deals especially with actions we can make to improve and tighten performance of `OpenLDAP` Lightweight Directory Access Protocol (`LDAP`) server. Take a note that we refer to the features available within the base installed program.

Get some fast `SCSI` hard disk

One of the most important parts of optimizing `OpenLDAP` server as well as for the majority of all `SQL` database servers is the speed of your hard disk, the faster it is, the faster your database will run. Consider a `SCSI` disk with low seek times, like `4.2ms`, this can make all the difference, much greater performance can also be made with `RAID` technology.

Skip the updating of the last access time

As you're supposed to know now, the **`noatime`** attribute of Linux eliminates the need by the system to make writes to the file system for files. Mounting the file system where your `OpenLDAP` Lightweight Directory Access Protocol (`LDAP`) server live with the `noatime` attribute will avoid some disk seeks and will improve the performance of you directory server.

If you want to mount the file system of the OpenLDAP Lightweight Directory Access Protocol (LDAP) server with the noatime attribute, it's important to create and install its databases in this partition. In our example, we have create this partition early in the chapter 2 of this book named "Linux Installation" and this partition is located on /var/lib.

Step 1
To mount the file system of OpenLDAP Lightweight Directory Access Protocol (LDAP) server with the noatime option, you must edit the **fstab** file (vi /etc/fstab) and add into the line that refer to /var/lib file system the noatime option after the defaults option as show below:

- Edit the **fstab** file (vi /etc/fstab), and change the line:

```
LABEL=/var/lib    /var/lib        ext2    defaults        1 2
```

To read:

```
LABEL=/var/lib    /var/lib        ext2    defaults,noatime        1 2
```

NOTE: The line related to /var/lib into your /etc/fstab file could be different from the one I show you above, this is just an example. Also, if you are running OpenLDAP in chroot jail environment, the file system to mount with the noatime option will be /chroot and not /var/lib.

Step 2
Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the Linux system about the modification.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /var/lib -oremount
```

Each file system that has been modified must be remounted with the command as shown above. In our example we have modified the /var/lib file system and it is for this reason that we remount this file system with the above command.

Step 3
After you file system has been remounted, it is important to verify if the modification into the fstab file has been correctly applied to the Linux system.

- You can verify if the modification has been correctly applied with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root    /           ext2 rw 0 0
/proc        /proc       proc rw 0 0
/dev/sda1    /boot       ext2 rw 0 0
/dev/sda10   /cache      ext2 rw 0 0
/dev/sda9    /chroot     ext2 rw 0 0
/dev/sda8    /home       ext2 rw 0 0
/dev/sda13   /tmp        ext2 rw 0 0
/dev/sda7    /usr        ext2 rw 0 0
/dev/sda11   /var        ext2 rw 0 0
/dev/sda12   /var/lib    ext2 rw,noatime 0 0
none /dev/pts devpts rw 0 0
```

This command will show you all file system in your Linux server with parameters applied to them. If you see something like:

```
/dev/sda12 /var/lib   ext2    rw,noatime 0 0
```

Congratulations!

> **NOTE:** Look under chapter related to Linux Kernel in this book for more information about the `noatime` attribute and other tunable parameters.

Further documentation

For more details, there are several manual pages for `OpenLDAP` that you can read; below I show you just the most important:

```
$ man ldapd (8)                 - LDAP X.500 Protocol Daemon
$ man ldapdelete (1)            - LDAP delete entry tool
$ man ldapfilter.conf (5)       - Configuration file for LDAP get filter routines
$ man ldapfriendly (5)          - Data file for LDAP friendly routines
$ man ldapmodify, ldapadd (1)   - LDAP modify entry and ldap add entry tools
$ man ldapmodrdn (1)            - LDAP modify entry RDN tool
$ man ldappasswd (1)            - Change the password of an LDAP entry
$ man ldapsearch (1)            - LDAP search tool
$ man ldapsearchprefs.conf (5)  - Configuration file for LDAP search preference routines
$ man ldaptemplates.conf (5)    - Configuration file for LDAP display template routines
$ man ldif (5)                  - LDAP Data Interchange Format
$ man slapd (8)                 - Stand-alone LDAP Daemon
$ man slapd.conf (5)            - Configuration file for slapd, the stand-alone LDAP daemon
$ man slurpd (8)                - Standalone LDAP Update Replication Daemon
$ man ud (1)                    - Interactive LDAP Directory Server query program
```

OpenLDAP Administrative Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages of `OpenLDAP` and documentation for more information.

Creating an LDMB backend database

There are two methods to create a database for `LDAP`, the first is off-line with the `slapadd` command utility and the other is on-line with the `ldapadd` command utility.

Usually you use the off-line method when you have many thousands of entries to insert into your database and the on-line method when you have only a small number of entries to put into your database. It is also important to note that the off-line method requires that `slapd` daemon is not running and the on-line method requires that `slapd` daemon of `OpenLDAP` is running.

slapadd

When you install `OpenLDAP` for the first time and have big entries to put in your backend database, it's always a good idea to put all these information into a text file and add them to your backend database with the `slapadd` command utility. This command is used to create the `LDMB` backend database off-line. To do it, the first thing will be to create an `LDIF` (**L**DAP **D**ata **I**nterchange **F**ormat) input file containing a text representation of your entries. To summarize, the `slapadd` tool of `OpenLDAP` converts an `LDIF` file into an `LDBM` back-end database.

Step 1
The text file named "datafiles" below can be used as an example file (of course, your real LDIF input file will handle much more information than this example). A blank line indicates that the entry is finished and that another entry is about to begin.

- Create the **datafiles** file (touch /tmp/datafiles) and add as an example in this file the following lines:

```
# Organization's Entry
dn: dc=openna,dc=com
dc: openna
objectclass: dcObject
objectclass: organization
o: OpenNA.com Inc.
#
# Gerhard's Entry
dn: cn=Gerhard Mourani,dc=openna,dc=com
cn: Gerhard Mourani
sn: Mourani
objectclass: organizationalRole
objectclass: organizationalPerson
#
# Ted's Entry
dn: cn=Ted Nakad,dc=openna,dc=com
cn: Ted Nakad
sn: Nakad
description: Agent & Sales Manager
objectclass: organizationalRole
objectclass: organizationalPerson
```

The above entries shows you some very basic example about how to convert your information into LDIF files before adding them to your new backend directory. Consult the OpenLDAP documentation and especially good books for more information.

NOTE: Before adding any objects under the database, you have to add an entry for your organization, first. This is done with the following entries in the above example.

```
dn: dc=openna,dc=com
dc: openna
objectclass: dcObject
objectclass: organization
o: OpenNA.com Inc.
```

Please note that these entries must be entered only one time to create your organization, after that all you have to do is to add additional information as we do for Gerhard's and Ted's.

Step 2
Once the LDIF input file containing our entries has been created, we must insert them into the Lightweight Directory Access Protocol (LDAP) server.

- To insert the LDIF input file and create the database off-line, use the following command if OpenLDAP runs in no chroot jail environment:

```
[root@deep /]# cd /tmp/
[root@deep tmp]# slapadd -l datafiles
```

The "-l" option specifies the location of the LDIF input file (datafiles) containing the entries in text form to add.

- To insert the LDIF input file and create the database off-line, use the following command if OpenLDAP runs in chroot jail environment:
  [root@deep /]# **cd /tmp/**
  [root@deep tmp]# **slapadd -l datafiles –f /chroot/opendla/etc/openldap/slapd_chroot.conf**

The "-l" option specifies the location of the LDIF input file (datafiles) containing the entries in text form to add and the "-f" option specifies where the slapd.conf configuration file reside. In our case and since the server runs in a chroot jail environment, this file is located under our jail structure and called slapd_chroot.conf, which is a copy of the original slapd.conf file containing path of our chroot jail.

To summarize, if you run OpenLDAP in chroot jail, you must have slpad.conf and slapd_chroot.conf files into /chroot/openldap/etc/openldap directory. The slapd.conf file is the original one and slapd_chroot.conf file is a modified copy of the original file containing the path of our chroot jail environment for the slapadd command utility of OpenLDAP to work off-line.

Below is a working example of the content of the modified copy of slapd.conf file, which I called "slapd_chroot.conf". Of course I suppose that your chroot jail reside under /chroot/openldap directory:

```
# See slapd.conf(5) for details on configuration options.
# This file should NOT be world readable.
#
include         /chroot/openldap/etc/openldap/schema/core.schema

# Define global ACLs to disable default read access.

# Do not enable referrals until AFTER you have a working directory
# service AND an understanding of referrals.
#referral       ldap://root.openldap.org

# Enable TLS/SSL connections with OpenLDAP
TLSCertificateFile      /chroot/openldap/usr/share/ssl/certs/ldap.cert
TLSCertificateKeyFile   /chroot/openldap/usr/share/ssl/private/ldap.key

#######################################################################
# ldbm database definitions
#######################################################################

database        ldbm
readonly        off
suffix          "dc=openna,dc=com"
rootdn          "cn=Manager,dc=openna,dc=com"

# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd.conf(5) for details.
# Use of strong authentication encouraged.
rootpw          {CRYPT}SdmwctNoMkNgQ

# The database directory MUST exist prior to running slapd AND
# should only be accessable by the slapd/tools. Mode 700 recommended.
directory       /chroot/openldap/var/lib/ldap
```

```
# ldbm indexed attribute definitions
index default pres,eq
index objectClass,uid
index cn,sn eq,sub

# ldbm access control definitions
defaultaccess read
access to attr=userpassword
        by self write
        by dn="cn=Manager, dc=openna, dc=com" write
        by * compare
```

NOTE: The `slapd` daemon of `OpenLDAP` is not started in this creation mode. Be sure to replace all required information with the appropriate domain components of your domain name.

`ldapadd`

If the entries in your directory server are already created or if you have only a small amount of information to insert into your backend database, you'll prefer to use the `ldapadd` command utility to do your job on-line. The `ldapadd` utility is used to add entries to your directory while the `LDAP` server is running and expects input in `LDIF` (**L**DAP **D**ata **I**nterchange **F**ormat) form.

Step 1

For example, to add the "`Europe Mourani`" entry using the `ldapadd` tool, you could create a file called "`entries`" with input in `LDIF` form into your `/tmp` directory.

- Create the **`entries`** file (`touch /tmp/entries`) and add as an example in this file the following contents:

```
# Organization's Specifications
dn: dc=openna,dc=com
dc: openna
objectclass: dcObject
objectclass: organization
o: OpenNA.com Inc.
#
# Europe's Entry
dn: cn=Europe Mourani,dc=openna,dc=com
cn: Europe Mourani
sn: Mourani
description: Marketing Representatif
objectclass: organizationalRole
objectclass: organizationalPerson
```

Step 2
Once the **entries** file has been created, we must add its content into the OpenLDAP Lightweight Directory Access Protocol (LDAP) server.

- To actually create the entry on-line in the backend database, use the following command:

```
[root@deep /]# cd /tmp/
[root@deep tmp]# ldapadd -f entries -D "cn=Manager, dc=openna, dc=com" -W
Enter LDAP Password :
adding new entry "dc=openna,dc=com"

adding new entry "cn=Europe Mourani,dc=openna,dc=com"
```

The above command assumes that you have set your rootdn to "cn=Manager,dc=openna, dc=com" and rootpw to an encrypted root password. You will be prompted to enter the encrypted root password.

NOTE: The slapd daemon of OpenLDAP is started in this creation mode. Be sure to replace all required information with the appropriate domain components of your domain name.

ldapmodify
Contrary to relational databases where data is constantly changed, the directory server contains information that is rarely modified once inserted. But, some times you need to modify information, and the ldapmodify tool will help you in your tasks. The ldapmodify command allows you to modify entries on the backend directory server.

Step 1
Assuming that we want to replace the contents of the "Europe Mourani" entry's description attribute with the new value "Marketing Representative", the following steps will do it.

- Create the **lnew** file (touch /tmp/lnew) and add as an example in this file the following contents:

```
dn: cn=Europe Mourani,dc=openna,dc=com
changetype: modify
replace: description
description: Marketing Representative
```

Step 2
Once the lnew file has been created, we must replace the entry in the OpenLDAP directory server with the one contained in this file (lnew).

- To modify the contents of backend database, use the following command:

```
[root@deep /]# cd /tmp/
[root@deep tmp]# ldapmodify -f lnew -D 'cn=Manager, dc=openna, dc=com' -W
Enter LDAP Password:
modifying entry "cn=Europe Mourani,dc=openna,dc=com"
```

OpenLDAP Users Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages of OpenLDAP and documentation for more information.

ldapsearch

The ldapsearch utility searches through the backend database of LDAP directory for information/entries you have requested.

- To search on LDAP directory for entries, use the following command:

```
[root@deep /]# ldapsearch -b 'dc=openna, dc=com' 'cn=europe*'
version: 2

#
# filter: cn=europe*
# requesting: ALL
#

# Europe Mourani,dc=openna,dc=com
dn: cn=Europe Mourani,dc=openna,dc=com
cn: Europe Mourani
sn: Mourani
objectClass: organizationalRole
objectClass: person
description: Marketing Representative

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

This command will retrieve all entries and values for the name **europe** and will print the result to standard output in your terminal.

Some possible uses of OpenLDAP software

OpenLDAP can be used as:

- ✓ A Web Catalogue Server
- ✓ A White Pages Server
- ✓ A Certificate Server
- ✓ An Access Control Server
- ✓ A Network Name Server

List of installed OpenLDAP files on your system

> /etc/rc.d/init.d/ldap
> /etc/openldap
> /etc/openldap/ldap.conf
> /etc/openldap/ldapfilter.conf
> /etc/openldap/ldaptemplates.conf
> /etc/openldap/ldapsearchprefs.conf
> /etc/openldap/slapd.conf
> /etc/openldap/schema
> /etc/openldap/schema/corba.schema
> /etc/openldap/schema/core.schema
> /etc/openldap/schema/cosine.schema
> /usr/share/man/man3/ldap_init_templates.3
> /usr/share/man/man3/ldap_init_templates_buf.3
> /usr/share/man/man3/ldap_free_templates.3
> /usr/share/man/man3/ldap_first_disptmpl.3
> /usr/share/man/man3/ldap_next_disptmpl.3
> /usr/share/man/man3/ldap_oc2template.3
> /usr/share/man/man3/ldap_tmplattrs.3
> /usr/share/man/man3/ldap_first_tmplrow.3
> /usr/share/man/man3/ldap_next_tmplrow.3
> /usr/share/man/man3/ldap_first_tmplcol.3
> /usr/share/man/man3/ldap_next_tmplcol.3

> /etc/openldap/schema/inetorgperson.schema
> /etc/openldap/schema/java.schema
> /etc/openldap/schema/krb5-kdc.schema
> /etc/openldap/schema/misc.schema
> /etc/openldap/schema/nadf.schema
> /etc/openldap/schema/nis.schema
> /etc/openldap/schema/openldap.schema
> /usr/bin/ud
> /usr/bin/ldapsearch
> /usr/bin/ldapmodify
> /usr/bin/ldapdelete
> /usr/bin/ldapmodrdn
> /usr/bin/ldapadd
> /usr/bin/ldappasswd
> /usr/include/lber.h
> /usr/include/lber_types.h
> /usr/include/ldap.h
> /usr/include/ldap_cdefs.h
> /usr/include/ldap_features.h
> /usr/include/ldap_schema.h
> /usr/include/disptmpl.h
> /usr/include/srchpref.h
> /usr/lib/liblber.so.2.0.5
> /usr/lib/liblber.so.2
> /usr/lib/liblber.so
> /usr/lib/liblber.la
> /usr/lib/liblber.a
> /usr/lib/libldap.so.2.0.5
> /usr/lib/libldap.so.2
> /usr/lib/libldap.so
> /usr/lib/libldap.la
> /usr/lib/libldap.a
> /usr/lib/libldap_r.so.2.0.5
> /usr/lib/libldap_r.so.2
> /usr/lib/libldap_r.so
> /usr/lib/libldap_r.la
> /usr/lib/libldap_r.a
> /usr/sbin/in.xfingerd
> /usr/sbin/go500gw
> /usr/sbin/go500
> /usr/sbin/mail500
> /usr/sbin/rp500
> /usr/sbin/fax500
> /usr/sbin/xrpcomp
> /usr/sbin/rcpt500
> /usr/sbin/maildap
> /usr/sbin/slapd
> /usr/sbin/slapadd
> /usr/sbin/slapcat
> /usr/sbin/slapindex
> /usr/sbin/slappasswd
> /usr/share/man/man1/ud.1
> /usr/share/man/man1/ldapdelete.1
> /usr/share/man/man1/ldapmodify.1
> /usr/share/man/man1/ldapadd.1
> /usr/share/man/man1/ldapmodrdn.1
> /usr/share/man/man1/ldappasswd.1
> /usr/share/man/man1/ldapsearch.1
> /usr/share/man/man3/lber-decode.3
> /usr/share/man/man3/ber_get_next.3
> /usr/share/man/man3/ber_skip_tag.3
> /usr/share/man/man3/ber_peek_tag.3
> /usr/share/man/man3/ber_scanf.3
> /usr/share/man/man3/ber_get_int.3
> /usr/share/man/man3/ber_get_stringa.3
> /usr/share/man/man3/ber_get_stringb.3
> /usr/share/man/man3/ber_get_null.3
> /usr/share/man/man3/ber_get_enum.3
> /usr/share/man/man3/ber_get_boolean.3
> /usr/share/man/man3/ldap_entry2text.3
> /usr/share/man/man3/ldap_entry2text_search.3
> /usr/share/man/man3/ldap_vals2text.3
> /usr/share/man/man3/ldap_entry2html.3
> /usr/share/man/man3/ldap_entry2html_search.3
> /usr/share/man/man3/ldap_vals2html.3
> /usr/share/man/man3/ldap_error.3
> /usr/share/man/man3/ldap_perror.3
> /usr/share/man/man3/ld_errno.3
> /usr/share/man/man3/ldap_result2error.3
> /usr/share/man/man3/ldap_errlist.3
> /usr/share/man/man3/ldap_err2string.3
> /usr/share/man/man3/ldap_first_attribute.3
> /usr/share/man/man3/ldap_next_attribute.3
> /usr/share/man/man3/ldap_first_entry.3
> /usr/share/man/man3/ldap_next_entry.3
> /usr/share/man/man3/ldap_count_entries.3
> /usr/share/man/man3/ldap_friendly.3
> /usr/share/man/man3/ldap_friendly_name.3
> /usr/share/man/man3/ldap_free_friendlymap.3
> /usr/share/man/man3/ldap_get_dn.3
> /usr/share/man/man3/ldap_explode_dn.3
> /usr/share/man/man3/ldap_explode_rdn.3
> /usr/share/man/man3/ldap_dn2ufn.3
> /usr/share/man/man3/ldap_getfilter.3
> /usr/share/man/man3/ldap_init_getfilter.3
> /usr/share/man/man3/ldap_init_getfilter_buf.3
> /usr/share/man/man3/ldap_getfilter_free.3
> /usr/share/man/man3/ldap_getfirstfilter.3
> /usr/share/man/man3/ldap_getnextfilter.3
> /usr/share/man/man3/ldap_setfilteraffixes.3
> /usr/share/man/man3/ldap_build_filter.3
> /usr/share/man/man3/ldap_get_values.3
> /usr/share/man/man3/ldap_get_values_len.3
> /usr/share/man/man3/ldap_value_free.3
> /usr/share/man/man3/ldap_value_free_len.3
> /usr/share/man/man3/ldap_count_values.3
> /usr/share/man/man3/ldap_count_values_len.3
> /usr/share/man/man3/ldap_modify.3
> /usr/share/man/man3/ldap_modify_s.3
> /usr/share/man/man3/ldap_modify_ext.3
> /usr/share/man/man3/ldap_modify_ext_s.3
> /usr/share/man/man3/ldap_mods_free.3
> /usr/share/man/man3/ldap_modrdn.3
> /usr/share/man/man3/ldap_modrdn_s.3
> /usr/share/man/man3/ldap_modrdn2.3
> /usr/share/man/man3/ldap_modrdn2_s.3
> /usr/share/man/man3/ldap_open.3
> /usr/share/man/man3/ldap_init.3
> /usr/share/man/man3/ldap_result.3
> /usr/share/man/man3/ldap_msgfree.3
> /usr/share/man/man3/ldap_msgtype.3
> /usr/share/man/man3/ldap_msgid.3
> /usr/share/man/man3/ldap_schema.3
> /usr/share/man/man3/ldap_str2syntax.3
> /usr/share/man/man3/ldap_syntax2str.3
> /usr/share/man/man3/ldap_syntax2name.3
> /usr/share/man/man3/ldap_syntax_free.3
> /usr/share/man/man3/ldap_str2matchingrule.3
> /usr/share/man/man3/ldap_matchingrule2str.3
> /usr/share/man/man3/ldap_matchingrule2name.3
> /usr/share/man/man3/ldap_matchingrule_free.3
> /usr/share/man/man3/ldap_str2attributetype.3
> /usr/share/man/man3/ldap_attributetype2str.3
> /usr/share/man/man3/ldap_attributetype2name.3
> /usr/share/man/man3/ldap_attributetype_free.3
> /usr/share/man/man3/ldap_str2objectclass.3
> /usr/share/man/man3/ldap_objectclass2str.3
> /usr/share/man/man3/ldap_objectclass2name.3

> /usr/share/man/man3/ber_get_bitstring.3
> /usr/share/man/man3/ber_first_element.3
> /usr/share/man/man3/ber_next_element.3
> /usr/share/man/man3/lber-encode.3
> /usr/share/man/man3/ber_alloc_t.3
> /usr/share/man/man3/ber_flush.3
> /usr/share/man/man3/ber_printf.3
> /usr/share/man/man3/ber_put_int.3
> /usr/share/man/man3/ber_put_ostring.3
> /usr/share/man/man3/ber_put_string.3
> /usr/share/man/man3/ber_put_null.3
> /usr/share/man/man3/ber_put_enum.3
> /usr/share/man/man3/ber_start_set.3
> /usr/share/man/man3/ber_put_seq.3
> /usr/share/man/man3/ber_put_set.3
> /usr/share/man/man3/lber-memory.3
> /usr/share/man/man3/lber-types.3
> /usr/share/man/man3/ldap.3
> /usr/share/man/man3/ldap_abandon.3
> /usr/share/man/man3/ldap_abandon_ext.3
> /usr/share/man/man3/ldap_add.3
> /usr/share/man/man3/ldap_add_s.3
> /usr/share/man/man3/ldap_add_ext.3
> /usr/share/man/man3/ldap_add_ext_s.3
> /usr/share/man/man3/ldap_bind.3
> /usr/share/man/man3/ldap_bind_s.3
> /usr/share/man/man3/ldap_simple_bind.3
> /usr/share/man/man3/ldap_simple_bind_s.3
> /usr/share/man/man3/ldap_sasl_bind.3
> /usr/share/man/man3/ldap_sasl_bind_s.3
> /usr/share/man/man3/ldap_kerberos_bind_s.3
> /usr/share/man/man3/ldap_kerberos_bind1.3
> /usr/share/man/man3/ldap_kerberos_bind1_s.3
> /usr/share/man/man3/ldap_kerberos_bind2.3
> /usr/share/man/man3/ldap_kerberos_bind2_s.3
> /usr/share/man/man3/ldap_unbind.3
> /usr/share/man/man3/ldap_unbind_ext.3
> /usr/share/man/man3/ldap_unbind_s.3
> /usr/share/man/man3/ldap_unbind_ext_s.3
> /usr/share/man/man3/ldap_set_rebind_proc.3
> /usr/share/man/man3/ldap_cache.3
> /usr/share/man/man3/ldap_enable_cache.3
> /usr/share/man/man3/ldap_disable_cache.3
> /usr/share/man/man3/ldap_destroy_cache.3
> /usr/share/man/man3/ldap_flush_cache.3
> /usr/share/man/man3/ldap_uncache_entry.3
> /usr/share/man/man3/ldap_uncache_request.3
> /usr/share/man/man3/ldap_set_cache_options.3
> /usr/share/man/man3/ldap_compare.3
> /usr/share/man/man3/ldap_compare_s.3
> /usr/share/man/man3/ldap_compare_ext.3
> /usr/share/man/man3/ldap_compare_ext_s.3
> /usr/share/man/man3/ldap_delete.3
> /usr/share/man/man3/ldap_delete_s.3
> /usr/share/man/man3/ldap_delete_ext.3
> /usr/share/man/man3/ldap_delete_ext_s.3
> /usr/share/man/man3/ldap_disptmpl.3
> /usr/share/man/man3/ldap_objectclass_free.3
> /usr/share/man/man3/ldap_scherr2str.3
> /usr/share/man/man3/ldap_search.3
> /usr/share/man/man3/ldap_search_s.3
> /usr/share/man/man3/ldap_search_st.3
> /usr/share/man/man3/ldap_search_ext.3
> /usr/share/man/man3/ldap_search_ext_s.3
> /usr/share/man/man3/ldap_searchprefs.3
> /usr/share/man/man3/ldap_init_searchprefs.3
> /usr/share/man/man3/ldap_init_searchprefs_buf.3
> /usr/share/man/man3/ldap_free_searchprefs.3
> /usr/share/man/man3/ldap_first_searchobj.3
> /usr/share/man/man3/ldap_next_searchobj.3
> /usr/share/man/man3/ldap_sort.3
> /usr/share/man/man3/ldap_sort_entries.3
> /usr/share/man/man3/ldap_sort_values.3
> /usr/share/man/man3/ldap_sort_strcasecmp.3
> /usr/share/man/man3/ldap_ufn.3
> /usr/share/man/man3/ldap_ufn_search_s.3
> /usr/share/man/man3/ldap_ufn_search_c.3
> /usr/share/man/man3/ldap_ufn_search_ct.3
> /usr/share/man/man3/ldap_ufn_setprefix.3
> /usr/share/man/man3/ldap_ufn_setfilter.3
> /usr/share/man/man3/ldap_ufn_timeout.3
> /usr/share/man/man3/ldap_url.3
> /usr/share/man/man3/ldap_is_ldap_url.3
> /usr/share/man/man3/ldap_url_parse.3
> /usr/share/man/man3/ldap_free_urldesc.3
> /usr/share/man/man3/ldap_url_search.3
> /usr/share/man/man3/ldap_url_search_s.3
> /usr/share/man/man3/ldap_url_search_st.3
> /usr/share/man/man5/ldap.conf.5
> /usr/share/man/man5/ldapfilter.conf.5
> /usr/share/man/man5/ldapfriendly.5
> /usr/share/man/man5/ldapsearchprefs.conf.5
> /usr/share/man/man5/ldaptemplates.conf.5
> /usr/share/man/man5/ldif.5
> /usr/share/man/man5/slapd.conf.5
> /usr/share/man/man5/slapd.replog.5
> /usr/share/man/man5/ud.conf.5
> /usr/share/man/man8/go500.8
> /usr/share/man/man8/go500gw.8
> /usr/share/man/man8/in.xfingerd.8
> /usr/share/man/man8/mail500.8
> /usr/share/man/man8/fax500.8
> /usr/share/man/man8/rcpt500.8
> /usr/share/man/man8/slapadd.8
> /usr/share/man/man8/slapcat.8
> /usr/share/man/man8/slapd.8
> /usr/share/man/man8/slapindex.8
> /usr/share/man/man8/slappasswd.8
> /usr/share/man/man8/slurpd.8
> /usr/share/openldap
> /usr/share/openldap/ldapfriendly
> /usr/share/openldap/go500gw.help
> /usr/share/openldap/rcpt500.help
> /var/lib/ldap

Part XI Gateway Server Related Reference

In this Part

Other Server - `Squid` Proxy Server
Other Server - `FreeS/WAN` VPN Server

This part of the book will exclusively deal with two programs that are less know or used than all the other that we can see in the UNIX world. In general this happen because there are used for specific needs and often by companies.

Usually, end users don't need to install them but this will surely change in the future with the increase of attacks on the Internet. Therefore here is a step-by-step guide on how to configure, secure, optimize and install them.

26 Gateway Server - Squid Proxy Server

In this Chapter

Linux Squid Proxy Server

Abstract

Proxy-servers, with their capability to save bandwidth, improve security, and increase web-surfing speed are becoming more popular than ever. At this time only a few proxy-server programs are on the market. These proxy-servers have two main drawbacks: they are commercial, and they don't support ICP (ICP is used to exchange hints about the existence of URLs in neighbour caches). Squid is the best choice for a proxy-cache server since it is robust, free, and can use ICP features.

Derived from the "cached" software from the ARPA-funded Harvest research project, developed at the National Laboratory for Applied Network Research and funded by the National Science Foundation, Squid offers high-performance caching of web clients, and also supports FTP, Gopher, HTTP and HTTPS data objects.

It stores hot objects in RAM, maintains a robust database of objects on disk, has a complex access control mechanism, and supports the SSL protocol for proxying secure connections. In addition, it can be hierarchically linked to other Squid-based proxy servers for streamlined caching of pages.

In our compilation and configuration we'll show you how to configure Squid depending of your needs. Two different set-ups are available.

The first will be to configure it to run as an **httpd-accelerator** to get more performance out of our Web Server. In accelerator mode, the Squid server acts as a reverse proxy cache: it accepts client requests, serves them out of cache, if possible, or requests them from the origin server for which it is the reverse proxy.

The second will be to configure Squid as a **proxy-caching** server to be able to let all users in your corporate network use Squid to access the Internet. This is a very interesting addition when you run a Gateway Server in your corporate. A Gateway Server as described in this book plus a Squid server mounted on it, will improve the security and performance speed of this system. This is also the solution to control and restrict what can be viewed on the Internet.

With a Squid Server configured as a proxy-caching server on a Gateway Server, you will be able to block for example porno sites, underground sites, warez (if you want ☺), etc. many possibilities exist like authorizing access to the Internet based on specific hours or days.

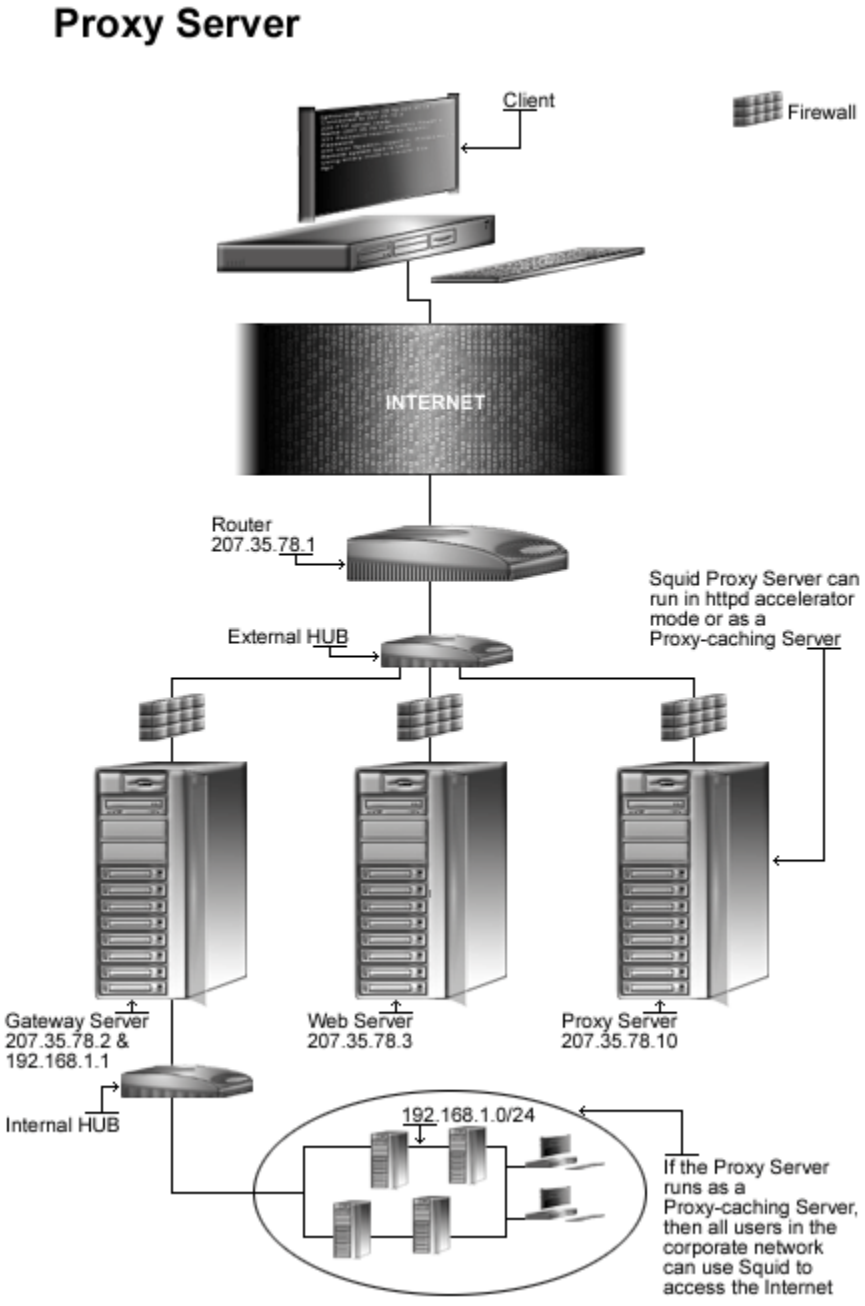
Recommended RPM packages to be installed for a `Proxy` Server

A minimal configuration provides the basic set of packages required by the Linux operating system. Minimal configuration is a perfect starting point for building secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a Proxy (`SQUID`) server running on `Squid` software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install `Squid` by RPM package. Therefore, `squid` RPM package is already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

basesystem	e2fsprogs	**iptables**	openssh-server	slocate
bash	ed	kernel	openssl	**squid**
bdflush	file	less	pam	sysklogd
bind	filesystem	libstdc++	passwd	syslinux
bzip2	fileutils	libtermcap	popt	SysVinit
chkconfig	findutils	lilo	procps	tar
console-tools	gawk	logrotate	psmisc	termcap
cpio	gdbm	losetup	pwdb	textutils
cracklib	gettext	MAKEDEV	qmail	tmpwatch
cracklib-dicts	glib	man	readline	utempter
crontabs	glibc	mingetty	rootfiles	util-linux
db1	glibc-common	mktemp	rpm	vim-common
db2	grep	mount	sed	vim-minimal
db3	groff	ncurses	setup	vixie-cron
dev	gzip	net-tools	sh-utils	words
devfsd	info	newt	shadow-utils	which
diffutils	initscripts	openssh	slang	zlib

Tested and fully functional on OpenNA.com.


Proxy Server
Client
Firewall
INTERNET
Router
207.35.78.1
Squid Proxy Server can run in httpd accelerator mode or as a Proxy-caching Server
External HUB
Gateway Server
207.35.78.2 &
192.168.1.1
Web Server
207.35.78.3
Proxy Server
207.35.78.10
Internal HUB
192.168.1.0/24
If the Proxy Server runs as a Proxy-caching Server, then all users in the corporate network can use Squid to access the Internet

These installation instructions assume

Commands are Unix-compatible.
The source path `is /var/tmp` (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest `Squid` version number is `2.4.STABLE1`

Packages

The following are based on information as listed by `Squid` as of 2001/03/20. Please regularly check at www.squid-cache.org for the latest status.

Source code is available from:
`Squid` Homepage: http://www.squid-cache.org/
`Squid` FTP Site: `206.168.0.9`
You must be sure to download: `squid-2.4.STABLE1-src.tar.gz`

Pristine source

If you don’t use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `Squid`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > Squid1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > Squid2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff Squid1 Squid2 > Squid-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing `Squid`

required steps Below are the required steps that you must make to configure, compile and optimize the `Squid` server software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the `/var/tmp` directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
  ```
  [root@deep /]# cp squid-version-src.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf squid-version-src.tar.gz
  ```

Step 2
To avoid security risks, we must create a new user account called "squid" to be the owner of the Squid database cache files and daemon.

- To create this special Squid user account, use the following command:
  ```
  [root@deep tmp]# useradd -r -d /var/lib/squid -s /bin/false -c "Squid
  Server" -u 23 squid >/dev/null 2>&1 || :
  ```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID.

Step 3
After that, move into the newly created Squid source directory and perform the following steps to configure and optimize the software for your system.

- To move into the newly created Squid source directory use the command:
  ```
  [root@deep tmp]# cd squid-2.4.STABLE1/
  ```

Step 4
There are some source files to modify before going in configuration and compilation of the program; the changes allow us to fix some problems and to configure the program for our PATH environment variable under Linux.

- Edit the **Makefile.in** file (`vi +18 icons/Makefile.in`) and change the line:

  ```
  DEFAULT_ICON_DIR  =  $(sysconfdir)/icons
  ```

  To read:

  ```
  DEFAULT_ICON_DIR  =  $(libexecdir)/icons
  ```

We change the variable (sysconfdir) to become (libexecdir). With this modification, the /icons directory of Squid will be located under the /usr/lib/squid directory.

- Edit the **Makefile.in** file (`vi +39 src/Makefile.in`) and change the lines:

  ```
  DEFAULT_CACHE_LOG  =  $(localstatedir)/logs/cache.log
  ```

  To read:

  ```
  DEFAULT_CACHE_LOG  =  $(localstatedir)/log/squid/cache.log


  DEFAULT_ACCESS_LOG  =  $(localstatedir)/logs/access.log
  ```

  To read:

  ```
  DEFAULT_ACCESS_LOG  =  $(localstatedir)/log/squid/access.log
  ```

```
DEFAULT_STORE_LOG  =  $(localstatedir)/logs/store.log
```

To read:

```
DEFAULT_STORE_LOG  =  $(localstatedir)/log/squid/store.log
```

```
DEFAULT_PID_FILE  =  $(localstatedir)/logs/squid.pid
```

To read:

```
DEFAULT_PID_FILE  =  $(localstatedir)/run/squid.pid
```

```
DEFAULT_SWAP_DIR  =  $(localstatedir)/cache
```

To read:

```
DEFAULT_SWAP_DIR  =  $(localstatedir)/lib/squid
```

```
DEFAULT_ICON_DIR  =  $(sysconfdir)/icons
```

To read:

```
DEFAULT_ICON_DIR  =  $(libexecdir)/icons
```

We change the default location of "`cache.log`", "`access.log`", and "`store.log`" files to be located under `/var/log/squid` directory. Then, we put the pid file of `Squid` under `/var/run` directory, and finally, locate the `/icons` directory of `Squid` under `/usr/lib/squid/icons` with the variable (`libexecdir`) above.

One important note here is the location of the cache directory of `Squid`. As we can see, we relocate it under `/var/lib/squid` directory since this directory `(/var/lib)` is on its own partition. This allows us to isolate this file system from the rest of our operating system and to eliminate possible buffer overflow attack. Also having the directory where `Squid` cache will reside on its own partition will allow us to improve performance by tuning parameters of this separate partition with commands like `ulimit`, etc.

Using `GNU malloc` library to improve cache performance of `Squid`

If you're suffering from memory limitations on your system, the cache performance of `Squid` will be affected. To reduce this problem, you can link `Squid` with an external `malloc` library such as `GNU malloc`. This library must be installed before compiling `Squid` on the server. To make `Squid` use `GNU malloc` as an external library follows these steps:

Packages

`GNU malloc` Homepage: http://www.gnu.org/order/ftp.html
You must be sure to download: `malloc.tar.gz`

```
[root@deep /]# cp malloc.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf malloc.tar.gz
```

Step 1
Compile and install GNU malloc on your system by executing the following commands:

```
[root@deep tmp]# cd malloc
[root@deep malloc]# export CC=gcc
[root@deep malloc]# export CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-
loops -fomit-frame-pointer"
[root@deep malloc]# make
```

Step 2
Copy the "libmalloc.a" file to your system library directory and be sure to name it "libgnumalloc.a".

```
[root@deep malloc]# cp libmalloc.a /usr/lib/libgnumalloc.a
```

Step 3
Copy the "malloc.h" file to your system include directory and be sure to name it "gnumalloc.h".

```
[root@deep malloc]# cp malloc.h /usr/include/gnumalloc.h
```

With the files "libgnumalloc.a" and "gnumalloc.h" installed to the appropriate location on your system, Squid will be able detect them automatically during its compile time, and will use them to improve its cache performance.

Step 4
Once the required modifications have been made into the related source files of Squid as explained previously and the GNU malloc library has been installed on the system to the appropriate location, it is time configure and optimize Squid for our system.

- To configure and optimize Squid use the following compilation lines:

```
CC="gcc" \
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
./configure \
--prefix=/usr \
--exec-prefix=/usr \
--bindir=/usr/sbin \
--libexecdir=/usr/lib/squid \
--localstatedir=/var \
--sysconfdir=/etc/squid \
--enable-delay-pools \
--enable-cache-digests \
--enable-poll \
--disable-ident-lookups \
--enable-truncate \
--enable-removal-policies="heap" \
--enable-auth-modules="PAM"
--enable-xmalloc-statistics \
--enable-cachemgr-hostname=www \
--enable-linux-netfilter \
--enable-stacktraces
```

This tells Squid to set itself up for this particular configuration setup with:

- Use the delay pools feature of Squid to limit and control bandwidth usage for users.
- Use Cache Digests to improve client response time and network utilization.
- Enable poll() instead of select() since it's preferred over select.
- Disable ident-lookups to remove code that performs Ident (RFC 931) lookups and reduce possible denial-of-service.
- Enable truncate to clean some performance improvements when removing cached files.
- Use the heap-replacement feature of Squid to have the choice of various cache replacement algorithms, instead of the standard LRU algorithm for better performance.
- Enable PAM proxy authentication backend modules.
- Show malloc statistics in status page.
- Make cachemgr.cgi default to this host. If you run Squid as a httpd-accelerator, you can omit this option, but if you run Squid as proxy-caching, you must keep it and specify the hostname of your Web Server (usually www) since a gateway/proxy server doesn't have to run a Web Server on the machine.
- Enable transparent proxy support for Linux kernel 2.4.
- Enable automatic call backtrace on fatal errors.

NOTE: Pay special attention to the compile CFLAGS line above. We optimize Squid for an i686 CPU architecture with the parameter "-march=i686 and -mcpu=i686". Please don't forget to adjust this CFLAGS line to reflect your own system and architecture.

Step 5
Now, we must make a list of all existing files on the system before installing the software, and one afterwards, then compare them using the **diff** utility tool of Linux to find out what files are placed where and finally install Squid Proxy Server:

```
[root@deep squid-2.4.STABLE1]# make
[root@deep squid-2.4.STABLE1]# cd
[root@deep /root]# find /* > Squid1
[root@deep /root]# cd /var/tmp/squid-2.4.STABLE1/
[root@deep squid-2.4.STABLE1]# make install
[root@deep squid-2.4.STABLE1]# mkdir -p /var/lib/squid
[root@deep squid-2.4.STABLE1]# mkdir -p /var/log/squid
[root@deep squid-2.4.STABLE1]# chown squid.squid /var/lib/squid/
[root@deep squid-2.4.STABLE1]# chown squid.squid /var/log/squid/
[root@deep squid-2.4.STABLE1]# chmod 750 /var/lib/squid/
[root@deep squid-2.4.STABLE1]# chmod 750 /var/log/squid/
[root@deep squid-2.4.STABLE1]# rm -rf /var/logs/
[root@deep squid-2.4.STABLE1]# rm -f /usr/sbin/RunCache
[root@deep squid-2.4.STABLE1]# rm -f /usr/sbin/RunAccel
[root@deep squid-2.4.STABLE1]# strip /usr/sbin/squid
[root@deep squid-2.4.STABLE1]# strip /usr/sbin/client
[root@deep squid-2.4.STABLE1]# strip /usr/lib/squid/*
[root@deep squid-2.4.STABLE1]# /sbin/ldconfig
[root@deep squid-2.4.STABLE1]# cd
[root@deep /root]# find /* > Squid2
[root@deep /root]# diff Squid1 Squid2 > Squid-Installed
```

The **make** command will compile all source files into executable binaries, and **make install** will install the binaries and any supporting files into the appropriate locations. The **mkdir** command will create two new directories named "squid" under /var/lib and /var/log.

The **rm** command will remove the /var/logs directory since this directory has been created to handle the log files related to Squid that we have relocated during compile time into /var/log/squid.

The **chown** will change the owner of /var/lib/squid and /var/log/squid to be the user squid, and the **chmod** command will make the mode of both squid directories (0750/drwxr-x---) for security reasons.

Take note that we remove the small scripts named "RunCache" and "RunAccel" which start Squid in either caching mode or accelerator mode, since we use a better script named "squid" located under /etc/rc.d/init.d directory that takes advantage of Linux system V. The **strip** command will reduce the size of binaries for optimum performance.

Step 6
Once configuration, optimization, compilation, and installation of the Proxy Server software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Squid and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf squid-version/
[root@deep tmp]# rm -rf malloc/
[root@deep tmp]# rm -f squid-version-src.tar.gz
[root@deep tmp]# rm -f malloc.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install Squid and GNU malloc. It will also remove the Squid and GNU malloc compressed archive from the /var/tmp directory.

Configuring Squid

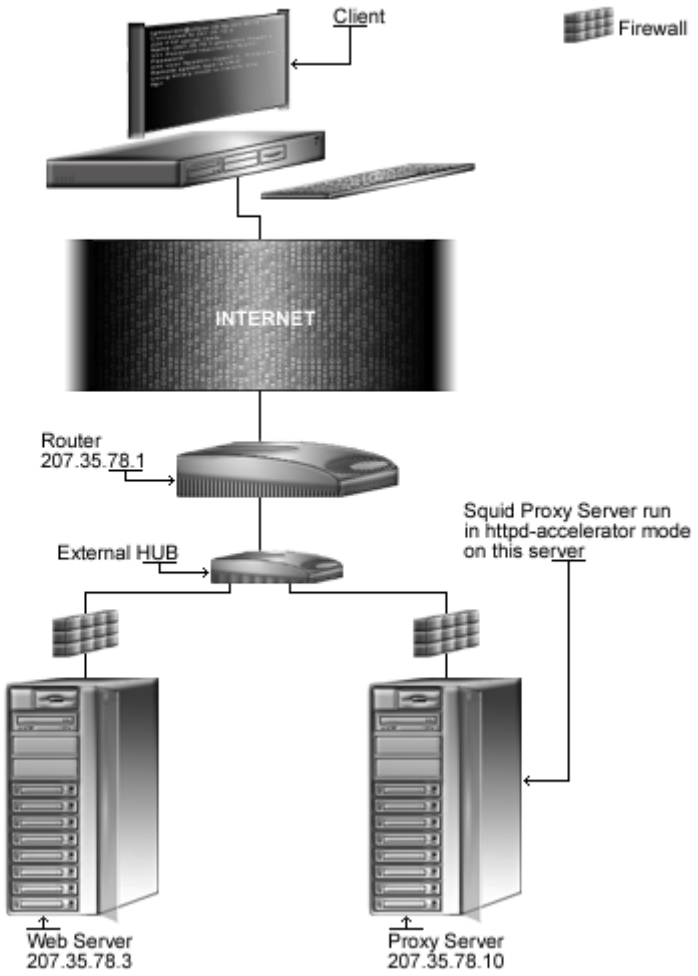
After Squid has been built and installed successfully in your system, your next step is to configure and customize all the required parameters in the different Squid configuration files as prudently as possible:

- ✓ /etc/squid/squid.conf (The Squid Configuration File)
- ✓ /etc/sysconfig/squid (The Squid System Configuration File)
- ✓ /etc/logrotate.d/squid (The Squid Log Rotation File)
- ✓ /etc/rc.d/init.d/squid (The Squid Initialization File)

Running Squid in a httpd-accelerator mode

The squid.conf file is used to set and configure all the different options for your Squid proxy server. In the configuration file below, we'll configure the /etc/squid/squid.conf file to be in httpd-accelerator mode. In this acceleration mode, if the Web Server runs on the same server where Squid is installed, you must set its daemon to run on port 81. With the Apache Web Server, you can do it by assign the line (Port 80) to (Port 81) in its httpd.conf file. If the Web Server runs on other servers in your network like we do, you can keep the same port number (80) for Apache, since Squid will bind on a different IP number where port (80) is not already in use.

Proxy httpd accelerator mode Server



/etc/squid/squid.conf: The Squid Configuration File

The /etc/squid/squid.conf file is the main configuration file for squid. Though there are hundred of option tags in this file, you should only need to change some options to get Squid up and running. The other options give you amazing flexibility, but you can learn about them once you have Squid running. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add/change the following options. Below is what we recommend you:

```
http_port 80
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 42 MB
redirect_rewrites_host_header off
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir ufs /var/lib/squid 200 16 256
emulate_httpd_log on
acl all src 0.0.0.0/0.0.0.0
http_access allow all
cache_mgr root
cache_effective_user squid
cache_effective_group squid
httpd_accel_host 207.35.78.3
httpd_accel_port 80
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

This tells squid.conf file to set itself up for this particular configuration with:

http_port 80
The option "http_port" specifies the port number where Squid will listen for HTTP client requests. If you set this option to port 80, the client will have the illusion of being connected to the Apache Web Server. Since we are running Squid in accelerator mode, we must listen on port 80.

icp_port 0
The option "icp_port" specifies the port number where Squid will sends and receive ICP requests from neighboring caches. We must set the value of this option to "0" to disable it, since we are configuring Squid to be in accelerator mode for the Web Server. The ICP feature is needed only in a multi-level cache environment with multiple siblings and parent caches. Using ICP in an accelerator mode configuration would add unwanted overhead to Squid. This is an optimization feature.

hierarchy_stoplist cgi-bin ?
The options "hierarchy_stoplist cgi-bin ?" is used to not query neighbour cache for certain objects. The above line is recommended.

```
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
```

The options "`acl QUERY urlpath_regex cgi-bin \?`" and "`no_cache deny QUERY`" are used to force certain objects to never be cached, like files under "`cgi-bin`" directory. This is a security feature.

```
cache_mem 42 MB
```

The option "`cache_mem`" specifies the amount of memory (RAM) to be used for caching the so called: In-Transit objects, Hot Objects, Negative-Cached objects. It's important to note that `Squid` can use much more memory than the value you specify in this parameter. For example, if you have 256 MB free for `Squid`, you must put 256/3 = 85 MB here. This is an optimization feature.

```
redirect_rewrites_host_header off
```

The option "`redirect_rewrites_host_header`", if set to "`off`", tells `Squid` to not rewrites any Host: header in redirected requests. It's recommended to set this option to "`off`" if you are running `Squid` in accelerator mode.

```
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
```

The options "cache_`replacement_policy`" and "`memory_replacement_policy heap GDSF`" specify the cache policy `Squid` will use to determine which objects in the cache must be replaced when the proxy need to make disk space and which objects are purged from memory when memory space is needed. In our configuration, we choose the `GDSF` (Greedy-Dual Size Frequency) policy as our default policy. See http://www.hpl.hp.com/techreports/1999/HPL-1999-69.html and http://fog.hpl.external.hp.com/techreports/98/HPL-98-173.html for more information.

```
cache_dir ufs /var/lib/squid 200 16 256
```

The option "`cache_dir`" specifies in order: which kind of storage system to use (`ufs`), the name of the cache directory (`/var/lib/squid`) for `Squid`, the disk space in megabytes to use under this directory (`200 Mbytes`), the number of first-level subdirectories to be created under the cache directory (`16 Level-1`), and the number of second-level subdirectories to be created under each first-level cache directory (`256 Level-2`). In accelerator mode, this option is directly related to the size and number of files that you want to serve with your `Apache` Web Server.

```
emulate_httpd_log on
```

The option "`emulate_httpd_log`" if set to "`on`" specifies that `Squid` should emulate the log file format of the `Apache` Web Server. This is very useful if you want to use a third party program like `Webalizer` to analyze and produce static report on the Web Server (`httpd`) log file.

```
acl all src 0.0.0.0/0.0.0.0
http_access allow all
```

The options "`acl`" and "`http_access`" specify and define an access control list to be applied on the `Squid` Proxy Server. Our "`acl`" and "`http_access`" options are not restricted, and allow every one to connect on the proxy server since we use this proxy to accelerate the public `Apache` Web Server. See your `Squid` documentation for more information when using `Squid` in non-accelerator mode.

```
cache_mgr root
```

The option "`cache_mgr`" specify the email-address of the administrator responsible for the `Squid` Proxy Server. This person is the one who will receive mail if `Squid` encounter problems. You can specify the name or the complete email address in this option.

```
cache_effective_user squid
cache_effective_group squid
```
The options "cache_effective_user" and "cache_effective_group" specify the UID/GID that the cache will run on. Don't forget to never run Squid as "root". In our configuration we use the UID "squid" and the GID "squid". This is a security feature.

```
httpd_accel_host 207.35.78.3
httpd_accel_port 80
```
The options "httpd_accel_host" and "httpd_accel_port" specify to Squid the IP address and port number where the real HTTP Server (i.e. Apache) reside. In our configuration, the real HTTP Web Server is on IP address 207.35.78.3 (www.openna.com) and on port (80). "www.openna.com" is another FDQN on our network, and since the Squid Proxy Server doesn't reside on the same host of Apache HTTP Web Server, we can use port (80) for our Squid Proxy Server, and port (80) for our Apache Web Server, and the illusion is perfect.

```
logfile_rotate 0
```
The option "logfile_rotate" specifies the number of logfile rotations that we want the Squid program to make. Setting the value to 0 will disable the default rotation and will let us control this feature through our personal logrotate script file on Linux. This is what we need to do on Linux and use our own log script file to make the appropriate rotation of Squid log files.

```
log_icp_queries off
```
The option "log_icp_queries" specifies if you want ICP queries (ICP is used to exchange hints about the existence of URLs in neighbour caches) to be logged to the "access.log" file or not. Since we don't use the ICP feature in Squid accelerator mode, we can safely set this option to "off".

```
cachemgr_passwd my-secret-pass all
```
The option "cachemgr_passwd" specifies a password that will be required for accessing the operations of the "cachemgr.cgi" program utility. This CGI utility program is designed to run through a web interface and outputs statistics about the Squid configuration and performance. The <my-secret-pass> is the password that you have chosen, and the keyword <all> specifies to set this password to be the same for all actions you can perform with this program. See "The cachemgr.cgi program utility of Squid", below in this chapter for more information.

```
buffered_logs on
```
The option "buffered_logs", if turned "on", can speed up the writing of some log files slightly. This is an optimization feature.

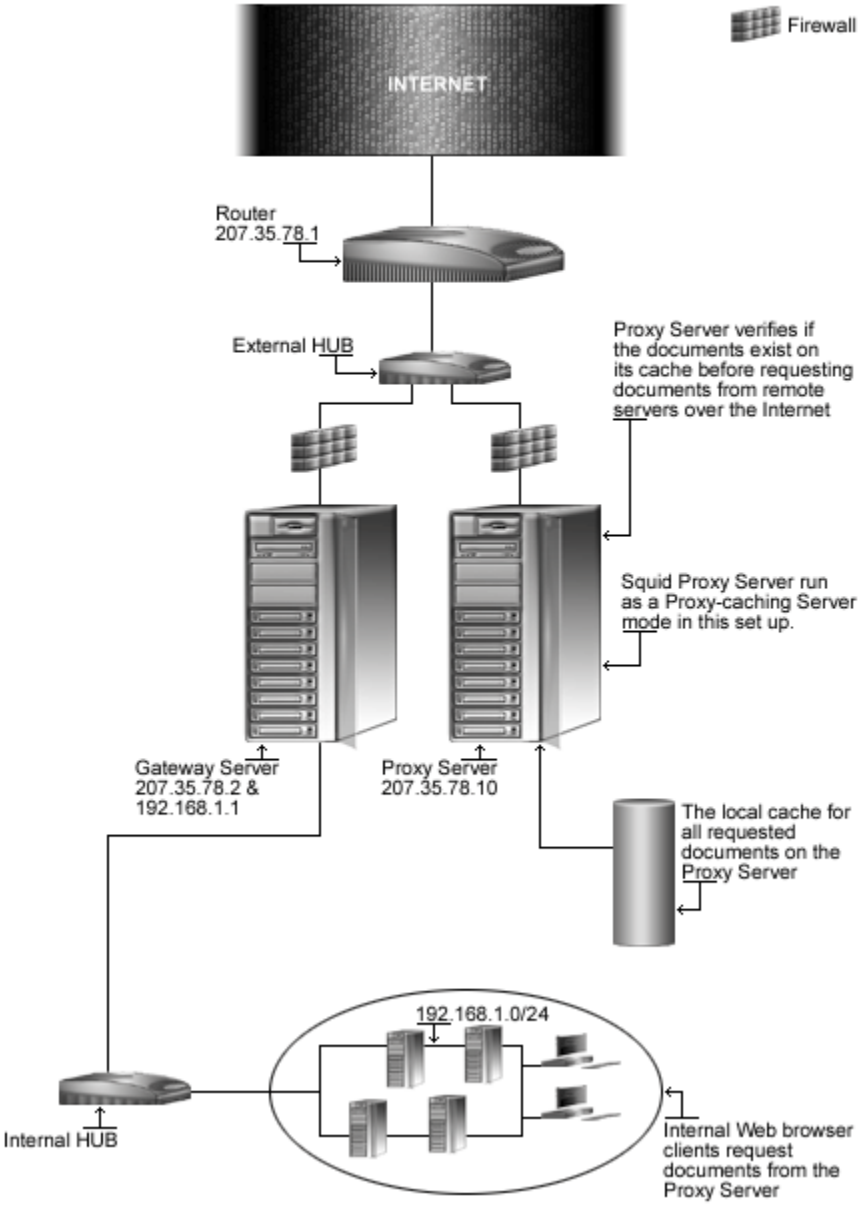
Running Squid in a proxy-caching mode

With some minor modification to the squid.conf file we have defined above to run in httpd-accelerator mode, we can run Squid as a proxy-caching server. With a proxy-caching server, all users in your corporate network will use Squid to access the Internet.

With this configuration, you can have complete control, and apply special policies on what can be viewed, accessed, and downloaded. You can also control bandwidth usage, connection time, and so on. A proxy cache server can be configured to run as stand-alone server for your corporation, or to use and share caches hierarchically with other proxy servers around the Internet.

With the first example below we show you how to configure Squid as a stand-alone server, and then speak a little bit about a cache hierarchy configuration, where two or more proxy-cache servers cooperate by serving documents to each other.

Proxy-caching Server



/etc/squid/squid.conf: The Squid Configuration File

To set up Squid as a proxy-cahing server, we use the same configuration file as above but with some addition and modification to the default related to Squid in httpd-accelerator mode. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs. The rest of the parameters are the same as for Squid in httpd-accelerator mode and I recommend you to read the configuration section related to Squid in accelerator mode for more information on each options.

- Edit the **squid.conf** file (vi /etc/squid/squid.conf) and add/change the following options for proxy cache that run as a stand-alone server. Below is what we recommend you:

```
icp_port 0
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY
cache_mem 42 MB
cache_replacement_policy heap GDSF
memory_replacement_policy heap GDSF
cache_dir ufs /var/lib/squid 200 16 256
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
cache_mgr root
cache_effective_user squid
cache_effective_group squid
logfile_rotate 0
log_icp_queries off
cachemgr_passwd my-secret-pass all
buffered_logs on
```

NOTE: In the above configuration example, the default Proxy port 3128 will be used. If you prefer to use another port like 8080, all you will have to do will be to add the parameter "http_port 8080" and configure your clients accordable.

The big difference with the httpd-accellerator mode configuration is the use of access control lists (ACL). This feature allows you to restrict access based on source IP address (src), destination IP address (dst), source domain, destination domain, time, and so on. Many types exist with this feature, and you should consult the "squid.conf" file for a complete list.

The four most used types are as follows:

```
acl  name      type       data
 |     |         |          |
acl some-name src a.b.c.d/e.f.g.h       # ACL restrict access based on source IP address
acl some-name dst a.b.c.d/e.f.g.h       # ACL restrict access based on destination IP address
acl some-name srcdomain foo.com         # ACL restrict access based on source domain
acl some-name dstdomain foo.com         # ACL restrict access based on destination domain
```

As an example, to restrict access to your `Squid` proxy server to only your internal clients, and to a specific range of designated ports, something like the following will make the job:

```
acl localnet src 192.168.1.0/255.255.255.0
acl localhost src 127.0.0.1/255.255.255.255
acl Safe_ports port 80 443 210 70 21 1025-65535
acl CONNECT method CONNECT
acl all src 0.0.0.0/0.0.0.0
http_access allow localnet
http_access allow localhost
http_access deny !Safe_ports
http_access deny CONNECT
http_access deny all
```

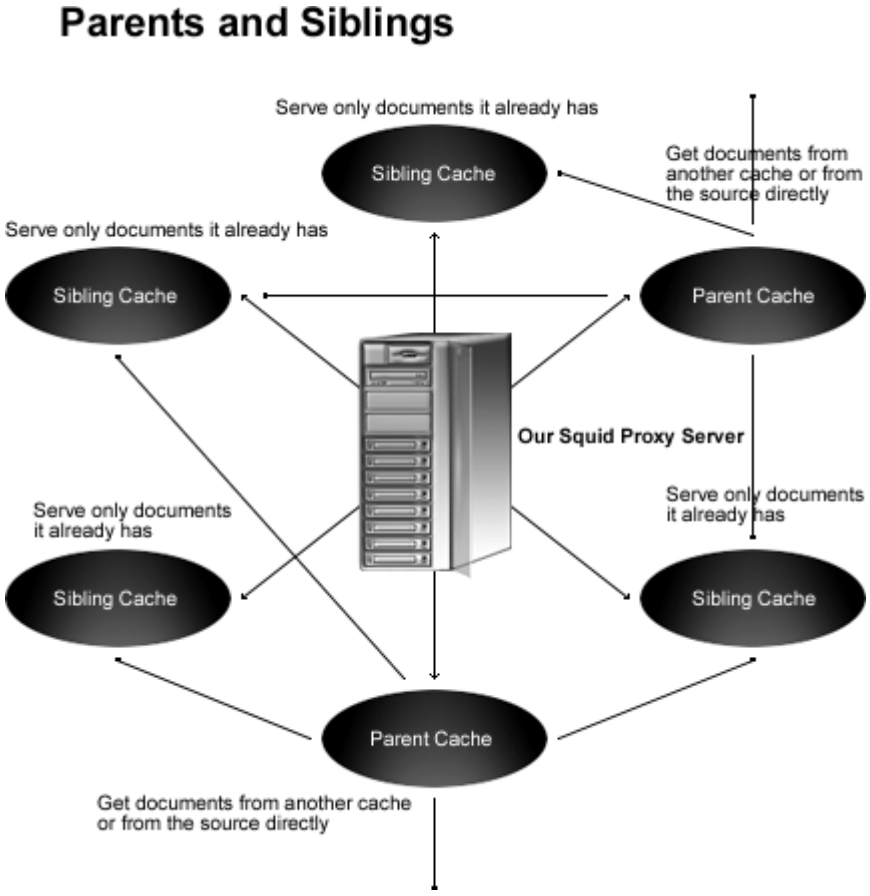
This `acl` configuration will allow all internal clients from the private class `C 192.168.1.0` to access the proxy server; it's also recommended that you allow the localhost IP (a special IP address used by your own server) to access the proxy.

After we choose a range of ports (80=http, 443=https, 210=wais, 70=gopher, and 21=ftp) which our internal clients can use to access the Internet, we deny the `CONNECT` method to prevent outside people from trying to connect to the proxy server, and finally, we deny all source IP address and ports on the proxy server.

Multi-level Web Caching

The second method of proxy cache is the so-called "Multi-level Web Caching" where you choose to share and cooperate with more proxy-cache servers on the Internet. With this method, your organization uses the cache of many others proxy cache servers, and to compensate, the other cache server can use yours.

It's important to note that in this situation, the proxy cache can play two different roles in the hierarchy. It can be configured to be a **sibling** cache, and be able to only serve documents it already has, or it can be configured as a **parent** cache, and be able to get documents from another cache or from the source directly.



NOTE: A good strategy to avoid generating more network traffic than without web caching is to choose to have several **sibling** caches and only a small number of **parent** caches.

`/etc/sysconfig/squid`: The `Squid` System Configuration File

The `/etc/sysconfig/squid` file is used to specify `Squid` system configuration information, such as if `Squid` should enable initial `DNS` checks at start-up, and the value of time to wait for `Squid` to shut down when asked.

- Create the **`squid`** file (`touch /etc/sysconfig/squid`) and add the following lines:

```
# default squid options
# -D disables initial dns checks. If you most likely will not to have an
#    internet connection when you start squid, uncomment this
#SQUID_OPTS="-D"

# Time to wait for Squid to shut down when asked. Should not be necessary
# most of the time.
SQUID_SHUTDOWN_TIMEOUT=100
```

/etc/logrotate.d/squid: The Squid Log Rotation Configuration File

The /etc/logrotate.d/squid file is responsible to rotate log files related to Squid software automatically each week via syslog. If you are not familiar with syslog, look at the syslog.conf (5) manual page for a description of the syslog configuration file, or the syslogd (8) manual page for a description of the syslogd daemon.

- Create the **squid** file (touch /etc/logrotate.d/squid) and add the following lines:

```
/var/log/squid/access.log {
    weekly
    rotate 5
    copytruncate
    compress
    notifempty
    missingok
}

/var/log/squid/cache.log {
    weekly
    rotate 5
    copytruncate
    compress
    notifempty
    missingok
}

/var/log/squid/store.log {
    weekly
    rotate 5
    copytruncate
    compress
    notifempty
    missingok
# This script asks squid to rotate its logs on its own.
# Restarting squid is a long process and it is not worth
# doing it just to rotate logs
    postrotate
      /usr/sbin/squid -k rotate
    endscript
}
```

/etc/rc.d/init.d/squid: The Squid Initialization File

The /etc/rc.d/init.d/squid script file is responsible to automatically start and stop the Squid Internet Object Cache on your server. Loading the squid daemon, as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Step 1
Create the **squid** script file (touch /etc/rc.d/init.d/squid) and add the following lines:

```
#!/bin/bash
# squid         This shell script takes care of starting and stopping
#               Squid Internet Object Cache
#
# chkconfig: - 90 25
# description: Squid - Internet Object Cache. Internet object caching is \
#       a way to store requested Internet objects (i.e., data available \
#       via the HTTP, FTP, and gopher protocols) on a system closer to the \
```

```
#       requesting site than to the source. Web browsers can then use the \
#       local Squid cache as a proxy HTTP server, reducing access time as \
#       well as bandwidth consumption.
# pidfile: /var/run/squid.pid
# config: /etc/squid/squid.conf

PATH=/usr/bin:/sbin:/bin:/usr/sbin
export PATH

# Source function library.
. /etc/rc.d/init.d/functions

# Source networking configuration.
. /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# check if the squid conf file is present
[ -f /etc/squid/squid.conf ] || exit 0

if [ -f /etc/sysconfig/squid ]; then
  . /etc/sysconfig/squid
else
  SQUID_OPTS="-D"
  SQUID_SHUTDOWN_TIMEOUT=100
fi

# determine the name of the squid binary
[ -f /usr/sbin/squid ] && SQUID=squid
[ -z "$SQUID" ] && exit 0

prog="$SQUID"

# determine which one is the cache_swap directory
CACHE_SWAP=`sed -e 's/#.*//g' /etc/squid/squid.conf | \
        grep cache_dir |  awk '{ print $3 }'`
[ -z "$CACHE_SWAP" ] && CACHE_SWAP=/var/lib/squid

RETVAL=0

start() {
    for adir in $CACHE_SWAP; do
        if [ ! -d $adir/00 ]; then
             echo -n "init_cache_dir $adir... "
             $SQUID -z -F 2>/dev/null
        fi
    done
    echo -n $"Starting $prog: "
    $SQUID $SQUID_OPTS 2> /dev/null &
    RETVAL=$?
    [ $RETVAL -eq 0 ] && touch /var/lock/subsys/$SQUID
    [ $RETVAL -eq 0 ] && echo_success
    [ $RETVAL -ne 0 ] && echo_failure
    echo
    return $RETVAL
}

stop() {
    echo -n  $"Stopping $prog: "
    $SQUID -k check >/dev/null 2>&1
    RETVAL=$?
    if [ $RETVAL -eq 0 ] ; then
```

```
        $SQUID -k shutdown &
        rm -f /var/lock/subsys/$SQUID
        timeout=0
        while : ; do
                [ -f /var/run/squid.pid ] || break
                if [ $timeout -ge $SQUID_SHUTDOWN_TIMEOUT ]; then
                    echo
                    return 1
                fi
                sleep 2 && echo -n "."
                timeout=$((timeout+2))
        done
        echo_success
        echo
    else
        echo_failure
        echo
    fi
    return $RETVAL
}

reload() {
    $SQUID $SQUID_OPTS -k reconfigure
}

restart() {
    stop
    start
}

condrestart() {
    [ -e /var/lock/subsys/squid ] && restart || :
}

rhstatus() {
    status $SQUID
    $SQUID -k check
}

probe() {
    return 0
}

case "$1" in
start)
    start
    ;;

stop)
    stop
    ;;

reload)
    reload
    ;;

restart)
    restart
    ;;

condrestart)
    condrestart
    ;;
```

```
status)
    rhstatus
    ;;

probe)
    exit 0
    ;;

*)
    echo $"Usage: $0 {start|stop|status|reload|restart|condrestart}"
    exit 1
esac

exit $?
```

Step 2
Once the `squid` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/rc.d/init.d/squid
  [root@deep /]# chown 0.0 /etc/rc.d/init.d/squid
  ```

- To create the symbolic `rc.d` links for `Squid`, use the following commands:
  ```
  [root@deep /]# chkconfig --add squid
  [root@deep /]# chkconfig --level 345 squid on
  ```

- To start `Squid` software manually, use the following command:
  ```
  [root@deep /]# /etc/rc.d/init.d/squid start
  Starting squid:          [OK]
  ```

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Securing Squid

This section deals especially with actions we can make to improve and tighten security under Squid. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

More control on mounting the cache directory of Squid

If you have created the cache directory of Squid in a separate partition in your Linux system (i.e. /var/lib), like we have done during the install set-up of Linux, then you can use the noexec, nodev, and nosuid features to improve and consolidate the cache security.

These features can be set up in the /etc/fstab file to inform the system to not allow execution of any binaries (noexec), to not interpret character or block special devices (nodev), and to not allow set-user-identifier or set-group-identifier bits to take effect (nosuid) on the mounted file system (/var/lib in our example).

Applying this procedure on the partition where the Squid Cache resides will help to eliminate the possibility of DEV, SUID/SGID, and execution of any binaries.

Step 1

- Edit the **fstab** file (vi /etc/fstab) and add in the line that refer to /var/lib file system the following options after the defaults option as show below:

```
LABEL=/var/lib    /var/lib    ext2    defaults,noexec,nodev,nosuid    1 2
```

Step 2

Once you have made the necessary adjustments to the /etc/fstab file, it is time to inform the Linux system about the modification.

- This can be accomplished with the following commands:

```
[root@deep /]# mount /var/lib -oremount
```

Each file system that has been modified must be remounted with the command show above. In our example we have modified the /var/lib file system and it is for this reason that we remount this file system with the above command.

Step 3

- You can verify if the modifications have been correctly applied to the Linux system with the following command:

```
[root@deep /]# cat /proc/mounts
/dev/root /   ext2        rw 0 0
/proc /proc   proc        rw 0 0
/dev/sda1     /boot       ext2    rw 0 0
/dev/sda9     /chroot     ext2    rw 0 0
/dev/sda8     /home       ext2    rw 0 0
/dev/sda13    /tmp        ext2    rw 0 0
/dev/sda7     /usr        ext2    rw 0 0
/dev/sda11    /var        ext2    rw 0 0
/dev/sda12    /var/lib    ext2    rw,noexec,nosuid,nodev 0 0
none  /dev/pts    devpts  rw 0 0
```

This command will show you all file system in your Linux server with parameters applied to them. If you see something like the following, congratulations!

```
/var/lib   /var/lib   ext2   rw,noexec,nosuid,noatime 0 0
```

Immunize the Squid configuration file

As we already know, the immutable bit can be used to prevent deletion, overwriting, or creation of a symbolic link to a file. Once your squid.conf file has been configured, it's a good idea to immunize it with the following command:

```
[root@deep /]# chattr +i /etc/squid/squid.conf
```

Optimizing Squid

This section deals especially with actions we can make to improve and tighten performance of Squid. Take a note that we refer to the features available within the base installed program.

The atime and noatime attributes

The atime and noatime attributes can be used to get a measurable performance gain in the Squid cache directory. See the chapter related to Linux kernel in this book for more information on this issue.

Physical memory

The most important resource for Squid is physical memory. Your processor does not need to be ultra-fast. Your disk system will be the major bottleneck, so fast disks are important for high-volume caches. Do not use IDE disks if you can help it.

The cachemgr.cgi program utility of Squid

The cachemgr.cgi utility program, which is available by default when you compile and install Squid into your system, is designed to run through a web interface, and outputs various statistics about Squid configuration and performance.

This program is located by default under the /usr/lib/squid directory, and you must put it in your "cgi-bin" directory (eg, /home/httpd/cgi-bin) on your Web server to be able to use it. Follow the simple steps below to use this program.

Step 1

Remember that during our configuration step, we have added to the Squid configuration time the option "--enable-cachemgr-hostname=www" to inform the program to run this script from the specified host name, which is in our case www (our Web Server hostname) on the network. This is an important point since there is no reason to run a Web Server on a Gateway/Proxy Server to be able to use this script if we already have a Web Server on our network to make this job.

The first step will be to move the "cachemgr" CGI file from the machine where Squid run to your Web Server under /home/httpd/cgi-bin directory by FTP transport, floppy, etc.

Step 2

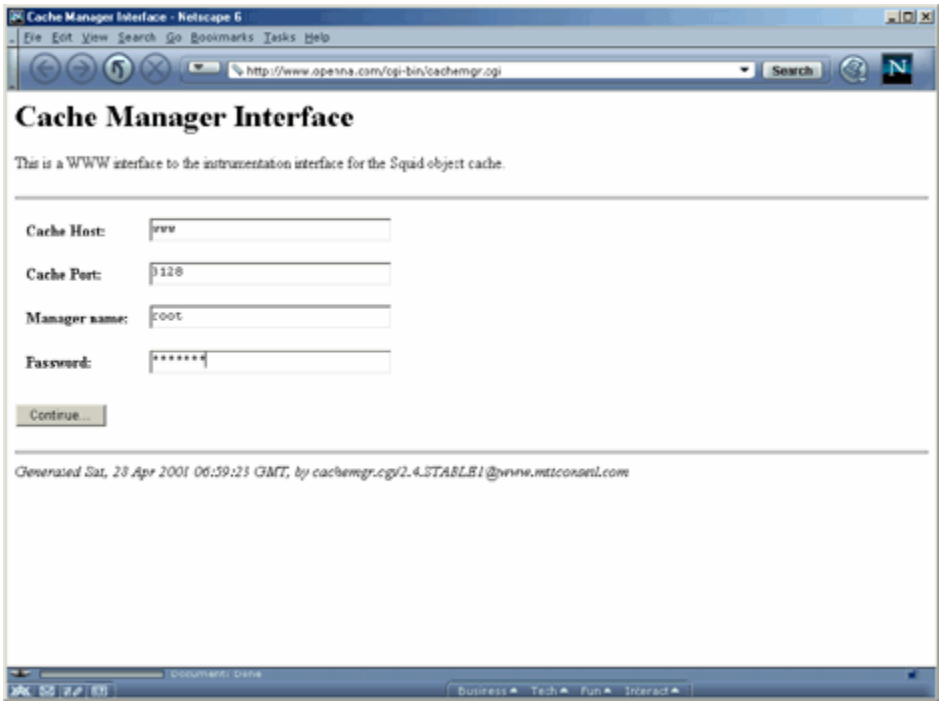
Once you've put the "cachemgr.cgi" program into your /cgi-bin directory on the remote Web Server, it is time to change its default mode permission and owner.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cd /home/httpd/cgi-bin/
[root@deep cgi-bin]# chown www.www cachemgr.cgi
[root@deep cgi-bin]# chmod 744 cachemgr.cgi
```

Step 3
Finally, you can point your web browser to the following address (http://my-web-server/cgi-bin/cachemgr.cgi) to be able to use the various features of this program.

The <my-web-server> is the address where your Apache web server lives, and <cachemgr.cgi> is the Squid utility program we have just placed in our "cgi-bin" directory to display information and the configuration of our Squid Proxy Linux server.



If you have configured the squid.conf file to use password authentication for cachemgr.cgi, you'll be asked to enter the Cache Host, Cache Port, Manager name, and Password information before you are able to access the cachemgr.cgi program. See the configuration of the /etc/squid/squid.conf file above for more information.

List of installed Squid files on your system

> /etc/init.d/squid
> /etc/logrotate.d/squid
> /etc/sysconfig/squid
> /etc/squid
> /etc/squid/mib.txt
> /etc/squid/squid.conf.default
> /etc/squid/squid.conf
> /etc/squid/mime.conf.default
> /etc/squid/mime.conf
> /etc/squid/errors
> /etc/squid/errors/ERR_ACCESS_DENIED
> /etc/squid/errors/ERR_CACHE_ACCESS_DENIED
> /etc/squid/errors/ERR_CACHE_MGR_ACCESS_DENIED
> /etc/squid/errors/ERR_CANNOT_FORWARD
> /etc/squid/errors/ERR_CONNECT_FAIL
> /etc/squid/errors/ERR_DNS_FAIL
> /etc/squid/errors/ERR_FORWARDING_DENIED
> /etc/squid/errors/ERR_FTP_DISABLED
> /etc/squid/errors/ERR_ZERO_SIZE_OBJECT
> /usr/lib/squid
> /usr/lib/squid/unlinkd
> /usr/lib/squid/cachemgr.cgi
> /usr/lib/squid/icons
> /usr/lib/squid/icons/anthony-binhex.gif
> /usr/lib/squid/icons/anthony-bomb.gif
> /usr/lib/squid/icons/anthony-box.gif
> /usr/lib/squid/icons/anthony-box2.gif
> /usr/lib/squid/icons/anthony-c.gif
> /usr/lib/squid/icons/anthony-compressed.gif
> /usr/lib/squid/icons/anthony-dir.gif
> /usr/lib/squid/icons/anthony-dirup.gif
> /usr/lib/squid/icons/anthony-dvi.gif
> /usr/lib/squid/icons/anthony-f.gif
> /usr/lib/squid/icons/anthony-image.gif
> /usr/lib/squid/icons/anthony-image2.gif
> /usr/lib/squid/icons/anthony-layout.gif

> /etc/squid/errors/ERR_FTP_FAILURE
> /etc/squid/errors/ERR_FTP_FORBIDDEN
> /etc/squid/errors/ERR_FTP_NOT_FOUND
> /etc/squid/errors/ERR_FTP_PUT_CREATED
> /etc/squid/errors/ERR_FTP_PUT_ERROR
> /etc/squid/errors/ERR_FTP_PUT_MODIFIED
> /etc/squid/errors/ERR_FTP_UNAVAILABLE
> /etc/squid/errors/ERR_INVALID_REQ
> /etc/squid/errors/ERR_INVALID_URL
> /etc/squid/errors/ERR_LIFETIME_EXP
> /etc/squid/errors/ERR_NO_RELAY
> /etc/squid/errors/ERR_ONLY_IF_CACHED_MISS
> /etc/squid/errors/ERR_READ_ERROR
> /etc/squid/errors/ERR_READ_TIMEOUT
> /etc/squid/errors/ERR_SHUTTING_DOWN
> /etc/squid/errors/ERR_SOCKET_FAILURE
> /etc/squid/errors/ERR_TOO_BIG
> /etc/squid/errors/ERR_UNSUP_REQ
> /etc/squid/errors/ERR_URN_RESOLVE
> /etc/squid/errors/ERR_WRITE_ERROR
> /usr/lib/squid/icons/anthony-link.gif
> /usr/lib/squid/icons/anthony-movie.gif
> /usr/lib/squid/icons/anthony-pdf.gif
> /usr/lib/squid/icons/anthony-portal.gif
> /usr/lib/squid/icons/anthony-ps.gif
> /usr/lib/squid/icons/anthony-quill.gif
> /usr/lib/squid/icons/anthony-script.gif
> /usr/lib/squid/icons/anthony-sound.gif
> /usr/lib/squid/icons/anthony-tar.gif
> /usr/lib/squid/icons/anthony-tex.gif
> /usr/lib/squid/icons/anthony-text.gif
> /usr/lib/squid/icons/anthony-unknown.gif
> /usr/lib/squid/icons/anthony-xbm.gif
> /usr/lib/squid/icons/anthony-xpm.gif
> /usr/lib/squid/pam_auth
> /usr/sbin/squid
> /usr/sbin/client
> /var/lib/squid
> /var/log/squid

27 Gateway Server - FreeS/WAN VPN Server

In this Chapter

Linux FreeS/WAN VPN

Abstract

Protection of client-to-server and vice versa with SSL solutions is an excellent choice but sometime for enterprise environments establishing secure communication channels, assuring full privacy, authenticity and data integrity in between two firewalls over the Internet are vital. For this, IPSEC has been created.

IPSEC is **I**nternet **P**rotocol **SEC**urity. It uses strong cryptography to provide both authentication and encryption services. Authentication ensures that packets are from the right sender and have not been altered in transit. Encryption prevents unauthorized reading of packet contents. IPSEC can protect any protocol running above IP and any medium used below IP. IPSEC can also provide some security services "in the background", with no visible impact on users. More to the point, it can protect a mixture of protocols running over a complex combination of media (i.e. IMAP/POP etc.) without having to change them in any ways, since the encryption occurs at the IP level.

IPSEC services allow you to build secure tunnels through untrusted networks like the Internet. Everything passing through the untrusted net is encrypted by the IPSEC gateway machine and decrypted by the gateway at the other end. The result is **V**irtual **P**rivate **N**etwork or VPN. This is a network, which is effectively private even though it includes machines at several different sites connected by the insecure Internet.

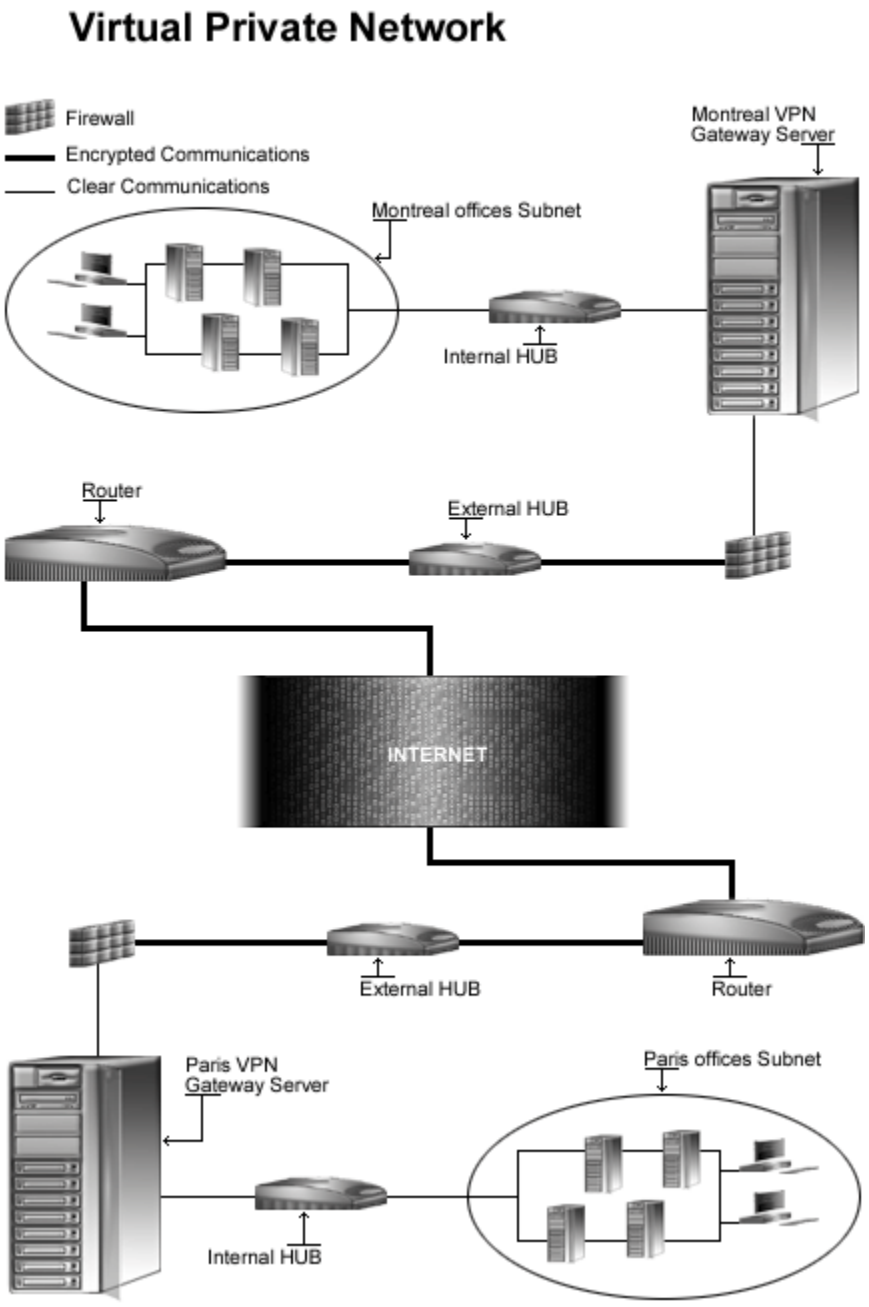
Recommended RPM packages to be installed for a VPN Server

A minimal configuration provides the basic set of packages required by the Linux operating system. Minimal configuration is a perfect starting point for building secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a VPN (FreeS/WAN) server running on FreeS/WAN software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install FreeS/WAN by RPM package. Therefore, freeswan RPM package is already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

basesystem	e2fsprogs	initscripts	openssh	slang
bash	ed	iptables	openssh-server	slocate
bdflush	file	**kernel**	openssl	sysklogd
bind	filesystem	less	pam	syslinux
bzip2	fileutils	libstdc++	passwd	SysVinit
chkconfig	findutils	libtermcap	popt	tar
console-tools	**freeswan**	lilo	procps	termcap
cpio	gawk	logrotate	psmisc	textutils
cracklib	gdbm	losetup	pwdb	tmpwatch
cracklib-dicts	gettext	MAKEDEV	qmail	utempter
crontabs	glib	man	readline	util-linux
db1	glibc	mingetty	rootfiles	vim-common
db2	glibc-common	mktemp	rpm	vim-minimal
db3	grep	mount	sed	vixie-cron
dev	groff	ncurses	setup	words
devfsd	gzip	net-tools	sh-utils	which
diffutils	info	newt	shadow-utils	zlib

Tested and fully functional on OpenNA.com.


Virtual Private Network
Firewall
Encrypted Communications
Clear Communications
Montreal VPN
Gateway Server
Montreal offices Subnet
Internal HUB
Router
External HUB
INTERNET
External HUB
Router
Paris VPN
Gateway Server
Paris offices Subnet
Internal HUB

These installation instructions assume

Commands are Unix-compatible.
The source path `is /var/tmp` (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: Yes
Latest `FreeS/WAN` VPN version number is `1.9`

Packages

The following are based on information as listed by `FreeS/WAN` as of 2001/03/27. Please regularly check at www.freeswan.org for the latest status.

Source code is available from:
`FreeS/WAN` VPN Homepage Site: http://www.freeswan.org/
`FreeS/WAN` VPN FTP Site: 194.109.6.26
You must be sure to download: `freeswan-1.9.tar.gz`

Pristine source

If you don’t use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `FreeS/WAN`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  `[root@deep /root]#` **`find /* > Freeswan1`**
- And the following one after you install the software:
  `[root@deep /root]#` **`find /* > Freeswan2`**
- Then use the following command to get a list of what changed:
  `[root@deep /root]#` **`diff Freeswan1 Freeswan2 > Freeswan-Installed`**

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Prerequisites

The installation of `IPSEC FreeS/WAN` Virtual Private Network software requires some modification of your original kernel since `FreeS/WAN` must be included and incorporated in your kernel before you can use it.

For this reason the first step in installing `FreeS/WAN` software is to go to the Linux Kernel section in this book and follow the instructions on how to install the Linux Kernel on your system (even if you have already done this before) and come back to “Linux `FreeS/WAN VPN`” (this section) after you have executed the “**`make dep; make clean`**” commands, but before the “**`make bzImage`**” command in the Linux Kernel section.

CAUTION: It is highly recommended to not compile anything in the kernel with optimization flags if you intend to use and install the FreeSWAN software on your system. Any optimization flags added to the Linux kernel will produce errors messages in the FreeSWAN IPSEC software when it tries to run; this is an important warning to note, or else nothing will work with FreeSWAN. The optimization flags documented in Chapter related to Linux Kernel, "Securing & Optimizing Kernel" apply without any problems to all sections and chapters of this book with the single exception of the FreeSWAN IPSEC software. Once again, I repeat, don't use or add any optimization options or flags into your Linux kernel when compiling and patching it to support FreeSWAN.

Compiling - Optimizing & Installing FreeS/WAN

Below are the required steps that you must make to compile and optimize the FreeS/WAN software before installing it into your Linux system. Don't forget that your Linux kernel must be pre-configured as described previously before going into the following steps.

Step 1
Once Linux Kernel is pre-configured and you get the FreeS/WAN program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp freeswan-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf freeswan-version.tar.gz
```

Step 2
After that, move into the newly created FreeS/WAN directory then configure, compile and optimize it.

- To move into the top-level directory of FreeS/WAN distribution use the command:

```
[root@deep tmp]# cd freeswan-1.9/
```

Step 3
You must modify the Makefile under the FreeS/WAN source directory and subdirectories named utils, klips/utils, Pluto, and lib to specify installation paths. We must modify these files to be compliant with Linux file system structure and install FreeS/WAN files under our PATH environment variable.

- Edit the **Makefile** file (vi Makefile) and change all of the targeted lines in the order shown below:

```
PUBDIR=$(DESTDIR)/usr/local/sbin
```

To read:

```
PUBDIR=$(DESTDIR)/usr/sbin
```

```
REALPRIVDIR=/usr/local/lib/ipsec
```

To read:

```
REALPRIVDIR=/usr/lib/ipsec
```

```
MANTREE=$(DESTDIR)/usr/local/man
```

To read:

```
MANTREE=$(DESTDIR)/usr/share/man
```

```
CONFDIR=$(DESTDIR)/etc
```

To read:

```
CONFDIR=/etc
```

Step 3.1

- Edit the **`Makefile`** file of the subdirectory `utils` (`vi utils/Makefile`) and change all of the targeted lines in the order shown below:

```
PUBDIR=/usr/local/sbin
```

To read:

```
PUBDIR=/usr/sbin
```

```
PRIVDIR=/usr/local/lib/ipsec
```

To read:

```
PRIVDIR=/usr/lib/ipsec
```

```
REALPRIVDIR=/usr/local/lib/ipsec
```

To read:

```
REALPRIVDIR=/usr/lib/ipsec
```

```
MANTREE=/usr/local/man
```

To read:

```
MANTREE=/usr/share/man
```

Step 3.2

- Edit the **Makefile** file of the subdirectory `klips/utils` (`vi klips/utils/Makefile`) and change all of the targeted lines in the order shown below:

  ```
  CFLAGS=-O2 -I../net/ipsec -I../../lib -g
  ```

  To read:

  ```
  CFLAGS=-O3 -I../net/ipsec -I../../lib -g
  ```

  ```
  BINDIR=/usr/local/lib/ipsec
  ```

  To read:

  ```
  BINDIR=/usr/lib/ipsec
  ```

  ```
  MANTREE=/usr/local/man
  ```

  To read:

  ```
  MANTREE=/usr/share/man
  ```

Step 3.3

- Edit the **Makefile** file of the subdirectory `pluto` (`vi pluto/Makefile`) and change all of the targeted lines in the order shown below:

  ```
  BINDIR=/usr/local/lib/ipsec
  ```

  To read:

  ```
  BINDIR=/usr/lib/ipsec
  ```

  ```
  MANTREE=/usr/local/man
  ```

  To read:

  ```
  MANTREE=/usr/share/man
  ```

Step 3.4

- Edit the **Makefile** file of the subdirectory `lib` (`vi lib/Makefile`) and change the following line:

  ```
  MANTREE=/usr/local/man
  ```

  To read:

  ```
  MANTREE=/usr/share/man
  ```

Step 3.5

- Edit the **Makefile** file of the subdirectory libdes (vi libdes/Makefile) and change all of the targeted lines in the order shown below:

  LIBDIR=/usr/local/lib

  To read:

  LIBDIR=/usr/lib

  BINDIR=/usr/local/bin

  To read:

  BINDIR=/usr/bin

  INCDIR=/usr/local/include

  To read:

  INCDIR=/usr/include

  MANDIR=/usr/local/man

  To read:

  MANDIR=/usr/share/man

All of the above changes (step3 to step 3.5), will relocate all files related to the FreeS/WAN software to the destination target directories we have chosen in order to be compliant with the Linux file system structure.

Step 4
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install FreeS/WAN in the server:

```
[root@deep freeswan-1.9]#  make insert
[root@deep freeswan-1.9]#  make programs
[root@deep freeswan-1.9]# cd
[root@deep /root]# find /* > Freeswan1
[root@deep /root]# cd /var/tmp/freeswan-1.6/
[root@deep freeswan-1.9]#  make install
[root@deep freeswan-1.9]# cd
[root@deep /root]# find /* > Freeswan2
[root@deep /root]# diff Freeswan1 Freeswan2 > Freeswan-Installed
```

The **make insert** command will creates a symbolic link /usr/src/linux/net/ipsec, pointing to the KLIPS source directory. It patches some kernel files, where necessary, to know about KLIPS and/or to fix bugs. It also adds its default configuration to the kernel configuration file, and finally, it makes the KLIPS communication file, /dev/ipsec, if it's not already there.

The **make programs** command builds the libraries, Pluto, and various user-level utilities. The **make install** will install the Pluto daemon and user-level utilities, and set things up for boot-time start-up.

Step 5
Once the program is installed into the system, we must return to the /usr/src/linux directory and execute the following commands to reconfigure and install the kernel with FreeS/WAN VPN support enable:

```
[root@deep freeswan-1.9]#  cd /usr/src/linux/
[root@deep linux]#  make config
```

WARNING: The difference with the make config command we used before to configure the kernel is that now a new section related to FreeS/WAN has been included in our kernel configuration, and for this reason we must reconfigure the kernel to customize the IPSec options to be a part of the kernel.

The first thing you need to do is ensure that your kernel has been built with FreeS/WAN support enabled. In the 2.4 kernel version, a new section related to Frees/WAN VPN support named "IPSec options (FreeS/WAN)" should appear in your kernel configuration after you have patched the kernel with the FreeS/WAN program as descibed above. You need ensure that you have answered **Y** to the following questions under the new kernel section: IPSec options (FreeS/WAN).

```
IPSec options (FreeS/WAN)
IP Security Protocol (FreeS/WAN IPSEC) (CONFIG_IPSEC) [Y/n/?]
IPSEC: IP-in-IP encapsulation (CONFIG_IPSEC_IPIP) [Y/n/?]
IPSEC: PF_KEYv2 kernel/user interface (CONFIG_IPSEC_PFKEYv2) [Y/n/?]
IPSEC: Enable ICMP PMTU messages (CONFIG_IPSEC_ICMP) [Y/n/?]
IPSEC: Authentication Header (CONFIG_IPSEC_AH) [Y/n/?]
HMAC-MD5 authentication algorithm (CONFIG_IPSEC_AUTH_HMAC_MD5) [Y/n/?]
HMAC-SHA1 authentication algorithm (CONFIG_IPSEC_AUTH_HMAC_SHA1) [Y/n/?]
IPSEC: Encapsulating Security Payload (CONFIG_IPSEC_ESP) [Y/n/?]
3DES encryption algorithm (CONFIG_IPSEC_ENC_3DES) [Y/n/?]
IPSEC Debugging Option (DEBUG_IPSEC) [Y/n/?]
```

NOTE: All the customizations you made to your kernel the first time you ran the make config, make dep, and make clean commands will be preserved, so you don't need to reconfigure every part of your kernel; Just the new section added by FreeS/WAN named "IPSec options (FreeS/WAN)" is required, as shown above.

Some networking options will get turned on automatically, even if you previously turned them off; This is because IPSEC needs them. Whichever configuration program you are using, you should pay careful attention to a few issues: in particular, do NOT disable any of the following under the "Networking Options" of your kernel configuration:

```
Kernel/User netlink socket (CONFIG_NETLINK) [Y/n/?]
Netlink device emulation (CONFIG_NETLINK_DEV) [Y/n/?]
```

Step 6
Now that we have included in the kernel the support for FreeS/WAN VPN, we can compile and install the new kernel.

- Return to the /usr/src/linux directory and run the following commands again:
```
[root@deep linux]#  make dep; make clean; make bzImage
```

After execution of the above commands, follow the rest of the instructions in the Linux Kernel chapter of this book as normal to install the kernel. At this point, after you have copied and installed your new kernel image, system.map, or modules (if necessary), and set the lilo.conf file to load the new kernel, you must edit and customize the configuration files related to FreeS/WAN "ipsec.conf" and "ipsec.secrets" before rebooting your system.

Step 7
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete FreeS/WAN and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf freeswan-version/
[root@deep tmp]# rm -f freeswan-version.tar.gz
```

The rm command as used above will remove all the source files we have used to compile and install FreeS/WAN. It will also remove the FreeS/WAN compressed archive from the /var/tmp directory.

Configuring FreeS/WAN

After building FreeS/WAN, your next step is to verify or change, if necessary options in your FreeS/WAN configuration files. Those files are:

- ✓ /etc/ipsec.conf (The FreeS/WAN Configuration File)
- ✓ /etc/ipsec.secrets (The FreeS/WAN Configuration File to store secret keys)

/etc/ipsec.conf: The FreeS/WAN Configuration File

The configuration file for FreeS/WAN (/etc/ipsec.conf) allows you to configure your IPSEC configurations, control information and connections types. IPSEC currently supports two types of connections: Manually keyed and Automatically keyed.

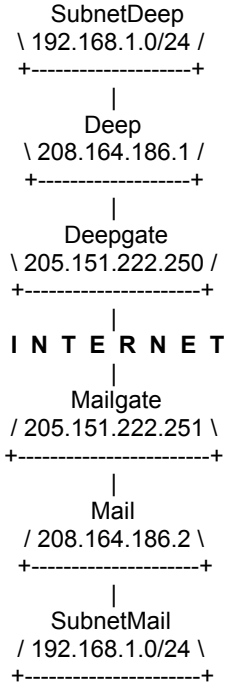
The difference is strictly in how they are keyed. Manually keyed connections use keys stored in the /etc/ipsec.conf file. This type of connection is less secure than automatically keyed. Automatically keyed connections use keys automatically generated by the Pluto key negotiation daemon. The key negotiation protocol, used by default and named IKE, authenticates the other system using shared secrets stored in /etc/ipsec.secrets file. For these reasons, we will use and show you the automatically keyed connection that is more secure than the manually keyed connection (it is highly recommended that you use the automatically keyed connection).

In our example configuration below, we configure a sample tunnel with a firewall-penetrating tunnel, and we assume that firewalling is being done on the left and right side. We choose to show you this configuration since we assume it is what most users and companies will use.

Also, it allows us to play with more options in the configuration file `ipsec.conf` for automatically keyed connections. Different configurations exist and you may consult the "`doc/examples`" file under the subdirectory "`doc`" of the `Frees/WAN` source directory for more information and other possible configurations.

SubnetDeep======Deep------Deepgate.....**Mailgate-------Mail======SubnetMail**
Untrusted net

leftsubnet = **SubnetDeep** (192.168.1.0/24)
left = **Deep** (deep.openna.com)
leftnexthop = **Deepgate** (the first router in the direction or ISP router for deep.openna.com)
Internet = **Untrusted net**
rightnexthop = **Mailgate** (the first router in the direction or ISP router for mail.openna.com)
right = **Mail** (mail.openna.com)
rightsubnet = **SubnetMail** (192.168.1.0/24)



SubnetDeep is the IP network address of your private internal network on the first gateway. eth1 is attached to the internal network.

Deep is the IP address of your first Gateway. eth0 is attached to the Internet.

Deepgate is the IP address of the first router in the direction of your second gateway (mail.openna.com) or your ISP router.

INTERNET is the untrusted network.

Mailgate is the IP address of the first router in the direction of your first gateway (deep.openna.com) or your ISP router.

Mail is the IP address of your second Gateway. eth0 is attached to the Internet.

SubnetMail is the IP network address of your private internal network on the second gateway. eth1 is attached to the internal network.

We must edit the **`ipsec.conf`** file (`vi /etc/ipsec.conf`) and change the default values to fit our specifications for `IPSEC` configuration and communication. Currently there are two types of section in this file (`/etc/ipsec.conf`): a "**`config`**" section, which specifies general configuration information for `IPSEC`, and a "**`conn`**" section which specifies an `IPSEC` connection. Its contents are not security-sensitive unless manual keying is being done (recall, manual keying is not recommended for security reasons).

The first section type, named **`config setup`**, is the only **`config`** section known to the `IPSEC` software containing overall setup parameters for `IPSEC` that apply to all connections, and information used when the software is being started.

The second type, named **conn,** contains a connection specification defining a network connection to be made using IPSEC. The name it is given is arbitrary, and is simply used to identify the connection to ipsec_auto(8) and ipsec_manual(8).

```
# /etc/ipsec.conf - FreeS/WAN IPSEC configuration file

# More elaborate and more varied sample configurations can be found
# in doc/examples.

# basic configuration
config setup
        interfaces="ipsec0=eth0"
        klipsdebug=none
        plutodebug=none
        plutoload=%search
        plutostart=%search

# sample connection
conn deep-mail
        left=208.164.186.1
        leftsubnet=192.168.1.0/24
        leftnexthop=205.151.222.250
        right=208.164.186.2
        rightsubnet=192.168.1.0/24
        rightnexthop=205.151.222.251
        keyingtries=0
        auth=ah
        auto=start
```

This tells ipsec.conf file to set itself up for this particular configuration setup with:

interfaces="ipsec0=eth0"
This option specifies which appropriate virtual and physical interfaces for IPSEC to use. The default setting, "interfaces=%defaultroute", will look for your default connection to the Internet, or your corporate network. Also, you can name one or more specific interfaces to be used by FreeS/WAN. For example:

```
interfaces="ipsec0=eth0"
interfaces="ipsec0=eth0 ipsec1=ppp0"
```

Both set the eth0 interface as ipsec0. The second one, however, also supports IPSEC over a PPP interface. If the default setting "interfaces=%defaultroute" is not used, then the specified interfaces will be the only ones this gateway machine can use to communicate with other IPSEC gateways.

klipsdebug=none
This option specifies the debugging output for KLIPS (the kernel IPSEC code). The default value **none**, means no debugging output and the value **all** means full output.

plutodebug=none
This option specifies the debugging output for the Pluto key. The default value, **none**, means no debugging output, and the value **all** means full output.

plutoload=%search
This option specifies which connections (by name) to load automatically into memory when Pluto starts. The default is none and the value **%search** loads all connections with auto=add or auto=start.

plutostart=%search
This option specifies which connections (by name) to automatically negotiate when Pluto starts. The default is none and the value **%search** starts all connections with auto=start.

conn deep-mail
This option specifies the name given to identify the connection specification to be made using IPSEC. It's a good convention to name connections by their ends to avoid mistakes. For example, the link between deep.openna.com and mail.openna.com gateways server can be named "deep-mail", or the link between your Montreal and Paris offices, "montreal-paris".

Note that the names "deep-mail" or whatever you have chosen should be the same in the ipsec.conf file on both gateways. In other words, the only change you should make in the /etc/ipsec.conf file on the second gateway is changing the "interfaces=" line to match the interface the second gateway uses for IPSEC connection, if, of course, it's different from the first gateway. For example, if the interface eth0 is used on the both gateways for IPSEC communication, you don't need to change the line "interfaces=" on the second gateway. On the other hand, if the first gateway use eth0 and the second use eth1, you must change the line "interfaces=" on the second gateway to match the interface eth1.

left=208.164.186.1
This option specifies the IP address of the gateway's external interface used to talk to the other gateway.

leftsubnet=192.168.1.0/24
This option specifies the IP network or address of the private subnet behind the gateway.

leftnexthop=205.151.222.250
This option specifies the IP address of the first router in the appropriate direction or ISP router.

right=208.164.186.2
This is the same explanation as "left=" but for the right destination.

rightsubnet=192.168.1.0/24
This is the same explanation as "leftsubnet=" but for the right destination.

rightnexthop=205.151.222.251
This is the same explanation as "leftnexthop=" but for the right destination.

keyingtries=0
This option specifies how many attempts (an integer) should be made in (re)keying negotiations. The default value 0 (retry forever) is recommended.

auth=ah
This option specifies whether authentication should be done separately using AH (Authentication Header), or be included as part of the ESP (Encapsulated Security Payload) service. This is preferable when the IP headers are exposed to prevent man-in-the-middle attacks.

auto=start
This option specifies whether automatic startup operations should be done at IPSEC startup.

NOTE: A data mismatch anywhere in this configuration "ipsec.conf" will cause FreeS/WAN to fail and to log various error messages.

/etc/ipsec.secrets: The FreeS/WAN File to store Secret Keys

The file ipsec.secrets stores the secrets used by the pluto daemon to authenticate communication between both gateways. Two different kinds of secrets can be configured in this file, which are preshared secrets and RSA private keys. You must check the modes and permissions of this file to be sure that the super-user "root" owns the file, and its permissions are set to block all access by others.

Step 1
An example secret is supplied in the ipsec.secrets file by default. You should change it by creating your own. With automatic keying you may have a shared secret up to 256 bits, which is then used during the key exchanges to make sure a man in the middle attack does not occur.

- To create a new shared secret, use the following commands:
  `[root@deep /]# ipsec ranbits 256 > temp`

New, random keys are created with the ranbits(8) utility in the file named "temp". The ranbits utility may pause for a few seconds if not enough entropy is available immediately. Don't forget to delete the temporary file as soon as you are done with it.

Step 2
Now that our new shared secret key has been created in the "temp" file, we must put it in the /etc/ipsec.secrets file. When editing the ipsec.secrets file, you should see something like the following appearing in your text editor. Each line has the IP addresses of the two gateways plus the secret. It should look something like this:

```
# This file holds shared secrets which are currently the only inter-Pluto
# authentication mechanism.  See ipsec_pluto(8) manpage.  Each secret is
# (oversimplifying slightly) for one pair of negotiating hosts.

# The shared secrets are arbitrary character strings and should be both
# long and hard to guess.

# Note that all secrets must now be enclosed in quotes, even if they have
# no white space inside them.

10.0.0.1 11.0.0.1         "jxVS1kVUTTulkVRRTnTujSm444jRuU1mlkklku2nkW3nnVu
V2WjjRRnulmlkmU1Run5VSnnRT"
```

- Edit the **ipsec.secrets** file (vi /etc/ipsec.secrets) and change the default secrets keys:

```
10.0.0.1 11.0.0.1  " jxVS1kVUTTulkVRRTnTujSm444jRuU1mlkklku2nkW3nnVu
V2WjjRRnulmlkmU1Run5VSnnRT "
```

To read:

```
208.164.186.1 208.164.186.2
"0x9748cc31_2e99194f_d230589b_cd846b57_dc070b01_74b66f34_19c40a1a_804906ed"
```

Where "208.164.186.1" and "208.164.186.2" are the IP addresses of the two gateways and "0x9748cc31_2e99194f_d230589b_cd846b57_dc070b01_74b66f34_19c40a1a_804906ed" (note that the quotes are required) is the shared secret we have generated above with the command "ipsec ranbits 256 > temp" in the "temp" file.

Step 3
The files ipsec.conf, and ipsec.secrets must be copied to the second gateway machine so as to be identical on both ends. The only exception to this is the ipsec.conf file, which must have in it a section labeled by the line **config setup** with the correct interface settings for the second gateway, if they differ from the first. The ipsec.secrets file, contrary to the RSA private key, should have the same-shared secrets on the two gateways.

WARNING: The file /etc/ipsec.secrets should have permissions rw------- (600) and be owned by the super-user "root". The file /etc/ipsec.conf is installed with permissions rw-r--r— (644) and must be owned also by "root".

Configuring RSA private keys secrets

Recall that currently with FreeSWAN software there are two kinds of secrets: preshared secrets and RSA private keys. The preshared secrets are what we have configured in our ipsec.conf and ipsec.secrets example, above. Some people may prefer to use RSA private keys for authentication by the Pluto daemon of the other hosts. If you are in this situation, you will have to make some minor modifications to your ipsec.conf and ipsec.secrets files as described in the following steps:

You need to create a separate RSA key for *each* gateway. Each one gets its private key in its own ipsec.secrets file, and the public keys go in leftrsasigkey and rightrsasigkey parameters in the **conn** description of ipsec.conf file, which goes to both.

Step 1
Create a separate RSA key for *each* gateway:

- On the first gateway (e.i. deep), use the following commands:

```
[root@deep /]# cd /
[root@deep /]# ipsec rsasigkey --verbose 1024 > deep-keys
computing primes and modulus...
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 30 tries
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 230 tries
swapping primes so p is the larger
computing (p-1)*(q-1)...
computing d...
computing exp1, exp1, coeff...
output...
```

- On the second gateway (e.i. `mail`), use the following commands:

```
[root@mail /]# cd /
[root@mail /]# ipsec rsasigkey --verbose 1024 > mail-keys
computing primes and modulus...
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 30 tries
getting 64 random bytes from /dev/random
looking for a prime starting there
found it after 230 tries
swapping primes so p is the larger
computing (p-1)*(q-1)...
computing d...
computing exp1, exp1, coeff...
output...
```

The `rsasigkey` utility generates an RSA public and private key pair of a 1024-bit signature, and puts it in the file `deep-keys` (`mail-keys` for the second command on the second gateway). The private key can be inserted verbatim into the `ipsec.secrets` file, and the public key into the `ipsec.conf` file.

WARNING: The `rsasigkey` utility may pause for a few seconds if not enough entropy is available immediately. You may want to give it some bogus activity such as random mouse movements. The temporary RSA "`deep-keys`" and "`mail-keys`" files should be deleted as soon as you are done with it. Don't forget to delete the `deep-keys` and `mail-keys` RSA files.

Step 2
Modify your `/etc/ipsec.conf` files to use RSA public keys in *each* gateway:

Edit you original **`ipsec.conf`** file (`vi /etc/ipsec.conf`) and add the following parameters related to RSA in the **`conn`** desciption of your `ipsec.conf` file on both gateway:

```
# sample connection
conn deep-mail
        left=208.164.186.1
        leftsubnet=192.168.1.0/24
        leftnexthop=205.151.222.250
        right=208.164.186.2
        rightsubnet=192.168.1.0/24
        rightnexthop=205.151.222.251
        keyingtries=0
        auth=ah
        authby=rsasig
        leftrsasigkey=<Public key of deep>
        rightrsasigkey=<Public key of mail>
        auto=start
```

`authby=rsasig`
This parameter specifies how the two security gateways should authenticate each other. The default value is secret for shared secrets. We must specify rsasig for RSA since we have decided to use RSA digital signatures.

leftrsasigkey=<Public key of deep>
This parameter specifies the left participant's public key for RSA signature authentication. In our example, left is 208.164.186.1, and represents deep.openna.com, so we must put the RSA public key for deep on this line.

rightrsasigkey=<Public key of mail>
This parameter specifies the right participant's public key for RSA signature authentication. In our example, right is 208.164.186.2, and represents mail.openna.com, so we must put the RSA public key of mail on this line.

You can retrieve the public key of deep in the RSA key file named "deep-keys", and the public key of mail in the RSA key file named "mail-keys", that we have created in step 1 above. These files will look like this:

RSA keys for gateway deep (deep-keys):

```
[root@deep /]# cd /
[root@deep /]# vi deep-keys

    # 1024 bits, Fri Feb  4 05:05:19 2000
    # for signatures only, UNSAFE FOR ENCRYPTION
    #pubkey=0x010395daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74
bcfb51a6ecc08890d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb
4d7689f2da7a69199e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16
a76bab772888f1fd71aa08f08502a141b611f
    Modulus:
0x95daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74bcfb51a6ecc08890
d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb4d7689f2da7a6919
9e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16a76bab772888f1fd
71aa08f08502a141b611f
    PublicExponent: 0x03
    # everything after this point is secret
    PrivateExponent:
0x63e74967eaea2025c98c69f6ef0753a6a3ff6764157dbdf1f50013471324dd352366f48805b0b
37f232384b2b52ce2ee85d173468b62eaa052381a9588a317b3a1324d01a531a41fa7add6c5efbd
d88f4718feed2bc0246be924e81bb90f03e49ceedf7af0dd48f06f265b519600bd082c6e6bd27ea
a71cc0288df1ecc3b062b
    Prime1:
0xc5b471a88b025dd09d4bd7b61840f20d182d9b75bb7c11eb4bd78312209e3aee7ebfe632304db
6df5e211d21af7fee79c5d45546bea3ccc7b744254f6f0b847f
    Prime2:
0xc20a99feeafe79767122409b693be75f15e1aef76d098ab12579624aec708e85e2c5dd62080c3
a64363f2f45b0e96cb4aef8918ca333a326d3f6dc2c72b75361
    Exponent1:
0x83cda11b0756e935be328fcebad5f6b36573bcf927a80bf2328facb6c0697c9eff2a9976cade7
9ea3ec0be1674fff4512e8d8e2f29c2888524d818df9f5d02ff
    Exponent2:
0x815c66a9f1fefba44b6c2b124627ef94b9411f4f9e065c7618fb96dc9da05f03ec83e8ec055d7
c42ced4ca2e75f0f3231f5061086ccd176f37f9e81da1cf8ceb
    Coefficient:
0x10d954c9e2b8d11f4db1b233ef37ff0a3cecfffad89ba5d515449b007803f577e3bd7f0183ced
dfd805466d62f767f3f5a5731a73875d30186520f1753a7e325
```

RSA keys for gateway mail (mail-keys):

```
[root@mail /]# cd /
[root@mail /]# vi mail-keys
```

```
    # 1024 bits, Fri Feb  4 04:46:59 2000
    # for signatures only, UNSAFE FOR ENCRYPTION
    #pubkey=0x01037631b81f00d5e6f888c542d44dbb784cd3646f084ed96f942d341c7c4686c
bd405b805dc728f8697475f11e8b1dd797550153a3f0d4ff0f2b274b70a2ebc88f073748d1c1c88
21dc6be6a2f0064f3be7f8e4549f8ab9af64944f829b014788dd202cf7d2e320cab666f5e7a197e
64efe0bfee94e92ce4dad82d5230c57b89edf
    Modulus:
0x7631b81f00d5e6f888c542d44dbb784cd3646f084ed96f942d341c7c4686cbd405b805dc728f8
697475f11e8b1dd797550153a3f0d4ff0f2b274b70a2ebc88f073748d1c1c8821dc6be6a2f0064f
3be7f8e4549f8ab9af64944f829b014788dd202cf7d2e320cab666f5e7a197e64efe0bfee94e92c
e4dad82d5230c57b89edf
    PublicExponent: 0x03
    # everything after this point is secret
    PrivateExponent:
0x4ecbd014ab3944a5b08381e2de7cfadde242f4b03490f50d737812fd8459dd3803d003e84c5fa
f0f84ea0bf07693a64e35637c2a08dff5f721a324b1747db09f62c871d5e11711251b845ae76753
d4ef967c494b0def4f5d0762f65da603bc04c41b4c6cab4c413a72c633b608267ae2889c162a3d5
bc07ee083b1c6e038400b
    Prime1:
0xc7f7cc8feaaac65039c39333b878bffd8f95b0dc22995c553402a5b287f341012253e9f25b839
83c936f6ca512926bebee3d5403bf9f4557206c6bbfd9aac899
    Prime2:
0x975015cb603ac1d488dc876132d8bc83079435d2d3395c03d5386b5c004eadd4d7b01b3d86aad
0a2275d2d6b791a2abe50d7740b7725679811a32ca22db97637
    Exponent1:
0x854fddb5471c84357bd7b777d0507ffe5fb92092c1bb92e37801c3cc5aa22b5616e29bf6e7ad1
028624a486e0c619d47f428e2ad2a6a2e3a159d9d2a911c85bb
    Exponent2:
0x64e00e87957c81385b3daf9621e5d302050d7937377b92ad38d04792aadf1e8de52012290471e
06c1a3e1e47a61171d435e4f807a4c39a6561177316c9264ecf
    Coefficient:
0x6f087591becddc210c2ee0480e30beeb25615a3615203cd3cef65e5a1d476fd9602ca0ef10d9b
858edb22db42c975fb71883a470b43433a7be57df7ace4a0a3f
```

Extract and copy the public RSA key files of `deep` and `mail` to your `ipsec.conf` files as shown below. You can locate the line related to the public key by a sentence beginning with the commented-out: "`#pubkey=`" line.

```
# sample connection
conn deep-mail
    left=208.164.186.1
    leftsubnet=192.168.1.0/24
    leftnexthop=205.151.222.250
    right=208.164.186.2
    rightsubnet=192.168.1.0/24
    rightnexthop=205.151.222.251
    keyingtries=0
    auth=ah
    authby=rsasig
    leftrsasigkey=0x010395daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801ce
    a9cb74bcfb51a6ecc08890d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef9
    8a7f29edcb4d7689f2da7a69199e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5d
    eac3b19d561c16a76bab772888f1fd71aa08f08502a141b611f
    rightrsasigkey=0x01037631b81f00d5e6f888c542d44dbb784cd3646f084ed96f942d341c
    7c4686cbd405b805dc728f8697475f11e8b1dd797550153a3f0d4ff0f2b274b70a2ebc88f07
    3748d1c1c8821dc6be6a2f0064f3be7f8e4549f8ab9af64944f829b014788dd202cf7d2e320
    cab666f5e7a197e64efe0bfee94e92ce4dad82d5230c57b89edf
    auto=start
```

NOTE: Don't forget that, in this example, the "leftrsasigkey=" parameter contains the public key of deep and the "rightrsasigkey=" parameter contains the public key of mail.

Step 3
Modify your /etc/ipsec.secrets files to use RSA private keys in *each* gateway:

Edit your original **ipsec.secrets** file (vi /etc/ipsec.secrets) and add the RSA private key for authentication on both gateways:

The ipsec.secrets file for gateway deep:

```
[root@deep /]# vi /etc/ipsec.secrets

208.164.186.1 208.164.186.2
"0x9748cc31_2e99194f_d230589b_cd846b57_dc070b01_74b66f34_19c40a1a_804906ed"
```

You must change your original ipsec.secrets file as shown above to look like the following on both gateways. It is important to note that the private keys are not the same on both gateways, deep and mail. The private key for deep comes from the RSA key file "deep-keys", while the private key for mail comes from the RSA key file "mail-keys":

```
208.164.186.1 208.164.186.2: RSA {
        Modulus:
0x95daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74bcfb51a6ecc08890
d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb4d7689f2da7a6919
9e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16a76bab772888f1fd
71aa08f08502a141b611f
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent:
0x63e74967eaea2025c98c69f6ef0753a6a3ff6764157dbdf1f50013471324dd352366f48805b0b
37f232384b2b52ce2ee85d173468b62eaa052381a9588a317b3a1324d01a531a41fa7add6c5efbd
d88f4718feed2bc0246be924e81bb90f03e49ceedf7af0dd48f06f265b519600bd082c6e6bd27ea
a71cc0288df1ecc3b062b
        Prime1:
0xc5b471a88b025dd09d4bd7b61840f20d182d9b75bb7c11eb4bd78312209e3aee7ebfe632304db
6df5e211d21af7fee79c5d45546bea3ccc7b744254f6f0b847f
        Prime2:
0xc20a99feeafe79767122409b693be75f15e1aef76d098ab12579624aec708e85e2c5dd62080c3
a64363f2f45b0e96cb4aef8918ca333a326d3f6dc2c72b75361
        Exponent1:
0x83cda11b0756e935be328fcebad5f6b36573bcf927a80bf2328facb6c0697c9eff2a9976cade7
9ea3ec0be1674fff4512e8d8e2f29c2888524d818df9f5d02ff
        Exponent2:
0x815c66a9f1fefba44b6c2b124627ef94b9411f4f9e065c7618fb96dc9da05f03ec83e8ec055d7
c42ced4ca2e75f0f3231f5061086ccd176f37f9e81da1cf8ceb
        Coefficient:
0x10d954c9e2b8d11f4db1b233ef37ff0a3cecfffad89ba5d515449b007803f577e3bd7f0183ced
dfd805466d62f767f3f5a5731a73875d30186520f1753a7e325
    }
```

The ipsec.secrets file for gateway mail:

```
[root@mail /]# vi /etc/ipsec.secrets

208.164.186.1 208.164.186.2: RSA {
        Modulus:
0x95daee1be05f3038ae529ef2668afd79f5ff1b16203c9ceaef801cea9cb74bcfb51a6ecc08890
d3eb4b5470c0fc35465c8ba2ce9d1145ff07b5427e04cf4a38ef98a7f29edcb4d7689f2da7a6919
9e4318b4c8d0ea25d33e4f084186a2a54f4b4cec12cca1a5deac3b19d561c16a76bab772888f1fd
71aa08f08502a141b611f
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent:
0x63e74967eaea2025c98c69f6ef0753a6a3ff6764157dbdf1f50013471324dd352366f48805b0b
37f232384b2b52ce2ee85d173468b62eaa052381a9588a317b3a1324d01a531a41fa7add6c5efbd
d88f4718feed2bc0246be924e81bb90f03e49ceedf7af0dd48f06f265b519600bd082c6e6bd27ea
a71cc0288df1ecc3b062b
        Prime1:
0xc5b471a88b025dd09d4bd7b61840f20d182d9b75bb7c11eb4bd78312209e3aee7ebfe632304db
6df5e211d21af7fee79c5d45546bea3ccc7b744254f6f0b847f
        Prime2:
0xc20a99feeafe79767122409b693be75f15e1aef76d098ab12579624aec708e85e2c5dd62080c3
a64363f2f45b0e96cb4aef8918ca333a326d3f6dc2c72b75361
        Exponent1:
0x83cda11b0756e935be328fcebad5f6b36573bcf927a80bf2328facb6c0697c9eff2a9976cade7
9ea3ec0be1674fff4512e8d8e2f29c2888524d818df9f5d02ff
        Exponent2:
0x815c66a9f1fefba44b6c2b124627ef94b9411f4f9e065c7618fb96dc9da05f03ec83e8ec055d7
c42ced4ca2e75f0f3231f5061086ccd176f37f9e81da1cf8ceb
        Coefficient:
0x10d954c9e2b8d11f4db1b233ef37ff0a3cecfffad89ba5d515449b007803f577e3bd7f0183ced
dfd805466d62f767f3f5a5731a73875d30186520f1753a7e325
    }
```

Authentication by RSA Signatures requires that each host have its own private key. The key part of an entry may start with a token indicating the kind of key. "RSA" signifies RSA private key and "PSK" (which is the default) signifies PreShared Key. Since "PSK" is the default, we must specify "RSA", so that we'll be able to use RSA private keys in this file (ipsec.secrets). The super-user "root" should own the file ipsec.secrets, and its permissions should be set to block all access by others.

Requiring network setup for IPSec

There are some considerations you must ensure are correct before running FreeS/WAN software. These considerations are important if you don't want to receive error messages during start up of your VPN. The following are the steps to follow:

Step1
You will need to enable TCP/IP forwarding on the both gateway servers. In Linux, this is accomplished by adding the following line:

- To enable IPv4 forwarding on your Linux system, edit the /etc/sysctl.conf file (vi /etc/sysctl.conf) and add the following line:

  ```
  # Enable packet forwarding
  net.ipv4.ip_forward = 1
  ```

You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

Step 2
Recall that automatically keyed connections use keys automatically generated by the Pluto key negotiation daemon. The pluto daemon will start up, try to connect to the Pluto daemon at the other end of the tunnel, and establish a connection. For this reason, an IPSEC gateway should have packet filters rules (in the firewall script file) permitting the following protocols to traverse the gateway when talking to other IPSEC gateway:

- ✓ UDP port 500 for IKE implemented by the Pluto daemon
- ✓ Protocol 50 for ESP encryption and/or authentication
- ✓ Protocol 51 for AH packet-level authentication

- Edit the **iptables** script file (vi /etc/rc.d/init.d/iptables) on both gateway machines, and add the following lines to allow IPSEC packets to traverse the remote network gateway to your network gateway and vice versa:

```
# FreeS/WAN IPSec VPN
# -------------------

# If you are using the FreeSWAN IPSec VPN, you will need to fill in the
# addresses of the gateways in the IPSECSG and the virtual interfaces for
# FreeS/Wan IPSEC in the FREESWANVI parameters. Look at the beginning of
# this firewall script rules file to set the parameters.

# IPSECSG is a Space separated list of remote gateways. FREESWANVI is a
# Space separated list of virtual interfaces for FreeS/Wan IPSEC
# implementation. Only include those that are actually used.

# Allow IPSEC protocol from remote gateways on external interface
# IPSEC uses three main types of packet:
# IKE uses the UDP protocol and port 500,
# ESP use the protocol number 50, and
# AH use the protocol number 51

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p udp \
         -s $IPSECSG --source-port -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p udp \
         -d $IPSECSG --destination-port -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p 50 \
         -s $IPSECSG --source-port -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p 50 \
         -d $IPSECSG --destination-port -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p 51 \
         -s $IPSECSG --source-port -j ACCEPT

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p 51 \
```

```
        -d $IPSECSG --destination-port -j ACCEPT

# Allow all traffic to FreeS/WAN Virtual Interface
iptables -A INPUT  -i $FREESWANVI \
        --source-port \
        --destination-port -j ACCEPT

iptables -A OUTPUT -o $FREESWANVI \
        --source-port \
        --destination-port -j ACCEPT

# Forward anything from the FreeS/WAN virtual interface IPSEC tunnel
iptables -A FORWARD -i $FREESWANVI \
        --source-port \
        --destination-port -j ACCEPT
```

Where `EXTERNAL_INTERFACE="eth0"` # You external interface to the Internet.
Where `IPSECSG="208.164.186.2"` # Space separated list of remote `VPN` gateways.
Where `FREESWANVI="ipsec0"` # Space separated list of virtual interfaces for `FreeS/Wan`.

NOTE: See Chapter related to "Networking Firewall", for more information. Don't forget to add/check these firewall rules in the other gateway as well.

Step 3
The `rp_filter` subsystem (related to `IP` spoofing protection) must be turned off on both gateways for `IPSEC` to work properly. This is accomplished by checking if the value 0 (off) is set in the `/proc/sys/net/ipv4/conf/ipsec0/rp_filter` and `/proc/sys/net/ipv4/conf/eth0/rp_filter` files respectively:

- To check if the value 0 (off) is set in the `rp_filter` files, use the commands:
```
[root@deep /]# cat /proc/sys/net/ipv4/conf/ipsec0/rp_filter
0
[root@deep /]# cat /proc/sys/net/ipv4/conf/eth0/rp_filter
0
```

NOTE: The subdirectory "`ipsec0`" in our example will be created only after the reboot of your system. So you may check the value of the "`rp_filter`" file in the "`ipsec0`" directory after your system has been restarted.

- To set the value 0 (off) in the both `rp_filter` files manually, use the commands:
```
[root@deep /]# echo 0 > /proc/sys/net/ipv4/conf/ipsec0/rp_filter
[root@deep /]# echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter
```

Also you can put lines like the following in your firewall script files `/etc/rc.d/init.d/iptables` on the both gateways to automatically set these values to 0 (off) and avoid making them manually:

```
# Disable IP spoofing protection to allow IPSEC to work properly
echo 0 > /proc/sys/net/ipv4/conf/ipsec0/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/eth0/rp_filter
```

NOTE: In the example of the firewall script file above, we assume that `eth0` is the interface you use for your connection. Of course if you use `eth1` you must change `eth0` to `eth1`, and so on.

If you forget this step you will receive error messages on your terminal such as the following during the start up of `FreeSWAN IPSEC`:

```
ipsec_setup: WARNING: ipsec0 has route filtering turned on, KLIPS may not work
ipsec_setup:  (/proc/sys/net/ipv4/conf/ipsec0/rp_filter = `1', should be 0)
ipsec_setup: WARNING: eth0 has route filtering turned on, KLIPS may not work
ipsec_setup:  (/proc/sys/net/ipv4/conf/eth0/rp_filter = `1', should be 0)
```

Step 4
It's important to note that any masquerading rules for internal networks that use `IPSEC` must come after the rules allowing `IPSEC` related traffic (The step 2 and 3 above), or the machine will try to masquerade the packets, instead of them being passed over to `IPSEC`.

Edit the **`iptables`** script file (`vi /etc/rc.d/init.d/iptables`) on both gateway machines and add/check the following lines to allow masqueraded packets to traverse the remote network gateway to your network gateway and vice versa:

```
# Masquerade internal traffic.

    # All internal traffic is masqueraded externally.

    iptables -A POSTROUTING -t nat -o $EXTERNAL_INTERFACE -j MASQUERADE
```

Where `EXTERNAL_INTERFACE="eth0"` # You external interface to the Internet.
Where `INTRANET=" 192.168.1.0/24"` # whatever private range you use.

NOTE: See chapter related to "Networking Firewall with Masquerading and Forwarding support" for more information.

Now, you can reboot your system, and the machines on Gateway A should be able to talk to the machines on Gateway B with no problems.

Testing the `FreeS/WAN` installation

- Reboot the both gateways to get `FreeS/WAN` started.

- Examine the `/var/log/messages` file for any signs of trouble. If all goes well you should see something like this in the `/var/log/messages` file:

```
Feb  2 05:22:35 deep ipsec_setup: Starting FreeS/WAN IPSEC
snap2000jan31b...
Feb  2 05:22:35 deep ipsec_setup: KLIPS debug `none'
Feb  2 05:22:35 deep ipsec_setup: KLIPS ipsec0 on eth0
192.168.1.1/255.255.255.0 broadcast 192.168.1.255
Feb  2 05:22:36 deep ipsec_setup: Disabling core dumps:
Feb  2 05:22:36 deep ipsec_setup: Starting Pluto (debug `none'):
Feb  2 05:22:37 deep ipsec_setup: Loading Pluto database `deep-mail':
```

```
Feb  2 05:22:37 deep ipsec_setup: Enabling Pluto negotiation:
Feb  2 05:22:37 deep ipsec_setup: Routing for Pluto conns `deep-mail':
Feb  2 05:22:37 deep ipsec_setup: Initiating Pluto tunnel `deep-mail':
Feb  2 05:22:39 deep ipsec_setup: 102 "deep-mail" #1: STATE_MAIN_I1:
initiate
Feb  2 05:22:39 deep ipsec_setup: 104 "deep-mail" #1: STATE_MAIN_I2: from
STATE_MAIN_I1; sent MI2, expecting MR2
Feb  2 05:22:39 deep ipsec_setup: 106 "deep-mail" #1: STATE_MAIN_I3: from
STATE_MAIN_I2; sent MI3, expecting MR3
Feb  2 05:22:39 deep ipsec_setup: 004 "deep-mail" #1: STATE_MAIN_I4: SA
established
Feb  2 05:22:39 deep ipsec_setup: 110 "deep-mail" #2: STATE_QUICK_I1:
initiate
Feb  2 05:22:39 deep ipsec_setup: 004 "deep-mail" #2: STATE_QUICK_I2: SA
established
Feb  2 05:22:39 deep ipsec_setup: ...FreeS/WAN IPSEC started
```

- Examine the `/var/log/secure` file for any signs of trouble. If all goes well you should see something like the following:

```
Feb 21 14:45:42 deep Pluto[432]: Starting Pluto (FreeS/WAN Version 1.3)
Feb 21 14:45:43 deep Pluto[432]: added connection description "deep-mail"
Feb 21 14:45:43 deep Pluto[432]: listening for IKE messages
Feb 21 14:45:43 deep Pluto[432]: adding interface ipsec0/eth0 192.168.1.1
Feb 21 14:45:43 deep Pluto[432]: loading secrets from
"/etc/ipsec.secrets"
Feb 21 14:45:43 deep Pluto[432]: "deep-mail" #1: initiating Main Mode
Feb 21 14:45:44 deep Pluto[432]: "deep-mail" #1: ISAKMP SA established
Feb 21 14:45:44 deep Pluto[432]: "deep-mail" #2: initiating Quick Mode
POLICY_RSASIG+POLICY_ENCRYPT+POLICY_AUTHENTICATE+POLICY_TUNNEL+POLICY_PFS
Feb 21 14:45:46 deep Pluto[432]: "deep-mail" #2: sent QI2, IPsec SA
established
Feb 21 14:45:47 deep Pluto[432]: "deep-mail" #3: responding to Main Mode
Feb 21 14:45:49 deep Pluto[432]: "deep-mail" #3: sent MR3, ISAKMP SA
established
Feb 21 14:45:49 deep Pluto[432]: "deep-mail" #4: responding to Quick Mode
Feb 21 14:45:50 deep Pluto[432]: "deep-mail" #4: IPsec SA established
```

- On both gateways, the following entries should now exist in the `/proc/net/` directory:

```
[root@deep /]# ls -l /proc/net/ipsec_*
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_eroute
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_klipsdebug
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_spi
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_spigrp
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_spinew
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_tncfg
-r--r--r--   1 root     root  0 Feb  2 05:30 /proc/net/ipsec_version
```

- The `IPSEC` interfaces should be attached on top of the specified physical interfaces. Confirm that with:

```
[root@deep /]# cat /proc/net/ipsec_tncfg
ipsec0 -> eth0 mtu=16260 -> 1500
ipsec1 -> NULL mtu=0 -> 0
ipsec2 -> NULL mtu=0 -> 0
ipsec3 -> NULL mtu=0 -> 0
```

- Now execute the following command to show minimal debugging information and see if the output looks something like this:

```
[root@deep /]# ipsec look
deep.openna.com Fri Feb  4 17:25:17 EST 2000
============-============
192.168.1.1/32     -> 192.168.1.2/32     => tun0x106@192.168.1.2
esp0x4450894d@192.168.1.2 ah0x4450894c@192.168.1.2
------------=------------
ah0x3350f551@192.168.1.1 AH_HMAC_MD5: dir=in ooowin=32 seq=115
bit=0xffffffff alen=128 aklen=16
life(c,s,h)=bytes(16140,0,0)add(51656,0,0)use(54068,0,0)packets(115,0,0)
idle=499
ah0x4450894c@192.168.1.2 AH_HMAC_MD5: dir=out ooowin=32 seq=2828 alen=128
aklen=16
life(c,s,h)=bytes(449488,0,0)add(51656,0,0)use(51656,0,0)packets(2828,0,0
) idle=6
esp0x3350f552@192.168.1.1 ESP_3DES: dir=in ooowin=32 seq=115
bit=0xffffffff eklen=24
life(c,s,h)=bytes(13380,0,0)add(51656,0,0)use(54068,0,0)packets(115,0,0)
idle=499
esp0x4450894d@192.168.1.2 ESP_3DES: dir=out ooowin=32 seq=2828 eklen=24
life(c,s,h)=bytes(381616,0,0)add(51656,0,0)use(51656,0,0)packets(2828,0,0
) idle=6
tun0x105@192.168.1.1 IPIP: dir=in 192.168.1.2 -> 192.168.1.1
life(c,s,h)=add(51656,0,0)
tun0x106@192.168.1.2 IPIP: dir=out 192.168.1.1 -> 192.168.1.2
life(c,s,h)=bytes(327581,0,0)add(51656,0,0)use(51656,0,0)packets(2828,0,0
) idle=6
192.168.1.0     0.0.0.0   255.255.255.0   U   0 0 0 eth0
192.168.1.0     0.0.0.0   255.255.255.0   U   0 0 0 ipsec0
192.168.1.1     0.0.0.0   255.255.255.255 UH  0 0 0 eth0
192.168.1.2     192.168.1.2  255.255.255.255 UGH 0 0 0 ipsec0
Destination     Gateway   Genmask Flags   MSS Window  irtt Iface
```

- Try pinging **192.168.1.2** from the **192.168.1.1** client. If this works then you have set it up correctly. If it does not work check your network to make sure 208.164.186.1 can reach 208.164.186.2, and that TCP-IP forwarding is enabled, and make sure that no firewall rules are blocking the packets, or trying to masquerade them before the rules allowing IPSec related traffic. For this test to work, it is important to use pings that go from one subnet to the other.

```
208.164.186.1 ---- 205.151.222.250 ---- 205.151.222.251 ---- 208.164.186.2
  |                                                            |
192.168.1.0/24                                          192.168.1.0/24
  |                                                            |
192.168.1.1                                             192.168.1.2
```

A last note about testing the installation of FreeSWAN IPSEC, if you encounter a problem that you are unable to resolve, you can use the following command to view a collection of debugging information (contents of files, selections from logs, etc.) related to the IPSEC encryption/authentication system that you should send to the Linux-IPSEC Mailing List (linux-ipsec@clinet.fi) to help you.

- Use the following command to make an output of a collection of debugging information:

```
[root@deep /]# ipsec barf > result
```

This command is primarily provided as a convenience for remote debugging; A single command which packages up (and labels) all information that might be relevant to diagnosing a problem in IPSEC.

Further documentation

For more details, there are several man pages you can read:

```
$ man ipsec (8)                          - invoke IPSEC utilities
$ man ipsec atoaddr, addrtoa (3)         - convert Internet addresses to and from ASCII
$ man ipsec atoasr (3)                   - convert ASCII to Internet address, subnet, or range
$ man ipsec atobytes, bytestoa (3)       - convert binary data bytes from and to ASCII formats
$ man ipsec atodata, datatoa (3)         - convert binary data from and to ASCII formats
$ man ipsec atosa, satoa (3)             - convert IPSEC Security Association IDs to and from ASCII
$ man ipsec atosubnet, subnettoa (3)     - convert subnet/mask ASCII form to and from addresses
$ man ipsec atoul, ultoa (3)             - convert unsigned-long numbers to and from ASCII
$ man ipsec auto (8)                     - control automatically-keyed IPSEC connections
$ man ipsec barf (8)                     - spew out collected IPSEC debugging information
$ man ipsec bitstomask (3)               - convert bit count to Internet subnet mask
$ man ipsec eroute (8)                   - manipulate IPSEC extended routing tables
$ man ipsec goodmask (3)                 - is this Internet subnet mask a valid one?
$ man ipsec hostof (3)                   - given Internet address and subnet mask, return host part
$ man ipsec klipsdebug (8)               - set Klips (kernel IPSEC support) debug features and level
$ man ipsec look (8)                     - show minimal debugging information
$ man ipsec manual (8)                   - take manually-keyed IPSEC connections up and down
$ man ipsec masktobits (3)               - convert Internet subnet mask to bit count
$ man ipsec optionsfrom (3)              - read additional ``command-line'' options from file
$ man ipsec pluto (8)                    - IPsec IKE keying daemon
$ man ipsec ranbits (8)                  - generate random bits in ASCII form
$ man ipsec rangetoa (3)                 - convert Internet address range to ASCII
$ man ipsec rsasigkey (8)                - generate RSA signature key
$ man ipsec setup (8)                    - control IPSEC subsystem
$ man ipsec spi (8)                      - manage IPSEC Security Associations
$ man ipsec spigrp (8)                   - group/ungroup IPSEC Security Associations
$ man ipsec subnetof (3)                 - given Internet address and subnet mask, return subnet number
$ man ipsec tncfg (8)                    - associate IPSEC virtual interface with real interface
$ man ipsec whack (8)                    - control interface for IPSEC keying daemon
$ man ipsec.conf (5)                     - IPSEC configuration and connections
$ man ipsec.secrets (5)                  - secrets for IKE/IPsec authentication
$ man ipsec (8)                          - invoke IPSEC utilities
$ man ipsec atoaddr, addrtoa (3)         - convert Internet addresses to and from ASCII
$ man ipsec atoasr (3)                   - convert ASCII to Internet address, subnet, or range
$ man ipsec atobytes, bytestoa (3)       - convert binary data bytes from and to ASCII formats
$ man ipsec atodata, datatoa (3)         - convert binary data from and to ASCII formats
$ man ipsec atosa, satoa (3)             - convert IPSEC Security Association IDs to and from ASCII
$ man ipsec atosubnet, subnettoa (3)     - convert subnet/mask ASCII form to and from addresses
$ man ipsec atoul, ultoa (3)             - convert unsigned-long numbers to and from ASCII
$ man ipsec auto (8)                     - control automatically-keyed IPSEC connections
$ man ipsec barf (8)                     - spew out collected IPSEC debugging information
$ man ipsec bitstomask (3)               - convert bit count to Internet subnet mask
$ man ipsec eroute (8)                   - manipulate IPSEC extended routing tables
$ man ipsec goodmask (3)                 - is this Internet subnet mask a valid one?
$ man ipsec hostof (3)                   - given Internet address and subnet mask, return host part
$ man ipsec klipsdebug (8)               - set Klips (kernel IPSEC support) debug features and level
$ man ipsec look (8)                     - show minimal debugging information
$ man ipsec manual (8)                   - take manually-keyed IPSEC connections up and down
$ man ipsec masktobits (3)               - convert Internet subnet mask to bit count
$ man ipsec optionsfrom (3)              - read additional ``command-line'' options from file
$ man ipsec pluto (8)                    - IPsec IKE keying daemon
$ man ipsec ranbits (8)                  - generate random bits in ASCII form
$ man ipsec rangetoa (3)                 - convert Internet address range to ASCII
$ man ipsec rsasigkey (8)                - generate RSA signature key
$ man ipsec setup (8)                    - control IPSEC subsystem
$ man ipsec spi (8)                      - manage IPSEC Security Associations
$ man ipsec spigrp (8)                   - group/ungroup IPSEC Security Associations
$ man ipsec subnetof (3)                 - given Internet address and subnet mask, return subnet number
$ man ipsec tncfg (8)                    - associate IPSEC virtual interface with real interface
$ man ipsec whack (8)                    - control interface for IPSEC keying daemon
```

$ man ipsec.conf (5) - IPSEC configuration and connections
$ man ipsec.secrets (5) - secrets for IKE/IPsec authentication

List of installed `FreeS/WAN` files on your system

> /etc/rc.d/init.d/ipsec
> /etc/rc.d/rc0.d/K68ipsec
> /etc/rc.d/rc1.d/K68ipsec
> /etc/rc.d/rc2.d/S47ipsec
> /etc/rc.d/rc3.d/S47ipsec
> /etc/rc.d/rc4.d/S47ipsec
> /etc/rc.d/rc5.d/S47ipsec
> /etc/rc.d/rc6.d/K68ipsec
> /etc/ipsec.conf
> /etc/ipsec.secrets
> /usr/lib/ipsec
> /usr/lib/ipsec/spi
> /usr/lib/ipsec/eroute
> /usr/lib/ipsec/spigrp
> /usr/lib/ipsec/tncfg
> /usr/lib/ipsec/klipsdebug
> /usr/lib/ipsec/pluto
> /usr/lib/ipsec/whack
> /usr/lib/ipsec/ipsec
> /usr/lib/ipsec/barf
> /usr/lib/ipsec/manual
> /usr/lib/ipsec/auto
> /usr/lib/ipsec/look
> /usr/lib/ipsec/showdefaults
> /usr/lib/ipsec/_include
> /usr/lib/ipsec/_confread
> /usr/lib/ipsec/_keycensor
> /usr/lib/ipsec/_secretcensor
> /usr/lib/ipsec/_updown
> /usr/lib/ipsec/ranbits
> /usr/lib/ipsec/rsasigkey
> /usr/lib/ipsec/setup
> /usr/man/man3/ipsec_atoaddr.3
> /usr/man/man3/ipsec_addrtoa.3
> /usr/man/man3/ipsec_atosubnet.3
> /usr/man/man3/ipsec_subnettoa.3
> /usr/man/man3/ipsec_atoasr.3
> /usr/man/man3/ipsec_rangetoa.3
> /usr/man/man3/ipsec_atodata.3
> /usr/man/man3/ipsec_atobytes.3
> /usr/man/man3/ipsec_bytestoa.3
> /usr/man/man3/ipsec_datatoa.3
> /usr/man/man3/ipsec_atosa.3
> /usr/man/man3/ipsec_satoa.3
> /usr/man/man3/ipsec_atoul.3
> /usr/man/man3/ipsec_ultoa.3
> /usr/man/man3/ipsec_goodmask.3
> /usr/man/man3/ipsec_masktobits.3
> /usr/man/man3/ipsec_bitstomask.3
> /usr/man/man3/ipsec_optionsfrom.3
> /usr/man/man3/ipsec_subnetof.3
> /usr/man/man3/ipsec_hostof.3
> /usr/man/man3/ipsec_broadcastof.3
> /usr/man/man5/ipsec.secrets.5
> /usr/man/man5/ipsec.conf.5
> /usr/man/man8/ipsec_spi.8
> /usr/man/man8/ipsec.8
> /usr/man/man8/ipsec_eroute.8
> /usr/man/man8/ipsec_spigrp.8
> /usr/man/man8/ipsec_tncfg.8
> /usr/man/man8/ipsec_klipsdebug.8
> /usr/man/man8/ipsec_pluto.8
> /usr/man/man8/ipsec_whack.8
> /usr/man/man8/ipsec_barf.8
> /usr/man/man8/ipsec_look.8
> /usr/man/man8/ipsec_manual.8
> /usr/man/man8/ipsec_auto.8
> /usr/man/man8/ipsec_setup.8
> /usr/man/man8/ipsec_ranbits.8
> /usr/man/man8/ipsec_rsasigkey.8
> /usr/sbin/ipsec

Part XII Other Server Related Reference

In this Part

Other Server - `Wu-ftpd` FTP Server
Other Server - `Apache` Web Server
Other Server - `Samba` File Sharing Server

This part of the book will exclusively deal with three important programs in the Unix world. These programs are the most used on server environment and run since many years ago on the Internet. Most of us usually use one of them every time to surf on the Internet, transfer file between computers on the Internet or internally via `LAN` in the enterprises.

28 Other Server - Wu-ftpd FTP Server

In this Chapter

Linux Wu-ftpd FTP Server

Abstract

Despite its age, using the File Transfer Protocol (FTP) is one of the most popular way to transfer files from machine to machine across a network. Clients and servers have been written for each of the popular platforms on the market, thereby making FTP the most convenient way to perform file transfers.

Many different ways exist to configure your FTP servers. One is as a private user-only site, which is the default configuration for an FTP server; a private FTP server allows users on the Linux system only to be able to connect via FTP and access their files.

Other kinds exist, like the anonymous FTP server. An anonymous FTP server allows anyone on the network to connect to it and transfer files without having an account. Due to the potential security risk involved with this setup, precautions should be taken to allow access only to certain directories on the system.

The configuration we will cover here is an FTP server that allows FTP to semi-secure areas of a Unix file system (chroot'd Guest FTP access). This configuration allows users to have access to the FTP server directories without allowing them to get into higher levels. This is the most secure setup for an FTP server and it is a useful way for remote clients to maintain their Web accounts.

The steps I describe in this chapter allow you to setup any of the three types of FTP server available. At the end of this tutorial you'll find a section about anonymous FTP configuration.

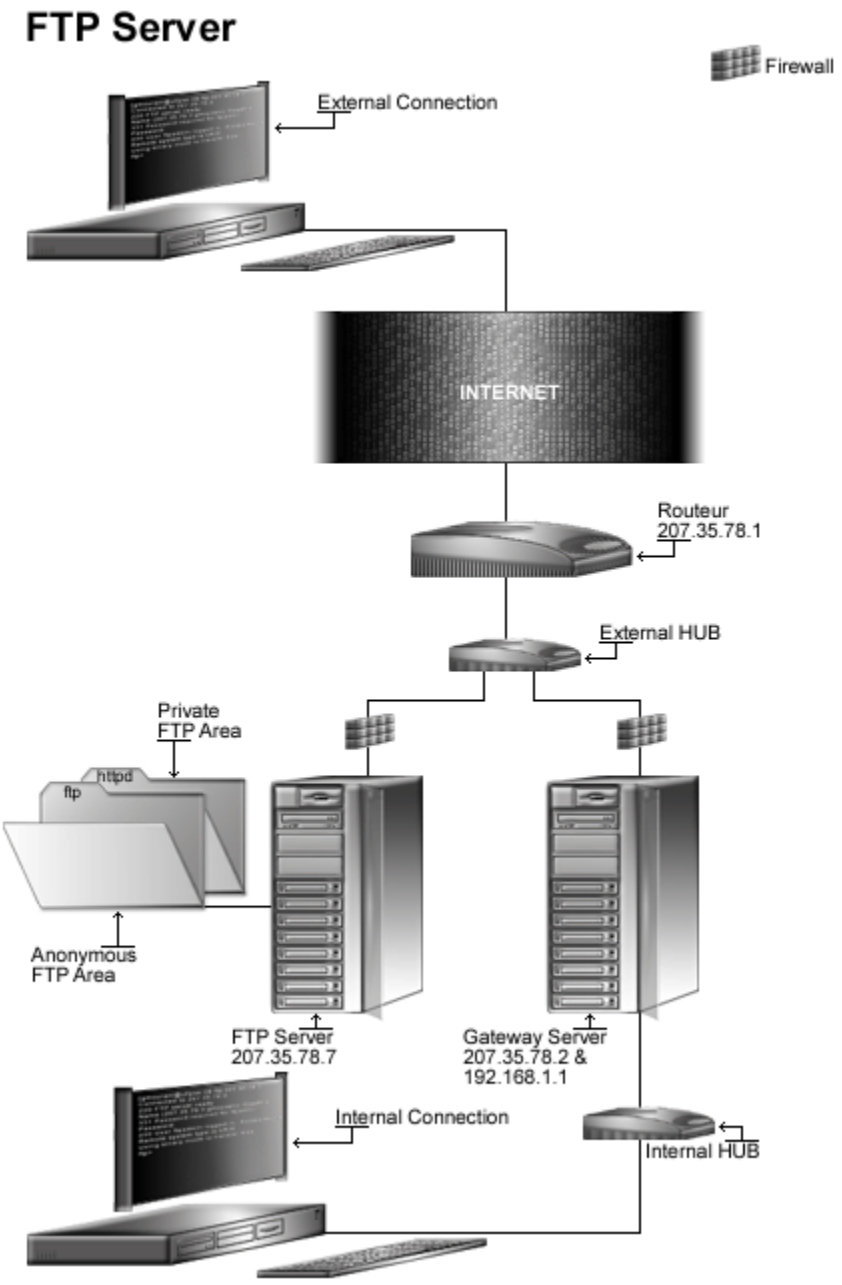
Recommended RPM packages to be installed for a FTP Server

A minimal configuration provides the basic set of packages required by the Linux operating system. Minimal configuration is a perfect starting point for building secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a FTP Server (FTP) running on wu-ftpd software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install wu-ftpd by RPM package. Therefore, wu-ftpd RPM package is already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

basesystem	ed	less	passwd	SysVinit
bash	file	libstdc++	popt	tar
bdflush	filesystem	libtermcap	procps	termcap
bind	fileutils	lilo	psmisc	textutils
bzip2	findutils	logrotate	pwdb	tmpwatch
chkconfig	gawk	losetup	**quota**	utempter
console-tools	gdbm	MAKEDEV	qmail	util-linux
cpio	gettext	man	readline	vim-common
cracklib	glib	mingetty	rootfiles	vim-minimal
cracklib-dicts	glibc	mktemp	rpm	vixie-cron
crontabs	glibc-common	mount	sed	words
db1	grep	ncurses	setup	which
db2	groff	net-tools	sh-utils	**wu-ftpd**
db3	gzip	newt	shadow-utils	zlib
dev	info	openssh	slang	
devfsd	initscripts	openssh-server	slocate	
diffutils	iptables	openssl	sysklogd	
e2fsprogs	kernel	pam	syslinux	

Tested and fully functional on OpenNA.com.


FTP Server
Firewall
External Connection
INTERNET
Routeur
207.35.78.1
External HUB
Private
FTP Area
httpd
ftp
Anonymous
FTP Area
FTP Server
207.35.78.7
Gateway Server
207.35.78.2 &
192.168.1.1
Internal Connection
Internal HUB

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest Wu-ftpd version number is 2.6.1

Packages

The following are based on information as listed by Wu-ftpd as of 2001/03/16. Please regularly check at www.wu-ftpd.org for the latest status.

Source code is available from:
Wu-ftpd Homepage: http://www.wu-ftpd.org/
Wu-ftpd FTP Site: 205.133.13.68
You must be sure to download: wu-ftpd-2.6.1.tar.gz

Pristine source

If you don’t use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Wu-ftpd, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > Wu-ftpd1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > Wu-ftpd2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff Wu-ftpd1 Wu-ftpd2 > Wu-ftpd-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing Wu-ftpd

Below are the required steps that you must make to configure, compile and optimize the Wu-ftpd software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
  ```
  [root@deep /]# cp wu-ftpd-version.tar.gz /var/tmp/
  [root@deep /]# cd /var/tmp/
  [root@deep tmp]# tar xzpf wu-ftpd-version.tar.gz
  ```

Step 2
After that, move into the newly created Wu-ftpd directory.

- To move into the newly created Wu-ftpd directory use the following command:
  [root@deep tmp]# **cd wu-ftpd-2.6.1/**

Step 3
Now it is time to configure the software for our system in the most secure and optimized manner available. As you will be notified in many documentation files into the Wu-ftpd source directory, beginning with version 2.6.0 of Wu-ftpd, the WU-FTPD Development Group is moving the build process to use GNU Autoconf.

At this time and because for many platforms, the autoconf build is experimental, I recommend to try first ./configure and if that fails try the old method ./build.

- To configure Wu-ftpd with recommended securities and speed, use the compile lines:

```
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
./configure \
--prefix=/usr \
--sysconfdir=/etc \
--localstatedir=/var \
--mandir=/usr/share/man \
--enable-quota \
--enable-ratios \
--disable-rfc931 \
--disable-logtoomany \
--disable-plsm
```

This tells Wu-ftpd to set itself up for this particular configuration setup with:

- Add QUOTA mechanisms support (if your OS supports it in the kernel).
- Compile in support for upload-download ratios.
- Do not do RFC931 lookups to be faster.
- Do not log failed attempts (for busy servers to prevent to fill up the log file and put high load on syslog).
- Disable PID lock sleep messages causing the daemon to sleep (for busy sites only).

WARNING: Pay special attention to the compile CFLAGS line above. We optimize Wu-ftpd for an i686 CPU architecture with the parameter "-march=i686 and -mcpu=i686". Please don't forget to adjust this CFLAGS line to reflect your own system.

Step 4
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install Wu-ftpd in the server:

```
[root@deep wu-ftpd-2.6.1]# make
[root@deep wu-ftpd-2.6.1]# cd
[root@deep /root]# find /* > Wu-ftpd1
[root@deep /root]# cd /var/tmp/wu-ftpd-2.6.1/
[root@deep wu-ftpd-2.6.1]# make install
[root@deep wu-ftpd-2.6.1]# install –m100 util/xferstats /usr/sbin/
[root@deep wu-ftpd-2.6.1]# touch /var/log/xferlog
[root@deep wu-ftpd-2.6.1]# chmod 600 /var/log/xferlog
[root@deep wu-ftpd-2.6.1]# cd /usr/sbin/
[root@deep sbin]# ln -sf in.ftpd wu.ftp
[root@deep sbin]# ln -sf in.ftpd in.wuftp
[root@deep sbin]# cd
[root@deep /root]# find /* > Wu-ftpd2
[root@deep /root]# diff Wu-ftpd1 Wu-ftpd2 > Wu-ftpd-Installed
```

The **install -m** command will install the binary xferstats used to see static information about transferred files, and the **touch** command will create the log file for xferstats under /var/log directory in the system. The **chmod** will change the mode of xferlog file to be readable and writable only by the owner.

Step 5
After that, we will change some default properties of Wu-ftpd binaries to be more restritive and more secure.

```
[root@deep /]# chmod 100 /usr/bin/ftpcount
[root@deep /]# chmod 100 /usr/bin/ftpwho
[root@deep /]# chmod 100 /usr/sbin/ftprestart
[root@deep /]# chmod 100 /usr/sbin/ftpshut
[root@deep /]# chmod 100 /usr/sbin/privatepw
[root@deep /]# chmod 110 /usr/sbin/in.ftpd
[root@deep /]# chown bin.bin /usr/bin/ftpcount
[root@deep /]# chown bin.bin /usr/bin/ftpwho
[root@deep /]# chown bin.bin /usr/sbin/ftprestart
[root@deep /]# chown bin.bin /usr/sbin/ftpshut
[root@deep /]# chown bin.bin /usr/sbin/privatepw
[root@deep /]# chown bin.bin /usr/sbin/in.ftpd
[root@deep /]# chown bin.bin /usr/sbin/wu.ftp
[root@deep /]# chown bin.bin /usr/sbin/in.wuftp
[root@deep /]# chown bin.bin /var/log/xferlog
```

Step 6
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Wu-ftpd and its related source directory, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf wu-ftpd-version/
[root@deep tmp]# rm -f wu-ftpd-version.tar.gz
```

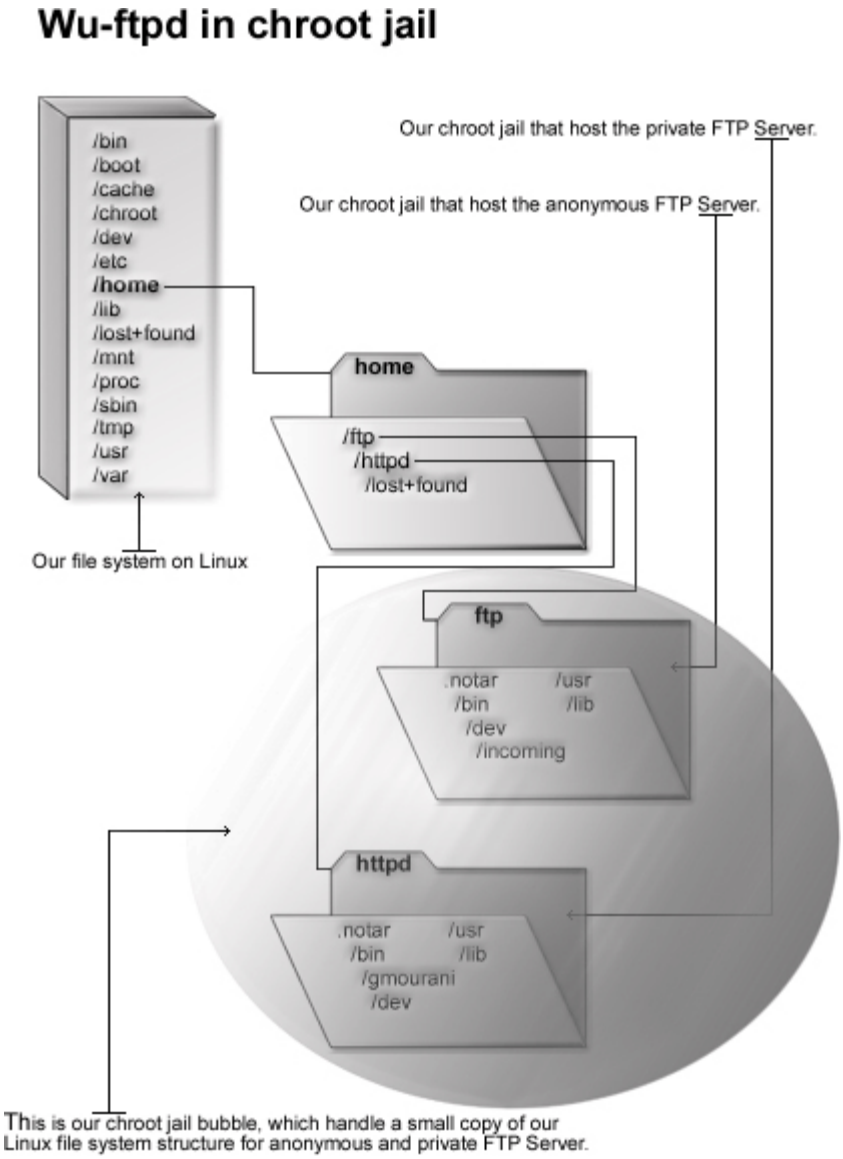
The `rm` command as used above will remove all the source files we have used to compile and install `Wu-ftpd`. It will also remove the `Wu-ftpd` compressed archive from the `/var/tmp` directory.

Running `Wu-ftpd` in a chroot jail

This part focuses on preventing `Wu-ftpd` from being used as a point of break-in to the system hosting it. The potential for bugs that affect security is rather high with this software therefore an additional step can be taken - that is, **running `Wu-ftpd` in a chroot jail**.

The main benefit of a chroot jail is that the jail will limit the portion of the file system the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support `Wu-ftpd`, the programs available in the jail can be extremely limited.

Most importantly, there is no need for setuid-root programs, which can be used to gain root access and break out of the jail. By running `Wu-ftpd` in a chroot jail you can improve the security significantly in a Unix environment.


Wu-ftpd in chroot jail
/bin
/boot
/cache
/chroot
/dev
/etc
/home
/lib
/lost+found
/mnt
/proc
/sbin
/tmp
/usr
/var
Our file system on Linux
Our chroot jail that host the private FTP Server.
Our chroot jail that host the anonymous FTP Server.
home
/ftp
/httpd
/lost+found
ftp
.notar
/usr
/bin
/lib
/dev
/incoming
httpd
.notar
/usr
/bin
/lib
/gmourani
/dev
This is our chroot jail bubble, which handle a small copy of our
Linux file system structure for anonymous and private FTP Server.

Necessary steps to run `Wu-ftpd` in a chroot jail:

What you're essentially doing is creating a skeleton root file system with enough components necessary (binaries, libraries, etc.) to allow Unix to do a chroot when the user logs in.

Step 1
It's important to give to your strictly `FTP` users no real shell account on the Linux system. In this manner, if for any reasons someone could successfully get out of the `FTP` chrooted environment, it would not have the possibility of using a shell to gain access via other protocols like `telnel`, `ssh`, etc.

First, create new users for this purpose; these users will be the users allowed to connect to your `FTP` server. This has to be separate from a regular user account with unlimited access because of how the "chroot" environment works. Chroot makes it appear from the user's perspective as if the level of the file system you've placed them in is the top level of the file system.

- Use the following command to create users in the `/etc/passwd` file. This step must be done for each additional new user you allow to access your `FTP` server.

```
[root@deep /]# useradd -d /home/httpd/gmourani -s /bin/false gmourani
2>/dev/null || :

[root@deep /]# passwd gmourani
Changing password for user gmourani
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The `useradd` command will add the new `guest` user named `gmourani` to our Linux server and will set its home directory to be located under `/home/httpd/gmourani` directory since it is a useful location for remote clients to maintain their Web accounts. Finally, the `passwd` command will set the password for this user `gmourani`.

Step 2
Now, edit the **`shells`** file (`vi /etc/shells`) and add a non-existent shell name like "`/bin/false`", which is the one we used in the `passwd` command above.

```
[root@deep /]# vi /etc/shells
/bin/bash2
/bin/bash
/bin/sh
/bin/false ← This is our added no-existent shell
```

Step 3
Then, create all the necessary chrooted environment directories as shown below:

```
[root@deep /]# mkdir /home/httpd/bin
[root@deep /]# mkdir /home/httpd/dev
[root@deep /]# mkdir /home/httpd/lib
[root@deep /]# mkdir /home/httpd/usr
[root@deep /]# mkdir /home/httpd/usr/bin
```

Step 4
After that, we must change the mode permission on the chroot glue directories to mode (0111/d--x--x--x) for security reasons:

```
[root@deep /]# chmod 0111 /home/httpd/bin/
[root@deep /]# chmod 0111 /home/httpd/dev/
[root@deep /]# chmod 0111 /home/httpd/lib/
[root@deep /]# chmod 0111 /home/httpd/usr/
[root@deep /]# chmod 0111 /home/httpd/usr/bin/
```

Step 5
Once all permission modes of the supporting glues have been changed, it is time to copy the require binaries programs to the related directories in the chroot area for Wu-ftpd to work. Those programs are necessary to allow guest users to chmod, ls, tar, compress, and rename files on the FTP chroot jail server. If there are features you don't want any users to be able to use, then don't copy them to the chroot area.

```
[root@deep /]# cp /bin/ls /home/httpd/bin/
[root@deep /]# cp /bin/tar /home/httpd/bin/
[root@deep /]# cp /bin/chmod /home/httpd/bin/
[root@deep /]# cp /bin/cpio /home/httpd/bin/
[root@deep /]# cp /bin/gzip /home/httpd/bin/
[root@deep /]# cp /usr/bin/rename /home/httpd/usr/bin/
[root@deep /]# chmod 0111 /home/httpd/bin/*
[root@deep /]# chmod 0111 /home/httpd/usr/bin/*
[root@deep /]# cd /home/httpd/bin/
[root@deep /]# ln -sf gzip zcat
```

NOTE: The chmod commands above will change modes of those programs to be (0111 ---x--x–x) because we don't want users to be able to modify or read the binaries in the chroot area but just to execute them if necessary.

Step 6
The binaries we have copied to the chroot area have been compiled with shared libraries by default and for this reason it is important to find the shared libraries dependencies associated with them and copy them into the "lib" directory in the chroot jail area that we have created hearly during our steps.

As usually, to find the shared library dependencies of binaries, you have to use the ldd command of Linux. Because we have installed the most important FTP features, you must copy all the libraries below to the /home/httpd/lib directory of the chroot area. These libraries are part of libc, and needed by various programs in bin.

```
[root@deep /]# cp /lib/libcrypt.so.1 /home/httpd/lib/
[root@deep /]# cp /lib/libnsl.so.1 /home/httpd/lib/
[root@deep /]# cp /lib/libresolv.so.2 /home/httpd/lib/
[root@deep /]# cp /lib/libc.so.6 /home/httpd/lib/
[root@deep /]# cp /lib/ld-linux.so.2 /home/httpd/lib/
[root@deep /]# cp /lib/libtermcap.so.2 /home/httpd/lib/
[root@deep /]# cp /lib/libpthread.so.0 /home/httpd/lib/
[root@deep /]# cp /lib/librt.so.1 /home/httpd/lib/
[root@deep /]# strip -R .comment /home/httpd/lib/*
```

WARNING: Depending of what you have compiled with the program the required shared libraries may be more or different then the one as illustrated above. Please use the **ldd** command on each binary under /bin directory to find out the ones you need and copy them to the /lib directory of the chroot area.

The "strip -R .comment" command will remove all the named section ".comment" from the libraries files under the /lib directory and will make them smaller in size and can help in performance of them.

Step 7
Finally, create the /home/httpd/dev/null file and set its mode appropriately.

```
[root@deep /]# mknod /home/httpd/dev/null c 1 3
[root@deep /]# chmod 666 /home/httpd/dev/null
```

Configuring Wu-ftpd

After building Wu-ftpd, and all the require chroot glues environment and users, your next step is to verify or change, if necessary options in your Wu-ftpd configuration files. Those files are:

- /etc/ftpaccess (The Wu-ftpd Configuration File)
- /etc/ftphosts (The Wu-ftpd Hosts Configuration File)
- /etc/ftpconversion (The Wu-ftpd Compress Configuration File)
- /etc/logrotate.d/ftpd (The Wu-ftpd Log Rotation File)
- /etc/rc.d/init.d/ftpd (The Wu-ftpd Initialization File)

/etc/ftpaccess: The Wu-ftpd Configuration File

The /etc/ftpaccess file is the main configuration file used to configure the operation of the Wu-ftpd server. This file is the primary means of controlling what users, and how many users, can access your server, and other important points of the security configuration.

Each line in the file either defines an attribute or sets its value. Whatever you want to configure an anonymous FTP, private or Guest FTP server, this is the file that you must understand and configure. We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

Step 1

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following lines:

```
class   openna   guest  207.35.78.*

email admin@openna.com

limit openna 20 MoTuWeTh,Fr0000-1800 /.too_many.msg
loginfails 3

readme  README*    login
readme  README*    cwd=*
```

```
message /welcome.msg            login
message .message                cwd=*

compress        yes             all
tar             yes             all
chmod           no              anonymous
delete          no              anonymous
overwrite       no              anonymous
rename          no              anonymous

log transfers anonymous,real,guest inbound,outbound

# Specify which group of users will be treated as "guests".
guestuser *

# We don't want users being able to upload into these areas.
upload /home/httpd *           no
upload /home/httpd * /dev      no
upload /home/httpd * /bin      no
upload /home/httpd * /lib      no
upload /home/httpd * /usr      no
upload /home/httpd * /usr/bin  no

# Areas where upload clauses are allowed.
upload /home/httpd /gmourani yes gmourani gmourani 0644 dirs 0755
upload /home/httpd /gmourani/* yes gmourani gmourani 0644 dirs 0755

# We'll prevent downloads with noretrieve.
noretrieve /home/httpd/dev/
noretrieve /home/httpd/bin/
noretrieve /home/httpd/lib/
noretrieve /home/httpd/usr/
noretrieve /home/httpd/usr/bin/

log security anonymous,real,guest

guest-root /home/httpd gmourani
restricted-uid gmourani
restricted-gid gmourani

deny-uid %-99 %65535-
deby-gid %-99 %65535-

greeting terse
keepalive yes

passwd-check rfc822 warn
```

Step 2
Now, change its default permission to be (`0600/-rw-------`).

```
[root@deep /]# chmod 600 /etc/ftpaccess
```

This tells `ftpaccess` file to set itself up for this particular configuration setup with:

```
class   openna   guest  207.35.78.*
```
This option "class" specifies a class of users who can access the FTP server. You can define as many classes as you want in the ftpaccess file. In our example, we define the class name <openna>, and we allow only guest user <guest> with accounts on the FTP server to access their home directories via FTP if they are coming from the address 207.35.78.*.

It's important to note that three different kinds of users exist: **anonymous, guest,** and **real**. **Anonymous** users are anyone on the network who connect to the server and transfer files without having an account on it. **Guest** users are real users on the system for which their session is set up exactly as with anonymous FTP (this is the one we setup in our example), and **Real** users must have accounts and shells (this can pose a security risk) on the server to be able to access it.

```
limit openna 20 MoTuWeTh,Fr0000-1800 /.too_many.msg
```
This option "limit" specifies the number of users allowed to log in to the FTP server by class and time of day. In our example, we limit access to the FTP server for the class name <openna> to 20 users <20> from Monday through Thursday <MoTuWeTh>, all day, and Friday from midnight to 6:00 p.m <Fr0000-1800>.

Also, if the limit of 20 users is reached, the content of the file </.too_many.msg> is displayed to the connecting user. This can be a useful parameter when you need to control the resources of your server. Finally, it's important for security reason that the message file /.too_many.msg should be in someplace safety outside the chroot area.

```
loginfails 3
```
This option "loginfails" specifies the number of failed login attempts connection clients can make before being disconnected. In our example we disconnect a user from the FTP server after three failed attempts.

```
readme  README*    login
readme  README*    cwd=*
```
This option "readme" specifies to notify clients at login time, or upon using the change working directory command, that a certain file in their current directory was last modified. In our example, we set the name of the file to be relative to the FTP directory <README*>, and the condition under which to display the message to be either displayed upon a successful login <login> or displayed when a client enters the new default directory <cwd=*>.

```
message /welcome.msg            login
message .message                cwd=*
```
This option "message" specifies to display special messages to the client when they either log in, or upon using the change working directory command. In our example, we indicate the location and the name of the files to be displayed <welcome.msg or .message>, and the condition under which to display the files to be either displayed upon a successful login <login>, or displayed when a client enters a new directory <cwd=*>.

For the readme and message options above, remember that when you're specifying a path for anonymous users, the path must be relative to the anonymous FTP directory.

```
compress        yes             all
tar             yes             all
chmod           no              anonymous
delete          no              anonymous
overwrite       no              anonymous
rename          no              anonymous
```

These options, "compress", "tar", "chmod", "delete", "overwrite", and "rename", specify the permissions that you want to give to your users for these commands. In our example, we do not give permission to the anonymous users <anonymous> to chmod, delete, overwrite, and rename files, and allow everybody to use compress and tar commands <all>. If you don't specify the following directives, they default to "yes" for everybody. This is a security feature.

```
log transfers anonymous,real,guest inbound,outbound
```

This option "log transfers" specifies to log all FTP transfers for security purposes. In our example, we log all **anonymous**, **real** and **guest** users transfers <anonymous,real,guest> that are both **inbound** and **outbound** <inbound,outbound> which specify the direction that the transfers must take in order to be logged. The resulting logs are stored in the /var/log/xferlog file.

```
guestuser *
```

This option "guestuser" specifies all of your guest user names (or numeric ID) that are real users on the system, in which the session is set up exactly as with anonymous FTP. You can also use a wildcards (*) as a value to specify all guest user names in the system (as we do). If you prefer to add each user name, it's important that any additional guestuser you may add appears one per line in the configuration file.

Finally, it's appearing that guestgroup isn't the best way to make a user a guest. If you forget to explicitly add the user in /etc/group the user isn't a guest and it's for this reason that guestuser is recommended instead of guestgroup parameter. This is a security feature.

```
log security anonymous,real,guest
```

This option "log security" specifies to enable logging of violations of security rules for anonymous, real, and guest FTP clients. In our example, we specify to log violations for users using the FTP server to access anonymous accounts, real accounts, and for users using the FTP server to access guest accounts <anonymous,real,guest>. This is a security feature.

```
guest-root /home/httpd gmourani
restricted-uid gmourani
restricted-gid gmourani
```

These clauses "guest-root", "restricted-uid", and "restricted-gid" specify and control whether or not **guest** users will be allowed access to areas on the FTP server outside their home directories (this is an important security feature). In our example, we specified the chroot() path for user <gmourani> to be </home/httpd>, and that it cannot access other's files because it is restricted to his home directories <restricted-uid gmourani>, <restricted-gid gmourani>.

Multiple UID ranges may be given on the line. If a guest-root is chosen for the user, the user's home directory in the /etc/passwd file is used to determine the initial directory, and their home directory, in the system-wide /etc/passwd, is not used. This is a security feature.

```
deny-uid %-99 %65535-
deby-gid %-99 %65535-
```

These clauses allow specification of UID and GID values, which will be denied access to the FTP server. To summarize it ensures no login from privileged accounts on a Linux machine and in many cases, this can eliminate the need for the /etc/ftpusers file. If you want to allow anonymous FTP, then add the following two: allow-uid ftp and allow-gid ftp. This is a security feature.

```
greeting terse
```

This option "greeting" specifies how much system information will be displayed before the remote user logs in. There are three parameters you can choose: <full> is the default and shows the hostname and daemon version of the server, <brief> which shows only the hostname, and <terse>, which will simply says "FTP server ready" to your terminal. This is a security feature.

```
keepalive yes
```

This option "keepalive" specifies whether the system should send keep alive messages to the remote FTP server. If set to "yes", then death of the connection or crash of remote machines will be properly noticed.

/etc/ftphosts: The Wu-ftpd Hosts Configuration File

This file is used to define whether users are allowed to log in from certain hosts or whether there are denied access.

Step 1

- Create the **ftphosts** file (touch /etc/ftphosts) and add for example in this file the following lines:

```
# Host access configuration file
#
# Everything after a '#' is treated as comment,
# empty lines are ignored
allow ftpadmin 207.35.78.1 207.35.78.2 207.35.78.4
deny ftpadmin 207.35.78.5
```

In the above example, we allow the user <ftpadmin> to connect via FTP from the explicitly listed addresses <207.35.78.1 207.35.78.2 207.35.78.4>, and deny the specified <ftpadmin> user to connect from the site <207.35.78.5>.

Step 2

- Now, change its default permission to be (0600/-rw-------):

```
[root@deep /]# chmod 600 /etc/ftphosts
```

/etc/ftpconversions: The Wu-ftpd Compress Configuration File

This file contains instructions that permit you to compress files on demand before the transfer.

Step 1

- Edit the **ftpconversions** file (`vi /etc/ftpconversions`) and add or verify in this file the following lines:

```
 :.Z:  :  :/bin/compress -d -c %s:T_REG|T_ASCII:O_UNCOMPRESS:UNCOMPRESS
 :   : :.Z:/bin/compress -c %s:T_REG:O_COMPRESS:COMPRESS
 :.gz: :  :/bin/gzip -cd %s:T_REG|T_ASCII:O_UNCOMPRESS:GUNZIP
 :   : :.gz:/bin/gzip -9 -c %s:T_REG:O_COMPRESS:GZIP
 :   : :.tar:/bin/tar -c -f - %s:T_REG|T_DIR:O_TAR:TAR
 :   : :.tar.Z:/bin/tar -c -Z -f -
%s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+COMPRESS
 :   : :.tar.gz:/bin/tar -c -z -f -
%s:T_REG|T_DIR:O_COMPRESS|O_TAR:TAR+GZIP
 :   : :.crc:/bin/cksum %s:T_REG::CKSUM
 :   : :.md5:/bin/md5sum %s:T_REG::MD5SUM
```

Step 2

- Now, change its default permissions to be (0600/-rw-------):

```
[root@deep /]# chmod 600 /etc/ftpconversions
```

/etc/logrotate.d/ftpd: The Wu-ftpd Log Rotation File

Configure your `/etc/logrotate.d/ftpd` file to automatically rotate your log files each week.

- Create the **ftpd** file (`touch /etc/logrotate.d/ftpd`) and add the following lines:

```
/var/log/xferlog {
    # ftpd doesn't handle SIGHUP properly
    nocompress
}
```

/etc/rc.d/init.d/ftpd: The Wu-ftpd Initialization File

The `/etc/rc.d/init.d/ftpd` script file is responsible to automatically start and stop the `Wu-ftpd` daemon on your server. Loading `ftpd` daemon, as a standalone daemon will eliminate load time as well as the `ftpaccess` file load time too and will even reduce swapping since non-library code will be shared.

Step 1
Create the **ftpd** script file (`touch /etc/rc.d/init.d/ftpd`) and add the following lines inside it:

```
#! /bin/sh
#
# ftpd        This starts and stops ftpd.
#
# chkconfig: 345 50 50
# description: Wu-ftpd is one of the most widely \
# used daemons on the Internet. \
#
```

```
# processname: /usr/sbin/in.ftpd
# config: /etc/sysconfig/network
# config: /etc/ftpaccess
# pidfile: /var/run/ftpd.pid

PATH=/sbin:/bin:/usr/bin:/usr/sbin

# Source function library.
. /etc/init.d/functions

# Get config.
test -f /etc/sysconfig/network && . /etc/sysconfig/network

# Check that networking is up.
[ ${NETWORKING} = "yes" ] || exit 0

[ -f /usr/sbin/in.ftpd ] || exit 1
[ -f /etc/ftpaccess ] || exit 1

RETVAL=0

start(){
    echo -n "Starting ftpd: "
    daemon in.ftpd -l -a -S
    RETVAL=$?
    echo
    touch /var/lock/subsys/ftpd
    return $RETVAL
}

stop(){
    echo -n "Stopping ftpd: "
    killproc in.ftpd
    RETVAL=$?
    echo
    rm -f /var/lock/subsys/ftpd
    return $RETVAL

}

reload(){
    echo -n "Reloading ftpd: "
    killproc in.ftpd -USR2
    RETVAL=$?
    echo
    return $RETVAL
}

restart(){
    stop
    start
}

condrestart(){
    [ -e /var/lock/subsys/ftpd ] && restart
    return 0
}


# See how we were called.
case "$1" in
    start)
        start
```

```
        ;;
    stop)
        stop
        ;;
    status)
        status in.ftpd
        ;;
    restart)
        restart
        ;;
    reload)
        reload
        ;;
    condrestart)
        condrestart
        ;;
    *)
        echo "Usage: ftpd {start|stop|status|restart|condrestart|reload}"
        RETVAL=1
esac

exit $RETVAL
```

Step 2
Once the `ftpd` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/rc.d/init.d/ftpd
  [root@deep /]# chown 0.0 /etc/rc.d/init.d/ftpd
  ```

- To create the symbolic `rc.d` links for `Wu-ftpd`, use the following commands:
  ```
  [root@deep /]# chkconfig --add ftpd
  [root@deep /]# chkconfig --level 345 ftpd on
  ```

- To start `Wu-ftpd` software manually, use the following command:
  ```
  [root@deep /]# /etc/rc.d/init.d/ftpd start
  Starting ftpd:                [OK]
  ```

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Securing Wu-ftpd

This section deals especially with actions we can make to improve and tighten security under Wu-ftpd. Note that we refer to the features available within the base installed program and not to any additional software.

The upload command

By default, the Wu-ftpd server will grant upload privileges to all users. The upload parameter allow remote clients to load and place files on the FTP server. For optimal security, we don't want users being able to upload into "/", "/bin", "/dev", "/lib", "/usr", and "/usr/bin", directories in the /home/httpd/ chrooted directory.

In our /etc/ftpaccess file we have already chroot'd users to /home/httpd, and they cannot access any area of the file system outside that directory structure, but in case something happens to the permissions on them you should deny upload privileges in your /etc/ftpaccess file into these areas (/home/httpd, /home/httpd/bin, /home/httpd/dev, /home/httpd/lib, /home/httpd/usr, and /home/httpd/usr/bin) then allowing only what we want to:

Step 1

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following lines to deny upload privileges into these areas.

```
# We don't want users being able to upload into these areas.
upload /home/httpd *           no
upload /home/httpd * /dev      no
upload /home/httpd * /bin      no
upload /home/httpd * /lib      no
upload /home/httpd * /usr      no
upload /home/httpd * /usr/bin  no

# Areas where upload clauses are allowed.
upload /home/httpd /gmourani yes gmourani gmourani 0644 dirs 0755
upload /home/httpd /gmourani/* yes gmourani gmourani 0644 dirs 0755
```

The above lines specify to deny upload feature into the "/", "/dev", "/bin", "/lib", "/usr", and "/usr/bin" directories of the chroot'd "/home/httpd" directory structure. If you have other directories in the chroot area that you want to protect with the upload clause, then add them to the list.

The last line in our example, allow uploads into the directory and one subdirectories of /gmourani area with permission files set to 644 and the creation of new directories with permission set to 755 for guest user and group named gmourani.

WARNING: If you want to allow upload into more than one subdirectory of /gmourani area, then you will have to add a new lines for each additional allowed subdirectories. For example if I want to allow upload into /gmourani/folder1/folder2/folder3, I will add the following additional lines:

```
# Areas where upload clauses are allowed.
upload /home/httpd /gmourani yes gmourani gmourani 0644 dirs 0755
upload /home/httpd /gmourani/* yes gmourani gmourani 0644 dirs 0755
upload /home/httpd /gmourani/*/* yes gmourani gmourani 0644 dirs 0755
upload /home/httpd /gmourani/*/*/* yes gmourani gmourani 0644 dirs 0755
```

Step 2
Restart the `Wu-ftpd` server for the changes to take effect.

- To restart `Wu-ftpd`, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/ftpd restart
Stopping ftpd:            [OK]
Starting ftpd:            [OK]
```

The special file `.notar`

Whether you allow on-the-fly tarring of directories or not, you should make sure an end-run cannot be made using `tar` command in all areas where the `upload` parameter is not permit.

Step 1
To do so, create the special file `.notar` in each directory and in the `FTP` directory. Don't use the `touch` command to create `.notar`. Use the `echo`, as in:

```
[root@deep /]# echo "Tarring is denied" > /home/httpd/.notar
[root@deep /]# echo "Tarring is denied" > /home/httpd/dev/.notar
[root@deep /]# echo "Tarring is denied" > /home/httpd/bin/.notar
[root@deep /]# echo "Tarring is denied" > /home/httpd/lib/.notar
[root@deep /]# echo "Tarring is denied" > /home/httpd/usr/.notar
[root@deep /]# echo "Tarring is denied" > /home/httpd/usr/bin/.notar
```

WARNING: Don't forget to add in this list any additional directories where the `upload` parameter is not allowed into the `FTP` area. Also don't forget to create the `.notar` file inside it.

Step 2
It's appear that using the "`noretrieve .notar`" paramater in the `/etc/ftpaccess` file breaks IE. Some mirrors will copy your `.notar` rather than detect and create, so you'll want it readable and retrievable.

```
[root@deep /]# chmod 0444 /home/httpd/.notar
[root@deep /]# chmod 0444 /home/httpd/dev/.notar
[root@deep /]# chmod 0444 /home/httpd/bin/.notar
[root@deep /]# chmod 0444 /home/httpd/lib/.notar
[root@deep /]# chmod 0444 /home/httpd/usr/.notar
[root@deep /]# chmod 0444 /home/httpd/usr/bin/.notar
```

Step 3
Restart the `Wu-ftpd` server for the changes to take effect.

- To restart `Wu-ftpd`, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/ftpd restart
Stopping ftpd:            [OK]
Starting ftpd:            [OK]
```

WARNING: It's important to NOT add the "`noretrieve .notar`" parameter inside the `/etc/ftpaccess` file of `Wu-ftpd` or any `.notar` files will not be retrievable.

The `noretrieve` command

The `noretrieve` parameter of `Wu-ftpd` server allows you to deny transfer of the sectected directories or files. It is also a good idea to prevent downloads of those subdirectories (`/dev`, `/bin`, `/lib`, `/usr`, and `/usr/bin`) in the `/home/httpd` directory with the command `noretrieve` in your `/etc/ftpaccess` file.

Step 1

- Edit the **`ftpaccess`** file (`vi /etc/ftpaccess`) and add the following lines to deny transfer into these areas.

```
# We'll prevent downloads with noretrieve.
noretrieve /home/httpd/dev/
noretrieve /home/httpd/bin/
noretrieve /home/httpd/lib/
noretrieve /home/httpd/usr/
noretrieve /home/httpd/usr/bin/
```

Step 2

Restart the `Wu-ftpd` server for the changes to take effect.

- To restart `Wu-ftpd`, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/ftpd restart
Stopping ftpd:            [OK]
Starting ftpd:            [OK]
```

NOTE: If you have others directories into the chroot area which you want to deny transfer of files inside them, you must add theses directories to the `noretrieve` clause list above.

Setup an `Anonymous FTP` server

For administrators who want to setup an `anonymous FTP` server, all you have to do is just to add the `FTP anonymous` user into the `/etc/passwd` file and setup the appropriate, parameters and authorizations. Of course, don't forget to add the new require directories of the `anonymous` server to the chroot'd jail.

In our example we'll first give `anonymous` users only an access to get files from the `FTP anonymous` directory on the `FTP` server. Why? Because if we want to give them the possibility to be able to `upload` contents to the `anonymous` server area, we should create a special separate file system to receive their uploaded files or be prepared to some **D**enial **o**f **S**ervice (`DoS`) attack into your system.

If you want to allow `upload` into the `FTP anonymous` server area, then create a new file system like for example `/home/ftp/incoming` for this purpose and mount it. In this case, please refer to the indicated steps later.

Step 1
First, we must add a new user to the /etc/passwd file for the anonymous FTP connection. Pay special attention to the UID number used for this user. That is, a user with UID lower than value of UID_MIN defined in the /etc/login.defs.

The directory where we'll setup the anonymous users areas will be separate from the one we have created for the guest users areas.

- To create the anonymous FTP user, use the following command:
  ```
  [root@deep /]# useradd -c "FTP Anonymous Users" -u 95 -d /home/ftp -s
  /bin/false ftp 2>/dev/null || :
  ```

WARNING: Don't create a password for this user, it's an anonymous user, therefore every one should be able to log on with this account.

Step 2
Secondly, it's important to change the owner of the /home/ftp directory to by someone like the bin user of the Linux system, but never the super-user root or the anonymous user.

- To change the owner of the /ftp directory, use the following commands:
  ```
  [root@deep /]# chown bin /home/ftp/
  [root@deep /]# chmod 755 /home/ftp/
  ```

The above command will change the owner of the /home/ftp directory to become the user named "bin". It's important to check and be sure that every added directories under the /home/ftp chroot'd area are owns by user like "bin" but not "ftp" user.

Without this verification, anonymous users will be able to delete, chmod, create directories, etc inside the chroot jail of the anonymous FTP server (very dangerous).

Step 3
After that it's important to edit the /etc/ftpaccess file to inform the FTP server to allow the anonymous user named ftp which we have added to the password file to connect to the server.

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following lines:

  ```
  allow-uid ftp
  allow-gid ftp
  ```

Step 4
Once the anonymous user has been added to the password file and allowed to connect, it's time to create a new defined FTP users class line inside the /etc/ftpaccess file to allows anonymous access.

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following line to allow anonymous FTP users from anywhere:

  ```
  class   anonftp  anonymous *
  ```

Step 5
Now we need to create the chroot'd glues areas for the anonymous FTP users in the system.

```
[root@deep /]# mkdir /home/ftp/bin
[root@deep /]# mkdir /home/ftp/dev
[root@deep /]# mkdir /home/ftp/lib
[root@deep /]# mkdir /home/ftp/usr
[root@deep /]# mkdir /home/ftp/usr/bin
```

Step 6
After that, we must change the mode permission on the chroot glue directories to mode (0111/d--x--x--x) for security reasons:

```
[root@deep /]# chmod 0111 /home/ftp/bin/
[root@deep /]# chmod 0111 /home/ftp/dev/
[root@deep /]# chmod 0111 /home/ftp/lib/
[root@deep /]# chmod 0111 /home/ftp/usr/
[root@deep /]# chmod 0111 /home/ftp/usr/bin/
```

Step 7
Once all permission modes of the supporting anonymous glues have been changed, it is time to copy the require binaries programs to the related directories in the chroot area for Wu-ftpd to work. Those programs are necessary to allow anonymous users to ls, tar, and compress files on the anonymous FTP chroot jail server.

```
[root@deep /]# cp /bin/ls /home/ftp/bin/
[root@deep /]# cp /bin/tar /home/ftp/bin/
[root@deep /]# cp /bin/cpio /home/ftp/bin/
[root@deep /]# cp /bin/gzip /home/ftp/bin/
[root@deep /]# cp /usr/bin/compress /home/ftp/usr/bin/
[root@deep /]# chmod 0111 /home/ftp/bin/*
[root@deep /]# chmod 0111 /home/ftp/usr/bin/compress
[root@deep /]# cd /home/ftp/bin/
[root@deep /]# ln –sf gzip zcat
```

NOTE: The chmod commands above will change modes of those programs to be (0111 ---x--x–x) because we don't want users to be able to modify or read the binaries in the anonymous chroot area but just to execute them if necessary.

Step 8
Find the shared libraries dependencies associated with binaries and copy them to the "lib" directory in the anonymous chroot jail. These libraries are part of libc, and needed by various programs in bin.

```
[root@deep /]# cp /lib/libcrypt.so.1 /home/ftp/lib/
[root@deep /]# cp /lib/libnsl.so.1 /home/ftp/lib/
[root@deep /]# cp /lib/libc.so.6 /home/ftp/lib/
[root@deep /]# cp /lib/libtermcap.so.2 /home/ftp/lib/
[root@deep /]# cp /lib/libnss_files.so.2 /home/ftp/lib/
[root@deep /]# cp /lib/libpthread.so.0 /home/ftp/lib/
[root@deep /]# cp /lib/librt.so.1 /home/ftp/lib/
[root@deep /]# cp /lib/ld-linux.so.2 /home/ftp/lib/
[root@deep /]# strip -R .comment /home/ftp/lib/*
```

Step 9
Finally, create the /home/ftp/dev/null file and set its mode appropriately.

```
[root@deep /]# mknod /home/ftp/dev/null c 1 3
[root@deep /]# chmod 666 /home/ftp/dev/null
```

Step 10
This step applies only if you choose to permit upload feature with anonymous FTP connection on the server. If you have created a special file system as explained above to allow upload into the anonymous FTP server area, then you'll need to have an FTP site administrator user to owns the files inside the /incoming directory, which is the area where we want to allow FTP anonymous users to uploads (this is a security feature).

- Use the following command to create the FTP Site Administrator user.

```
[root@deep /]# useradd -c "FTP Site Administrator" -u 96 -d /home/ftp -s
/bin/false ftpadmin 2>/dev/null || :

[root@deep /]# passwd ftpadmin
Changing password for user ftpadmin
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID.

- Now set permissions of the /home/ftp and /home/ftp/incoming areas to be:

```
chown ftpadmin /home/ftp/
chown ftpadmin.ftpadmin /home/ftp/incoming/
chmod 755 /home/ftp/
chmod 3773 /home/ftp/incoming/
```

NOTE: The /home/ftp/incoming location is a Linux file system which has already been created early by you for this purpose or which will be created now to allow you to change its security permission.

Step 11
This step applies only if you choose to permit upload feature with anonymous FTP connection on the server. Once the FTP Site Administrator has been added to the password file, we must allow it to connect to the anonymous FTP server through the /etc/ftpaccess file:

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following lines:

```
allow-uid ftpadmin
allow-gid ftpadmin
```

Step 12
This step applies to both download and upload features with anonymous FTP connection on the server. Once permission have been granted to the ftpadmin user, it's time to deny upload capability into the /ftp areas and to allow it just from the /incoming area of the anonymous FTP server.

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following clauses:

```
upload /home/ftp *            no
upload /home/ftp * /dev       no
upload /home/ftp * /bin       no
upload /home/ftp * /lib       no
upload /home/ftp * /usr       no
upload /home/ftp * /usr/bin   no

# Areas where upload clauses are allowed.
upload /home/ftp /incoming yes ftpadmin ftpadmin 0440 nodirs
```

The above lines specifies to deny upload feature into all the "/ftp" areas, and to allow uploads <yes> into the directory </incoming> of the anonymous FTP server to everyone with permission files set to <0440> but without the possibility to create new directories <nodirs> inside this /incoming area. It's important to note that the owner and group permission of all files inside this directory will be the FTP Site Administrator user named <ftpadmin>.

Step 13
This step applies to both download and upload features with anonymous FTP connection on the server. As you're supposed to know now, every new directories you may want to create and add inside the protected bubble of the FTP server and especially directories for anonymous ftp users must be examined to know if we need to allow download or not from these new directories. For anonymous FTP connection we must ensure the following:

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following clauses to make sure no downloads occur from all of the following areas of the anonymous ftp server:

```
noretrieve /home/ftp/dev/
noretrieve /home/ftp/bin/
noretrieve /home/ftp/lib/
noretrieve /home/ftp/usr/
noretrieve /home/ftp/usr/bin/
noretrieve /home/ftp/incoming/
```

Also because the /incoming area of the anonymous ftp server is a Linux file system, there will be another directory inside it named loast+found, it's wise to prevent possible downloading of this directory too:

- Edit the **ftpaccess** file (vi /etc/ftpaccess) and add the following clause:

```
noretrieve /home/ftp/incoming/lost+found/
```

Step 14
Finally, don't forget to create the special files .notar in each anonymous users directory and make them readable and retrievable too. Related to our anonymous example directories, these must be done in all of the following locations:

```
[root@deep /]# echo "Tarring is denied" > /home/ftp/.notar
[root@deep /]# echo "Tarring is denied" > /home/ftp/dev/.notar
[root@deep /]# echo "Tarring is denied" > /home/ftp/bin/.notar
[root@deep /]# echo "Tarring is denied" > /home/ftp/lib/.notar
[root@deep /]# echo "Tarring is denied" > /home/ftp/usr/.notar
[root@deep /]# echo "Tarring is denied" > /home/ftp/usr/bin/.notar
[root@deep /]# echo "Tarring is denied" > /home/ftp/incoming/.notar
[root@deep /]# chmod 0444 /home/ftp/.notar
[root@deep /]# chmod 0444 /home/ftp/dev/.notar
[root@deep /]# chmod 0444 /home/ftp/bin/.notar
[root@deep /]# chmod 0444 /home/ftp/lib/.notar
[root@deep /]# chmod 0444 /home/ftp/usr/.notar
[root@deep /]# chmod 0444 /home/ftp/usr/bin/.notar
[root@deep /]# chmod 0444 /home/ftp/incoming/.notar
```

Step 15
Restart the `Wu-ftpd` server for the changes to take effect.

- To restart `Wu-ftpd`, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/ftpd restart
Stopping ftpd:            [OK]
Starting ftpd:            [OK]
```

Further documentation
For more details, there are several manual pages related to `Wu-ftpd` that you could read:

```
$ man ftpcount (1)          - Show current number of users for each class
$ man ftpwho (1)            - Show current process information for each ftp user
$ man ftpaccess (5)         - ftpd configuration file
$ man ftphosts (5)          - ftpd individual user host access file
$ man ftpconversions (5)    - ftpd conversions database
$ man xferlog (5)           - FTP server logfile
$ man ftpd (8)              - Internet File Transfer Protocol server
$ man ftpshut (8)           - Close down the ftp servers at a given time
$ man ftprestart (8)        - Restart previously shutdown ftp servers
$ man privatepw (8)         - Change WU-FTPD Group Access File Information
```

`Wu-ftpd` Administrative Tools
The commands listed belows are some of the most used in regular use of this software, but many more exist. Check the manual pages for more details.

ftpwho
The `ftpwho` program utility displays all active `ftp` users, and their current process information on the system. The output of the command is in the format of the `/bin/ps` command. The format of this command is:

- To displays all active `ftp` users and their current process, use the following command:
```
[root@deep /]# ftpwho
Service class openna:
 5443 ?        S      0:00 ftpd: station1.openna.com: ftpadmin: IDLE
   -   1 users ( 20 maximum)
```

Here, you can see that one user is logged in, 20 users are allowed to connect, and this user has the username "`ftpadmin`" who claims to be from `station1.openna.com`.

ftpcount

The `ftpcount` program utility, which is a simplified version of `ftpwho`, shows only the current number of users logged in to the system, and the maximum number of users allowed.

- To shows only the current number of users logged in to the system and the maximum number of users allowed, use the following command:

```
[root@deep /]# ftpcount
Service class openna            -   1 users ( 20 maximum
```

List of installed `Wu-ftpd` files on your system

> /etc/rc.d/init.d/wuftpd
> /etc/ftpaccess
> /etc/ftpconversions
> /etc/ftphosts
> /etc/logrotate.d/ftpd
> /usr/bin/ftpcount
> /usr/bin/ftpwho
> /usr/sbin/in.ftpd
> /usr/sbin/wu.ftpd
> /usr/sbin/in.wuftpd
> /usr/sbin/ftpshut
> /usr/sbin/ckconfig
> /usr/sbin/ftprestart
> /usr/sbin/privatepw
> /usr/sbin/xferstats
> /usr/share/man/man1/ftpcount.1
> /usr/share/man/man1/ftpwho.1
> /usr/share/man/man5/ftpaccess.5
> /usr/share/man/man5/ftphosts.5
> /usr/share/man/man5/ftpconversions.5
> /usr/share/man/man5/ftpservers.5
> /usr/share/man/man5/xferlog.5
> /usr/share/man/man8/ftpd.8
> /usr/share/man/man8/ftpshut.8
> /usr/share/man/man8/ftprestart.8
> /usr/share/man/man8/privatepw.8
> /var/log/xferlog

29 Other Server - Apache Web Server

In this Chapter

Linux MM - Shared Memory Library

Abstract

I recommend that you compile and install this small program only if you intend to install and use the Apache web server with third party modules like mod_ssl for encrypted data, mod_perl for the Perl programming language, or mod_php for the PHP server-side scripting language. This program will provide a significant performance to Apache modules. For instance if you need to install Apache with SSL support for your electronic commerce on the Internet, this will allows the SSL protocol to use a high-performance RAM-based session cache instead of a disk-based one.

As explained in the [MM Shared Memory Library web site]:
The MM library is a 2-layer abstraction library, which simplifies the usage of shared memory between forked (and, in this example, strongly related) processes under Unix platforms. On the first layer it hides all platform dependent implementation details (allocation and locking) when dealing with shared memory segments, and on the second layer it provides a high-level malloc(3)-style API for a convenient and well known way to work with data-structures inside those shared memory segments.

The library is released under the term of an open-source (BSD-style) license, because it was originally written as a proposal for use inside the next version of the Apache web server as a base library for providing shared memory pools to Apache modules (because currently, Apache modules can only use heap-allocated memory, which isn't shared across the pre-forked server processes). The requirement actually comes from comprehensive modules like mod_ssl, mod_perl and mod_php, which would benefit a lot from easy to use shared memory pools.

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest MM version number is 1.1.3

Packages

The following is based on information as listed by MM Shared Memory Library as of 01/07/2000. Please regularly check at www.engelschall.com/sw/mm/ for the latest status.

Source code is available from:
MM Homepage: http://www.engelschall.com/sw/mm/
You must be sure to download: mm-1.1.3.tar.gz

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install MM, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > MM1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > MM2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff MM1 MM2 > MM-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing MM

Below are the required steps that you must make to compile and optimize the MM Shared Memory Library software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp mm-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf mm-version.tar.gz
```

Step 2
After that, move into the newly created MM directory then configure, compile and optimize it.

- To move into the newly created MM directory use the following command:
```
[root@deep tmp]# cd mm-1.1.3/
```

- To configure, compile and optimize MM use the following compilation lines:
```
CFLAGS="-O3 -march=i686 -funroll-loops -fomit-frame-pointer" \
./configure \
--prefix=/usr \
--mandir=/usr/share/man \
--disable-shared
```

This tells MM to set itself up for this particular configuration setup with:

- Disable shared libraries.

WARNING: Pay special attention to the compile `CFLAGS` line above. We compile optimize `MM` for an i686 CPU architecture with the parameter "`-march=i686` and `-mcpu=i686`". Please don't forget to adjust this `CFLAGS` line to reflect your own system and CPU architecture.

Step 3
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `MM Shared Memory Library` in the server:

```
[root@deep mm-1.1.3]# make
[root@deep mm-1.1.3]# make test
[root@deep mm-1.1.3]# cd
[root@deep /root]# find /* > MM1
[root@deep /root]# cd /var/tmp/mm-1.1.3/
[root@deep mm-1.1.3]# make install
[root@deep mm-1.1.3]# cd
[root@deep /root]# find /* > MM2
[root@deep /root]# diff MM1 MM2 > MM-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and then install the binaries and any supporting files into the appropriate locations.

NOTE: The **`make test`** command will make some important tests on the program to verify that it works, and respond properly before the installation.

Step 4
Now, it's time to use the following command to verify and be sure that the "`--disable-shared`" option has been properly applied during compile time to the program.

- To verify if the program has been compiled statically, use the following command:
```
[root@deep tmp]# ldd /usr/bin/mm-config
not a dynamic executable
```

If you receive a message like "`not a dynamic executable`", then congratulations!

Step 5
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete `MM` and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf mm-version/
[root@deep tmp]# rm -f mm-version.tar.gz
```

The `rm` command as used above will remove all the source files we have used to compile and install `MM`. It will also remove the `MM` compressed archive from the `/var/tmp` directory.

Further documentation

For more details, there are two manual pages related to this software that you could read:

```
MM (3)              - Shared Memory Library
mm-config (1)       - MM library configuration/build utility
```

List of installed `MM Shared Memory Library` files on your system

```
> /usr/bin/mm-config
> /usr/include/mm.h
> /usr/lib/libmm.la
> /usr/lib/libmm.a
> /usr/share/man/man1/mm-config.1
> /usr/share/man/man3/mm.3
```

Linux Apache Web Server

Abstract

Apache is the most widely used HTTP-server in the world today. It surpasses all free and commercial competitors on the market, and provides a myriad of features; more than the nearest opponent could give you on a UNIX variant. It is also the most used web server for a Linux system. A web server like Apache, in its simplest function, is software that displays and serves HTML pages hosted on a server to a client browser that understands the HTML code. Mixed with third party modules and programs, it can become powerful software, which will provide strong and useful services to a client browser.

I expect that most of the users that read this book will be especially interested in knowing how to install the Apache web server in the most secure, and optimized, way. In its base install, Apache is no more difficult to install then the other software we have installed on our Linux server. The procedures can become tricky when we want to add some third party modules or programs.

There are a lot of possibilities, variants and options for installing Apache. Therefore, in the following, we provide some step-by-step examples where you can see how to build Apache with other third-party modules and programs like mod_ssl, mod_perl, PHP4, SQL database, etc.

Of course, the building of these programs is optional, and you are free to compile only what you want (i.e., you may want to compile Apache with support for PHP4, but without SSL or SQL database connectivity). For simplification we assume some prerequisites for each example. If these don't fit your situation, simply adjust the steps.

In this chapter, we explain and cover some of the basic ways in which you can adjust the configuration to improve the server's performance. Also, for the interested users, we'll provide a procedure to be able to run Apache as a non root-user and in a chrooted environment for optimal security.

Some statistics about Apache and Linux

People like to see statistics and benchmark of different kind. It is always interesting to know the last milliseconds, bits we can take from our software and servers. The following pages explains and show you another one about Apache and Linux but not in the way you are accustomed in general. The moral is that: it is not always good to try or trust benchmarks, technologies limit, unthinking factor, etc that may influence results, but stability of your system is something you must have and keep.

What are some of the actual facts that the tests came up with?

- With 1 CPU and 256 MB RAM, Linux & Apache achieved 1,314 http requests per second.

First of, let's just look at an approximation of the situation that this represents:

- 1,314 hits/sec * 3600 sec/hour * 24 hours/day = 113,529,600 hits/day.

So Linux/Apache should be able to handle your site on a 1 CPU 256 MB RAM machine if you get 113 million hits per day or less. Of course, this only works if your access is 100% even, which is extremely unrealistic. Let's assume that your busy times get ten times more hits per second than your average hits/second. That means that a single CPU Linux machine with 256 meg of RAM should work for you if you get about 11 million hits every day (113/10 = 11.3).

Heck, let's be more conservative. Let's say that your busy times get 100 times more hits/second than your average hits/second. That means that if you get 1.1 million hits per day or less, that same machine will serve your site just fine (113/100 = 1.13).

OK, there's that way of looking at it, but it's not really a good way. It's a very coarse approximation of access patterns and what a site needs. Let's try another way of looking at this. Let's do some simple calculations to see what sort of bandwidth these numbers mean. Bandwidth will be a better and more constant method of determining whom these numbers apply to than guessed at hit ratios.

The files served must be of "varying sizes", so we'll have to make some assumptions about the average size of the files being served. Since over 1000 files were served per second, it is pretty safe to work by averages.

Some numbers:

- 1,314 hits/sec * 1 kilobyte/hit * 8192 bits/kilobyte = 10764288 bits/sec = 10 MBits/sec.
- 1,314 hits/sec * 2 kilobytes/hit * 8192 bits/kilobyte = 21528576 bits/sec = 21 MBits/sec.
- 1,314 hits/sec * 5 kilobytes/hit * 8192 bits/kilobyte = 53821440 bits/sec = 53 MBits/sec.
- 1,314 hits/sec * 10 kilobytes/hit * 8192 bits/kilobyte = 107642880 bits/sec = 107 MBits/sec.
- 1,314 hits/sec * 25 kilobytes/hit * 8192 bits/kilobyte = 269107200 bits/sec = 269 MBits/sec.

Just as a reference, a T1 line is worth approximately 1.5 MBits/sec, these numbers don't include TCP/IP & HTTP overhead.

Now, what does this tell us? Well, that if you are serving up 1,314 pages per second where the average page is only 1 kilobyte, you'll need ten (10) T1 lines or the equivalent until the computer is the limiting factor. What site on earth is going to be getting a sustained >1000 hits per second for 1 kilobyte files? Certainly not one with any graphics in it.

Let's assume that you're running a site with graphics in it and that you're average file is 5 kilobytes - not too conservative or too liberal. This means that if you're serving up 1,314 of them a second, you'll need 53 MBits of bandwidth. And there are no peak issues here; you can't peak out more than your bandwidth.

Let's go at it another way, this time starting with our available bandwidth:

1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 184 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 92 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 37 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 19 hits/sec.
1 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 8 hits/sec.

5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 916 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 458 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 183 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 92 hits/sec.
5 T1 Line * 1.5 MBits/T1 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 36 hits/sec.

1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 5,494 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 2747 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 1099 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 550 hits/sec.
1 T3 Line * 45 MBits/T3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 220 hits/sec.

1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/kilobyte = 18,921 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/2 kilobytes = 9461 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/5 kilobytes = 3785 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/10 kilobytes = 1,893 hits/sec.
1 OC3 Line * 155 MBits/OC3 * 1,000,000 bits/MBit * 1 kilobyte/8192 bits * 1 hit/25 kilobytes = 757 hits/sec.

NOTE: These numbers don't include `TCP/IP` or `HTTP` overhead.

It is clear that the numbers are only significant when you have the equivalent bandwidth of over 6 T1 lines. Let's be clear about this: if you have only **five (5) T1 lines** or less, a single CPU Linux machine with 256 MB RAM will **wait on your internet connection** and not be able to serve up to its full potential.

Let me re-emphasize this: A single CPU Linux machine with 256 MB RAM running `Apache` will **run faster than your internet connection**! Put another way, if your site runs on five (5) T1 lines or less, a single CPU Linux machine with 256 MB RAM will **more than fulfill your needs with CPU cycles left over**.

Let's make an assumption that you either (a) have pages with more than about a screen of text or (b) black and white pictures that make your average file size 5K. Given this, would indicate that a single CPU Linux machine with only 256 MB RAM running `Apache` would be **constantly waiting on your T3 line**. In other words, a single CPU Linux machine with 256 MB RAM will **serve your needs with room to grow** if your site is served by a **T3 line** or less.

One might also conclude that if you serve things like colour pictures (other than small buttons and doodads) and thus your average file size is 25K, a single CPU Linux machine with 256 MB RAM will serve your site just fine even if you are served by an OC3 line that you have all to your self.

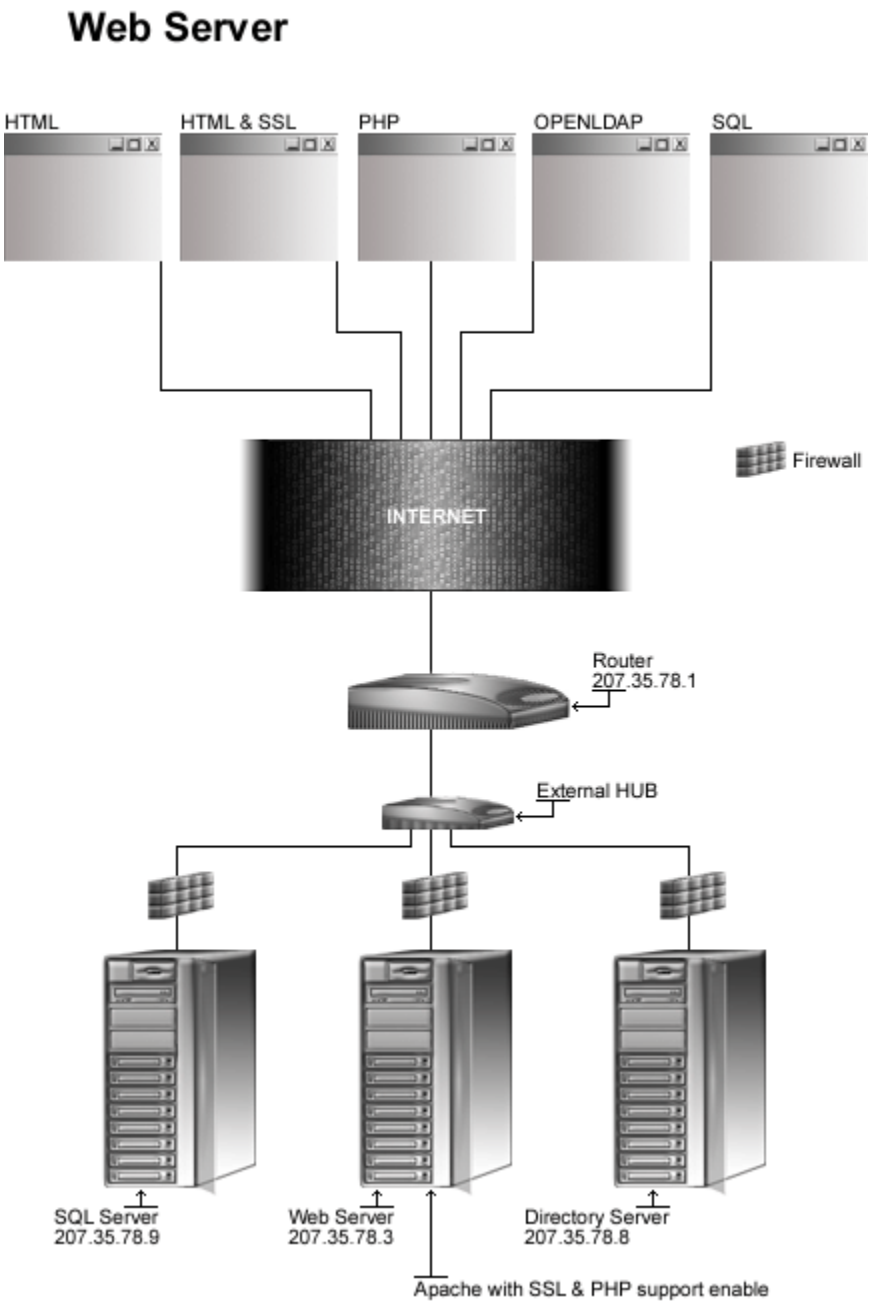
Recommended RPM packages to be installed for a Web Server

A minimal configuration provides the basic set of packages required by the Linux operating system. Minimal configuration is a perfect starting point for building secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a Web Server running on Apache software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install Apache by RPM package. Therefore, apache RPM package is already included in the list below as you can see. There are seven other interesting RPM packages to install with Apache. These packages freetype, gd, libjpeg, libpng, libtool-libs, aspell and pspell will allow the Web Server to run fine with external programs that you might install in the future. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

apache	e2fsprogs	iptables	openssh	slocate
aspell	ed	kernel	openssh-server	sysklogd
basesystem	file	less	openssl	syslinux
bash	filesystem	**libjpeg**	pam	SysVinit
bdflush	fileutils	**libpng**	passwd	tar
bind	findutils	libstdc++	**perl**	termcap
bzip2	**freetype**	libtermcap	popt	textutils
chkconfig	gawk	**libtool-libs**	procps	tmpwatch
console-tools	**gd**	lilo	psmisc	utempter
cpio	gdbm	logrotate	**pspell**	util-linux
cracklib	gettext	losetup	pwdb	vim-common
cracklib-dicts	glib	MAKEDEV	qmail	vim-minimal
crontabs	glibc	man	readline	vixie-cron
db1	glibc-common	mingetty	rootfiles	words
db2	grep	mktemp	rpm	which
db3	groff	mount	sed	zlib
dev	gzip	ncurses	sh-utils	
devfsd	info	net-tools	shadow-utils	
diffutils	initscripts	newt	slang	

Tested and fully functional on OpenNA.com.


Web Server
HTML
HTML & SSL
PHP
OPENLDAP
SQL
Firewall
INTERNET
Router
207.35.78.1
External HUB
SQL Server
207.35.78.9
Web Server
207.35.78.3
Directory Server
207.35.78.8
Apache with SSL & PHP support enable

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account “root”.
Whether kernel recompilation may be required: No
Latest Apache version number is 1.3.20
Latest Mod_SSL version number is 2.8.4-1.3.20
Latest Mod_Perl version number is 1.25
Latest PHP version number is 4.0.5

Packages

The following are based on information as listed by Apache as of 2001/05/26, mod_ssl as of 2001/05/26, mod_perl as of 2001/03/16, and PHP as of 2001/05/16. Please regularly check at www.apache.org, www.modssl.org, perl.apache.org, and www.php.net for the latest status.

Source codes are available from:
Apache Homepage: http://www.apache.org/
Apache FTP Site: 198.3.136.138
You must be sure to download: apache_1.3.20.tar.gz
Mod_SSL Homepage: http://www.modssl.org/
Mod_SSL FTP Site: 129.132.7.171
You must be sure to download: mod_ssl-2.8.4-1.3.20.tar.gz
Mod_Perl Homepage: http://perl.apache.org/
You must be sure to download: mod_perl-1.25.tar.gz
PHP Homepage: http://www.php.net/
You must be sure to download: php-4.0.5.tar.gz

Prerequisites

Apache requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install them from your Linux CD-ROM or source archive files. Please make sure you have all of these programs installed on your machine before you proceed with this chapter.

- ✓ OpenSSL should be already installed on your system if you want Apache and SSL encryption support.
- ✓ An SQL database of your choice should be already installed on your system if you want Apache with PHP4 and SQL database connectivity support.
- ✓ MM Shared Memory Library should be already installed on your system if you want Apache and MM high-performance RAM-based session cache support.
- ✓ OpenLDAP should be already installed on your system if you want Apache with PHP4 and LDAP directory connectivity support.
- ✓ Sendmail or qmail should be already installed on your system if you want Apache with mail capability.
- ✓ IMAP & POP should be already installed on your system if you want Apache with PHP4 and IMAP & POP capability.

- ✓ libjpeg package, which contains a library of functions for manipulating JPEG images.
- ✓ libpng package, which contains a library of functions for creating and manipulating PNG image format files.
- ✓ freetype package, a library which can open and manages font files as well as efficiently load, hint and render individual glyphs.
- ✓ gd package, which is a graphics library for drawing GIF files.
- ✓ aspell package, which is a spelling checker program.
- ✓ pspell package, which is a portable spell checker interface library.
- ✓ libtool-libs package, which contains the runtime libraries for GNU libtool.

- ➢ To verify if libjpeg package is installed on your system, use the command:
```
[root@deep /]# rpm -q libjpeg
package libjpeg is not installed
```
- ➢ To verify if libpng package is installed on your system, use the command:
```
[root@deep /]# rpm -q libpng
package libpng is not installed
```
- ➢ To verify if freetype package is installed on your system, use the command:
```
[root@deep /]# rpm -q freetype
package freetype is not installed
```
- ➢ To verify if gd package is installed on your system, use the command:
```
[root@deep /]# rpm -q gd
package gd is not installed
```
- ➢ To verify if aspell package is installed on your system, use the command:
```
[root@deep /]# rpm -q aspell
package aspell is not installed
```
- ➢ To verify if pspell package is installed on your system, use the command:
```
[root@deep /]# rpm -q pspell
package pspell is not installed
```
- ➢ To verify if libtool-libs package is installed on your system, use the command:
```
[root@deep /]# rpm -q libtool-libs
package libtool-libs is not installed
```

- To mount your CD-ROM drive before installing all require packages, use the command:
```
[root@deep /]# mount /dev/cdrom /mnt/cdrom/
mount: block device /dev/cdrom is write-protected, mounting read-only
```
- To install the libjpeg package on your Linux system, use the following command:
```
[root@deep /]# cd /mnt/cdrom/RedHat/RPMS/
[root@deep RPMS]# rpm -Uvh libjpeg-version.i386.rpm
libjpeg      ##################################################
```

- To install the `libpng` package on your Linux system, use the following command:
```
[root@deep RPMS]# rpm -Uvh libpng-version.i386.rpm
libpng       ##################################################
```

- To install the `freetype` package on your Linux system, use the following command:
```
[root@deep RPMS]# rpm -Uvh freetype-version.i386.rpm
freetype     ##################################################
```

- To install the `gd` package on your Linux system, use the following command:
```
[root@deep RPMS]# rpm -Uvh gd-version.i386.rpm
gd           ##################################################
```

- To install the `aspell` package on your Linux system, use the following command:
```
[root@deep RPMS]# rpm -Uvh aspell-version.i386.rpm
aspell       ##################################################
```

- To install the `pspell` package on your Linux system, use the following command:
```
[root@deep RPMS]# rpm -Uvh pspell-version.i386.rpm
pspell       ##################################################
```

- To install the `libtool-libs` package on your Linux system, use the command:
```
[root@deep RPMS]# rpm -Uvh libtool-libs-version.i386.rpm
libtool-libs ##################################################
```

- To unmount your CD-ROM drive, use the following command:
```
[root@deep RPMS]# cd /; umount /mnt/cdrom/
```

NOTE: For more information on the required software, see their related chapters in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install `Apache`, and one afterwards, and then compare them using the **`diff`** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
```
[root@deep /root]#  find /* > Apache1
```

- And the following one after you install the software:
```
[root@deep /root]#  find /* > Apache2
```

- Then use the following command to get a list of what changed:
```
[root@deep /root]#  diff Apache1 Apache2 > Apache-Installed
```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the `/root` directory of the system to stock all generated list files.

Compiling - Optimizing & Installing Apache

Below are the required steps that you must make to compile, configure and optimize the Apache software with other third-party modules and programs (if needed) before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the entire needed programs from the main software site you must copy them to the /var/tmp directory and change to this location before expanding the archives. Below I suppose all of the following: Apache, mod_ssl, mod_perl, and PHP4.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp apache_version.tar.gz /var/tmp/
[root@deep /]# cp mod_ssl-version-version.tar.gz /var/tmp/
[root@deep /]# cp mod_perl-version.tar.gz /var/tmp/
[root@deep /]# cp php-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf apache_version.tar.gz
[root@deep tmp]# tar xzpf mod_ssl-version-version.tar.gz
[root@deep tmp]# tar xzpf mod_perl-version.tar.gz
[root@deep tmp]# tar xzpf php-version.tar.gz
```

Step 2
Apache web server, like many applications that we have installed, cannot be run as super-user "root" for security reasons. We must create a special user that has minimal access to the system, and still functions enough to run the Apache web Server. It is best to choose and create a new user just for the purpose of running the Web Server daemon.

- To create the Apache user, use the following command:

```
[root@deep tmp]# useradd -c "Apache Server" -u 80 -s /bin/false -r -d
/home/httpd www 2>/dev/null || :
```

The above command will create a null account, with no password, no valid shell, no files owned-nothing but a UID and a GID.

Step 3

Apply mod-ssl to Apache source tree

This section applies only if you choose to install mod_ssl with Apache in your system. If you want to use and include the SSL data encryption support in your Apache Web Server, then move into the new mod_ssl source directory and type the following commands on your terminal:

- To move into the new mod_ssl source directory, use the following command:

```
[root@deep tmp]# cd mod_ssl-2.8.4-1.3.20/
```

- To configure mod_ssl and include its codes into Apache, use the following compilation:

```
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
./configure \
--with-apache=../apache_1.3.20 \
--with-crt=/usr/share/ssl/certs/my.domain.com.crt \
--with-key=/usr/share/ssl/private/my.domain.com.key
```

The "--with-apache" option specifies the location of the Apache source directory (it's important to note that we suppose your Apache version in this example is 1.3.20), the "--with-crt" option specifies the location of your existing public key for SSL encryption, and the "--with-key" option specifies the location of your existing private key for SSL encryption.

WARNING: OpenSSL software must already be installed on your server, and your public and private keys must already be existent or be created on your server, or you'll receive an error message during the configuration time of mod_ssl. See the chapter related to OpenSSL in this book for more information on the subject.

Step 4

Improve the MaxClients Parameter of Apache

By default in the Apache configuration file (httpd.conf) the maximum number you can set for the MaxClients Parameter is 256. For a busy site, and for better performance, it's recommended that you increase the limit of this parameter. You can do it by editing the src/include/httpd.h file in the source directory of Apache and change the default value.

- To move into the Apache source directory use the following command:
  ```
  [root@deep mod_ssl-2.8.4-1.3.20]# cd ../apache_1.3.20/
  ```

- Edit the **httpd.h** file (vi +334 src/include/httpd.h), changing the line:

  ```
  #define HARD_SERVER_LIMIT 256
  ```

  To read:

  ```
  #define HARD_SERVER_LIMIT 1024
  ```

WARNING: If you configure Apache without mod_ssl support, then the line to edit to change the default value will be 317 instead of 334.

Step 5

Pre-configure Apache for PHP4's configure step

This section applies only if you chose to install and use PHP4 with Apache in your system. If you want to use and include the PHP4 server-side scripting language support on your Apache web server, then move into the new Apache source directory if you are not already in it and type the following commands on your terminal:

- To move into Apache source directory, use the following command:
  ```
  [root@deep /]# cd /var/tmp/apache_1.3.20/
  ```

- To pre-configure PHP4 and include its codes into Apache, use the following compilation:
  ```
  OPTIM="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
  CFLAGS="-DDYNAMIC_MODULE_LIMIT=0" \
  ./configure \
  --prefix=/home/httpd \
  --bindir=/usr/bin \
  --sbindir=/usr/sbin \
  --libexecdir=/usr/lib/apache \
  ```

```
--includedir=/usr/include/apache \
--sysconfdir=/etc/httpd/conf \
--localstatedir=/var \
--runtimedir=/var/run \
--logfiledir=/var/log/httpd \
--datadir=/home/httpd \
--proxycachedir=/var/cache/httpd \
--mandir=/usr/share/man
```

WARNING: This step is necessary only if you want to include PHP4 support in your Apache source code, since it'll pre-configure Apache for PHP4's configure step below. Take a note that the "-DDYNAMIC_MODULE_LIMIT=0" option will disable the use of dynamically loaded modules in the compilation of Apache, and will improve its performance.

Step 6

Configure PHP4 and apply it to the Apache source tree

This section applies only if you chose to install and use PHP4 with Apache in your system. Once we have pre-configured Apache to support PHP4 features, it is time to move into the new uncompressed PHP4 source directory then configure, optimize, compile and install it in the Linux server by using the following commands on your terminal:

- To move into the new PHP4 source directory, use the following command:
  ```
  [root@deep /]# cd /var/tmp/php-4.0.5
  ```

- To configure PHP4 and include its codes into Apache, use the following compilation lines:
  ```
  CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer -
  I/usr/include/openssl" \
  ./configure \
  --prefix=/usr \
  --with-exec-dir=/usr/bin \
  --with-apache=../apache_1.3.20 \
  --with-config-file-path=/etc/httpd \
  --with-gd \
  --with-ttf \
  --with-jpeg \
  --with-png \
  --with-mm \
  --with-imap-ssl \ (if you want SSL support in IMAP).
  --with-imap \ (if you want IMAP & POP support).
  --with-ldap \ (if you want LDAP database light directory support).
  --with-pgsql \ (if you want PostgreSQL database support).
  --with-mysql=/usr \ (if you want MySQL database support).
  --with-gettext \
  --with-zlib \
  --with-pspell \ (if you want a spell checker for specific applications)
  --enable-inline-optimization \
  --enable-bcmath
  ```

This tells PHP4 to set itself up for this particular configuration setup with:

- Include GD support.
- Include Freetype support.
- Include JPEG support for GD.
- Include PNG support for GD.

- Include mm support for session storage.
- Include SSL support in IMAP.
- Include IMAP & POP support.
- Include LDAP directory support.
- Include PostgresSQL database support.
- Include MySQL database support.
- Include GNU gettext support for Multilanguage.
- Include zlib support.
- Include PSPELL support for spell checker on third party program.
- Enable inline-optimization for better performance (only if you have much memory).
- Enable and compile with bc style precision math function for better performance.

Step 7
This section applies only if you chose to install and use PHP4 with Apache in your system. Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install PHP4 in the server.

```
[root@deep php-4.0.5]# make
[root@deep php-4.0.5]# cd
[root@deep /root]# find /* > PHP1
[root@deep /root]# cd /var/tmp/php-4.0.5/
[root@deep php-4.0.5]# make install
[root@deep php-4.0.51]# cd
[root@deep /root]# find /* > PHP2
[root@deep /root]# diff PHP1 PHP2 > PHP-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and finally install the binaries and any supporting files into the appropriate locations.

Step 8
Apply mod_perl to Apache source tree and build/install the Perl-side of mod_perl
This section applies only if you chose to install and use mod_perl with Apache in your system. If you want to use and include Perl programming language support in your Apache Web Server then, move into the new uncompressed mod_perl source directory and type the following commands on your terminal:

- To move into the new mod_perl source directory, use the following command:
  ```
  [root@deep /]# cd /var/tmp/mod_perl-1.25/
  ```

- To configure mod_perl and include its codes into Apache, use the compilation lines:
  ```
  perl Makefile.PL \
  EVERYTHING=1 \
  APACHE_SRC=../apache_1.3.20/src \
  USE_APACI=1 \
  PREP_HTTPD=1 \
  DO_HTTPD=1
  ```

The <Makefile.PL> command will search for Apache source trees to configure mod_perl, <DO_HTTPD=1> will avoid to configure and build httpd daemon, <EVERYTHING=1> will enable all callback hooks arguments.

Step 9

This section applies only if you chose to install and use mod_perl with Apache in your system. Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **diff** utility to find out what files are placed where and finally install mod_perl in the server.

```
[root@deep mod_perl-1.25]# make
[root@deep mod_perl-1.25]# cd
[root@deep /root]# find /* > ModPerl1
[root@deep /root]# cd /var/tmp/mod_perl-1.25/
[root@deep mod_perl-1.25]# make install
[root@deep mod_perl-1.25]# cd
[root@deep /root]# find /* > ModPerl2
[root@deep /root]# diff ModPerl1 ModPerl2 > ModPerl-Installed
```

The above commands will configure the software to ensure your system has the necessary libraries to successfully compile the package, compile all source files into executable binaries, and finally install the binaries and any supporting files into the appropriate locations.

Step 10

Build/Install Apache with/without mod_ssl +- PHP4 and/or mod_perl support

Once you have included in your Apache source the third party modules that you want to support and use, it is time to configure, compile, optimize and install them into your Linux system. The next step is to move into the Apache source directory and type the following commands on your terminal depending on what you want to install with Apache.

For people that just want to configure, compile and install Apache without any other third-party modules or programs, you must start directly from here.

- To move into Apache source directory, use the following command:
  ```
  [root@deep /]# cd /var/tmp/apache_1.3.20/
  ```

- To build Apache with all the require support programs, use the following compilation:
  ```
  SSL_BASE=SYSTEM \ (only for mod_ssl support).
  EAPI_MM=SYSTEM \ (only for mm Shared Memory Library support).
  OPTIM="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer" \
  CFLAGS="-DDYNAMIC_MODULE_LIMIT=0" \
  ./configure \
  --prefix=/home/httpd \
  --bindir=/usr/bin \
  --sbindir=/usr/sbin \
  --libexecdir=/usr/lib/apache \
  --includedir=/usr/include/apache \
  --sysconfdir=/etc/httpd/conf \
  --localstatedir=/var \
  --runtimedir=/var/run \
  --logfiledir=/var/log/httpd \
  --datadir=/home/httpd \
  --proxycachedir=/var/cache/httpd \
  --mandir=/usr/share/man \
  --add-module=src/modules/experimental/mod_mmap_static.c \ (only for mod_mmap).
  --add-module=src/modules/standard/mod_auth_db.c \ (only for mod_auth_db support).
  --enable-module=ssl \ (only for mod_ssl support).
  --enable-rule=SSL_SDBM \ (only for mod_ssl support).
  --disable-rule=SSL_COMPAT \ (only for mod_ssl support).
  --activate-module=src/modules/php4/libphp4.a \ (only for PHP4 support).
  --enable-module=php4 \ (only for PHP4 support).
  --activate-module=src/modules/perl/libperl.a \ (only for mod_perl support).
  --enable-module=perl \ (only for mod_perl support with Apache).
  ```

```
--disable-module=status \
--disable-module=userdir \
--disable-module=negotiation \
--disable-module=autoindex \
--disable-module=imap \
--server-uid=www \
--server-gid=www
```

This tells `Apache` to set itself up for this particular configuration setup with:

- Enable module `mod_mmap` to improve performance on download time.
- Enable module `mod_auth_db` for users password authentication security.
- Enable module `mod_ssl` for data encryptions and secure communication.
- Enable module `mod_php4` for `php` server-side scripting language.
- Enable module `mod_perl` for better security and performance than the default `cgi` scripts.
- Disable module status
- Disable module userdir
- Disable module negotiation
- Disable module autoindex
- Disable module imap

WARNING: It's important to note that removing all unneeded modules during the configure time of `Apache` will improve the performance of your Web Server. In our configuration, we've removed the most unused modules both to lower the load operation, and limit the security risks in our `Apache` Web Server. See your `Apache` documentation for information on each one.

Step 11
Now, we must make a list of files on the system before you install the software, and one afterwards, then compare them using the **`diff`** utility to find out what files are placed where and finally install `Apache` in the server.

```
[root@deep apache_1.3.20]# make
[root@deep apache_1.3.20]# cd
[root@deep /root]# find /* > Apache1
[root@deep /root]# cd /var/tmp/apache_1.3.20/
[root@deep apache_1.3.20]# make install
[root@deep apache_1.3.20]# rm -f /usr/sbin/apachectl
[root@deep apache_1.3.20]# rm -f /usr/share/man/man8/apachectl.8
[root@deep apache_1.3.20]# rm -rf /home/httpd/icons/
[root@deep apache_1.3.20]# rm -rf /home/httpd/htdocs/
[root@deep apache_1.3.20]# rm -f /home/httpd/cgi-bin/printenv
[root@deep apache_1.3.20]# rm -f /home/httpd/cgi-bin/test-cgi
[root@deep apache_1.3.20]# rm -rf /var/cache/httpd/
[root@deep apache_1.3.20]# rm -rf /etc/httpd/conf/ssl.crl/
[root@deep apache_1.3.20]# rm -rf /etc/httpd/conf/ssl.crt/
[root@deep apache_1.3.20]# rm -rf /etc/httpd/conf/ssl.csr/
[root@deep apache_1.3.20]# rm -rf /etc/httpd/conf/ssl.key/
[root@deep apache_1.3.20]# rm -rf /etc/httpd/conf/ssl.prm/
[root@deep apache_1.3.20]# rm -f /etc/httpd/conf/srm.conf
[root@deep apache_1.3.20]# rm -f /etc/httpd/conf/srm.conf.default
[root@deep apache_1.3.20]# rm -f /etc/httpd/conf/access.conf
[root@deep apache_1.3.20]# rm -f /etc/httpd/conf/access.conf.default
[root@deep apache_1.3.20]# rm -f /etc/httpd/conf/mime.types.default
[root@deep apache_1.3.20]# rm -f /etc/httpd/conf/magic.default
[root@deep apache_1.3.20]# cd /var/tmp/php-4.0.5/
[root@deep php-4.0.5]# install -m644 php.ini-dist /etc/httpd/php.ini
```

```
[root@deep php-4.0.5]# cd
[root@deep /root]# find /* > Apache2
[root@deep /root]# diff Apache1 Apache2 > Apache-Installed
```

The **make** command will compile all source files into executable binaries, and **make install** will install the binaries and any supporting files into the appropriate locations. The **rm -f** command will remove the small script "apachectl" responsible to start and stop the Apache daemon since we use a better script named "httpd" located under the /etc/rc.d/init.d/ directory that takes advantage of Linux system V.

We also remove the /home/httpd/icons directory needed under Apache when you use its automatic indexing feature. This feature can bring about a security risk, and for this reason we've disabled it in the configuration file. Therefore, we can safely remove the directory to make space on the Linux server. The /home/httpd/htdocs directory handles all documentation files related to Apache, so after we have finished reading the documentation we can remove it to gain space.

The **install -m** command will install the php.ini-optimized file under the /etc/httpd directory, and will rename it php.ini; this file controls many aspects of PHP's behavior and will exist only if you have configured Apache with PHP4 support. The ssl.crl, ssl.crt, ssl.csr, ssl.key, and ssl.prm directories under /etc/httpd/conf are all of the directories related to SSL, and handle private and public keys as well as other thing related to SSL features. Since we use another location, /usr/share/ssl, we can remove them safely. As for PHP4 support, these directories will exist only if you have configured Apache with mod_ssl support.

Finally, we remove the unneeded srm.conf, srm.conf.default, access.conf, mime.types.default, magic.default, and access.conf.default files, whose purposes are now handled by the httpd.conf Apache configuration file.

Step 12
Once compilation, optimization and installation of the software have been finished, we can free up some disk space by deleting the program tar archives and the related source directories since they are no longer needed.

- To delete all programs and their related source directories, use the following commands:

```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf apache-version/
[root@deep tmp]# rm -f apache-version.tar.gz
[root@deep tmp]# rm -rf mod_ssl-version-version/
[root@deep tmp]# rm -f mod_ssl-version-version.tar.gz
[root@deep tmp]# rm -rf php-version/
[root@deep tmp]# rm -f php-version.tar.gz
[root@deep tmp]# rm -rf mod_perl-version/
[root@deep tmp]# rm -f mod_perl-version.tar.gz
```

The rm commands as used above will remove all the source files we have used to compile and install Apache, mod_ssl, mod_perl, and PHP4. It will also remove the Apache, mod_ssl, mod_perl, and PHP4 compressed archives from the /var/tmp directory.

Configuring Apache

Configuration files for different services are very specific depending on your needs, and your network architecture. Someone might install Apache Server for showing web pages only; another might install it with database connectivity and e-commerce with SSL support, etc. Later, I provide a working httpd.conf file, with PHP4, Perl, SSL, and password authentication settings, to show you different possibilities but don't forget to use only the ones you need.

We'll focus on optimization and security of these files, and leave all specific adjustments to your tastes. You will need to read the documentation that comes with these programs, and hopefully understand them.

After building Apache, your next step is to verify or change, if necessary options in your Apache configuration files. Those files are:

- ✓ /etc/httpd/conf/httpd.conf (The Apache Configuration File)
- ✓ /etc/logrotate.d/httpd (The Apache Log Rotation File)
- ✓ /etc/rc.d/init.d/httpd (The Apache Initialization File)

/etc/httpd/conf/httpd.conf: The Apache Configuration File

The httpd.conf file is the main configuration file for the Apache Web Server. A lot options exist, and it's important to read the documentation that comes with Apache for more information on different settings and parameters.

The following configuration example is a full working configuration file for Apache, with SSL and PHP4 support. Also, it's important to note that I only comment parameters that relate to security and optimization, and leave all the others to your own research. We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Edit the **httpd.conf** file (vi /etc/httpd/conf/httpd.conf) and set your needs:

```
### Section 1: Global Environment
#
ServerType standalone
ServerRoot "/etc/httpd"
PidFile /var/run/httpd.pid
ResourceConfig /dev/null
AccessConfig /dev/null
Timeout 300
KeepAlive On
MaxKeepAliveRequests 0
KeepAliveTimeout 15
MinSpareServers 16
MaxSpareServers 64
StartServers 16
MaxClients 512
MaxRequestsPerChild 100000

### Section 2: 'Main' server configuration
#
<IfDefine SSL>
Listen 207.35.78.3:80
Listen 207.35.78.3:443
</IfDefine>

User www
```

```
Group www
ServerAdmin webadmin@openna.com
ServerName www.openna.com
DocumentRoot "/home/httpd/openna"

<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>

<Directory "/home/httpd/openna">
    Options None
    AllowOverride None
    Order allow,deny
    Allow from all
</Directory>

<Files .pl>
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Files>

<IfModule mod_dir.c>
DirectoryIndex index.htm index.html index.php index.php3 index.shtml
</IfModule>

#<IfModule mod_include.c>
#Include conf/mmap.conf
#</IfModule>

UseCanonicalName On

<IfModule mod_mime.c>
TypesConfig /etc/httpd/conf/mime.types
</IfModule>

DefaultType text/plain
HostnameLookups Off

ErrorLog /var/log/httpd/error_log
LogLevel warn
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\" \"%{User-Agent}i\""
combined
LogFormat "%h %l %u %t \"%r\" %>s %b" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent
CustomLog /var/log/httpd/access_log common

ServerSignature Off

<IfModule mod_alias.c>
ScriptAlias /cgi-bin/  "/home/httpd/cgi-bin/"
 <Directory "/home/httpd/cgi-bin">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
 </Directory>
</IfModule>
```

```
<IfModule mod_mime.c>
AddEncoding x-compress Z
AddEncoding x-gzip gz tgz
AddType application/x-tar .tgz
#AddType application/x-httpd-php .php
#AddType application/x-httpd-php .php3
#AddType application/x-httpd-php-source .phps
</IfModule>

ErrorDocument 404 http://www.openna.com/error.htm
ErrorDocument 403 "Access Forbidden -- Go away.

<IfModule mod_setenvif.c>
BrowserMatch "Mozilla/2" nokeepalive
BrowserMatch "MSIE 4\.0b2;" nokeepalive downgrade-1.0 force-response-1.0
BrowserMatch "RealPlayer 4\.0" force-response-1.0
BrowserMatch "Java/1\.0" force-response-1.0
BrowserMatch "JDK/1\.0" force-response-1.0
</IfModule>

### Section 3: Virtual Hosts
#
NameVirtualHost 207.35.78.3:80

<VirtualHost 207.35.78.3:80>
ServerAdmin webadmin@openna.com
ServerName www.openna.com
DocumentRoot "/home/httpd/openna"

ErrorLog /var/log/httpd/error_openna_log
TransferLog /var/log/httpd/access_openna_log
</VirtualHost>

## SSL Global Context
#
<IfDefine SSL>
AddType application/x-x509-ca-cert .crt
AddType application/x-pkcs7-crl    .crl
</IfDefine>

<IfModule mod_ssl.c>
SSLPassPhraseDialog     builtin
SSLMutex                sem
SSLRandomSeed startup   file:/dev/urandom 1024
SSLRandomSeed connect   builtin
SSLSessionCache         shm:/var/run/ssl_scache(512000)
SSLSessionCacheTimeout  300
SSLLog                  /var/log/httpd/ssl_engine_log
SSLLogLevel warn
</IfModule>

## SSL Virtual Host Context
#
<IfDefine SSL>
NameVirtualHost 207.35.78.3:443

<VirtualHost 207.35.78.3:443>
ServerAdmin webadmin@openna.com
ServerName www.openna.com
DocumentRoot "/home/httpd/openna"

ErrorLog /var/log/httpd/error_openna_log
```

```
TransferLog /var/log/httpd/access_openna_log

SSLEngine on

SSLCipherSuite ALL:!ADH:RC4+RSA:+HIGH:+MEDIUM:+LOW:+SSLv2:+EXP:+eNULL
SSLCertificateFile      /usr/share/ssl/certs/www.crt
SSLCertificateKeyFile   /usr/share/ssl/private/www.key
SSLVerifyClient none
SSLVerifyDepth  10

SetEnvIf User-Agent ".*MSIE.*" \
         nokeepalive ssl-unclean-shutdown \
         downgrade-1.0 force-response-1.0

CustomLog /var/log/httpd/ssl_request_log \
          "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
</VirtualHost>
</IfDefine>
```

This tells `httpd.conf` file to set itself up for this particular configuration setup with:

`ServerType standalone`
This option "`ServerType`" specifies how `Apache` should run on the system. You can run it from the super-server `xinetd`, or as standalone daemon. It's highly recommended to run `Apache` in standalone type for best performance and speed. Loading the `httpd` daemon, as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared. This is a performance feature.

`ServerRoot "/etc/httpd"`
This option "`ServerRoot`" specifies the directory in which the configuration files of the `Apache` server lives. It allows `Apache` to know where it can find its configuration files when it starts. In our setup, this file is located under `/etc/httpd/conf` directory and it's named `httpd.conf`.

`PidFile /var/run/httpd.pid`
This option "`PidFile`" specifies the location where the server will record the process id of the daemon when it starts. This option is only required when you configure `Apache` in standalone mode as we do.

`ResourceConfig /dev/null`
This option "`ResourceConfig`" specifies the location of the old `srm.conf` file that `Apache` read after it finished reading its `httpd.conf` file. When you set the location to `/dev/null`, `Apache` allows you to include the content of this file into the `httpd.conf` file, and in this manner, you have just one file that handles all your configuration parameters for simplicity.

`AccessConfig /dev/null`
This option "`AccessConfig`" specifies the location of the old `access.conf` file that `Apache` read after it finished reading the `srm.conf` file. As for the above "`ResourceConfig`" parameter, when you set the location to `/dev/null`, `Apache` allows you to include the content of this file into its `httpd.conf` file, and in this manner, you have just one file that handles all your configuration parameters for simplicity again.

`Timeout 300`
This option "`Timeout`" specifies the amount of time `Apache` will wait for a GET, POST, PUT request and ACKs on transmissions. You can safely leave this option on its default values.

`KeepAlive On`
This option "`KeepAlive`" if set to "`On`" enables persistent connections on the Web Server. For better performance, it's recommended to set this option to "`On`" and allow more than one request per connection. This is a performance feature.

`MaxKeepAliveRequests 0`
This option "`MaxKeepAliveRequests`" specifies the number of requests allowed per connection when the `KeepAlive` option above is set to "`On`". When the value of this option is set to "`0`" then unlimited requests are allowed on the server. For server performance, it's recommended to allow unlimited requests. This is a performance feature.

`KeepAliveTimeout 15`
This option "`KeepAliveTimeout`" specifies how much time, in seconds, `Apache` will wait for a subsequent request before closing the connection. The value of "`15`" seconds is a good average for server performance. This is a performance feature.

`MinSpareServers 16`
This option "`MinSpareServers`" specifies the minimum number of idle child server processes for `Apache`, which is not handling a request. This is an important tuning parameter regarding the performance of the `Apache` Web Server. For high load operation, a value of "`16`" is recommended by various benchmarks on the Internet. This is a performance feature.

`MaxSpareServers 64`
This option "`MaxSpareServers`" specifies the maximum number of idle child server processes for `Apache`, which is not handling a request. This is also an important tuning parameter regarding the performance of the `Apache` Web Server. For high load operation, a value of "`64`" is recommended by various benchmarks on the Internet. This is a performance feature.

`StartServers 16`
This option "`StartServers`" specifies the number of child server processes that will be created by `Apache` on start-up. This is, again, an important tuning parameter regarding the performance of the `Apache` Web Server. For high load operation, a value of "`16`" is recommended by various benchmarks on the Internet. This is a performance feature.

`MaxClients 512`
This option "`MaxClients`" specifies the number of simultaneous requests that can be supported by `Apache`. This is an important tuning parameter regarding the performance of the `Apache` Web Server. For high load operation, a value of "`512`" is recommended by various benchmarks on the Internet. This is a performance feature.

`MaxRequestsPerChild 100000`
This option "`MaxRequestsPerChild`" specifies the number of requests that an individual child server process will handle. This is an important tuning parameter regarding the performance of the Apache Web Server. This is a performance feature.

`User www`
This option "`User`" specifies the UID that `Apache` daemon will run as. It's important to create a new user that has minimal access to the system, and functions just for the purpose of running the Web Server daemon. Using a different UID that already exists on the system (i.e. `nobody`) can allow your services to access each other's resources. In our example, we use the `Apache` user we have created previously which is named "`www`".

```
Group www
```

This option "Group" specifies the GID the Apache daemon will run as. It's important to create a new group that has minimal access to the system and functions just for the purpose of running the Web Server daemon. In our example, we use the Apache group we have created previously which is named "www".

```
<Directory />
    Options None
    AllowOverride None
    Order deny,allow
    Deny from all
</Directory>
```

This block of options allows running a really tight ship by stopping users overriding system wide settings. This is because the default Apache access for <Directory /> is Allow from All, and this means that it will serve any file mapped from an URL. For this reason it is highly recommended that you change this block such as the one we have configured and then override this for directories you want accessible. This is a security feature.

```
DirectoryIndex index.htm index.html index.php index.php3 index.shtml
```

This option "DirectoryIndex" specifies the files to use by Apache as a pre-written HTML directory index. In other words, if Apache can't find the default index page to display, it'll try the next entry in this parameter, if available. To improve performance of the Web Server it's recommended to list the most used default index pages of your web site first and not to include too much. This is a performance feature.

```
<IfModule mod_include.c>
Include conf/mmap.conf
</IfModule>
```

This option "Include" specifies the location of other files that you can include from within the server configuration files httpd.conf. In our case, we include the mmap.conf file located under /etc/httpd/conf directory. This file mmap.conf maps files into memory for faster serving. See the section on "Optimizing Apache" in this chapter for more information. This is a performance feature.

```
HostnameLookups Off
```

This option "HostnameLookups" if set to "Off" specifies the disabling of DNS lookups. It's recommended to set this option to "Off" in order to save the network traffic time, and to improve the performance of your Apache Web Server. This is a performance feature.

> **NOTE:** If your httpd.conf file contains many <VirtualHost> sections that are substantially the same, then I recommend you to read the Apache "Dynamically configured mass virtual hosting" document, which describes how to efficiently serve an arbitrary number of virtual hosts. This is an online documentation, which can be retrieved from the Apache website at the following URL: http://httpd.apache.org/docs/vhosts/mass.html.

/etc/logrotate.d/httpd: The Apache Log rotation File

The /etc/logrotate.d/httpd file allows the Web Server to rotate each week all Apache log files automatically. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

- Create the **httpd** file (touch /etc/logrotate.d/httpd) and add the following lines:

```
/var/log/httpd/access_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}

/var/log/httpd/error_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}

/var/log/httpd/ssl_request_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}

/var/log/httpd/ssl_engine_log {
    missingok
    postrotate
        /usr/bin/killall -HUP httpd
    endscript
}
```

NOTE: Lines to automatically rotate the SSL log files named ssl_request_log and ssl_engine_log are included in this file. If you intend to run Apache without SSL support, you must remove the above lines related to SSL.

/etc/rc.d/init.d/httpd: The Apache Initialization File

The /etc/rc.d/init.d/httpd script file is responsible to automatically start and stop the Apache daemon on your server. Loading the httpd daemon, as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Step 1
Create the **httpd** script file (touch /etc/rc.d/init.d/httpd) and add the following lines:

```
#!/bin/sh
#
# Startup script for the Apache Web Server
#
# chkconfig: 345 85 15
# description: Apache is a World Wide Web server.  It is used to serve \
#          HTML files and CGI.
# processname: httpd
```

```
# pidfile: /var/run/httpd.pid
# config: /etc/httpd/conf/httpd.conf


# Source function library.
. /etc/rc.d/init.d/functions

# See how we were called.
case "$1" in
  start)
    echo -n "Starting httpd: "
    daemon httpd -DSSL
    echo
    touch /var/lock/subsys/httpd
    ;;
  stop)
    echo -n "Shutting down http: "
    killproc httpd
    echo
    rm -f /var/lock/subsys/httpd
    rm -f /var/run/httpd.pid
    ;;
  status)
    status httpd
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  reload)
    echo -n "Reloading httpd: "
    killproc httpd -HUP
    echo
    ;;
  *)
    echo "Usage: $0 {start|stop|restart|reload|status}"
    exit 1
esac

exit 0
```

Step 2
Once the `httpd` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the command:
```
[root@deep /]# chmod 700 /etc/rc.d/init.d/httpd
[root@deep /]# chown 0.0 /etc/rc.d/init.d/httpd
```

- To create the symbolic `rc.d` links for `Apache`, use the following command:
```
[root@deep /]# chkconfig --add httpd
[root@deep /]# chkconfig --level 345 httpd on
```

- To start `Apache` software manually, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/httpd start
Starting httpd:          [OK]
```

WARNING: The "**-DSSL**" option that we added by default in the initialization file above will start Apache in SSL mode. If you want to start it in regular mode, remove the "**-DSSL**" option near the line that reads "daemon httpd" in the Apache initialization file.

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, floppy-2.0.tgz for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example /var/tmp, assuming you have done this your directory structure will be /var/tmp/floppy-2.0. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Enable PHP4 server-side scripting language with the Web Server

If you intend to use PHP4 server-side scripting language support with your Apache Web Server don't forget to include in your /etc/httpd/conf/httpd.conf file the following lines to enable this feature:

Step 1
Edit the **httpd.conf** file (vi /etc/httpd/conf/httpd.conf), and add or uncomment the following lines between the section tags <IfModule mod_mime.c> and </IfModule>.

```
AddType application/x-httpd-php .php
AddType application/x-httpd-php .php3
AddType application/x-httpd-php-source .phps
```

Step 2
Once the above lines have been included or uncommented into the httpd.conf file of Apache to enable PHP4 feature with your Web Server, you must restart the Apache for the changes to take effect.

- To restart Apache, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/httpd restart
Shutting down http:            [OK]
Starting httpd:                [OK]
```

Step 3
After that the Web Server has been restarted, we must test the new PHP4 feature to be sure it's working. We'll create a small PHP file named php.php in our DocumentRoot, and then point our web browser to this PHP document to see if PHP4 work on the server.

- Create the **php.php** file in your DocumentRoot (touch /home/httpd/openna/php.php) and add the following line in the PHP file:

```
<?php phpinfo()?>
```

NOTE: This line will inform PHP4 program to display various pieces of information about the configuration of our Linux Web Server.

Step 4
Now, point your web browser to the following address: http://my.domain.com/php.php

The <my.domain.com> is the address where your Apache Web Server lives, and <php.php> is the PHP document we have created above to display the information and configuration of our Linux Web Server with PHP4 features enable.

If you see something like the above page appearing in your web browser… congratulations! Your PHP module is working.

Securing Apache

This section deals especially with actions we can make to improve and tighten security under Apache. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Change some important permissions on files and directories for your Web Server

When you install Apache, there are some files and directories that have too much permission set by default. The binary program httpd can be set to be read-only by the super-user "root", and executable by the owner, group, and others for better security. The /etc/httpd/conf and /var/log/httpd directories don't need to be readable, writable or executable by other people.

```
[root@deep /]# chmod 511 /usr/sbin/httpd
[root@deep /]# chmod 700 /etc/httpd/conf/
[root@deep /]# chmod 700 /var/log/httpd/
```

Automatic indexing

If you have enabled the automatic indexing of directories in your Apache configuration file, (IndexOptions in httpd.conf), then you'll have a security issue since any requests for a directory that don't find an index file will build an index of what is in the directory. In many cases, you may only want people seeing files that you specifically link to. To turn this off, you need to remove read permissions from the DocumentRoot directory (but not the files inside it).

```
[root@deep /]# cd /home/httpd/
[root@deep httpd]# chmod 311 openna
[root@deep httpd]# ls -la

d-wx--x--x   13 webadmin webadmin     1024 Jul 28 08:12 openna
```

Now, with this modification, any requests for this protected directory should return an error message like:

```
Forbidden
You don't have permission to access "/openna/" on this server.
```

NOTE: "openna" is the DocumentRoot (the directory out of which you will serve the documents). In our configuration file example (httpd.conf) the IndexOptions directive is not used, therefore we don't need to apply this security feature.

Immunize important configuration file like `httpd.conf`

As we already know, the immutable bit can be used to prevent deletion, overwriting or creation of a symbolic link to a file. Once the httpd.conf file has been configured, it's a good idea to immunize it with the following command:

```
[root@deep /]# chattr +i /etc/httpd/conf/httpd.conf
```

Create the `.dbmpasswd` password file for users authentication

This step is necessary only if you think that you'll use an access file authentication system for your web site. Access file authentication is used when you are in the need to protect some part of your web site with a user password. With Apache, a lot of options exist to protect your site with usernames and passwords.

Step 1
The dbmmanage program utility, which comes by default with Apache, can be used to create and update usernames and passwords of HTTP users. This method use a DBM format files that is the fastest mechanism when you have thousands users to manage in your password file. First of all, it's important to change the permission of this program to be (0750/-rwxr-x---), writable only by the super-user "root", readable and executable by group and nothing for the others.

- To change the permissions on the dbmmanage program, use the following command:
```
[root@deep /]# chmod 750 /usr/bin/dbmmanage
```

Step 2
Once the permission has been set to this program, we can create the DBM format file with username and password.

- To create a username and password, use the following command:
```
[root@deep /]# /usr/bin/dbmmanage  /etc/httpd/dbmpasswd adduser gmourani
New password:
Re-type new password:
User gmourani added with password encrypted to dtkTL83yvMbFQ using crypt
```

Where </etc/httpd/> is the location where we want to create and handle this password file, <dbmpasswd> is the name we give to the password file, and <gmourani> is the name of the user we want to add in our dbmpasswd file.

NOTE: Every user that we would like to add to the dbmpasswd file doesn't need to be a real user on the system. I mean that it is not necessary to have them in the /etc/passwd file.

Step 3
If you use the dbmmanage utility of Apache Web Server to create passwords and usernames, don't forget to include in your /etc/httpd/conf/httpd.conf configuration file the part of your web site you need to protect with user password authentication.

- Edit the **httpd.conf** file (vi /etc/httpd/conf/httpd.conf) and add the following lines to protect the "private" directory of your web site (in our example: openna) with user password authentication:

```
<Directory "/home/httpd/openna/private">
    Options None
    AllowOverride AuthConfig
    AuthName "Restricted Section"
    AuthType Basic
    AuthDBUserFile /etc/httpd/dbmpasswd
    require valid-user
</Directory>
```

The path </home/httpd/openna/private> specifies the directory we want to protect with a password and username, the </etc/httpd/dbmpasswd> specifies the location of the DBM password file.

WARNING: To add the DB password authentication module to your Apache Web Server, you must be sure to include it during the configuration time of Apache with the following parameter "--add-module=src/modules/standard/mod_auth_db.c". See your Apache documentation for more information.

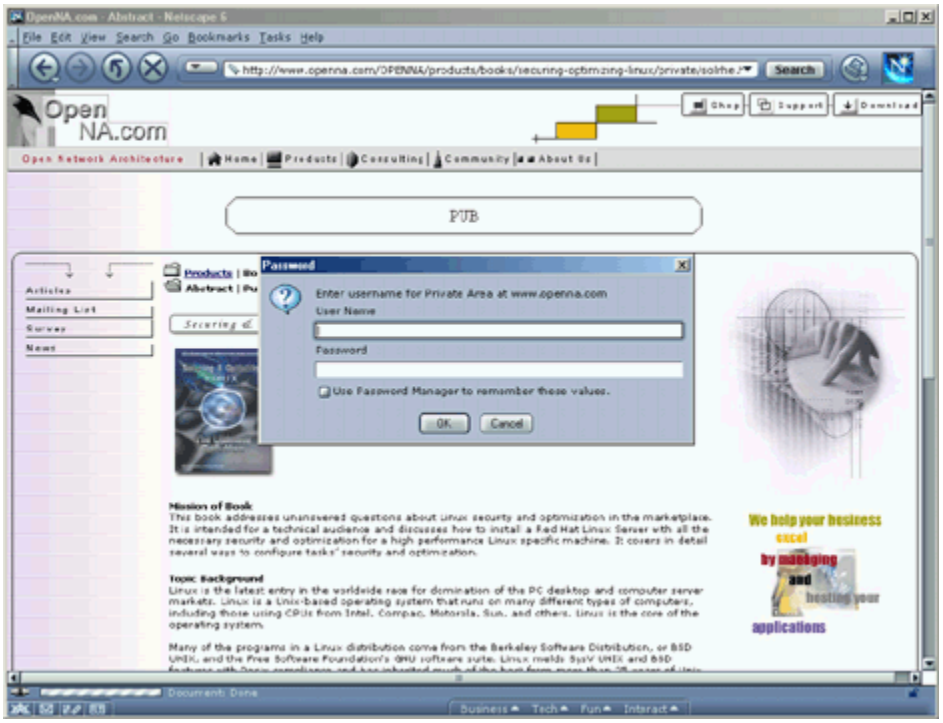
Step 4
Once the above lines have been included into the httpd.conf file of Apache to enable users password authentication feature, you must restart Apache for the changes to take effect.

- To restart Apache, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/httpd restart
Shutting down http:                [OK]
Starting httpd:                    [OK]
```

Step 5
Finally, we must test the new protected directory named private.

To verify that it works, points your web browser to the following address: http://my.domain.com/private/. The <my.domain.com> is the address where your Apache Web Server lives and </private/> is the directory protected with user password authentication.



Optimizing Apache

This section deals especially with actions we can make to improve and tighten performance of Apache. Take a note that we refer to the features available within the base installed Linux system as well as Apache program and additional software.

The mod_mmap_static module of Apache

There is a special module with the Apache distribution called mod_mmap_static that can by used to improve the performance of your Web Server. This module works by providing mappings of a statically configured list of frequently requested, but not changed, files in your RootDirectory. Therefore, if files displayed by Apache don't change often, you can use this useful module to memory-map the static documents and increase the speed of your Web Server. This means visitors to your sites get faster download times.

It's important to note that the **mod_mmap_static** module of Apache must be enabled during the configuration and compilation time of Apache before you can use it. If you have follow what were described in the previous configuration and compilation time section, this is already in Apache (--add-module=../mod_mmap_static.c).

Step 1

The magical command to map all files under a RootDirectory to a specific text file of your choice is shown below. Once again, this Apache module is only useful when you have a static web site, I mean by static, a web site where contents do not change often.

- To memory-map static documents, use the following command:

```
[root@deep /]# find /home/httpd/openna -type f -print | sed -e
's/.*/mmapfile &/' > /etc/httpd/conf/mmap.conf
```

The </home/httpd/openna> is the RootDirectory, or to be more precise, the directory out of which you will serve your documents, and the </etc/httpd/conf/mmap.conf> is the location where we want to create this file, mmap.conf, that contains a static memory-map of all documents under our RootDirectory.

> **WARNING:** If you add or update contents into your site, don't forget to reuse this command line again and restart you Web Server for the changes to take effect.

Step 2

Once the mmap.conf file has been create under the location where we have chosen to keep this file, we must include it in the httpd.conf file of Apache to be able to use its interesting features on our Web Server.

- Edit the **httpd.conf** file (vi /etc/httpd/conf/httpd.conf) and add or uncomment the following lines:

```
<IfModule mod_include.c>
Include conf/mmap.conf
</IfModule>
```

NOTE: See your Apache documentation for more information about the use of mod_mmap_static. Remember that this feature must be used only when you serve documents that don't change often on your web site.

Step 3
Finally, the last step to do is to restart the Apache Web Server for the changes to take effect:

- To restart Apache, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/httpd restart
Shutting down http:            [OK]
Starting httpd:                [OK]
```

The Zend Optimizer for PHP4 server-side scripting language

This section applies only if you chose to install and use PHP4 with Apache in your system. The Zend Optimizer is a small program located in the PHP4 Zend engine between the Zend run-time compiler and the executor that run as plugging with PHP4.

When used with PHP4, an application that uses the Zend Optimizer typically executes another 40% to 100% faster. If you intended to use this free program, you can download it from the Zend website and place its library file into your system after expanding the archive.

These installation instructions assume
Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Latest Zend Optimizer version number is 1.1.0

Packages
The following are based on information as listed by Zend as of 2001/06/05. Please regularly check at http://www.zend.com/ for the latest status.

Source codes are available from:
Zend Homepage: http://www.zend.com/
You must be sure to download: ZendOptimizer-1.1.0-PHP_4.0.5-Linux_glibc21-i386.tar.gz

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:
```
[root@deep /]# cp ZendOptimizer-version-i386.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf ZendOptimizer-version-i386.tar.gz
```

Step 2
After that, move into the newly created Zend directory and copy the file called ZendOptimizer.so under /usr/lib directory.

- To copy the ZendOptimizer.so file to your /usr/lib directory use the commands:

```
[root@deep tmp]# cd ZendOptimizer-1.1.0-PHP_4.0.5-Linux_glibc21-i386
[root@deep ZendOptimizer-1.1.0...]# cp ZendOptimizer.so /usr/lib/
```

Step 3
Now, edit your **php.ini** file (vi /etc/httpd/php.ini) and add the following two lines.

```
zend_optimizer.optimization_level=15
zend_extension="/usr/lib/ZendOptimizer.so"
```

Step 4
Finally, you must restart the Apache Web Server for the changes to take effect:

- To restart Apache, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/httpd restart
Shutting down http:            [OK]
Starting httpd:                [OK]
```

Step 5
Now, to verify if the Zend Optimizer is running use the php.php file that we have created previously by pointing your web browser to the following address: http://my.domain.com/php.php

The <my.domain.com> is the address where your Apache Web Server lives, and <php.php> is the PHP document we have created earlier to display the information and configuration of our Linux Web Server with PHP4 support.

The part of the output where the Zend Optimizer is listed will look something like this:

```
This program makes use of the Zend scripting language engine:
Zend Engine v1.0.5, Copyright (c) 1998-2001 Zend Technologies
    with Zend Optimizer v1.1.0, Copyright (c) 1998-2000, by Zend Technologies
```

The atime and noatime attributes

The atime and noatime attributes of Linux can be used to get measurable performance gains with Apache. See the chapter related to “General System Optimization” in this book for more information on the subject.

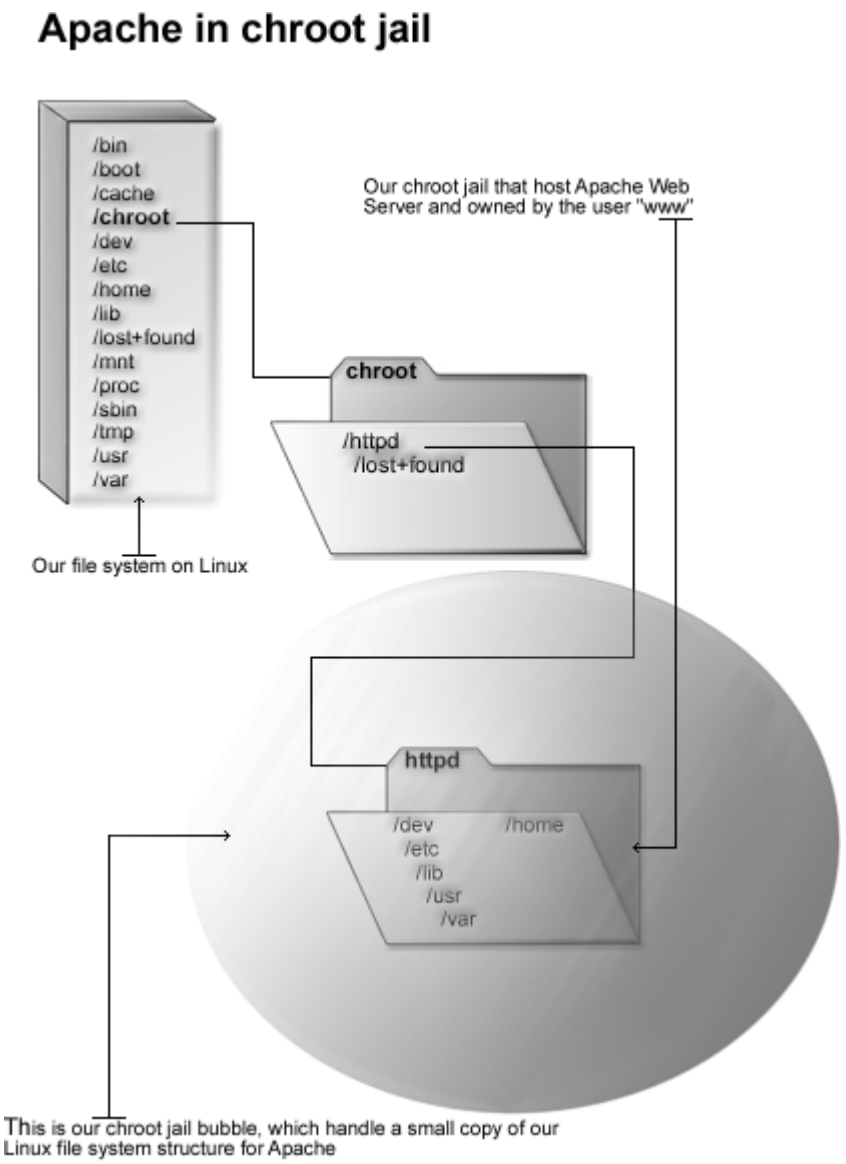
The ulimit parameter

The ulimit parameter of Linux that provide control over the resources available to the shell and to processes started by it can be used with its “-n” option to tune the maximum number of open file descriptors that may be used by a process with Apache. As for the atime and noatime attributes above, you can go to the related chapter in this book, “General System Optimization”, for more information on the subject.

Running `Apache` in a chroot jail

This part focuses on preventing `Apache` from being used as a point of break-in to the system hosting it. `Apache` by default runs **as a non-root user**, which will limit any damage to what can be done as a normal user with a local shell. Of course, allowing what amounts to an anonymous guest account falls rather short of the security requirements for most `Apache` servers, so an additional step can be taken - that is, **running `Apache` in a chroot jail**.

The main benefit of a chroot jail is that the jail will limit the portion of the file system the daemon can see to the root directory of the jail. Additionally, since the jail only needs to support `Apache`, the programs available in the jail can be extremely limited. Most importantly, there is no need for setuid-root programs (remember that `Perl` use `SUID`), which can be used to gain root access and break out of the jail. By running `Apache` in a chroot environment you can improve the security significantly in a Unix environment.



Chrooting Apache is not an easy task and has a tendency to break things. Before we embark on this, we need to first decide whether it is beneficial for you to do so. Some pros and cons are, but most certainly not limited to, the following:

Pros:

- ✓ If Apache is ever compromised, the attacker will not have access to the entire Linux file system.
- ✓ Poorly written CGI scripts that may allow someone to access your server will not work.

Cons:

- ✓ There are extra libraries you'll need to have in the chroot jail for Apache to work.
- ✓ If you use any Perl/CGI features with Apache, you will need to copy the needed binaries, Perl libraries and files to the appropriate spot within the chroot space. The same applies for SSL, PHP, and other third-party programs.

Necessary steps to run Apache with mod_ssl and PHP4 in a chroot jail:

The chrooted configuration listed below supposes that you've compiled the Apache server with the external programs **mod_ssl** and **PHP4** only. The differences in what you've compiled with Apache reside in which libraries and binaries program you'll need to copy to the chrooted directory.

Remember that if you've compiled Apache to use **mod_perl,** you must copy all the related binaries and Perl libraries to the chrooted directory. Perl resides by default in /usr/lib/perl5 and in case you use Perl features, copy the Perl directory and its subdirectories to /chroot/httpd/usr/lib/perl5. Personally I don't recommend to running Apache with Perl support in chroot jail. You can add these interpreters back in, but you lose some of the benefits of chroot.

Step 1
Add a new UID and a new GID if this is not already done for running Apache httpd daemon. This is important because running it as root defeats the purpose of the jail, and using a different UID that already exists on the system (i.e. nobody) can allow your services to access each others' resources.

Consider the scenario where a webserver is running as nobody, or any other overly used UID/GID and compromised. The cracker can now access any other processes running as nobody from within the chroot.

These are sample UID/GIDs. Check the /etc/passwd and /etc/group files for a free UID/GID number. In our configuration we'll use the numeric value "80" and UID/GID "www".

- To create the Apache user, use the following command:
  ```
  [root@deep /]# useradd -c "Apache Server" -u 80 -s /bin/false -r -d
  /home/httpd www 2>/dev/null || :
  ```

The above commands will create the group "www" with the numerical GID value 80, and the user "www" with the numerical UID value 80.

Step 2
Once the Apache user has been created, it is time to set up the chroot environment. First we need to create the chrooted Apache structure. We use /chroot/httpd for the chrooted Apache. The /chroot/httpd is just a directory on a different partition where we've decided to put Apache for more control and security.

```
[root@deep /]# /etc/rc.d/init.d/httpd stop ← Only if Apache daemon already run.
Shutting down http:             [OK]

[root@deep /]# mkdir /chroot/httpd
[root@deep /]# mkdir /chroot/httpd/dev
[root@deep /]# mkdir /chroot/httpd/lib
[root@deep /]# mkdir /chroot/httpd/etc
[root@deep /]# mkdir /chroot/httpd/home
[root@deep /]# mkdir /chroot/httpd/tmp
[root@deep /]# chmod 777 /chroot/httpd/tmp/
[root@deep /]# chmod +t /chroot/httpd/tmp/
[root@deep /]# mkdir -p /chroot/httpd/usr/sbin
[root@deep /]# mkdir -p /chroot/httpd/var/run
[root@deep /]# mkdir -p /chroot/httpd/var/log
```

We need all of the above directories because, from the point of the chroot, we're sitting at "/" and anything above this directory is inaccessible. Note that /chroot/httpd/tmp is required only if you use mod_ssl with Apache.

Step 3
After that, move the main configuration directory and all configuration files of Apache, the DocumentRoot directory and the httpd binary program of the Web Server to the chroot jail then create the special devices /dev/null and /dev/urandom which is/are require by the system to work properly. Note that /dev/urandom is requiring only if you use mod_ssl.

```
[root@deep /]# mv /etc/httpd /chroot/httpd/etc/
[root@deep /]# mv /home/httpd /chroot/httpd/home/
[root@deep /]# mv /var/log/httpd /chroot/httpd/var/log/
[root@deep /]# mv /usr/sbin/httpd /chroot/httpd/usr/sbin/
[root@deep /]# mknod /chroot/httpd/dev/null c 1 3
[root@deep /]# chmod 666 /chroot/httpd/dev/null
[root@deep /]# mknod /chroot/httpd/dev/urandom c 1 9 ← Only for mod_ssl support.
```

Step 4
This step is requiring only if you have compiled Apache with mod_ssl support. In this case, recreate a small copy of the /usr/share/ssl directory with certs, crl and private directories which handles all private and public keys to the chroot jail environment.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# mkdir –p /chroot/httpd/usr/share/ssl
[root@deep /]# cp -r /usr/share/ssl/certs /chroot/httpd/usr/share/ssl/
[root@deep /]# cp -r /usr/share/ssl/private /chroot/httpd/usr/share/ssl/
[root@deep /]# cp -r /usr/share/ssl/crl /chroot/httpd/usr/share/ssl/
```

WARNING: If you have other private and public keys related to other programs and applications into the certs and private directories, please don't copy them to the jail environment. Only copy the private and public keys related to Apache.

Step 5
This step is required only if you have compiled Apache with PHP4 support. In this case, move all of the following directories and PHP4 binaries files to the chroot jail environment and change all default permission modes of PHP4 binaries as execute-only for security reason.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# mkdir /chroot/httpd/usr/include
[root@deep /]# mkdir /chroot/httpd/usr/lib
[root@deep /]# mkdir /chroot/httpd/usr/bin
[root@deep /]# mv /usr/include/php /chroot/httpd/usr/include/
[root@deep /]# mv /usr/lib/ZendOptimizer.so /chroot/httpd/usr/lib/
[root@deep /]# mv /usr/lib/php /chroot/httpd/usr/lib/
[root@deep /]# mv /usr/bin/phpextdist /chroot/httpd/usr/bin/
[root@deep /]# mv /usr/bin/phpize /chroot/httpd/usr/bin/
[root@deep /]# mv /usr/bin/php-config /chroot/httpd/usr/bin/
[root@deep /]# mv /usr/bin/pear /chroot/httpd/usr/bin/
[root@deep /]# chmod 111 /chroot/httpd/usr/bin/*
```

Step 6
Now, we must find the shared library dependencies of httpd binary and install them into the chroot directory structure. Use the **ldd /chroot/httpd/usr/sbin/httpd** command to find out which libraries are needed. The output (depending on what you've compiled with Apache) will be something similar to:

- To find the shared library dependencies of httpd, execute the following command:

```
[root@deep /]# ldd /chroot/httpd/usr/sbin/httpd
libpam.so.0 => /lib/libpam.so.0 (0x4001b000)
libdl.so.2 => /lib/libdl.so.2 (0x40023000)
libz.so.1 => /usr/lib/libz.so.1 (0x40026000)
libpng.so.2 => /usr/lib/libpng.so.2 (0x40034000)
libgd.so.1.8 => /usr/lib/libgd.so.1.8 (0x40055000)
libresolv.so.2 => /lib/libresolv.so.2 (0x40086000)
libm.so.6 => /lib/libm.so.6 (0x40099000)
libcrypt.so.1 => /lib/libcrypt.so.1 (0x400b9000)
libnsl.so.1 => /lib/libnsl.so.1 (0x400e6000)
libc.so.6 => /lib/libc.so.6 (0x400fd000)
libttf.so.2 => /usr/lib/libttf.so.2 (0x40223000)
libjpeg.so.62 => /usr/lib/libjpeg.so.62 (0x4024a000)
/lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

What we can see here is the fact that depending of what programs have been compiled and included with Apache, the shared library dependencies may change. If you make attention to the above dependencies, you will see for example that libz.so.1, libpng.so.2, libgd.so.1.8, libttf.so.2, and libjpeg.so.62 files which have been compiled during the PHP4 configuration time are require in the chroot jail for the Web Server to work properly.

Therefore it is always important and vital to execute the **ldd** command to find which libraries are require depending of programs you may have compiled and included with your Web Server.

Step 7
Once the required libraries have been identified, copy them to the appropriate location into the chroot jail. In our example these are the shared libraries identified above.

```
[root@deep /]# cp /lib/libpam.so.0 /chroot/httpd/lib/
[root@deep /]# cp /lib/libdl.so.2 /chroot/httpd/lib/
[root@deep /]# cp /usr/lib/libz.so.1 /chroot/httpd/usr/lib/
[root@deep /]# cp /usr/lib/libpng.so.2 /chroot/httpd/usr/lib/
[root@deep /]# cp /usr/lib/libgd.so.1.8 /chroot/httpd/usr/lib/
[root@deep /]# cp /lib/libresolv.so.2 /chroot/httpd/lib/
[root@deep /]# cp /lib/libm.so.6 /chroot/httpd/lib/
[root@deep /]# cp /lib/libcrypt.so.1 /chroot/httpd/lib/
[root@deep /]# cp /lib/libnsl.so.1 /chroot/httpd/lib/
[root@deep /]# cp /lib/libc.so.6 /chroot/httpd/lib/
[root@deep /]# cp /usr/lib/libttf.so.2 /chroot/httpd/usr/lib/
[root@deep /]# cp /usr/lib/libjpeg.so.62 /chroot/httpd/usr/lib/
[root@deep /]# cp /lib/ld-linux.so.2 /chroot/httpd/lib/
[root@deep /]# strip -R .comment /chroot/httpd/usr/lib/*
```

You'll also need the following extra libraries for some network functions, like resolving:

```
[root@deep /]# cp /lib/libnss_compat* /chroot/httpd/lib/
[root@deep /]# cp /lib/libnss_dns* /chroot/httpd/lib/
[root@deep /]# cp /lib/libnss_files* /chroot/httpd/lib/
[root@deep /]# strip -R .comment /chroot/httpd/lib/*
```

NOTE: The "`strip -R .comment`" command will remove all the named section "`.comment`" from the libraries files under the `/lib` directory and will make them smaller in size and can help in performance of them.

Step 8
Now we need to copy the `passwd` and `group` files inside the `/chroot/httpd/etc` chrooted directory. Next, we'll remove all entries except for the user that `Apache` runs as in both files (`passwd` and `group`).

```
[root@deep /]# cp /etc/passwd /chroot/httpd/etc/
[root@deep /]# cp /etc/group /chroot/httpd/etc/
```

- Edit the **`passwd`** file under the chroot jail (`vi /chroot/httpd/etc/passwd`) and delete all entries except for the user `Apache` run as (in our configuration, it's "`www`"):

```
www:x:80:80:Apache Server:/home/httpd:/bin/false
```

- Edit the **`group`** file under the chroot jail (`vi /chroot/httpd/etc/group`) and delete all entries except the group `Apache` run as (in our configuration it's "`www`"):

```
www:x:80:
```

Step 9
You will also need **/etc/resolv.conf**, **/etc/nsswitch.conf**, **/etc/localtime**, and **/etc/hosts** files in your chroot jail structure.

```
[root@deep /]# cp /etc/resolv.conf /chroot/httpd/etc/
[root@deep /]# cp /etc/nsswitch.conf /chroot/httpd/etc/
[root@deep /]# cp /etc/localtime /chroot/httpd/etc/
[root@deep /]# cp /etc/hosts /chroot/httpd/etc/
```

Step 10
Now we must set some files in the chroot jail directory immutable for better security.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cd /chroot/httpd/etc/
[root@deep etc]# chattr +i passwd
[root@deep etc]# chattr +i group
[root@deep etc]# chattr +i /httpd/conf/httpd.conf
[root@deep etc]# chattr +i resolv.conf
[root@deep etc]# chattr +i hosts
[root@deep etc]# chattr +i nsswitch.conf
```

WARNING: Don't forget to remove the immutable bit on these files if you have some modifications to apport to them with the command "`chattr -i`".

Step 11
One of the last steps to do is to inform the `syslogd` daemon about the new `Apache` chrooted service. Normally, processes talk to `syslogd` through `/dev/log`. As a result of the chroot jail, this won't be possible, so program `syslogd` needs to be told to listen to the `/chroot/httpd/dev/log`. To do this, edit the `syslog` startup script to specify additional places to listen.

- Edit the **syslog** script (`vi +24 /etc/rc.d/init.d/syslog`) and change the line:

  ```
  daemon syslogd -m 0
  ```

  To read:

  ```
  daemon syslogd -m 0 -a /chroot/httpd/dev/log
  ```

Step 12
The default `httpd` script file of `Apache` starts the daemon "`httpd`" outside the chroot jail. We must change it to now start `httpd` from the chroot jail.

- Edit the **httpd** script file (`vi /etc/rc.d/init.d/httpd`) and change the lines:

  ```
  daemon httpd -DSSL
  ```

  To read:

  ```
  /usr/sbin/chroot /chroot/httpd/ /usr/sbin/httpd -DSSL
  ```

```
rm -f /var/run/httpd.pid
```

To read:

```
rm -f /chroot/httpd/var/run/httpd.pid
```

Step 13
Finally, we must test the new chrooted jail configuration of our Apache Web Server.

- The first thing to do is to restart our syslogd daemon with the following command:
```
[root@deep /]# /etc/rc.d/init.d/syslog restart
Shutting down kernel logger:            [OK]
Shutting down system logger:            [OK]
Starting system logger:                 [OK]
Starting kernel logger:                 [OK]
```

- Now, start the new chrooted jail Apache with the following command:
```
[root@deep /]# /etc/rc.d/init.d/httpd start
Starting httpd:                         [OK]
```

- If you don't get any errors, do a **ps ax | grep httpd** and see if we're running:
```
[root@deep /]# ps ax | grep httpd
14373 ?        S      0:00 httpd -DSSL
14376 ?        S      0:00 httpd -DSSL
14377 ?        S      0:00 httpd -DSSL
14378 ?        S      0:00 httpd -DSSL
14379 ?        S      0:00 httpd -DSSL
14380 ?        S      0:00 httpd -DSSL
14381 ?        S      0:00 httpd -DSSL
14382 ?        S      0:00 httpd -DSSL
14383 ?        S      0:00 httpd -DSSL
14384 ?        S      0:00 httpd -DSSL
14385 ?        S      0:00 httpd -DSSL
14386 ?        S      0:00 httpd -DSSL
14387 ?        S      0:00 httpd -DSSL
14388 ?        S      0:00 httpd -DSSL
14389 ?        S      0:00 httpd -DSSL
14390 ?        S      0:00 httpd -DSSL
14391 ?        S      0:00 httpd -DSSL
14397 ?        S      0:00 httpd -DSSL
14476 ?        S      0:00 httpd -DSSL
14477 ?        S      0:00 httpd -DSSL
14478 ?        S      0:00 httpd -DSSL
```

If so, lets check to make sure it's chrooted by picking out one of its process numbers and doing **ls -la /proc/that_process_number/root/**.

```
[root@deep /]# ls -la /proc/14373/root/
```

If you see something like the following, congratulations! Your Apache with mod_ssl and PHP4 in chroot jail is working.

```
dev
etc
home
lib
tmp
usr
var
```

/etc/logrotate.d/httpd: The New Apache Log Rotation File

With all modifications for running Apache in chroot jail, the Apache logs files resides now in the /chroot/httpd/var/log/httpd directory instead of /var/log/httpd, and for this reason we need to modify the existing /etc/logrotate.d/httpd file to point to the new chrooted directory. Also, because we've compiled Apache with mod_ssl, we'll add one more line to permit the logrotate program to rotate the ssl_request_log and ssl_engine_log files.

Configure your /etc/logrotate.d/httpd file to rotate your log files each week automatically. We must change the default one to fit our requirements and operating system. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy our needs.

Edit the **httpd** file (vi /etc/logrotate.d/httpd) and add or modify:

```
/chroot/httpd/var/log/httpd/access_log {
    missingok
    postrotate
        /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
    endscript
}

/chroot/httpd/var/log/httpd/error_log {
    missingok
    postrotate
        /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
    endscript
}

/chroot/httpd/var/log/httpd/ssl_request_log {
    missingok
    postrotate
        /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
    endscript
}

/chroot/httpd/var/log/httpd/ssl_engine_log {
    missingok
    postrotate
        /usr/bin/killall -HUP /chroot/httpd/usr/sbin/httpd
    endscript
}
```

List of installed Apache files on your system

> /etc/rc.d/init.d/httpd
> /etc/logrotate.d/httpd
> /etc/httpd
> /etc/httpd/conf
> /etc/httpd/conf/httpd.conf.default
> /etc/httpd/conf/httpd.conf
> /etc/httpd/conf/mime.types
> /etc/httpd/conf/magic
> /etc/httpd/php.ini
> /home/httpd
> /home/httpd/cgi-bin
> /usr/bin/htpasswd
> /usr/bin/htdigest
> /usr/bin/dbmmanage
> /usr/include/apache
> /usr/include/apache/xml
> /usr/include/apache/xml/asciitab.h
> /usr/include/apache/buff.h
> /usr/include/apache/compat.h
> /usr/include/apache/conf.h
> /usr/include/apache/explain.h
> /usr/include/apache/fnmatch.h
> /usr/include/apache/hsregex.h
> /usr/include/apache/http_conf_globals.h
> /usr/include/apache/http_config.h
> /usr/include/apache/http_core.h
> /usr/include/apache/httpd.h
> /usr/include/apache/http_log.h
> /usr/include/apache/http_main.h
> /usr/include/apache/http_protocol.h
> /usr/include/apache/http_request.h
> /usr/include/apache/http_vhost.h
> /usr/include/apache/multithread.h
> /usr/include/apache/rfc1413.h

> /usr/include/apache/xml/hashtable.h
> /usr/include/apache/xml/iasciitab.h
> /usr/include/apache/xml/latin1tab.h
> /usr/include/apache/xml/nametab.h
> /usr/include/apache/xml/utf8tab.h
> /usr/include/apache/xml/xmldef.h
> /usr/include/apache/xml/xmlparse.h
> /usr/include/apache/xml/xmlrole.h
> /usr/include/apache/xml/xmltok.h
> /usr/include/apache/xml/xmltok_impl.h
> /usr/include/apache/ap_alloc.h
> /usr/include/apache/ap_compat.h
> /usr/include/apache/ap_config_auto.h
> /usr/include/apache/ap_config.h
> /usr/include/apache/ap_ctx.h
> /usr/include/apache/ap_ctype.h
> /usr/include/apache/ap.h
> /usr/include/apache/ap_hook.h
> /usr/include/apache/ap_md5.h
> /usr/include/apache/ap_mm.h
> /usr/include/apache/ap_mmn.h
> /usr/include/apache/ap_sha1.h
> /usr/include/apache/scoreboard.h
> /usr/include/apache/util_date.h
> /usr/include/apache/util_md5.h
> /usr/include/apache/util_script.h
> /usr/include/apache/util_uri.h
> /usr/include/apache/os.h
> /usr/include/apache/os-inline.c
> /usr/lib/apache
> /usr/sbin/httpd
> /usr/sbin/ab
> /usr/sbin/logresolve
> /usr/sbin/rotatelogs
> /usr/sbin/apxs
> /usr/share/man/man1/htpasswd.1
> /usr/share/man/man1/htdigest.1
> /usr/share/man/man1/dbmmanage.1
> /usr/share/man/man8/httpd.8
> /usr/share/man/man8/ab.8
> /usr/share/man/man8/logresolve.8
> /usr/share/man/man8/rotatelogs.8
> /usr/share/man/man8/apxs.8
> /var/log/httpd

List of installed `PHP4` files on your system

> /usr/bin/phpextdist
> /usr/bin/phpize
> /usr/bin/php-config
> /usr/bin/pear
> /usr/include/php
> /usr/include/php/Zend
> /usr/include/php/Zend/acconfig.h
> /usr/include/php/Zend/FlexLexer.h
> /usr/include/php/Zend/modules.h
> /usr/include/php/Zend/zend_alloc.h
> /usr/include/php/Zend/zend_API.h
> /usr/include/php/Zend/zend_builtin_functions.h
> /usr/include/php/Zend/zend_compile.h
> /usr/include/php/Zend/zend_config.h
> /usr/include/php/Zend/zend_config.w32.h
> /usr/include/php/Zend/zend_constants.h
> /usr/include/php/Zend/zend_dynamic_array.h
> /usr/include/php/Zend/zend_errors.h
> /usr/include/php/Zend/zend_execute.h
> /usr/include/php/Zend/zend_execute_locks.h
> /usr/include/php/Zend/zend_extensions.h
> /usr/include/php/Zend/zend_fast_cache.h
> /usr/include/php/Zend/zend_globals.h
> /usr/include/php/Zend/zend_globals_macros.h
> /usr/include/php/Zend/zend.h
> /usr/include/php/Zend/zend_hash.h
> /usr/include/php/Zend/zend_highlight.h
> /usr/include/php/Zend/zend_indent.h
> /usr/include/php/Zend/zend_list.h
> /usr/include/php/Zend/zend_llist.h
> /usr/include/php/Zend/zend_operators.h
> /usr/include/php/Zend/zend-parser.h
> /usr/include/php/Zend/zend_ptr_stack.h
> /usr/include/php/Zend/zend-scanner.h
> /usr/include/php/Zend/zend_stack.h
> /usr/include/php/Zend/zend_static_allocator.h
> /usr/include/php/Zend/zend_variables.h
> /usr/include/php/TSRM
> /usr/include/php/TSRM/acconfig.h
> /usr/include/php/TSRM/readdir.h
> /usr/include/php/TSRM/tsrm_config_common.h
> /usr/include/php/TSRM/tsrm_config.h
> /usr/include/php/TSRM/tsrm_config.w32.h
> /usr/include/php/ext/xml/expat/xmlparse/expat_hashtable.h
> /usr/include/php/ext/xml/expat/xmlparse/xmlparse.h
> /usr/include/php/ext/xml/expat/xmltok
> /usr/include/php/ext/xml/expat/xmltok/asciitab.h
> /usr/include/php/ext/xml/expat/xmltok/iasciitab.h
> /usr/include/php/ext/xml/expat/xmltok/latin1tab.h
> /usr/include/php/ext/xml/expat/xmltok/nametab.h
> /usr/include/php/ext/xml/expat/xmltok/utf8tab.h
> /usr/include/php/ext/xml/expat/xmltok/xmldef.h
> /usr/include/php/ext/xml/expat/xmltok/xmlrole.h
> /usr/include/php/ext/xml/expat/xmltok/xmltok.h
> /usr/include/php/ext/xml/expat/xmltok/xmltok_impl.h
> /usr/include/php/ext/xml/php_xml.h
> /usr/include/php/main
> /usr/include/php/main/configuration-parser.h
> /usr/include/php/main/config.w32.h
> /usr/include/php/main/fdfdata.h
> /usr/include/php/main/fopen-wrappers.h
> /usr/include/php/main/internal_functions_registry.h
> /usr/include/php/main/logos.h
> /usr/include/php/main/php3_compat.h
> /usr/include/php/main/php_compat.h
> /usr/include/php/main/php_content_types.h
> /usr/include/php/main/php_globals.h
> /usr/include/php/main/php.h
> /usr/include/php/main/php_ini.h
> /usr/include/php/main/php_main.h
> /usr/include/php/main/php_network.h
> /usr/include/php/main/php_open_temporary_file.h
> /usr/include/php/main/php_reentrancy.h
> /usr/include/php/main/php_regex.h
> /usr/include/php/main/php_syslog.h
> /usr/include/php/main/php_ticks.h
> /usr/include/php/main/php_variables.h
> /usr/include/php/main/php_version.h
> /usr/include/php/main/rfc1867.h
> /usr/include/php/main/safe_mode.h
> /usr/include/php/main/SAPI.h
> /usr/include/php/main/snprintf.h
> /usr/include/php/main/win95nt.h
> /usr/include/php/regex
> /usr/include/php/regex/cclass.h
> /usr/include/php/regex/cname.h

> /usr/include/php/TSRM/TSRM.h
> /usr/include/php/TSRM/tsrm_strtok_r.h
> /usr/include/php/TSRM/tsrm_virtual_cwd.h
> /usr/include/php/ext
> /usr/include/php/ext/standard
> /usr/include/php/ext/standard/base64.h
> /usr/include/php/ext/standard/basic_functions.h
> /usr/include/php/ext/standard/cyr_convert.h
> /usr/include/php/ext/standard/datetime.h
> /usr/include/php/ext/standard/dl.h
> /usr/include/php/ext/standard/dns.h
> /usr/include/php/ext/standard/exec.h
> /usr/include/php/ext/standard/file.h
> /usr/include/php/ext/standard/flock_compat.h
> /usr/include/php/ext/standard/fsock.h
> /usr/include/php/ext/standard/head.h
> /usr/include/php/ext/standard/html.h
> /usr/include/php/ext/standard/info.h
> /usr/include/php/ext/standard/md5.h
> /usr/include/php/ext/standard/microtime.h
> /usr/include/php/ext/standard/pack.h
> /usr/include/php/ext/standard/pageinfo.h
> /usr/include/php/ext/standard/php_array.h
> /usr/include/php/ext/standard/php_assert.h
> /usr/include/php/ext/standard/php_browscap.h
> /usr/include/php/ext/standard/php_crypt.h
> /usr/include/php/ext/standard/php_dir.h
> /usr/include/php/ext/standard/php_ext_syslog.h
> /usr/include/php/ext/standard/php_filestat.h
> /usr/include/php/ext/standard/php_image.h
> /usr/include/php/ext/standard/php_incomplete_class.h
> /usr/include/php/ext/standard/php_iptc.h
> /usr/include/php/ext/standard/php_lcg.h
> /usr/include/php/ext/standard/php_link.h
> /usr/include/php/ext/standard/php_mail.h
> /usr/include/php/ext/standard/php_math.h
> /usr/include/php/ext/standard/php_metaphone.h
> /usr/include/php/ext/standard/php_output.h
> /usr/include/php/ext/standard/php_parsedate.h
> /usr/include/php/ext/standard/php_rand.h
> /usr/include/php/ext/standard/php_smart_str.h
> /usr/include/php/ext/standard/php_standard.h
> /usr/include/php/ext/standard/php_string.h
> /usr/include/php/ext/standard/php_var.h
> /usr/include/php/ext/standard/quot_print.h
> /usr/include/php/ext/standard/reg.h
> /usr/include/php/ext/standard/scanf.h
> /usr/include/php/ext/standard/type.h
> /usr/include/php/ext/standard/uniqid.h
> /usr/include/php/ext/standard/url.h
> /usr/include/php/ext/standard/url_scanner_ex.h
> /usr/include/php/ext/standard/url_scanner.h
> /usr/include/php/ext/xml
> /usr/include/php/ext/xml/expat
> /usr/include/php/ext/xml/expat/xmlparse
> /usr/include/php/regex/regex2.h
> /usr/include/php/regex/regex_extra.h
> /usr/include/php/regex/regex.h
> /usr/include/php/regex/utils.h
> /usr/include/php/acconfig.h
> /usr/include/php/build-defs.h
> /usr/include/php/php_config.h
> /usr/include/php/php_version.h
> /usr/lib/php
> /usr/lib/php/extensions
> /usr/lib/php/extensions/no-debug-non-zts-20000809
> /usr/lib/php/Benchmark
> /usr/lib/php/Benchmark/Iterate.php
> /usr/lib/php/Benchmark/Timer.php
> /usr/lib/php/DB
> /usr/lib/php/DB/common.php
> /usr/lib/php/DB/ibase.php
> /usr/lib/php/DB/msql.php
> /usr/lib/php/DB/mssql.php
> /usr/lib/php/DB/mysql.php
> /usr/lib/php/DB/oci8.php
> /usr/lib/php/DB/odbc.php
> /usr/lib/php/DB/pgsql.php
> /usr/lib/php/DB/storage.php
> /usr/lib/php/DB/sybase.php
> /usr/lib/php/File
> /usr/lib/php/File/Find.php
> /usr/lib/php/File/SearchReplace.php
> /usr/lib/php/HTML
> /usr/lib/php/HTML/Form.php
> /usr/lib/php/Net
> /usr/lib/php/Net/Socket.php
> /usr/lib/php/Payment
> /usr/lib/php/Payment/Verisign.php
> /usr/lib/php/PEAR
> /usr/lib/php/PEAR/Installer.php
> /usr/lib/php/XML
> /usr/lib/php/XML/Parser.php
> /usr/lib/php/build
> /usr/lib/php/build/pear.m4
> /usr/lib/php/build/fastgen.sh
> /usr/lib/php/build/library.mk
> /usr/lib/php/build/ltlib.mk
> /usr/lib/php/build/mkdep.awk
> /usr/lib/php/build/program.mk
> /usr/lib/php/build/rules.mk
> /usr/lib/php/build/rules_common.mk
> /usr/lib/php/build/rules_pear.mk
> /usr/lib/php/build/dynlib.mk
> /usr/lib/php/build/shtool
> /usr/lib/php/build/dynlib.m4
> /usr/lib/php/build/acinclude.m4
> /usr/lib/php/DB.php
> /usr/lib/php/HTTP.php
> /usr/lib/php/PEAR.php

List of installed `mod_perl` files on your system

> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/mod_perl.exp
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/typemap
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/Symbol
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/Symbol/Symbol.so
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/Symbol/Symbol.bs
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/Leak
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/Leak/Leak.so
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/Leak/Leak.bs
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include

> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/support
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/support/suexec.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/regex
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/regex/cclass.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/regex/utils.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/regex/regex2.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/regex/cname.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/xmlparse.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/utf8tab.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/xmltok_impl.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/latin1tab.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/hashtable.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/xmlrole.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/nametab.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/xmltok.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/asciitab.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/xmldef.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/expat-lite/iasciitab.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/sdbm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/sdbm/sdbm_tune.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/sdbm/sdbm_pair.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/lib/sdbm/sdbm.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_config.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_config.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/util_date.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/compat.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_mmn.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/util_script.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_md5.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_ctype.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_conf_globals.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/httpd.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_main.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_log.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_sha1.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/explain.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/rfc1413.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_protocol.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_request.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_hook.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_core.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/multithread.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/http_vhost.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/buff.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_mm.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_ctx.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_alloc.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/scoreboard.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_compat.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/hsregex.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/ap_config_auto.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/fnmatch.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/util_uri.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/util_md5.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/include/conf.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/unix
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/unix/os-inline.c
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/unix/os.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/tpf
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/tpf/os.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/tpf/ebcdic.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/tpf/os-inline.c
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/os390
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/os390/ebcdic.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/os390/os.h

> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/os390/os-inline.c
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/service.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/resource.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/getopt.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/os.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/readdir.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/passwd.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/registry.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/installer
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/installer/installdll
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/installer/installdll/test
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/installer/installdll/test/test.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/win32/installer/installdll/test/resource.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/netware
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/netware/os.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/netware/precomp.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/netware/test_char.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/netware/uri_delims.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/netware/getopt.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/os2
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/os2/os.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/os2/os-inline.c
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/mpeix
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/mpeix/os.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/mpeix/os-inline.c
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/bs2000
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/bs2000/ebcdic.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/bs2000/os.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/os/bs2000/os-inline.c
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/php4
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/php4/mod_php4.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/proxy
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/proxy/mod_proxy.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/perl
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/perl/mod_perl_xs.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/perl/apache_inc.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/perl/mod_perl_version.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/perl/perl_PL.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/perl/mod_perl.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/standard
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/standard/mod_rewrite.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/ssl
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/ssl/ssl_expr.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/ssl/ssl_expr_parse.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/ssl/ssl_util_sdbm.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/ssl/ssl_util_table.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/ssl/mod_ssl.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/Apache/include/modules/ssl/ssl_util_ssl.h
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/mod_perl
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/auto/mod_perl/.packlist
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl.pod
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl_hooks.pm.PL
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl_cvs.pod
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl_method_handlers.pod
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl_tuning.pod
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl_traps.pod
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/mod_perl_hooks.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/cgi_to_mod_perl.pod
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Bundle
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Bundle/Apache.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Registry.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/PerlSections.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/PerlRun.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Debug.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/src.pm

> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/RedirectLogFix.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Include.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/FakeRequest.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Options.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/RegistryLoader.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/MyConfig.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/ExtUtils.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Symdump.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Status.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/StatINC.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/RegistryBB.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/test.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/SizeLimit.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Resource.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/RegistryNG.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/httpd_conf.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/SIG.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Opcode.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Connection.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Constants.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/File.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Leak.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Log.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/ModuleConfig.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/PerlRunXS.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Server.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Symbol.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Table.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/URI.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Util.pm
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Constants
> /usr/lib/perl5/site_perl/5.6.0/i386-linux/Apache/Constants/Exports.pm
> /usr/share/man/man3/Apache.3pm
> /usr/share/man/man3/Apache::Constants.3pm
> /usr/share/man/man3/Apache::File.3pm
> /usr/share/man/man3/Apache::Leak.3pm
> /usr/share/man/man3/Apache::Log.3pm
> /usr/share/man/man3/Apache::PerlRunXS.3pm
> /usr/share/man/man3/Apache::Symbol.3pm
> /usr/share/man/man3/Apache::Table.3pm
> /usr/share/man/man3/Apache::URI.3pm
> /usr/share/man/man3/Apache::Util.3pm
> /usr/share/man/man3/mod_perl_cvs.3pm
> /usr/share/man/man3/Apache::Registry.3pm
> /usr/share/man/man3/Apache::SizeLimit.3pm
> /usr/share/man/man3/cgi_to_mod_perl.3pm
> /usr/share/man/man3/Apache::Resource.3pm
> /usr/share/man/man3/Apache::PerlSections.3pm
> /usr/share/man/man3/Apache::PerlRun.3pm
> /usr/share/man/man3/Apache::Debug.3pm
> /usr/share/man/man3/Apache::Symdump.3pm
> /usr/share/man/man3/mod_perl_tuning.3pm
> /usr/share/man/man3/Apache::Status.3pm
> /usr/share/man/man3/Apache::RedirectLogFix.3pm
> /usr/share/man/man3/Apache::ExtUtils.3pm
> /usr/share/man/man3/mod_perl_method_handlers.3pm
> /usr/share/man/man3/Apache::Include.3pm
> /usr/share/man/man3/Apache::StatINC.3pm
> /usr/share/man/man3/Apache::test.3pm
> /usr/share/man/man3/Apache::RegistryLoader.3pm
> /usr/share/man/man3/Apache::httpd_conf.3pm
> /usr/share/man/man3/Apache::FakeRequest.3pm
> /usr/share/man/man3/mod_perl.3pm
> /usr/share/man/man3/Apache::src.3pm
> /usr/share/man/man3/mod_perl_traps.3pm
> /usr/share/man/man3/Apache::SIG.3pm
> /usr/share/man/man3/Bundle::Apache.3pm
> /usr/share/man/man3/Apache::Options.3pm

30 Other Server - Samba File Sharing Server

In this Chapter

Linux Samba File Sharing Server

Abstract

Enterprise-level organizations often handle many kinds of different operating systems, and have the need to keep them in a networked environment for files sharing and printers. Employees may work on workstations like Linux, Microsoft Windows 95/98/2000/NT, OS/2 or Novel, and still need to access the server in their daily work. A Linux server with Samba support can be used to respond for these kinds of activities.

Samba is a strong network service for file and print sharing that works on the majority of operating systems available today. When well implemented by the administrator, it's faster and more secure than the native file sharing services available on Microsoft Windows machines.

As explained in the README file of Samba:
Samba is the protocol by which a lot of PC-related machines share files and printers, and other information, such as lists of available files and printers. Operating systems that support this natively include Windows 95/98/2000/NT, OS/2, and Linux, and add on packages that achieve the same thing are available for DOS, Windows, VMS, Unix of all kinds, MVS, and more.

Apple Macs and some Web Browsers can speak this protocol as well. Alternatives to SMB include Netware, NFS, AppleTalk, Banyan Vines, Decnet etc; many of these have advantages but none are both public specifications and widely implemented in desktop machines by default. Samba software includes an SMB server, to provide Windows NT and LAN Manager-style file and print services to SMB clients such as Windows 2000, Warp Server, smbfs and others, a Net BIOS (rfc1001/1002) name server, which amongst other things gives browsing support, an ftp-like SMB client so that you can access PC resources (disks and printers) from Unix, Netware and other operating systems, and finally, a tar extension to the client for backing up PCs.

In this chapter, we will explain and cover some of the basic ways in which you can adjust the configuration to improve the server's performance. Also, for the interested users, we'll provide a procedure to run Samba with SSL protocol support. Running Samba with SSL support will work perfectly for Unix-to-Unix platforms but not for Windows to Unix. This is in particularly due to the fact that at this time Microsoft has not reviewed its File Sharing system on Windows.

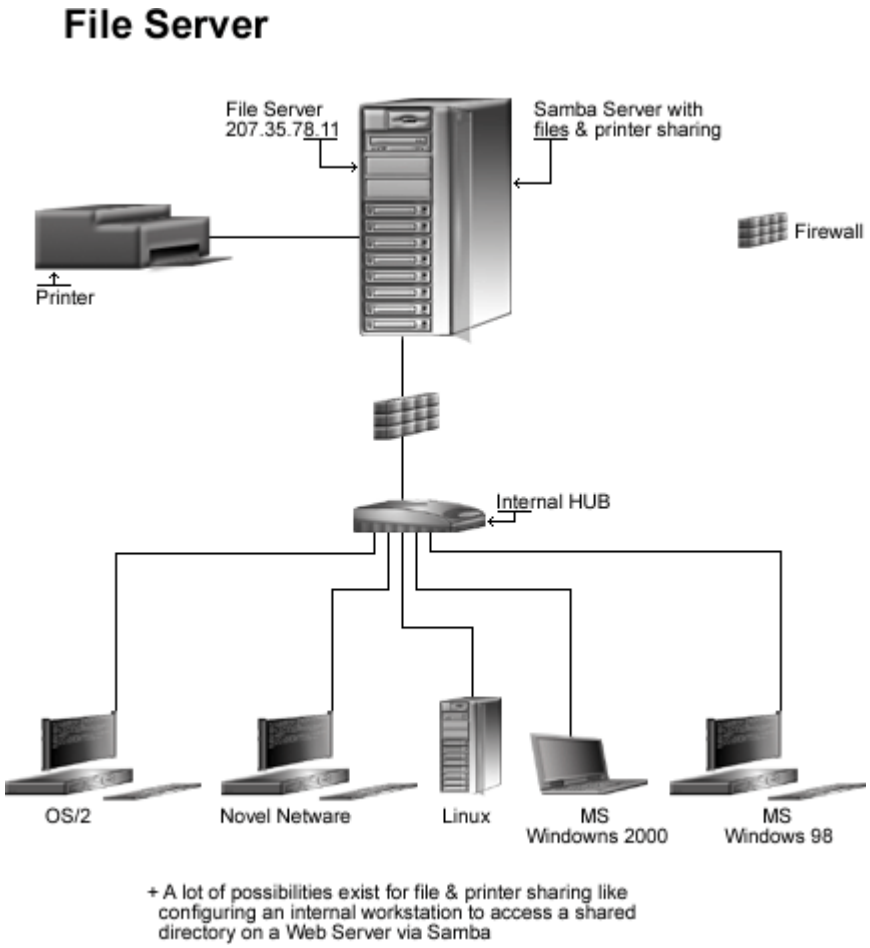
Recommended RPM packages to be installed for a Samba Server

A minimal configuration provides the basic set of packages required by the Linux operating system. Minimal configuration is a perfect starting point for building secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a File Sharing (Net BIOS) server running on Samba software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install Samba by RPM package. Therefore, samba, samba-common, and samba-client RPM packages are already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list.

basesystem	ed	less	passwd	slocate
bash	file	libstdc++	popt	sysklogd
bdflush	filesystem	libtermcap	procps	syslinux
bind	fileutils	lilo	psmisc	SysVinit
bzip2	findutils	logrotate	pwdb	tar
chkconfig	gawk	losetup	qmail	termcap
console-tools	gdbm	MAKEDEV	**quota**	textutils
cpio	gettext	man	readline	tmpwatch
cracklib	glib	mingetty	rootfiles	utempter
cracklib-dicts	glibc	mktemp	rpm	util-linux
crontabs	glibc-common	mount	**samba**	vim-common
db1	grep	ncurses	**samba-common**	vim-minimal
db2	groff	net-tools	**samba-client**	vixie-cron
db3	gzip	newt	sed	words
dev	info	openssh	setup	which
devfsd	initscripts	openssh-server	sh-utils	zlib
diffutils	iptables	openssl	shadow-utils	
e2fsprogs	kernel	pam	slang	

Tested and fully functional on OpenNA.com.


File Server
File Server
207.35.78.11
Samba Server with
files & printer sharing
Firewall
Printer
Internal HUB
OS/2
Novel Netware
Linux
MS
Windowns 2000
MS
Windows 98
+ A lot of possibilities exist for file & printer sharing like
configuring an internal workstation to access a shared
directory on a Web Server via Samba

These installation instructions assume

Commands are Unix-compatible.
The source path is /var/tmp (note that other paths are possible, as personal discretion).
Installations were tested on Red Hat 7.1.
All steps in the installation will happen using the super-user account "root".
Whether kernel recompilation may be required: No
Latest Samba version number is 2.2.0

Packages

The following are based on information as listed by Samba as of 2001/03/13. Please regularly check at www.samba.org for the latest status.

Source code is available from:
Samba Homepage: http://us1.samba.org/samba/samba.html
Samba FTP Sire: 167.216.222.41
You must be sure to download: samba-2.2.0.tar.gz

Prerequisites

Samba requires that the listed software below be already installed on your system to be able to compile successfully. If this is not the case, you must install it.

- ✓ To enable and use SSL encryption support into the software, OpenSSL library should be already installed on your system.

> **NOTE:** For more information on OpenSSL software, see its related chapter in this book.

Pristine source

If you don't use the RPM package to install this program, it will be difficult for you to locate all installed files into the system in the eventuality of an updated in the future. To solve the problem, it is a good idea to make a list of files on the system before you install Samba, and one afterwards, and then compare them using the **diff** utility of Linux to find out what files are placed where.

- Simply run the following command before installing the software:
  ```
  [root@deep /root]#  find /* > Samba1
  ```
- And the following one after you install the software:
  ```
  [root@deep /root]#  find /* > Samba2
  ```
- Then use the following command to get a list of what changed:
  ```
  [root@deep /root]#  diff Samba1 Samba2 > Samba-Installed
  ```

With this procedure, if any upgrade appears, all you have to do is to read the generated list of what files were added or changed by the program and remove them manually from your system before installing the new software. Related to our example above, we use the /root directory of the system to stock all generated list files.

Compiling - Optimizing & Installing Samba

Below are the required steps that you must make to compile and optimize the Samba software before installing it into your Linux system. First off, we install the program as user 'root' so as to avoid authorization problems.

Step 1
Once you get the program from the main software site you must copy it to the /var/tmp directory and change to this location before expanding the archive.

- These procedures can be accomplished with the following commands:

```
[root@deep /]# cp samba-version.tar.gz /var/tmp/
[root@deep /]# cd /var/tmp/
[root@deep tmp]# tar xzpf samba-version.tar.gz
```

Step 2
After that, move into the newly created Samba source subdirectory called "source" and perform the following steps before configuring and optimizing Samba for your system.

- To move into the newly created Samba source subdirectory use the command:

```
[root@deep tmp]# cd samba-2.2.0/source/
```

Step 3
There are some source files to modify before going in configuration and compilation of the program; the changes allow us to fix some problems. The first modification that we do is to relocate the lib directory of Samba to be under the /usr/bin directory.

- Edit the **smbsh.in** file (vi +3 smbwrapper/smbsh.in) and change the lines:

```
SMBW_LIBDIR=${SMBW_LIBDIR-@builddir@/smbwrapper}
```

To read:

```
SMBW_LIBDIR=${SMBW_LIBDIR-/usr/bin}
```

Step 4
After that, we must specify that our sbin directory for Samba binaries files will be located into /usr/sbin, and that /var directory for Samba log files will be under /var/log/samba.

- Edit the **Makefile.in** file (vi +33 Makefile.in) and change the following lines:

```
SBINDIR  =  @bindir@
```

To read:

```
SBINDIR  =  @sbindir@


VARDIR  =  @localstadir@
```

To read:

```
VARDIR  =  /var/log/samba
```

Step 5
Here we specify to use the GNU Linux version of the awk text processing utility instead of the Bell Labs research version of awk program for the "smbpasswd" file.

- Edit the **convert_smbpasswd** file (vi +10 script/convert_smbpasswd) and change the line:

  ```
  nawk 'BEGIN {FS=":"}
  ```

  To read:

  ```
  gawk 'BEGIN {FS=":"}
  ```

Step 6
Here we fix a small bug in the **configure.in** file below.

- Edit the **configure.in** file (vi +239 configure.in) and change the following line:

  ```
  # we need libcups for CUPS support...
  AC_CHECK_LIB(cups,httpConnect)
  ```

  To read:

  ```
  # we need libcups for CUPS support...
  dnl AC_CHECK_LIB(cups,httpConnect)
  ```

Step 7
Once the required modifications have been made into the related source files of Samba as explained previously, it is time configure and optimize it for our system.

- To configure and optimize Samba use the following compilation lines:

```
CFLAGS="-O3 -march=i686 -mcpu=i686 -funroll-loops -fomit-frame-pointer -
I/usr/include/openssl" \
./configure \
--prefix=/usr \
--libdir=/etc/samba \
--mandir=/usr/share/man \
--with-lockdir=/var/lock/samba \
--with-privatedir=/etc/samba \
--with-swatdir=/usr/share/swat \
--with-sslinc=/usr/include/openssl \
--with-ssl \
--with-pam \
--with-quotas
```

This tells Samba to set itself up for this particular configuration setup with:

- Include SSL support.
- Include PAM password database support.
- Include experimental disk-quota support.

Step 8
Now, we must make a list of all existing files on the system before installing the software, and one afterwards, then compare them using the **diff** utility tool of Linux to find out what files are placed where and finally install Samba on the server.

```
[root@deep source]# make all
[root@deep source]# cd
[root@deep /root]# find /* > Samba1
[root@deep /root]# cd /var/tmp/samba-2.2.0/source/
[root@deep source]# make install
[root@deep source]# install -m755 script/mksmbpasswd.sh  /usr/bin/
[root@deep source]# rm -rf /usr/share/swat/
[root@deep source]# rm -f /usr/sbin/swat
[root@deep source]# rm -f /usr/share/man/man8/swat.8
[root@deep source]# rm -rf /usr/private
[root@deep source]# mkdir -m 0755 /var/lock/samba
[root@deep source]# mkdir -m 1777 /var/spool/samba
[root@deep source]# chmod 700 /var/log/samba
[root@deep source]# strip /usr/sbin/smbd
[root@deep source]# strip /usr/sbin/nmbd
[root@deep source]# /sbin/ldconfig
[root@deep source]# cd
[root@deep /root]# find /* > Samba2
[root@deep /root]# diff Samba1 Samba2 > Samba-Installed
```

The **install** command will install the script "mksmbpasswd.sh" under /usr/bin directory. This script is needed to setup Samba users allowed to connect on our server via the "smbpasswd" file. See later in this documentation for how to setup and use Samba password.

The **rm** command will remove the /usr/share/swat directory and all the files under it, and it will also remove the swat binary program under /usr/sbin. The SWAT program is a web-based configuration utility that permits you to configure the smb.conf file of Samba via a web browser interface. Of course, in order to use the SWAT utility you will need to have a web server running, such as Apache. The SWAT utility can open a security breach on your server and for this reason I recommend that you remove and not use it.

The **mkdir -m 1777** command will create a /var/spool/samba directory on your system for all print sharing jobs you may have. Of course this directory is only necessary if you intend to use Samba print sharing over your LAN. Pay special attention to this command since it will set the "sticky" bit in /var/spool/samba so only the file's owner can delete a given file in this directory.

Step 9
Once configuration, optimization, compilation, and installation of the Samba Server software have been accomplished, we can free up some disk space by deleting the program tar archive and the related source directory since they are no longer needed.

- To delete Samba and its related source directory, use the following commands:
```
[root@deep /]# cd /var/tmp/
[root@deep tmp]# rm -rf samba-version/
[root@deep tmp]# rm -f samba-version.tgz
```

The rm command as used above will remove all the source files we have used to compile and install Samba. It will also remove the Samba compressed archive from the /var/tmp directory.

Configuring Samba

After Samba has been built and installed successfully in your system, your next step is to create, configure and customize all options and parameters in the different Samba configuration files. The different Samba configuration files to set up are:

- /etc/samba/smb.conf (The Samba Configuration File)
- /etc/samba/lmhosts (The Samba Net BIOS Mapping File)
- /etc/sysconfig/samba (The Samba System Configuration File)
- /etc/pam.d/samba (The Samba PAM Support Configuration File)
- /etc/logrotate.d/samba (The Samba Log Rotation File)
- /etc/rc.d/init.d/smb (The Samba Initialization File)

/etc/samba/smb.conf: The Samba Configuration File

The /etc/samba/smb.conf file is the main configuration file for the Samba suite and contains runtime configuration information, in which you can specify directories you want to access from Windows clients machines, IP addresses that are authorized to connect, how the File Sharing Server must run as, and so on through entries consisting of sections and parameters.

There are three special sections available with Samba. The first section called [global] contains global configuration directives common to all shares and become the defaults for sections, which do not specifically define certain items (unless they are over-ridden on a per-share basis).

The second section called [homes] allows services connecting clients to their home directory to be created on the fly by the File Sharing Server. This special section can represent any account on the machine, which isn't always desirable. For example, it can potentially create a share for root, bin, sys, and the like users. Therefore to eliminate this potential risk we must set an invalid users option in the [homes] section to protect against this.

The last section called [printers] works like the [homes] section but for printers. It allows users to connect to any printer specified in the configuration file.

A lot of options exist, and it's important to read the documentation that comes with Samba for more information on each of different settings and parameters available.

The following configuration example is a full working configuration file for Samba with encrypted password support. Also, it's important to note that I comment in this Samba configuration only parameters that relate to security and optimization, and leave all others to your own research.

In the example below, I have created just one directory called [tmp], and have allowed only class C machine IP address ranges to connect on the Samba server to this directory. Therefore don't forget to add your own directories from which you want your client machines to connect. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy your needs.

- Create the **smb.conf** file (`touch /etc/samba/smb.conf`) and add the following parameters:

```
[global]

        workgroup = OPENNA
        server string = OpenNA Samba Server
        encrypt passwords = True
        security = user
        smb passwd file = /etc/samba/smbpasswd
        log file = /var/log/samba/log.%m
        max log size = 0
        socket options = IPTOS_LOWDELAY TCP_NODELAY
        deadtime = 15
        getwd cache = Yes
        lpq cache time = 45
        domain master = Yes
        local master = Yes
        preferred master = Yes
        os level = 65
        dns proxy = Yes
        wins support = Yes
        name resolve order = wins lmhosts host bcast
        bind interfaces only = True
        interfaces = eth0 192.168.1.1/24 127.0.0.1
        hosts deny = ALL
        hosts allow = 192.168.1. 207.35.78. 127.0.0.1
        debug level = 1
        create mask = 0644
        directory mask = 0755
        oplocks = True
        level2 oplocks = True
        read raw = No
        write cache size = 262144

[homes]
        comment = Home Directories
        browseable = No
        read only = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[printers]
        comment = Remote Printers
        path = /var/spool/samba
        browseable = No
        printable = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[tmp]
        comment = Temporary File Space
        path = /tmp
        read only = No
        valid users = smbadmin
        invalid users = root bin daemon sync nobody sys tty disk mem kmem
```

This tells `smb.conf` file to set itself up for this particular configuration setup with:

```
[global]
```

```
workgroup = OPENNA
```
This parameter “workgroup” specifies the workgroup your server will appear to be in when queried by clients. It’s important to have the same workgroup name on both clients and servers machines. Therefore don’t forget to set the same workgroup name in the client part from which you want to connect to the server.

```
server string = OpenNA Samba Server
```
This parameter “server string” specifies the string that you wish to show to your users in the printer comment box in print manager, or to the IPC connection when using the "net view" command under Windows machines. It can be any string that you wish to show to your users.

```
encrypt passwords = True
```
This parameter “encrypt passwords” if set to “True” instructs Samba to use encrypted passwords instead of plain text password when negotiating with the client. Sniffer program will not be able to detect your password when it is encrypted. This option always must be set to “True” for security reasons. This is a security feature.

```
security = user
```
This parameter “security”, if set to “user”, specifies that a client must first "log-on" with a valid username and password, or the connection will be refused. This means that a valid username and password for the client must exist in your /etc/passwd file on the Linux server and in the /etc/smbpasswd file of the Samba server, or the connection from the client will fail. See “Securing Samba” in this chapter for more information about the “smbpasswd” file. This parameter is one of the most important settings in the smb.conf file. This is a security feature.

```
smb passwd file = /etc/samba/smbpasswd
```
This parameter “smb passwd file” specifies the path to the encrypted “smbpasswd” file. The “smbpasswd” file is a copy of the /etc/passwd file of the Linux system containing valid usernames and passwords of clients allowed to connect to the Samba server. The Samba software reads this file (smbpasswd) when a connection is requested.

```
log file = /var/log/samba/log.%m
```
This parameter “log file” specifies the locations and names of Samba log files. With the name extension “%m”, it allows you to have separate log files for each different user or machine that logs on your Samba server.

```
socket options = IPTOS_LOWDELAY TCP_NODELAY
```
This parameter “socket options” specifies parameters that you can include in your smb.conf configuration file to tune and improve your Samba server for optimal performance. By default we chose to tune the connection for a local network, and improve the performance of the Samba server for transferring files. This is a performance feature.

```
deadtime = 15
```
This parameter “deadtime” specifies the number of minutes to wait for client inactivity before considering that the connection is dead, and close it. A deadtime of zero (the default setting) indicates that no auto-disconnection should be performed. Using this parameter with a timeout of a few minutes is recommended for better performance of the systems. This is a performance feature.

`getwd cache = Yes`
This parameter "`getwd cache`" if set to "`Yes`" specifies to reduce the time taken for `getwd()` calls by using a caching algorithm. This is a performance feature.

`lpq cache time = 45`
This parameter "`lpq cache time`" specifies how long `lpq` info will be cached on memory to prevent the `lpq` command being called too often. A large value is recommended when your `lpq` command is very slow on the system. This is a performance feature.

`domain master = Yes`
This parameter "`domain master`" specifies to set "`nmbd`", which is the `Net BIOS` name server daemon, as a domain master browser for its given workgroup and enable WAN-wide browse list collation. This option usually must be set to "`Yes`" only on ONE `Samba` server for all OTHER `Samba` servers on the same network and workgroup.

`local master = Yes`
This parameter "`local master`" allows "`nmbd`", which is the `Net BIOS` name server daemon, to try to become a local master browser on a subnet. Like the above, usually this option must be set to "`Yes`" only on ONE `Samba` server that acts as a local master on a subnet for all the OTHER `Samba` servers on your network. Setting this parameter to "`Yes`" doesn't guaranty that Samba will become the local master browser on a subnet, it just ensure that `Samba` will participate in elections for local master browser. Use it in conjunction with parameters "`domain master`", and "`preferred master`".

`preferred master = Yes`
This parameter "`preferred master`" specifies and controls if "`nmbd`" is a preferred master browser for its workgroup. Once again, this must usually be set to "`Yes`" on ONE server for all the others on your network. Use it in conjunction with parameters "`domain master`", and "`local master`".

`os level = 65`
This parameter "`os level`" specifies by its integer value whether "`nmbd`" has a chance of becoming a local master browser for the Workgroup in the local broadcast area. The number 65 will win against any NT Server. If you have an NT Server on your network, and want to set your Linux `Samba` server to be a local master browser for the Workgroup in the local broadcast area then you must set the "`os level`" option to 65. Also, this option must be set only on ONE Linux `Samba` server, and must be disabled on all other Linux `Samba` servers you may have on your network. Use it in conjunction with parameters "`domain master`", "`local master`", and "`preferred master`".

`dns proxy = Yes`
This parameter "`dns proxy`" if set to "`Yes`" specifies that "`nmbd`" when acting as a `WINS` server and finding that a `Net BIOS` name has not been registered, should treat the `Net BIOS` name word-for-word as a `DNS` name and do a lookup with the `DNS` server for that name on behalf of the name-querying client. Configuring the `Samba` server to act as a `WINS` server is a good thing for its performance. I recommend to use your `Samba` server that runs with parameters "`domain master`", "`local master`", "`preferred master`", and "`os level`" set to "`Yes`" with this option "`dns proxy`" set to "`Yes`" too for better performance of your system.

```
wins support = Yes
```
This parameter “wins support” if set to “Yes” specifies that “nmbd” on the system will act as a WINS server. For better performance, it is recommended to set at least one Samba server in your network to be a WINS server. Note that you should NEVER set this to "Yes" on more than one machine in your network. It is a good idea to set your Samba server that runs with parameters “domain master”, “local master”, “preferred master”, and “os level” set to “Yes” to become the WINS server on the network (as we do here).

```
name resolve order = wins lmhosts host bcast
```
This parameter “name resolve order” specifies what naming services to use in order to resolve host names to IP addresses, and in what order. The parameters we chose cause the local “lmhosts” file of samba to be examined first, followed by the rest. This is a performance feature.

```
bind interfaces only = True
```
This parameter “bind interfaces only” if set to “True”, allows you to limit what interfaces on the server will serve “SMB” requests. This is a security feature. The configuration parameter “interfaces” below completes this option.

```
interfaces = eth0 192.168.1.1/24 127.0.0.1
```
This parameter “interfaces” allows you to override the default network interface list that Samba will use for browsing, name registration and other NBT traffic. By default, Samba will query the kernel for the list of all active interfaces and use any interface that it will find. With the above option, Samba will only listen on interface “eth0” on the IP addresses 192.168.1.1/24 and 127.0.0.1. This is a security feature, and completes the above configuration parameter “bind interfaces only”. Please note that if the network address 127.0.0.1 is not added to the "interfaces" parameter list then smbpasswd will fail to connect in it's default mode since we use the "bind interfaces only" parameter in conjunction with the "interfaces" parameter here. Therefore don't forget to add 127.0.0.1 to the "interfaces" parameter list above.

```
hosts deny = ALL
```
This parameter “hosts deny” specifies the list of hosts that are NOT permitted access to Samba services unless the specific services have their own lists to override this one. For simplicity, we deny access to all hosts by default, and allow specific hosts in the “hosts allow” parameter list as shown below. This is a security feature.

```
hosts allow = 192.168.1. 207.35.78. 127.0.0.1
```
This parameter “hosts allow” specifies which hosts are permitted to access a Samba service. In our example we allow by default all hosts from IP class C 192.168.1.*, 207.35.78.* and our localhost 127.0.0.1 to access the Samba server. Note that the localhost must always be set or you will receive some error messages. This is a security feature.

```
debug level = 1
```
This parameter “debug level” allows the logging level to be specified in the “smb.conf” file. If you set the debug level higher than 2 then you may suffer a large drop in performance. This is because the server flushes the log file after each operation, which can be very expensive. This is a performance feature.

```
create mask = 0644
```
This parameter “create mask” specifies and sets the necessary permissions according to the mapping from DOS modes to UNIX permissions. With this option set to 0644, all files copying or creating from a Windows system to the Unix system will have a permission of 0644 by default. This is a security feature.

directory mask = 0755
This parameter “directory mask” specifies and set the octal modes, which are used when converting DOS modes to UNIX modes when creating UNIX directories. With this option set to 0755, all directories copying or creating from a Windows system to the Unix system will have a permission of 0755 by default. This is a security feature.

oplocks = True
This parameter “oplocks”, tells smbd whether to issue oplocks (opportunistic locks) to file open requests. The oplock code can dramatically improve the speed of access to files on Samba servers and it is recommended to set this option to “True”. This is a performance feature.

level2 oplocks = True
This parameter “level2 oplocks”, if set to “True”, will increase the performance for many accesses of files that are not commonly written (such as .EXE application files). It is important for the "oplocks" (opportunistic locks) parameter to be set to "True" on this share in order for the "level2 oplocks " parameter to have any effect. This is a performance feature.

read raw = No
This parameter “read raw” controls whether or not the server will support the raw read SMB requests when transferring data to clients. Note that memory mapping is not used by the "read raw" operation. Thus, you may find memory mapping is more effective if you disable "read raw" using "read raw = No", like we do. This is a performance feature.

write cache size = 262144
This parameter “write cache size” allows Samba to improve performance on systems where the disk subsystem is a bottleneck. The value of this option is specified in bytes, and a size of 262,144 represents a 256k-cache size per file. It is to yours to set this parameter adequately related to the size of files that you hope to share with your server. If the majority of sharing files are between 512K in size, you could set the parameter to “524288”. This is a performance feature.

[tmp]

comment = Temporary File Space
This parameter “comment” allows you to specify a comment that will appear next to a share when a client does queries to the server either via the network neighborhood or via "net view" to list what shares are available.

path = /tmp
This parameter “path” specifies a directory to which the user of the service is to be given access. In our example this is the “tmp” directory of the Linux server.

read only = No
This parameter “read only” specifies if users should be allowed to only read files or not. In our example, since this is a configuration for the “tmp” directory of the Linux server, users can do more than just read files.

valid users = smbadmin
This parameter “valid users” specifies a list of users that should be allowed to login to this service. In our example only the user “smbadmin” is allowed to access the service.

```
invalid users = root bin daemon sync nobody sys tty disk mem kmem
```

This parameter "invalid users" specifies a list of users that should not be allowed to login to this service. This is really a "paranoid" check to ensure an improper setting does not breach your security. It is recommended that you include all default users that run daemons on the server. This is a security feature.

/etc/samba/lmhosts: The Samba Net BIOS Mapping File

The "lmhosts" file is the Samba Net BIOS name to IP address mapping file. It is very similar to the /etc/hosts file format, except that the hostname component must correspond to the Net BIOS naming format. The text in bold are the parts of the script initialization file that must be customized and adjusted to satisfy your needs.

- Create the **lmhosts** file (touch /etc/samba/lmhosts) and add the following lines:

```
# Sample Samba lmhosts file.
#
127.0.0.1       localhost
192.168.1.30    station1
192.168.1.31    station2
```

In our example, this file contains three IP to Net BIOS name mappings. The localhost (127.0.0.1), which is always require, the client machine called station1 (192.168.1.30) and another client machine called station2 (192.168.1.31).

/etc/sysconfig/samba: The Samba System Configuration File

The /etc/sysconfig/samba file is used to specify Samba system configuration information, such as if additional options are required to be passed to smbd and nmbd daemons at startup.

- Create the **samba** file (touch /etc/sysconfig/samba) and add the following lines:

```
# Options to smbd
SMBDOPTIONS="-D"
# Options to nmbd
NMBDOPTIONS="-D"
```

The "SMBDOPTIONS" and "NMBDOPTIONS" parameters with the "-D" options instructs samba server to operate as a daemon on the system. These values must be specified in this file since by default, the server will NOT operate as a daemon. Operating the server as a daemon is the recommended way of running Samba in your server.

/etc/pam.d/samba: The Samba PAM Support Configuration File

For better security of Samba, we will configure it to use PAM password authentication support. To do that, you must create the /etc/pam.d/samba file and add the following parameters inside it.

- Create the **samba** file (`touch /etc/pam.d/samba`) and add the following lines:

```
auth       required     /lib/security/pam_stack.so service=system-auth
auth       required     /lib/security/pam_nologin.so
account    required     /lib/security/pam_stack.so service=system-auth
account    required     /lib/security/pam_access.so
account    required     /lib/security/pam_time.so
password   required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_stack.so service=system-auth
session    required     /lib/security/pam_limits.so
session    optional     /lib/security/pam_console.so
```

/etc/logrotate.d/samba: The Samba Log Rotation File

This file allows the Samba server to automatically rotate its log files at the specified time. Here we'll configure the /etc/logrotate.d/samba file to rotate each week its log files automatically.

- Create the **samba** file (`touch /etc/logrotate.d/samba`) and add the lines:

```
/var/log/samba/log.* {
    notifempty
    missingok
    sharedscripts
    copytruncate
    postrotate
        /bin/kill -HUP `cat /var/lock/samba/*.pid 2> /dev/null` 2>
/dev/null || true
    endscript
}
```

/etc/rc.d/init.d/smb: The Samba Initialization File

The /etc/rc.d/init.d/smb script file is responsible to automatically start and stop the Samba smbd and nmbd daemons on your server. Loading the smbd and nmbd daemons, as a standalone daemon will eliminate load time and will even reduce swapping since non-library code will be shared.

Step 1
Create the **smb** script file (`touch /etc/rc.d/init.d/smb`) and add the following lines:

```
#!/bin/sh
#
# chkconfig: - 91 35
# description: Starts and stops the Samba smbd and nmbd daemons \
#              used to provide SMB network services.

# Source function library.
if [ -f /etc/init.d/functions ] ; then
  . /etc/init.d/functions
elif [ -f /etc/rc.d/init.d/functions ] ; then
  . /etc/rc.d/init.d/functions
else
  exit 0
fi

# Source networking configuration.
. /etc/sysconfig/network
```

```
if [ -f /etc/sysconfig/samba ]; then
   . /etc/sysconfig/samba
fi

# Check that networking is up.
[ ${NETWORKING} = "no" ] && exit 0

# Check that smb.conf exists.
[ -f /etc/samba/smb.conf ] || exit 0

RETVAL=0


start() {
        KIND="SMB"
        echo -n $"Starting $KIND services: "
        daemon smbd $SMBDOPTIONS
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Starting $KIND services: "
        daemon nmbd $NMBDOPTIONS
        RETVAL2=$?
        echo
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && touch /var/lock/subsys/smb || \
           RETVAL=1
        return $RETVAL
}

stop() {
        KIND="SMB"
        echo -n $"Shutting down $KIND services: "
        killproc smbd
        RETVAL=$?
        echo
        KIND="NMB"
        echo -n $"Shutting down $KIND services: "
        killproc nmbd
        RETVAL2=$?
        [ $RETVAL -eq 0 -a $RETVAL2 -eq 0 ] && rm -f /var/lock/subsys/smb
        echo ""
        return $RETVAL
}

restart() {
        stop
        start
}

reload() {
        echo -n $"Reloading smb.conf file: "
        killproc smbd -HUP
        RETVAL=$?
        echo
        return $RETVAL
}

status() {
        status smbd
        status nmbd
}

case "$1" in
```

```
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  reload)
        reload
        ;;
  status)
        status
        ;;
  condrestart)
        [ -f /var/lock/subsys/smb ] && restart || :
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart|status|condrestart}"
        exit 1
esac

exit $?
```

Step 2
Once the `smb` script file has been created, it is important to make it executable, change its default permissions, create the necessary links and start it. Making this file executable will allow the system to run it, changing its default permission is to allow only the root user to change this file for security reason, and creation of the symbolic links will let the process control initialization of Linux which is in charge of starting all the normal and authorized processes that need to run at boot time on your system to start the program automatically for you at each reboot.

- To make this script executable and to change its default permissions, use the commands:
  ```
  [root@deep /]# chmod 700 /etc/rc.d/init.d/smb
  [root@deep /]# chown 0.0 /etc/rc.d/init.d/smb
  ```

- To create the symbolic `rc.d` links for `Samba`, use the following commands:
  ```
  [root@deep /]# chkconfig --add smb
  [root@deep /]# chkconfig --level 345 smb on
  ```

- To start `Samba` daemons manually, use the following command:
  ```
  [root@deep /]# /etc/rc.d/init.d/smb start
  Starting SMB services:           [OK]
  Starting NMB services:           [OK]
  ```

NOTE: All the configuration files required for each software described in this book has been provided by us as a gzipped file, `floppy-2.0.tgz` for your convenience. This can be downloaded from this web address: ftp://ftp.openna.com/ConfigFiles-v2.0/floppy-2.0.tgz. You can unpack this to any location on your local machine, say for example `/var/tmp`, assuming you have done this your directory structure will be `/var/tmp/floppy-2.0`. Within this floppy directory each configuration file has its own directory for respective software. You can either cut and paste this directly if you are faithfully following our instructions from the beginning or manually edit these to modify to your needs. This facility is there though as a convenience but please don't forget ultimately it will be your responsibility to check, verify, etc. before you use them whether modified or as it is.

Running Samba with SSL support

This section applies only if you want to run Samba through SSL connection. Usually running Samba with SSL support is only required when you share files with the external through the Internet. For corporate network that runs Samba on an LAN for their Windows client machines, this is not useful since at this time Microsoft doesn't provide with their operating systems SSL support for File Sharing. There is from my knowledge one program named "stunnel" available from http://www.kuix.de/ssl/, which could help to solve this problem with Windows machines but I don't recommend you to use it. Unfortunately the best will be to wait and hope that Microsoft will provides SSL support with File Sharing in future upgrade of its operating systems. From now you can use this new feature of running Samba through SSL connection with operating systems like Linux with the use of its smbclient program that comes with Samba.

Below I show you how to set up the required certificate to be able to use Samba through SSL connection. The principle is exactly the same as for creating a certificate for a Web Server (refer to OpenSSL chapter if you have problem creating the certificates).

Step 1
First you have to know the **F**ully **Q**ualified **D**omain **N**ame (FQDN) of the File Sharing Server for which you want to request a certificate. When you want to access your File Sharing Server through smb.mydomain.com then the FQDN of your File Sharing Server is smb.mydomain.com.

Step 2
Second, select five large and relatively random files from your hard drive (compressed log files are a good start) and put them under your /usr/share/ssl directory. These will act as your random seed enhancers. We refer to them as random1: random2:...: random5 below.

- To select five random files and put them under /usr/share/ssl, use the commands:

```
[root@deep /]# cp /var/log/boot.log /usr/share/ssl/random1
[root@deep /]# cp /var/log/cron /usr/share/ssl/random2
[root@deep /]# cp /var/log/dmesg /usr/share/ssl/random3
[root@deep /]# cp /var/log/messages /usr/share/ssl/random4
[root@deep /]# cp /var/log/secure /usr/share/ssl/random5
```

Step 3
Third, create the RSA private key protected with a pass-phrase for your Samba File Sharing Server. The command below will generate 1024 bit RSA Private Key and stores it in the file smb.key. It will ask you for a pass-phrase: use something secure and remember it. Your certificate will be useless without the key. If you don't want to protect your key with a pass-phrase (only if you absolutely trust that server machine, and you make sure the permissions are carefully set so only you can read that key) you can leave out the -des3 option below.

- To generate the Key, use the following command:

```
[root@deep /]# cd /usr/share/ssl/
[root@deep ssl]# openssl genrsa -des3 -rand
random1:random2:random3:random4:random5 -out smb.key 1024
123600 semi-random bytes loaded
Generating RSA private key, 1024 bit long modulus
.....................+++++
.....+++++
e is 65537 (0x10001)
Enter PEM pass phrase:
Verifying password - Enter PEM pass phrase:
```

WARNING: Please backup your **smb.key** file and remember the pass-phrase you had to enter at a secure location. A good choice is to backup this information onto a diskette or other removable media.

Step 4
Finally, generate a **C**ertificate **S**igning **R**equest (CSR) with the server RSA private key. The command below will prompt you for the X.509 attributes of your certificate. Remember to give the name smb.mydomain.com when prompted for `Common Name'. Do not enter your personal name here. We are requesting a certificate for a File Sharing Server, so the Common Name has to match the FQDN of your website.

- To generate the CSR, use the following command:

```
[root@deep ssl]# openssl req -new -key smb.key -out smb.csr
Using configuration from /usr/share/ssl/openssl.cnf
Enter PEM pass phrase:
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a
DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [CA]:
State or Province Name (full name) [Quebec]:
Locality Name (eg, city) [Montreal]:
Organization Name (eg, company) [OpenNA.com]:
Organizational Unit Name (eg, section) [OpenNA.com File Sharing Server]:
Common Name (eg, YOUR name) [smb.openna.com]:
Email Address [noc@openna.com]:

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:.
An optional company name []:.
```

WARNING: Make sure you enter the FQDN (**F**ully **Q**ualified **D**omain **N**ame) of the server when OpenSSL prompts you for the "**CommonName**" (i.e. when you generate a CSR for a File Sharing Server which will be later accessed via smb.mydomain.com, enter smb.mydomain.com here).

After generation of your **C**ertificate **S**igning **R**equest (CSR), you could send this certificate to a commercial Certifying Authority (CA) like Thawte or Verisign for signing. You usually have to post the CSR into a web form, pay for the signing, await the signed Certificate and store it into a smb.crt file. The result is then a real Certificate, which can be used for Samba.

Step 5
You are not obligated to send your **C**ertificate **S**igning **R**equest (CSR) to a commercial Certifying Authority (CA) for signing. In some cases and with Samba File Sharing Server you can become your own Certifying Authority (CA) and sign your certificate by yourself. In the step below, I assume that your CA keys pair, which are required for signing certificate by yourself already exist on the server, if this is not the case, please refer to the chapter related to OpenSSL in this book for more information about how to create your CA keys pair and become your own Certifying Authority (CA).

- To sign server CSR's in order to create real SSL Certificates, use the following command:

```
[root@deep ssl]# /usr/share/ssl/misc/sign.sh smb.csr
CA signing: smb.csr -> smb.crt:
Using configuration from ca.config
Enter PEM pass phrase:
Check that the request matches the signature
Signature ok
The Subjects Distinguished Name is as follows
countryName           :PRINTABLE:'CA'
stateOrProvinceName   :PRINTABLE:'Quebec'
localityName          :PRINTABLE:'Montreal'
organizationName      :PRINTABLE:'OpenNA.com'
organizationalUnitName:PRINTABLE:'OpenNA.com File Sharing server'
commonName            :PRINTABLE:'smb.openna.com'
emailAddress          :IA5STRING:'noc@openna.com'
Certificate is to be certified until Mar 15 02:51:52 2002 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
CA verifying: smb.crt <-> CA cert
smb.crt: OK
```

This signs the CSR and results in a smb.crt file.

Step 6
Now, we must place the certificates files (smb.key and smb.crt) to the appropriate directories and change their default permission modes to be (0400/-r--------), owned by the super-user 'root' for Samba to be able to find and use them when it will start its daemon.

- To place the certificates into the appropriate directory, use the following commands:

```
[root@deep ssl]# mv smb.key private/
[root@deep ssl]# mv smb.crt certs/
[root@deep ssl]# chmod 400 private/smb.key
[root@deep ssl]# chmod 400 certs/smb.crt
[root@deep ssl]# chown 0.0 private/smb.key
[root@deep ssl]# chown 0.0 certs/smb.crt
[root@deep ssl]# rm -f smb.csr
```

First we move the smb.key file to the private directory and the smb.crt file to the certs directory. After that we change the permission mode and ownership of both certificates to be only readable and owned by the super-user 'root' for security reason. Finally we remove the smb.csr file from our system since it is no longer needed.

Step 7
To allow SSL-enabled connections with Samba, we must specify some new options into the smb.conf file. The text in bold are the parts of the lines that must be customized and adjusted to satisfy your needs.

- Edit the **smb.conf** file (vi /etc/samba/smb.conf), and add the following lines:

```
[global]

        workgroup = OPENNA
        server string = OpenNA Samba Server
        encrypt passwords = True
        security = user
        smb passwd file = /etc/samba/smbpasswd
        log file = /var/log/samba/log.%m
        max log size = 0
        socket options = IPTOS_LOWDELAY TCP_NODELAY
        deadtime = 15
        getwd cache = Yes
        lpq cache time = 45
        domain master = Yes
        local master = Yes
        preferred master = Yes
        os level = 65
        dns proxy = Yes
        wins support = Yes
        name resolve order = wins lmhosts host bcast
        bind interfaces only = True
        interfaces = eth0 192.168.1.1/24 127.0.0.1
        hosts deny = ALL
        hosts allow = 192.168.1. 207.35.78. 127.0.0.1
        debug level = 1
        create mask = 0644
        directory mask = 0755
        oplocks = True
        level2 oplocks = True
        read raw = No
        write cache size = 262144
        ssl = Yes
        ssl CA certFile = /usr/share/ssl/certs/ca.crt
        ssl server cert = /usr/share/ssl/certs/smb.crt
        ssl server key = /usr/share/ssl/private/smb.key

[homes]
        comment = Home Directories
        browseable = No
        read only = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[printers]
        comment = Remote Printers
        path = /var/spool/samba
        browseable = No
        printable = Yes
        invalid users = root bin daemon sync nobody sys tty disk mem kmem

[tmp]
        comment = Temporary File Space
        path = /tmp
        read only = No
```

```
valid users = smbadmin
invalid users = root bin daemon sync nobody sys tty disk mem kmem
```

The "ssl" variable enables the entire SSL mode on the Samba server. The second variable "ssl CA certFile" defines where to look up and find the Certification Authorities (CA). The "ssl server cert" will specify where the file containing the server's certificate is located. The "ssl server key" will specify where the file containing the server's private key is located.

NOTE: The "ssl CA certFile" variable is not needed if you don't verify client certificates. Please read your manual for more information on the subject

Step 8
The Samba SSL-enabled connections run by default on port 139 with smbd daemon. To allow external traffic through this port (139), we must add a new rule into our firewall script file for the File Sharing Server to accept external connections on the system. Please note that this is only required if you want to share your files through the Internet. For LAN this is not required at all.

- Edit the **iptables** script file (vi /etc/rc.d/init.d/iptables), and add/check the following lines to allow Samba packets with SSL support to traverse the network:

```
# Samba SSL server (139)
# ----------------

iptables -A OUTPUT -o $EXTERNAL_INTERFACE -p tcp \
         -s $IPADDR --source-port $UNPRIVPORTS \
         --destination-port 139 -j ACCEPT

iptables -A INPUT  -i $EXTERNAL_INTERFACE -p tcp ! --syn \
         --source-port 139 \
         -d $IPADDR --destination-port $UNPRIVPORTS -j ACCEPT
```

Where EXTERNAL_INTERFACE="eth0" # Internet connected interface
Where IPADDR="207.35.78.11" # Your IP address for eth0
Where UNPRIVPORTS="1024:" # Unprivileged port range

Step 9
Finally, we must restart our Samba server and firewall for the changes to take effect.

- To restart Samba use the following command:

```
[root@deep /]# /etc/rc.d/init.d/smb restart
Shutting down SMB services:       [OK]
Shutting down NMB services:       [OK]
Starting SMB services:            [OK]
Starting NMB services:            [OK]
```

- To restart you firewall use the following command:

```
[root@deep /]# /etc/rc.d/init.d/iptables restart
Shutting Firewalling: done
Starting Firewalling: done
done
```

NOTE: With SSL support acctivated into Samba, the smbd daemon of the program will ask you during startup to enter the pass phrase of the certificate, therefore don't forget it.

Step 10
Now that Samba is started, it is time to verify if everytinhg run as espected. A good way to test whether Samba is working properly is to use the smbclient program.

- On the Samba server, enter the following command, substituting the appropriate share and user for a connection:

```
[root@deep /]# smbclient //localhost/tmp -U smbadmin -I 192.168.1.1
    SSL: Certificate OK:
/C=CA/ST=Quebec/L=Montreal/O=OpenNA.com/OU=OpenNA.com File Sharing
Server/CN=smb.openna.com/Email=noc@openna.com
    SSL: Certificate OK:
/C=CA/ST=Quebec/L=Montreal/O=OpenNA.com/OU=OpenNA.com File Sharing
Server/CN=smb.openna.com/Email=noc@openna.com
    SSL: negotiated cipher: DES-CBC3-SHA
    Password:
    Domain=[OPENNA] OS=[Unix] Server=[Samba 2.2.0]
    smb: \> exit
```

If you see several debugging statements followed by a line indicating the negotiated cipher, such as: "SSL: negotiated cipher: DES-CBC3-SHA", congratulations, your Samba File Sharing Server is working with SSL support enable.

Securing Samba

This section deals especially with actions we can make to improve and tighten security under Samba. The interesting points here are that we refer to the features available within the base installed program and not to any additional software.

Create the encrypted Samba password file for your clients connections

The /etc/samba/smbpasswd file is where the Samba encrypted passwords are stored. It contains the username; Unix UID and SMB hashed passwords of the allowed users to your Samba server, as well as account flag information and the time the password was last changed.

It's important to create this password file and include all allowed users to it before your client machines try to connect to your File Sharing Server. Without this step, no one will be able to connect to your Samba server.

Step 1
To create new Samba users accounts on the system, you must first have a valid Linux account for them, therefore it is important before generating the "smbpasswd" file of Samba which will handle all Samba users allowed to connect to the system, to create in /etc/passwd file all users you want to be able to connect to your Samba server.

- Use the following command to create new users in the /etc/passwd file. This step must be done on each additional user that you allow to access the File Sharing Server.

```
[root@deep /]# useradd -s /bin/false smbadmin 2>/dev/null || :
[root@deep /]# passwd smbadmin
Changing password for user smbadmin
New UNIX password:
Retype new UNIX password:
passwd: all authentication tokens updated successfully
```

The useradd command will add the new Samba user named smbadmin to the File Sharing Server. The '-s' option specifies the name of the user's login shell, in our case we choose /bin/false and redirect it to /dev/null. Finally, the passwd command will set the password for this user 'smbadmin'.

Here it is important to make a special attention to the above command that I use to generate the Samba user account. If you remark, this user doesn't have a shell account on the system, he just have a valid username and password to log in and nothing else.

Step 2
Once we have added all Samba clients in our /etc/passwd file on the Linux server, we can now generate the "smbpasswd" file from the /etc/passwd file.

- To generate "smbpasswd" file from /etc/passwd file, use the following command:
  `[root@deep /]# cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd`

Step 3
Finally, the last step will be to create the same Samba user account in our new generated /etc/samba/smbpasswd file before we can use it.

- To create the same Samba user account, use the following command:

```
[root@deep /]# smbpasswd -a smbadmin
New SMB password:
Retype new SMB password:
Added user smbadmin.
Password changed for user smbadmin.
```

Step 4
Don't forget to change the permission of the new "smbpasswd" file to be readable and writable only by the super-user "root', and nothing for group and other (0600/-rw-------). This is a security measure.

```
[root@deep /]# chmod 600 /etc/samba/smbpasswd
[root@deep /]# testparm (this will verify the smb.conf file for possible error).
```

NOTE: See the file called "ENCRYPTION.txt" in samba/doc/texts/ for more information.

Immunize important configuration files

The immutable bit can be used to prevent accidentally deleting or overwriting a file that must be protected. It also prevents someone from creating a symbolic link to this file. Once your "smb.conf" and "lmhosts" files have been configured, it's a good idea to immunize them with a command like:

```
[root@deep /]# chattr +i /etc/samba/smb.conf
[root@deep /]# chattr +i /etc/samba/lmhosts
```

Optimizing Samba

This section deals especially with actions we can make to improve and tighten performance of the Samba server. Take a note that we refer to the features available within the base installed program.

Get some fast SCSI hard disk

Once again, one of the most important parts of optimizing Samba server as well as for the majority of all SQL database servers is the speed of your hard disk, the fastest it'll be, and the fastest your File Sharing Server will run. Considering a SCSI disk with low seek times like 4.2ms can make all the difference, much better performance can also be made with RAID technology.

Setting a "wide links" Samba parameter in configuration file

It is a big mistake to set the "wide links" Samba parameter to "No" in the Samba configuration file /etc/samba/smb.conf. This option, if set to "No", instructs Samba not to follow symbolic links outside of an area designated as being exported as a share point.

In order to determine if a link points is outside the shared area, Samba has to follow the link and then do a directory path lookup to determine where on the file system the link ended up. This ends up adding a total of six extra system calls per filename lookup, and Samba looks up filenames a lot. A test done was published that showed that setting this parameter would cause a 25- to 30-percent slowdown in Samba performance. Therefore setting this parameter to "No" can have a negative effect on your server performance due to the extra system calls that Samba will have to do in order to perform the link checks.

Tuning the buffer cache

The modification of the file system cache-tuning parameters can significantly improve Linux file-serving performance--up to a factor of two. Linux will attempt to use memory not being used for any other purpose for file system caching. A special daemon, called "bdflush", will periodically flush "dirty" buffers (buffers that contain modified file system data or metadata) to the disk.

The secret to good performance is to keep as much of the data in memory for as long as is possible. Writing to the disk is the slowest part of any file system. If you know that the file system will be heavily used, then you can tune this process for Linux Samba.

As with many kernel tunable options, this modification can be done on the fly by writing to special files in the /proc file system. The trick is you have to tell Linux you want it to do that. You do so by executing the following command.

The default setup for the "bdflush" parameters under Red Hat Linux is:
"30 64 64 256 500 3000 60 0 0"

Step 1
To change the values of **bdflush**, type the following command on your terminal:

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following line:

```
# Improve file system performance for Samba
vm.bdflush = 80 500 64 64 15 6000 6000 0 0
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all network devices manually on your system, use the following command:

```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

The above modifications in the /proc file system tells "bdflush" not to worry about writing out dirty blocks to the disk until the file system buffer cache is 80 percent full (80). The other values tune such things as the number of buffers to write out in one disk operation (500), how long to allow dirty buffers to age in the kernel (60*HZ), etc.

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:

```
[root@deep /]# sysctl -w vm.bdflush="80 500 64 64 15 6000 6000 0 0"
```

Tuning the buffermem

Another helpful tuning hint is to tell Linux the following: Use a minimum of 60 percent of memory for the buffer cache; only prune when the percentage of memory used for the buffer cache gets over 10 percent (this parameter is now unused); and allow the buffer cache to grow to 60 percent of all memory (this parameter is also unused now).

The default setup for the buffermem parameters under Red Hat Linux is:
"2 10 60"

Step 1
To change the values of **buffermem**, type the following command on your terminal:

- Edit the **sysctl.conf** file (vi /etc/sysctl.conf) and add the following line:

```
# Improve virtual memory performance for samba
vm.buffermem = 60 10 60
```

Step 2
You must restart your network for the change to take effect. The command to restart the network is the following:

- To restart all networks devices manually on your system, use the following command:
```
[root@deep /]# /etc/rc.d/init.d/network restart
Setting network parameters            [OK]
Bringing up interface lo              [OK]
Bringing up interface eth0            [OK]
Bringing up interface eth1            [OK]
```

Recall that the last two parameters (**10** and **60**) are unused by the system so we don't need to change the default ones.

NOTE: There is another way to update the entry without restarting the network by using the following command into your terminal screen:
```
[root@deep /]# sysctl -w vm.buffermem="60 10 60"
```

Further documentation
For more details about Samba program, there are several manual pages you can read:

```
$ man Samba (7)              - A Windows SMB/CIFS fileserver for UNIX
$ man smb.conf (5)           - The configuration file for the Samba suite
$ man smbclient (1)          - An ftp-like client to access SMB/CIFS resources on servers
$ man smbd (8)               - Server to provide SMB/CIFS services to clients
$ man smbmnt (8)             - Mount smb file system
$ man smbmount (8)           - Mount smb file system
$ man smbpasswd (5)          - The Samba encrypted password file
$ man smbpasswd (8)          - Change a users SMB password
$ man smbrun (1)             - Interface program between smbd and external programs
$ man smbsh (1)              - Allows access to Windows NT filesystem using UNIX commands
$ man smbstatus (1)          - Report on current Samba connections
$ man smbtar (1)             - Shell script for backing up SMB shares directly to UNIX tape drives
$ man smbumount (8)          - Umount for normal users
$ man testparm (1)           - Check an smb.conf configuration file for internal correctness
$ man testprns (1)           - Check printer name for validity with smbd
```

Samba Administrative Tools
The commands listed below are some that we use often, but many more exist. Check the manual pages and documentation of Samba for more information.

smbstatus
The smbstatus utility is a very simple program to list the current Samba connections.

- To report current Samba connections, use the following command:
```
[root@deep /]# smbstatus

Samba version 2.2.0
Service      uid      gid      pid    machine
----------------------------------------------
IPC$         smbadmin smbadmin 2688   station1 (192.168.1.30) Wed Mar 14
16:44:49 2001
No locked files
```

Samba Users Tools

The commands listed below are some that we use often, but many more exist. Check the manual pages and documentation that comes with Samba for more information.

smbclient

The smbclient program utility for Samba works much like the interface of the FTP program. This small program allow you to get files from the server to the local machine, put files from the local machine to the server, retrieve directory information from the server, and so on.

- To connect to a Windows machine with smbclient utility, use the following command:

```
[root@deep /]# smbclient //station1/Tmp -U smbadmin –I 192.168.1.1
Password:
Domain=[OPENNA] OS=[Windows NT 5.0] Server=[NT LAN Manager 5.0]
smb: \> ls
  .                   D        0   Tue Mar 14 15:31:50 2001
  ..                  D        0   Tue Mar 14 15:31:50 2001
  PostgreSQL          D        0   Tue Mar 14 15:32:22 2001
  Squid               D        0   Tue Mar 14 15:32:28 2001
  Imap                D        0   Tue Mar 14 15:32:38 2001
  E_comm              D        0   Tue Mar 14 15:32:42 2001
  StackGuard.pdf      A    61440   Tue Dec 21 20:41:34 2001

                65510 blocks of size 32768. 5295 blocks available
smb: \>exit
```

Where "//station1" is the name of the server you want to connect to. "/Tmp" is the directory on this server you want to connect to, and "smbadmin" is your username on this machine. The "-I" option indicates to use the specified network interface for the connection.

List of installed Samba files on your system

> /etc/rc.d/init.d/smb
> /etc/samba/smb.conf
> /etc/samba/lmhosts
> /etc/pam.d/samba
> /etc/logrotate.d/samba
> /etc/sysconfig/samba
> /etc/samba/codepages
> /etc/samba/codepages/codepage.437
> /etc/samba/codepages/unicode_map.437
> /etc/samba/codepages/codepage.737
> /etc/samba/codepages/unicode_map.737
> /etc/samba/codepages/codepage.775
> /etc/samba/codepages/codepage.850
> /etc/samba/codepages/unicode_map.850
> /etc/samba/codepages/codepage.852
> /etc/samba/codepages/unicode_map.852
> /etc/samba/codepages/codepage.861
> /etc/samba/codepages/unicode_map.861
> /etc/samba/codepages/codepage.932
> /etc/samba/codepages/unicode_map.932
> /etc/samba/codepages/codepage.866
> /etc/samba/codepages/unicode_map.866
> /etc/samba/codepages/codepage.949
> /etc/samba/codepages/unicode_map.949
> /etc/samba/codepages/codepage.950
> /etc/samba/codepages/unicode_map.950
> /etc/samba/codepages/codepage.936
> /etc/samba/codepages/unicode_map.936
> /usr/bin/smbpasswd
> /usr/bin/make_smbcodepage
> /usr/bin/make_unicodemap
> /usr/bin/rpcclient
> /usr/bin/nmblookup
> /usr/bin/make_printerdef
> /usr/bin/smbtar
> /usr/bin/addtosmbpass
> /usr/bin/convert_smbpasswd
> /usr/bin/mksmbpasswd.sh
> /usr/sbin/smbd
> /usr/sbin/nmbd
> /usr/share/man/man1/findsmb.1
> /usr/share/man/man1/make_smbcodepage.1
> /usr/share/man/man1/make_unicodemap.1
> /usr/share/man/man1/nmblookup.1
> /usr/share/man/man1/smbclient.1
> /usr/share/man/man1/smbcontrol.1
> /usr/share/man/man1/smbrun.1
> /usr/share/man/man1/smbsh.1
> /usr/share/man/man1/smbstatus.1
> /usr/share/man/man1/smbtar.1
> /usr/share/man/man1/testparm.1
> /usr/share/man/man1/testprns.1
> /usr/share/man/man1/wbinfo.1
> /usr/share/man/man5/lmhosts.5
> /usr/share/man/man5/smb.conf.5
> /usr/share/man/man5/smbpasswd.5

> /etc/samba/codepages/codepage.1251
> /etc/samba/codepages/unicode_map.ISO8859-1
> /etc/samba/codepages/unicode_map.ISO8859-2
> /etc/samba/codepages/unicode_map.ISO8859-5
> /etc/samba/codepages/unicode_map.ISO8859-7
> /etc/samba/codepages/codepage.857
> /etc/samba/codepages/unicode_map.857
> /etc/samba/codepages/unicode_map.ISO8859-9
> /usr/bin/smbclient
> /usr/bin/smbspool
> /usr/bin/testparm
> /usr/bin/testprns
> /usr/bin/smbstatus
> /usr/bin/smbcontrol
> /usr/share/man/man7/samba.7
> /usr/share/man/man8/nmbd.8
> /usr/share/man/man8/rpcclient.8
> /usr/share/man/man8/smbd.8
> /usr/share/man/man8/smbmnt.8
> /usr/share/man/man8/smbmount.8
> /usr/share/man/man8/smbpasswd.8
> /usr/share/man/man8/smbspool.8
> /usr/share/man/man8/smbumount.8
> /usr/share/man/man8/winbindd.8
> /var/log/samba
> /var/lock/samba
> /var/spool/samba

Part XIII Backup Related Reference

In this Part

Backup - `Tar` & `Dump`

Any serious networking topology required a backup policies and procedures. This is absolutely needed and you cannot pass through it if you want to protect valuable information and data for possible lost and errors. Now that everything is running smoothly and as you expect them to be in your secure servers, it is time to think a little bit about a procedure to ensure that your hard works to protect and secure your systems are not for nothing.

31 Backup - Tar & Dump

In this Chapter

Recommended RPM packages to be installed for a Backup Server

A minimal configuration provides the basic set of packages required by the Linux operating system. Minimal configuration is a perfect starting point for building secure operating system. Below is the list of all recommended RPM packages required to run properly your Linux server as a Backup server running on Amanda software.

This configuration assumes that your kernel is a monolithic kernel. Also I suppose that you will install Amanda by RPM package. Therefore, amanda, amanda-server, and amanda-client RPM packages are already included in the list below as you can see. All security tools are not installed, it is yours to install them as your need by RPM packages too since compilers packages are not installed and included in the list. Amanda is not presently discussed in this book, but you can install it from your CD-ROM vendor and run it as it comes.

amanda	devfsd	info	openssh	slocate
amanda-server	diffutils	initscripts	openssh-server	sysklogd
amanda-client	**dump**	iptables	openssl	syslinux
basesystem	e2fsprogs	kernel	pam	SysVinit
bash	ed	less	passwd	**tar**
bdflush	file	libstdc++	popt	termcap
bind	filesystem	libtermcap	procps	textutils
bzip2	fileutils	lilo	psmisc	tmpwatch
chkconfig	findutils	logrotate	pwdb	utempter
console-tools	gawk	losetup	qmail	util-linux
cpio	gdbm	MAKEDEV	readline	vim-common
cracklib	gettext	man	rootfiles	vim-minimal
cracklib-dicts	glib	mingetty	rpm	vixie-cron
crontabs	glibc	mktemp	sed	words
db1	glibc-common	mount	setup	which
db2	grep	ncurses	sh-utils	zlib
db3	groff	net-tools	shadow-utils	
dev	gzip	newt	slang	

Tested and fully functional on OpenNA.com.

Linux Tar & Dump

Abstract
A secure and reliable server is closely related to performing regular backups. Failures will probably occur sometimes. They may be caused by attacks, hardware failure, human error, power outages, etc. The safest method of doing backups is to record them in a location separate from your Linux system like over a network, from tape, removable drive, writable CD-ROM, etc.

Many methods of performing backups with Linux exist, such as "dump", "tar", "cpio", as well as "dd" commands that are each available by default on your Linux system. Also available are text-based utilities program, such as "Amanda", which is designed to add a friendlier user interface to the backup and restore procedures. Finally, commercial backup utilities are also available, such as "BRU".

The procedures for performing a backup and restore will differ depending on your choice of a backup solution. For this reason we will discuss methods for performing backups with the traditional UNIX tools: "tar", and "dump" which is a command-line backup tool.

What to backup
The idea of making a backup is to back up as much as possible on your system, but some exceptions exist as shown below. It is not logical to include these in your backup since you will lose time and space in your media for nothing.

The major exceptions to not include in your backup are:

- ✓ The /proc file system: since it only contains data that the kernel generates automatically, it is never a good idea to back it up.
- ✓ The /mnt file system, because it is where you mount your removable media like CD-ROM, floppy disk and other.
- ✓ The backup directory or media where you have placed your backup files, such as a tape, CD-ROM, NFS mounted file system, remote/local directory or other kind of media.
- ✓ Software that can be easily reinstalled, though they may have configuration files that are important to back up, lest you do all the work to configure them all over again. I will recommend putting them (the configuration files for software) on the floppy disk.

The tar backup program
The tar backup program is an archiving program designed to store and extract files from an archive file known as a tarfile. A tarfile may be made on a tape drive; however, it is also common to write a tarfile to a normal file.

A simple backup scheme
When you decide to make a backup of files on your system you must choose a backup scheme before the beginning of your backup procedure. A lot of strategic backup schemes exist, and depend on the backup policies you want to use. In the following, I will show you one backup scheme that you may use which takes advantage of the tar program's possibilities. This scheme is to first back up everything once, then back up everything that has been modified since the previous backup. The first backup is called a full backup; the subsequent ones are incremental backups.

Making backups with `tar`

With six tapes you can make backups every day; the procedure is to use tape 1 for the first full backup (Friday 1), and tapes 2 to 5 for the incremental backups (Monday through Thursday). Then, you make a new full backup on tape 6 (second Friday), and start doing incremental ones with tapes 2 to 5 again. It's important to keep tape 1 at its state until you've got a new full backup with tape 6. In the following example below, we assume that we write the backup to a `SCSI` tape drive named "`/dev/st0`", and we backup the home directory "`/home`" of our system.

First of all, we move to the file system "**/**" partition. When creating an archive file, `tar` will strip leading "`/`" (slash) characters from file path names. This means that restored files may not end up in the same locations they were backed up from. Therefore, to solve the problem, the solution is to change to the "**/**" root directory before making all backups and restorations.

- To move to the "`/`" root directory, use the command:
  ```
  [root@deep]# cd /
  ```

It is important to always start with a full backup (say, on a Friday), for example:

- Friday 1, (use tape 1 for the first full backup).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Monday, (use tapes 2 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Tuesday, (use tapes 3 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Wednesday, (use tapes 4 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Thursday, (use tapes 5 for the incremental backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpNf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Friday 2, (use tape 6 for the new full backups).
  ```
  [root@deep /]# cd /
  [root@deep /]# tar cpf /dev/st0 --label="full-backup created on \
  `date '+%d-%B-%Y'`." --directory / home
  ```

- Now, start doing incremental ones with tapes 2 to 5 again and so on.

The “c” option specifies that an archive file is beginning to be created.
The “p” option preserves permissions; file protection information will be “remembered”.
The “N” option does an incremental backup and only stores files newer than DATE.
The “f” option states that the very next argument will be the name of the archive file or device being written.

Notice how a filename, which contains the current date, is derived, simply by enclosing the “date” command between two back-quote characters. A common naming convention is to add a “tar” suffix for non-compressed archives, and a “tar.gz” suffix for compressed ones. Since we aren't able to specify a filename for the backup set, the "--label" option can be used to write some information about the backup set into the archive file itself. Finally, only the files contained in the /home are written to the tape.

Because the tape drive is a character device, it is not possible to specify an actual file name. Therefore the file name used as an argument to tar is simply the name of the device /dev/st0, the first tape device. The /dev/st0 device does not rewind after the backup set is written; Therefore, it is possible to write multiple sets on one tape. You may also refer to the device as /dev/st0, in which case the tape is automatically rewound after the backup set is written. When working with tapes you can use the following commands to rewind and eject your tape:

```
[root@deep /]# mt -f /dev/st0 rewind
[root@deep /]# mt -f /dev/st0 offline
```

WARNING: To reduce the space needed on a tar archive, the backups can be compressed with the “z” option of tar program. Unfortunately, using this option to compress backups can cause trouble. Due to the nature of how compression works, if a single bit in the compressed backup is wrong, all the rest of the compressed data will be lost. It’s recommended to NOT using compression (the “z” option) to make backups with the tar command.

- If your backup doesn't fit on one tape, you’ll have to use the --multi-volume (-M) option:
```
[root@deep /]# cd /
[root@deep /]# tar cMpf /dev/st0 /home
Prepare volume #2 for /dev/st0 and hit return:
```

- After you have made a backup, you should check that it is OK, using the --compare (-d) option as shown below:
```
[root@deep /]# cd /
[root@deep /]# tar dvf /dev/st0
```

- To perform a backup of your entire system, use the following command:
```
[root@deep /]# cd /
[root@deep /]# tar cpf /archive/full-backup-`date '+%d-%B-%Y'`.tar \
 --directory / --exclude=proc --exclude=mnt --exclude=archive \
 --exclude=cache --exclude=*/lost+found .
```

The ”--directory” option informs tar to first switch to the following directory path (the “/” directory in this example) prior to starting the backup. The “--exclude” options informs tar not to bother backing up the specified directories or files. Finally, the “.” character at the end of the command tells tar that it should back up everything in the current directory.

WARNING: When backing up your file systems, do not include the `/proc` pseudo-file-system! The files in `/proc` are not actually files but are simply file-like links which describe and point to kernel data structures. Also, do not include the `/mnt`, `/archive`, and all `lost+found` directories.

Automating tasks of backups made with `tar`

It is always interesting to automate the tasks of a backup. Automation offers enormous opportunities for using your Linux server to achieve the goals you set. The following example below is our backup script, named "`backup.cron`".

This script is designed to run on any computer by changing only the four variables: `COMPUTER`, `DIRECTORIES`, `BACKUPDIR`, and `TIMEDIR`. We suggest that you set this script up and run it at the beginning of the month for the first time, and then run it for a month before making major changes. In our example below we do the backup to a directory on the local server (`BACKUPDIR`), but you could modify this script to do it to a tape on the local server or via an `NFS` mounted file system.

Step 1
Create the backup script **`backup.cron`** file (`touch /etc/cron.daily/backup.cron`) and add the following lines to this backup file:

```
#!/bin/sh
# full and incremental backup script
# created 07 February 2000
# Based on a script by Daniel O'Callaghan <danny@freebsd.org>
# and modified by Gerhard Mourani <gmourani@openna.com>

#Change the 5 variables below to fit your computer/backup

COMPUTER=deep                   # Name of this computer
DIRECTORIES="/home"             # Directoris to backup
BACKUPDIR=/backups              # Where to store the backups
TIMEDIR=/backups/last-full      # Where to store time of full backup
TAR=/bin/tar                    # Name and location of tar

#You should not have to change anything below here

PATH=/usr/local/bin:/usr/bin:/bin
DOW=`date +%a`                  # Day of the week e.g. Mon
DOM=`date +%d`                  # Date of the Month e.g. 27
DM=`date +%d%b`                 # Date and Month e.g. 27 Sep

# On the 1st of the month a permanet full backup is made
# Every Sunday a full backup is made - overwriting last Sundays backup
# The rest of the time an incremental backup is made. Each incremental
# backup overwrites last weeks incremental backup of the same name.
#
# if NEWER = "", then tar backs up all files in the directories
# otherwise it backs up files newer than the NEWER date. NEWER
# gets it date from the file written every Sunday.


# Monthly full backup
if [ $DOM = "01" ]; then
        NEWER=""
        $TAR $NEWER -cf $BACKUPDIR/$COMPUTER-$DM.tar $DIRECTORIES
fi
```

```
# Weekly full backup
if [ $DOW = "Sun" ]; then
        NEWER=""
        NOW=`date +%d-%b`

        # Update full backup date
        echo $NOW > $TIMEDIR/$COMPUTER-full-date
        $TAR $NEWER -cf $BACKUPDIR/$COMPUTER-$DOW.tar $DIRECTORIES

# Make incremental backup - overwrite last weeks
else

        # Get date of last full backup
        NEWER="--newer `cat $TIMEDIR/$COMPUTER-full-date`"
        $TAR $NEWER -cf $BACKUPDIR/$COMPUTER-$DOW.tar $DIRECTORIES
fi
```

Here is an abbreviated look of the backup directory after one week:

```
[root@deep /]# ls -l /backups/
total 22217
-rw-r--r--  1 root     root         10731288    Feb  7 11:24 deep-01Feb.tar
-rw-r--r--  1 root     root             6879    Feb  7 11:24 deep-Fri.tar
-rw-r--r--  1 root     root             2831    Feb  7 11:24 deep-Mon.tar
-rw-r--r--  1 root     root             7924    Feb  7 11:25 deep-Sat.tar
-rw-r--r--  1 root     root         11923013    Feb  7 11:24 deep-Sun.tar
-rw-r--r--  1 root     root             5643    Feb  7 11:25 deep-Thu.tar
-rw-r--r--  1 root     root             3152    Feb  7 11:25 deep-Tue.tar
-rw-r--r--  1 root     root             4567    Feb  7 11:25 deep-Wed.tar
drwxr-xr-x  2 root     root             1024    Feb  7 11:20 last-full
```

WARNING: The directory where to store the backups (`BACKUPDIR`), and the directory where to store time of full backup (`TIMEDIR`) must exist or be created before the use of the backup-script, or you will receive an error message.

Also I recommend you to set the permission mode of these directories to be (`0700/-rwx------`) owned by the user making the backup. It is important that normal user cannot access in our example the `/backups` directory.

Step 2
If you are not running this backup script from the beginning of the month (01-month-year), the incremental backups will need the time of the Sunday backup to be able to work properly. If you start in the middle of the week, you will need to create the time file in the `TIMEDIR`.

- To create the time file in the `TIMEDIR` directory, use the following command:
  `[root@deep /]# date +%d%b > /backups/last-full/myserver-full-date`

Where <`/backups/last-full`> is our variable `TIMEDIR` where we want to store the time of the full backup, and <`myserver-full-date`> is the name of our server (e.g., `deep`), and our time file consists of a single line with the present date (i.e. `15-Feb`).

Step 3
Make this script executable and change its default permissions to be writable only by the super-user "root" (0700/-rwx------).

```
[root@deep /]# chmod 700 /etc/cron.daily/backup.cron
```

NOTE: Because this script is in the /etc/cron.daily directory, it will be automatically run as a cron job at one o'clock in the morning every day.

Restoring files with tar

More important than performing regular backups is having them available when we need to recover important files! In this section, we will discuss methods for restoring files, which have been backed up with "tar" command.

The following command will restore all files from the "full-backup-Day-Month-Year.tar" archive, which is an example backup of our /home directory created from the example tar commands shown above.

- To restore a full backup of the /home directory, use the following commands:

```
[root@deep /]# cd /
[root@deep /]# tar xpf /dev/st0/full-backup-Day-Month-Year.tar
```

The above command extracts all files contained in the compressed archive, preserving original file ownership and permissions.

The "x" option stands for extract.
The "p" option preserves permissions; file protection information will be "remembered".
The "f" option states that the very next argument will be the name of the archive file or device.

If you do not need to restore all the files contained in the archive, you can specify one or more files that you wish to restore:

- To specify one or more files that you wish to restore, use the following commands:

```
[root@deep]# cd /
[root@deep]# tar xpf /dev/st0/full-backup-Day-Month-Year.tar \
 home/wahib/Personal/Contents.doc home/quota.user
```

The above command restores the /home/wahib/Personal/Contents.doc and /home/quota.user files from the archive.

- If you just want to see what files are in the backup volume, Use the --list (-t) option:

```
[root@deep /]# tar tf /dev/st0
```

WARNING: If you have files on your system set with the immutable bit, using the "chattr" command, these files will not be remembered with the immutable bit from your restored backup. You must reset it immutable with the command "chattr +i" after the backup is completed.

Testing the ability to recover from backups

For many system administrators, recovering a file from a backup is an uncommon activity. This step assures that if you need to recover a file, the tools and processes will work. Performing this test periodically will help you to discover problems with the backup procedures so you can correct them before losing data. Some backup restoration software does not accurately recover the correct file protection and file ownership controls. Check the attributes of restored files to ensure they are being set correctly. Periodically test to ensure that you can perform a full system recovery from your backups.

Further documentation

For more details, there is one manual page that you can read:

`tar (1)` - The GNU version of the tar archiving utility

The `dump` backup program

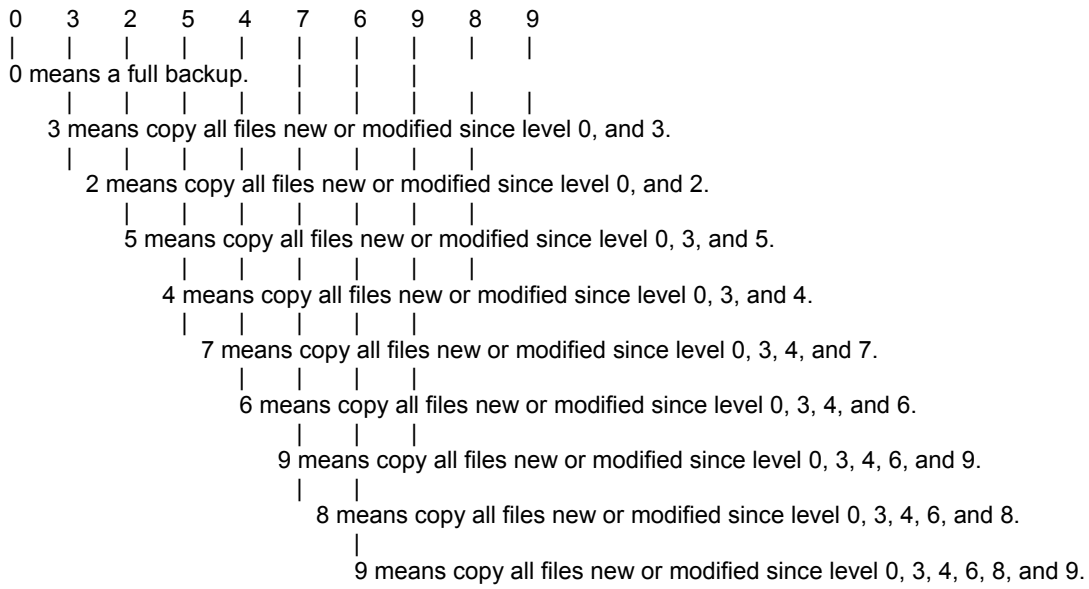
`Dump` is completely different from `tar`; it is a program for backing up and restoring file system. It backups up the entire file system - not the files. `Dump` does not care what file system is on the hard drive, or even if there are files in the file system. It examines files on an `ext2` file system, determines which ones need to be backed up, and copies those files to a specified disk, tape, file or other storage medium. It dumps one file system at a time quickly and efficiently.

Unfortunately, it does not do individual directories, and so it eats up a great deal more storage space than `tar`. It is also written specifically for backups. The `restore` command performs the inverse function of `dump`; It can restore a full backup of a file system. Subsequent incremental backups can then be layered on top of the full backup. Single files and directory sub trees may also be restored from full or partial backups. You can use `dump` if you need a procedure for both backing up file systems and restoring file systems after backups.

The `Dump` levels

`Dump` has several levels of backup procedures. The levels range from 0 to 9, where level number 0 means a full backup and guarantees the entire file system is copied. A level number above 0, incremental backup, tells `dump` to copy all files new or modified since the last `dump` of the same or lower level. To be more precise, at each incremental backup level you back up everything that has changed since the previous backup at the same or a previous level.

What are the advantages and the reasons to create and use several levels to make a backup? I try to explain it with the following schemas:



The advantages and reasons for doing this are that with multiple levels, the backup history can be extended more cheaply. A longer backup history is useful, since deleted or corrupted files are often not noticed for a long time. Even a version of a file that is not very up to date is better than no file at all. Also, backup levels are used to keep both the backup and restore times to a minimum (low).

The `dump` manual page suggests a good scheme to take the full advantage of backup levels: 3, 2, 5, 4, 7, 6, 9, 8, 9, etc as described by the table below. The most you have to backup is two day's worth of work. The number of tapes for a restore depends on how long you keep between full backups.

Tape	Level	Backup (days)	Restore tapes
1	0	n/a	1
2	3	1	1, 2
3	2	2	1, 3
4	5	1	1, 2, 4
5	4	2	1, 2, 5
6	7	1	1, 2, 5, 6
7	6	2	1, 2, 5, 7
8	9	1	1, 2, 5, 7, 8
9	8	2	1, 2, 5, 7, 9
10	9	1	1, 2, 5, 7, 9, 10

Making backups with dump

It's interesting to use the dump backup program if you want to take advantage of its several levels of backup procedures. Below, I show you a procedure to have a longer backup history, and to keep both the backup and restore times to a minimum.

In the following example, we assume that we write the backup to a tape drive named "/dev/st0" and we backup the /home directory of our system.

It is important to always start with a level 0 backup, for example:

- Friday 1, (use tape 1 for the first full backup).

```
[root@deep /]# dump -0u -f /dev/st0 /home
  DUMP: Date of this level 0 dump: Fri Mar 16 21:25:12 2001
  DUMP: Date of last level 0 dump: the epoch
  DUMP: Dumping /dev/sda6 (/home) to /dev/st0
  DUMP: mapping (Pass I) [regular files]
  DUMP: mapping (Pass II) [directories]
  DUMP: estimated 18582 tape blocks on 0.48 tape(s).
  DUMP: Volume 1 started at: Fri Mar 16 21:25:12 2001
  DUMP: dumping (Pass III) [directories]
  DUMP: dumping (Pass IV) [regular files]
  DUMP: DUMP: 18580 tape blocks on 1 volumes(s)
  DUMP: finished in 4 seconds, throughput 4645 KBytes/sec
  DUMP: Volume 1 completed at: Fri Mar 16 21:26:12 2001
  DUMP: Volume 1 took 0:00:04
  DUMP: Volume 1 transfer rate: 4645 KB/s
  DUMP: level 0 dump on Fri Fri Mar 16 21:25:12 2001
  DUMP: DUMP: Date of this level 0 dump: Fri Mar 16 21:25:12 2001
  DUMP: DUMP: Date this dump completed:  Fri Mar 16 21:25:18 2001
  DUMP: DUMP: Average transfer rate: 4645 KB/s
  DUMP: Closing /dev/st0
  DUMP: DUMP IS DONE
```

- Monday, (use tapes 2 for the incremental backups).

```
[root@deep /]# dump -3u -f /dev/st0 /home
```

- Tuesday, (use tapes 3 for the incremental backups).

```
[root@deep /]# dump -2u -f /dev/st0 /home
```

- Wednesday, (use tapes 4 for the incremental backups).

```
[root@deep /]# dump -5u -f /dev/st0 /home
```

- Thursday, (use tapes 5 for the incremental backups).

```
[root@deep /]# dump -4u -f /dev/st0 /home
```

- Friday 2, (use tape 6 for the incremental backups).

```
[root@deep /]# dump -7u -f /dev/st0 /home
```

- Monday, (use tapes 2 for the incremental backups).

```
[root@deep /]# dump -3u -f /dev/st0 /home
```

- Tuesday, (use tapes 3 for the incremental backups).

```
[root@deep /]# dump -2u -f /dev/st0 /home
```

- Wednesday, (use tapes 4 for the incremental backups).

```
[root@deep /]# dump -5u -f /dev/st0 /home
```

- Thursday, (use tapes 5 for the incremental backups).
  `[root@deep /]# dump -4u -f /dev/st0 /home`

- Friday 3, (use tape 7 for the incremental backups).
  `[root@deep /]# dump -6u -f /dev/st0 /home`

- Monday, (use tapes 2 for the incremental backups).
  `[root@deep /]# dump -3u -f /dev/st0 /home`

- Tuesday, (use tapes 3 for the incremental backups).
  `[root@deep /]# dump -2u -f /dev/st0 /home`

- Wednesday, (use tapes 4 for the incremental backups).
  `[root@deep /]# dump -5u -f /dev/st0 /home`

- Thursday, (use tapes 5 for the incremental backups).
  `[root@deep /]# dump -4u -f /dev/st0 /home`

- Friday 4, (use tape 8 for the incremental backups only if there have 5 Fridays in one month).
  `[root@deep /]# dump -9u -f /dev/st0 /home`

- Monday, (use tapes 2 for the incremental backups only if there have 5 Fridays in one month).
  `[root@deep /]# dump -3u -f /dev/st0 /home`

- Tuesday, (use tapes 3 for the incremental backups only if there have 5 Fridays in one month).
  `[root@deep /]# dump -2u -f /dev/st0 /home`

- Wednesday, (use tapes 4 for the incremental backups only if there have 5 Fridays in one month).
  `[root@deep /]# dump -5u -f /dev/st0 /home`

- Thursday, (use tapes 5 for the incremental backups only if there have 5 Fridays in one month).
  `[root@deep /]# dump -4u -f /dev/st0 /home`

- Month, (use another tape for a new full backup when the month change).
  `[root@deep /]# dump -0u -f /dev/st0 /home`

Where "`-0` to `-9`" is the backup level option you want to use, the "`u`" option means to update the file `/etc/dumpdates` after a successful `dump`, the "`-f`" option to write the backup to file; the file may be a special device file like `/dev/st0` (a tape drive), `/dev/rsd1c` (a disk drive), an ordinary file, or "`-`" (the standard output). Finally, you must specify what you want to backup. In our example, it is the `/home` directory.

You can see that we use the same tapes 2 to 5 for daily backups (Monday to Thursday = **4** tapes), tapes 6, 7, and 8 for weekly backups (other Fridays, 6 + 7 + 8 = **3** tapes; note that there can be five Fridays in one month) and tapes 1 and any subsequent new one for monthly backups (first Friday each month, **1** + any subsequent "11 months" = 12 tapes). In conclusion, if we use 8 tapes (**4** + **3** + **1** = **8**), we can have a full backup for one month and repeat the procedure with the 8 tapes to get our subsequent 11 months to come for a total of 1-year individual full backups.

The full backup should be done at set intervals, say once a month, and on a set of fresh tapes that are saved forever. With this kind of procedure, you will have 12 tapes for 12 months that handle histories and changes of your system for one year. Afterwards, you can copy the 12 tape backups onto a different computer designated to keep all yearly backups for a long time and be able to reuse them (12 tapes) to repeat the procedure for a new year. Thank you Gerhard!

Restoring files with dump

The restore command of the program performs the inverse function of dump(8). It restores files or file systems from backups made with dump. A full backup of a file system may be restored, and subsequent incremental backups layered on top of it. Single files and directory sub-trees may be restored from full, or partial, backups. You have a number of possible commands and options to restore backed up data with the dump program. Below, we show you a procedure that uses the full potential of the restore program with the most options possible. It is also done in interactive mode.

In an interactive restoration of files from a dump, the restore program provides a shell like interface that allows the user to move around the directory tree selecting files to be extracted, after reading in the directory information from the dump. The following is what we will see if we try to restore our /home directory:

First of all, we must move to the partition file system where we want to restore our backup. This is required, since the interactive mode of the restore program will restore our backups from the current partition file system where we have executed the restore command.

- To move to the partition file system we want to restore (the /home directory in our case), use the following command:
  ```
  [root@deep /]# cd /home
  ```

- To restore files from a dump in interactive mode, use the following command:
  ```
  [root@deep /home]# restore -i -f /dev/st0
  restore >
  ```

A prompt will appear in your terminal, to list the current, or specified, directory. Use the "**ls**" command as shown below:

```
restore > ls
.:
admin/       lost+found/  named/       quota.group  quota.user   wahib/

restore >
```

To change the current working directory to the specified one, use the "**cd**" commands (in our example, we change to *wahib* directory) as shown below:

```
restore > cd wahib
restore > ls
./wahib:
.Xdefaults     .bash_logout   .bashrc
.bash_history  .bash_profile  Personal/

restore >
```

To add the current directory or file to the list of files to be extracted, use the "**add**" command (If a directory is specified, then it and all its descendents are added to the extraction list) as shown below:

```
restore > add Personal/
restore >
```

Files that are on the extraction list are marked with a "*****" when they are listed by the "**ls**" command:

```
restore > ls
./wahib:
 .Xdefaults      .bash_logout    .bashrc
 .bash_history   .bash_profile  *Personal/
```

To delete the current directory or specified argument from the list of files to be extracted, use the "**delete**" command (If a directory is specified, then it and all its descendents are deleted from the extraction list) as shown below:

```
restore > cd Personal/
restore > ls
./wahib/Personal:
*Ad?le_Nakad.doc        *Overview.doc
*BIMCOR/                *Resume/
*My Webs/               *SAMS/
*Contents.doc           *Templates/
*Divers.doc             *bruno universite.doc
*Linux/                 *My Pictures/

restore > delete Resume/
restore > ls
./wahib/Personal:
*Ad?le_Nakad.doc        *Overview.doc
*BIMCOR/                Resume/
*My Webs/               *SAMS/
*Contents.doc           *Templates/
*Divers.doc             *bruno universite.doc
*Linux/                 *My Pictures/
```

NOTE: The most expedient way to extract most of the files from a directory is to add the directory to the extraction list and then delete those files that are not needed.

To extract all files in the extraction list from the dump, use the "**extract**" command (Restore will ask which volume the user wishes to mount. The fastest way to extract a few files is to start with the last volume and work towards the first volume) as shown below:

```
restore > extract
You have not read any tapes yet.
Unless you know which volume your file(s) are on you should start
with the last volume and work towards the first.
Specify next volume #: 1
set owner/mode for '.'? [yn] y
```

To exit from the interactive restore mode after you have finished extracting your directories or files, use the "**quit**" command as shown below.

```
/sbin/restore > quit
```

NOTE: Other methods of restoration exist with the `dump` program; consult the manual page of `dump` for more information.

Further documentation

For more details, there is some manual pages related to program `dump` that you can read:

```
$ man dump (8)          - ext2 file system backup
$ man restore (8)       - Restore files or file systems from backups made with dump
```

Backing up and restoring over the network

Backups allow you to restore the availability and integrity of information resources following security breaches and accidents. Without a backup, you may be unable to restore a computer's data after system failures and security breaches.

It is important to develop a plan that is broad enough to cover all the servers you plan to deploy. We must determine what categories of files will be backed up. For example, you may choose to back up only user data files (i.e. `/home`) because damaged system files should be reloaded from the original distribution media.

There are common technological approaches to file backups. For network servers, an authoritative version of the informational content of the server is created and maintained on a secure machine that is backed up. If the server is compromised and its content damaged, it can be reloaded from the secure system maintaining the authoritative version. This approach is typically used for public servers, such as Web servers, because the content changes at more predictable intervals.

It is important to ensure that backups are performed in a secure manner and that the contents of the backups remain secure. We recommend that the plan specify that:

- ✓ The source data is encrypted before being transmitted to the storage medium.
- ✓ The data remains encrypted on the backup storage media.
- ✓ The storage media are kept in a physically secure facility that is protected from man-made and natural disasters.

Transfer your backup in a secure manner over the network

In the previous sections, we have shown you how to make a backup onto both a tape and files from the same system where you execute the backup procedure, with utilities like `tar` and `dump`. These programs (`tar` and `dump`) are capable of making backups over the network as well.

To be able to backup over the network, usually you must ensure that the insecure RPM packages named "`rmt`" and "`rsh`" are installed on your system. The "`rmt`" utility provides remote access to tape devices for programs like `dump`, and `tar`. To complement this, the "`rsh`" package contains a set of programs, which allow users to run commands on remote machines, login to other machines and copy files between machines (`rsh`, `rlogin` and `rcp` are this set of programs).

Since "rsh" can be easily hacked, and "rmt" depends on "rsh" to be able to work, we have chosen to not install them in our setup installation (see chapter related to Linux installation in this book for more information on the subject) for security reasons. Therefore, we must find another way to make backups over the network in a secure manner.

SSH technology is the solution for our problem (see chapter related to OpenSSH in this book for more information on the subject) because it also has the ability to copy data across the network with its "scp" command, through encryption. The following is a method that permits us to use the potential of SSH software to transfer our backups made with tar or dump in a secure manner via the "scp" SSH utility.

Using the scp command of SSH to transfer backups over the network

The scp command copies files between hosts on a network. It uses SSH for data transfer, and uses the same authentication, and provides the same security, as SSH. Unlike the "rcp" utility that comes with the RPM package "rsh", "scp" will transmit your data over the network encrypted. In our example below, we transfer a backup file made with the tar archive program; the procedure to transfer a backup file or tape made with dump program is exactly the same.

Step 1
Before going into the command line that will transfer our data encrypted through the network, it is important to recall that scp command like any other SSH command used for encrypted connection between servers will ask us by default to enter a pass-phrase. This is not useful when we want to automate backup using SSH for the transfer. Fortunately, it is possible to configure SSH to not ask for the pass-phrase before establishing the remote encrypted connection. We do it my creating a new SSH user without a pass-phrase. Of course I suppose that this user already exist in your Unix /etc/passwd file. If you don't understand what I mean, please refer to the chapter related to OpenSSH in this book for more information on the subject.

- To create a new SSH user without a pass-phrase, use the following commands:

```
[root@deep /]# su backadmin
[backadmin@deep /]$ ssh-keygen -d
Generating DSA parameter and key.
Enter file in which to save the key (/home/backadmin/.ssh/id_dsa):
Created directory '/home/backadmin/.ssh'.
Enter passphrase (empty for no passphrase): < Here you press enter
Enter same passphrase again: < Here you press enter again
Your identification has been saved in /home/backadmin/.ssh/id_dsa.
Your public key has been saved in /home/backadmin/.ssh/id_dsa.pub.
The key fingerprint is:
1f:af:aa:22:0a:21:85:3c:07:7a:5c:ae:c2:d3:56:64 backadmin@deep
```

As we can see here, our new SSH user is named "backadmin" and already exist into the /etc/passwd file of the Linux system. We su to this user and generate a new keys pair for him. The most important part here, is when the program ask us to enter a pass-phrase, therefore we just press [Enter] to inform it that we don't want a pass-phrase for this new SSH user.

Step 2
Once the keys pair of our new SSH user have been generated, we must copy its local public key **id_dsa.pub** from its /home/backadmin/.ssh directory remotely into the server from where we want to make the secure connection for transferring the backup files under the name, say, "authorized_keys2". One way to copy the file is to use the ftp command or you might need to send the public key in electronic mail to the administrator of the system. Just include the contents of the ~/.ssh/id_dsa.pub file in the message.

WARNING: Don't forget that the same username in our case "backadmin" must exist on the other server side. This is required only to create the ~/.ssh directory required to place the public key.

Step 3
Now, we must edit the /etc/ssh/ssh_config file on the REMOTE host from where we have sent our **id_dsa.pub** key which has become **authorized_keys2** and add some additional lines to its ssh_config file to allow our new SSH user to connect and transfer backup files without a pass-phrase to the server. The text in bold are the parts of the configuration file that must be customized and adjusted to satisfy your needs

- Edit the **ssh_config** file (vi /etc/ssh/ssh_config) on REMOTE server and add the following lines:

```
# Site-wide defaults for various options

 Host *
   ForwardAgent no
   ForwardX11 no
   RhostsAuthentication no
   RhostsRSAAuthentication no
   RSAAuthentication yes
   PasswordAuthentication no
   FallBackToRsh no
   UseRsh no
   BatchMode no
   CheckHostIP yes
   StrictHostKeyChecking yes
   IdentityFile ~/.ssh/identity
   IdentityFile ~/.ssh/id_dsa
   IdentityFile ~/.ssh/id_rsa1
   IdentityFile ~/.ssh/id_rsa2
   Port 22
   Protocol 2,1
   Cipher blowfish
   EscapeChar ~

Host 207.35.78.13
   ForwardAgent no
   ForwardX11 no
   RhostsAuthentication no
   RhostsRSAAuthentication no
   RSAAuthentication no
   PasswordAuthentication no
   FallBackToRsh no
   UseRsh no
   BatchMode yes
   CheckHostIP no
   StrictHostKeyChecking yes
   IdentityFile ~/.ssh/identity
   IdentityFile ~/.ssh/id_dsa
   IdentityFile ~/.ssh/id_rsa1
   IdentityFile ~/.ssh/id_rsa2
   Port 22
   Protocol 2,1
   Cipher blowfish
   EscapeChar ~
```

From what we can see, is that we have added a copy of the first configuration but have changed two important options. The "**BatchMode yes**" option allow to connect without a pass-phrase and the "**Host 207.35.78.13**" option specifies that only connection coming from IP address 207.35.78.13 (this is the one that we will use with the scp command to transfer the backup files) is allowed to use this configuration where users can connect without a pass-phrase. The other settings are the same as for the original one. Finally we keep the original setting for regular connection to the server where pass-phrase is required.

Step 4
After that, we edit the /etc/ssh/sshd_config file on REMOTE again, and add to the "AllowUsers" option, our new SSH user to allow him to connect to the REMOTE server.

- Edit the **sshd_config** file (vi /etc/ssh/sshd_config) on REMOTE server and change for example the following lines:

  ```
  AllowUsers gmourani
  ```

  To read:

  ```
  AllowUsers gmourani backadmin
  ```

Here we add our user named "backadmin" to the list of allowed user on the REMOTE host.

NOTE: Step 1 to step 4 must be made on each servers from where you want to establish an encrypted remote connection without a pass-phrase to transfer backup over the network.

Step 5
Finally, everything is supposed to be fine now and we are ready to transfer backup over the network in a secure way.

- To use scp to copy a backup tape or file to a remote secure system, use the command:
  ```
  [backadmin@deep /]# scp <localdir/to/filelocation>\
  <user@host:/dir/for/file>
  ```

Where <localdir/to/filelocation> is the directory where your backup file resides on your LOCAL server, and <user@host:/dir/for/file> represents, in order, the username (user) of the person on the REMOTE site that will hold the backup file, the hostname (host) of the remote host where you want to send the backup file, and the remote directory of this host where you want to place the transferred backup file.

A real example will look like this:

```
[backadmin@deep /]# scp -Cp /backups/deep-01Feb.tar \
backadmin@backupserver:/archive/deep/deep-01Feb.tar
deep-01Feb.tgz      |      10479 KB | 154.1 kB/s | ETA: 00:00:00 | 100%
```

NOTE: The "C" option enables compression for fast data transfer over the encrypted session, the "p" option indicates that the modification and access times as well as modes of the source file should be preserved on the copy. This is usually desirable. It is important to note that the <dir/for/file> directory on the remote host (/archive/deep in our example) must be owned by the "username" you specify in your scp command ("admin" is this username in our example) or you may receive error message like: scp: /archive/deep/deep-01Feb.tar: Permission denied.

- To use scp to copy a remote tape or file to the local system, use the command:
  [backadmin@deep /]# **scp <user@host:/dir/for/file>**
  <localdir/to/filelocation>

Where <user@host:/dir/for/file> represents, in order, the username (user) of the person on the REMOTE site that holds the backup file, the hostname (host) of the REMOTE host where you want to get the backup file, and the REMOTE directory of this host where the backup file is kept, and <localdir/to/filelocation> is the LOCAL directory on your system where your want to place the backup file that you get from the REMOTE host.

A real example would look like this:

```
[backadmin@deep /]# scp -Cp admin@backupserver:/archive/deep/deep-
01Feb.tar /backups
admin@backupserver's password:
deep-01Feb.tgz      |      10479 KB | 154.1 kB/s | ETA: 00:00:00 | 100%
```

NOTE: It is important to note that the <localdir/to/filelocation> directory on the LOCAL host ("/backups" in our example) must be owned by the "username" you specify in your scp command ("admin" is this username in our example) or you may receive an error message like: scp: /backups/deep-01Feb.tar: Permission denied.

Alternatives to tar and dump backups programs

AMANDA
AMANDA Homepage: http://www.cs.umd.edu/projects/amanda/

BRU
BRU Homepage: http://www.bru.com/

Part XIV APPENDIXES

In this part

APPENDIX A
Tweaks, Tips and Administration tasks

Tweaks, Tips and Administration tasks

Some of the tips in this section are specific to Linux systems. Most are applicable to UNIX system in general. I make this section available since I think that it can be useful in daily administrative tasks from most of us.

1.0 The `du` utility command

You can use the **`du`** utility to estimate file space usage. For example, to determine in megabyte the sizes of the `/var/log` and `/home` directories trees, type the following command:

```
[root@deep /]# du -sh /var/log /home
3.5M        /var/log
350M        /home
```

Keep in mind that the above command will report the actual size of your data. Now that you know for example that `/home` is using 350M you can move into it and `du -sh *` to locate where the largest files are.

```
[root@deep /]# cd /home/
[root@deep /home]# du -sh *
343M        admin
11k         ftp
6.8M        httpd
12k         lost+found
6.0k        named
6.0k        smbclient
6.0k        test
8.0k        www
```

NOTE: You can add this command to your crontab so that every day you get emailed the desired disk space list, and you'll be able to monitor it without logging in constantly.

1.1 Find the route that the packets sent from your machine to a remote host

If you want to find out the route that the packets sent from your machine to a remote host, simply issue the following command:

```
[root@deep /]# traceroute www.redhat.com
traceroute to www.portal.redhat.com (206.132.41.202), 30 hops max, 38 byte packets
 1  portal.openna.com (207.253.108.5) 98.584 ms 1519.806 ms  109.911 ms
 2  fa5-1-0.rb02-piex.videotron.net (207.96.135.1) 149.888 ms 89.830 ms 109.914 ms
 3  ia-tlpt-bb01-fec1.videotron.net (207.253.253.53) 149.896 ms 99.873 ms 139.930 ms
 4  ia-cduc-bb02-ge2-0.videotron.net (207.253.253.61) 99.897 ms 169.863 ms 329.926 ms
 5  if-4-1.core1.Montreal.Teleglobe.net (207.45.204.5) 409.895 ms 1469.882 ms 109.902 ms
 6  if-1-1.core1.NewYork.Teleglobe.net (207.45.223.109) 189.920 ms 139.852 ms 109.939 ms
 7  206.132.150.133 (206.132.150.133) 99.902 ms 99.724 ms 119.914 ms
 8  pos1-0-2488M.wr2.CLE1.gblx.net (206.132.111.89) 189.899 ms 129.873 ms 129.934 ms
 9  pos8-0-2488m.kcy1.globalcenter.net (206.132.111.82) 169.890 ms 179.884 ms  169.933 ms
10  206.132.114.77 (206.132.114.77) 199.890 ms 179.771 ms 169.928 ms
11  pos8-0-2488M.wr2.SFO1.gblx.net (206.132.110.110) 159.909 ms 199.959 ms 179.837 ms
12  pos1-0-2488M.cr1.SNV2.gblx.net (208.48.118.118) 179.885 ms 309.855 ms 299.937 ms
13  pos0-0-0-155M.hr2.SNV2.gblx.net (206.132.151.46) 329.905 ms 179.843 ms 169.936 ms
14  206.132.41.202 (206.132.41.202)  2229.906 ms 199.752 ms 309.927 ms
```

Where <`www.redhat.com`> is the name or ip address of the host that you want to trace.

1.2 Display the number of times your Web pages have been accessed:
To display quickly the number of times your web page has been accessed use this command:

```
[root@deep /]# grep "GET / HTTP" /var/log/httpd/access_log | wc -l
      467
```

1.3 Shut down most services altogether
As root, you can shut down most services altogether with the following command:

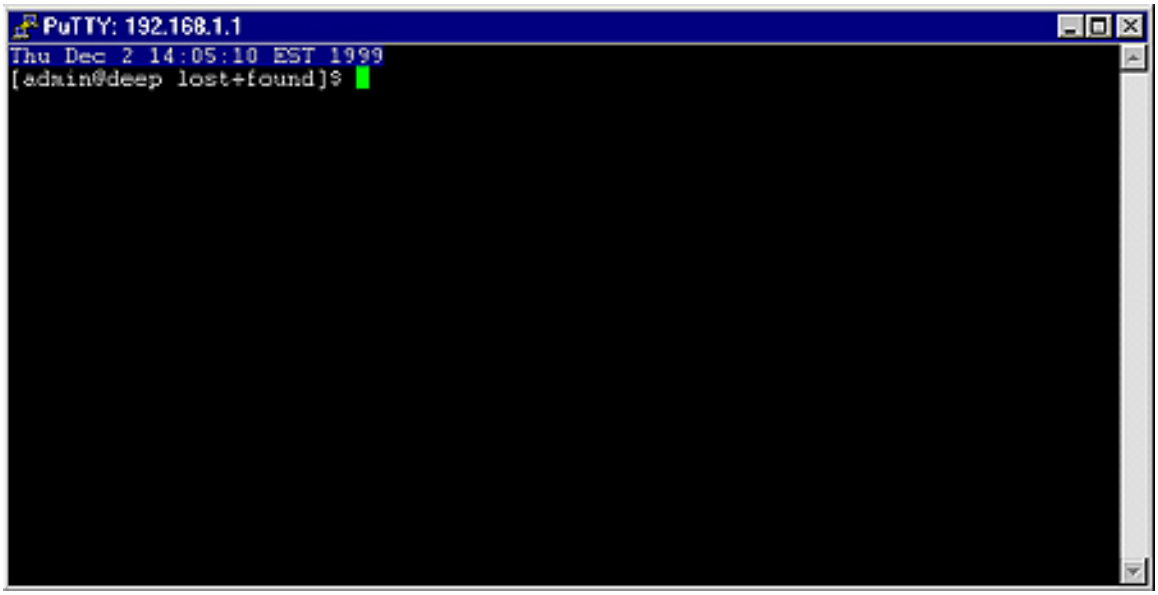
```
[root@deep /]# killall httpd smbd nmbd slapd named
```

The above command will shut down the Apache server, Samba services, LDAP server, and DNS server respectively.

1.4 Want a clock on the top of your terminal for all user?
Edit the **`profile`** file (`vi /etc/profile`) and add the following line:

```
PROMPT_COMMAND='echo -ne
"\0337\033[2;999r\033[1;1H\033[00;44m\033[K"`date`"\033[00m\0338"'
```

The result will look like:



1.5 Do you have `lsof` installed on your server?
If not, install it and execute `lsof -i`. This should list which ports you have open on your machine. The `lsof` program is a great tool as it will tell you which processes are listening on a given port.

```
[root@deep /]# lsof -i
COMMAND     PID     USER    FD  TYPE    DEVICE  SIZE    NODE    NAME
Inetd       344     root    4u          IPv4    327     TCP *:ssh (LISTEN)
```

1.6 Run commands on remote servers via `ssh` protocol without logging in

The `ssh` command can also be used to run commands on remote systems without logging in. The output of the command is displayed, and control returns to the local system. Here is an example which will display all the users logged in on the remote system.

```
[admin@deep /]$ ssh boreas.openna.com who
admin@boreas.openna.com's password:
root     tty1     Dec  2 14:45
admin    tty2     Dec  2 14:45
wahib    pts/0    Dec  2 11:38
```

1.7 Filename Completion

Tab filename completion allows you to type in portions of a filename or program, and then press `[TAB]`, and it will complete the filename for you. If there's more than one file or program that starts with what you already typed in, it will beep, and then when you press `[TAB]` again it will list all the files that start with what you initially typed.

> **NOTE:** AFAIK, filename completion works only for `bash` by default but not for e.g. `ksh`. If you use `ksh` instead of `bash` as the command shell then to enable "Filename Completion" in `ksh`, you have to set the following:
>
> ```
> set -o vi-tabcomplete
> ```

1.8 Special Characters

You can quickly accomplish tasks that you perform frequently by using shortcut keys — one or more keys you press on the keyboard to complete a task. For example, special characters can be used on the Linux shell like the following:

`Control-d` : If you are in the shell and hit `control-d` you get logged off.

`Control-l`: If you are in the shell and hit `control-l` you clear the screen.

`?` : This is a wildcard. This can represent a single character. If you specified something at the command line like "`m?b`" Linux would look for `mob`, `mib`, `mub`, and every other letter/number between a-z, 0-9.

`*` : This can represent any number of characters. If you specified a "`mi*`" it would use `mit`, `mim`, `miiii`, `miya`, and ANYTHING that starts with "`mi`". "`m*l`" could by `mill`, `mull`, `ml`, and anything that starts with an "`m`" and ends with an "`l`".

`[]` - Specifies a range. if I did `m[o,u,i]m` Linux would think: `mim`, `mum`, `mom` if I did: `m[a-d]m` Linux would think: `mam`, `mbm`, `mcm`, `mdm`. Get the idea? The `[]`, `?`, and `*` are usually used with copying, deleting, and directory listings.

NOTE: EVERYTHING in Linux is CASE sensitive. This means "`Bill`" and "`bill`" are not the same thing. This allows for many files to be able to be stored, since "`Bill`" "`bill`" "`bIll`" "`biLl`", etc. can be different files. So, when using the `[]` stuff, you have to specify capital letters if any files you are dealing with have capital letters. Much of everything is lower case in UNIX, though.

1.9 Freeze a process ID temporally

The UNIX `kill` command name is misleading: Only some incantations of the `kill` command actually terminate the target process. "`kill -STOP`" suspends the target process immediately and unconditionally. The process can still be resumed with "`kill -CONT`" as if nothing happened. This command can be useful when you want for example to freeze a suspicious process running on your system and conduct any further investigations at leisure.

```
[root@deep /]# kill -STOP 401
```

The above command will suspend the process ID 401, which is related to the `sshd` daemon on my running system. Of course the process number will be different on your server, therefore take this process number as an example only.

```
[root@deep /]# kill -CONT 401
```

The above command will resume the process ID 401, which is related to the `sshd` daemon on my running system.

APPENDIX B
Contributor Users

Contributor Users

This is a list of all Linux users around the world who have participated in the development of this book in a voluntary base by providing good comments, ideas, helps, suggestions, correction and any other information of this kind. To thanks them, I make this section available and list in a non-alphabetically order their names. Sorry if I left anyone out.

Brain Jensen
Rob Egelink
John Constantine
Carl Friedberg
Bart Van Pelt
Liang Ge
Ivan Darmawan
Jerome Alet
Arthur de Pauw
Sigfus Oddsson
Tim Stoop
Wolf aliase Paul
Pekka Saari
P Tiili
Catalin Russen
Raphael Quoilin
Bruce W. Mohler
Eugene Teo
Ivan Kolemanov
Jorge Bianquetti
Flavio Domingos
Wolf
Arthur de Pauw
Carl Friedberg
Brian Flemming Jensen
Vinh Nguyen
Xu Ying
Syamsul Hidayat
David Rousseau
Madhusudan Madhu
George Toft
Werner Puschitz
Mathieu Sebastien
Chris de Vidal

Naïf
ISM Kolemanor
John Francis Lee
Tim Groenwals
Greg Walsh
Shawn Duffy
Hilton Travis
Sylvain Rivest
Timur Snoke
Nelson
Eric Gerbier
Andre
Peter
Carlos A. Molina G
J
Paco Gracia
Jame Saffeld
Scott England-Sullivan
Frederic Faure
Teeguh Iskanto
Sebastien Letard
David Tillery
Jim Cornelson
Walker White
John Crain
Giuseppe
Sinisa
Charles Cosby
Stapleton Bernard
Neal Dias
Nathan Hopper
Olafur Gudmundsson
Matt Roberts

Sendy Harris
Steve Snyder
Mark Farey
Ligu Song
Jens Kerle
Serge Rodrigues
Oden Erikson
Michael Moore
Randy Jordan
Radu Coroi
Tou Brian
Brian Richardson
Mike Baker
Fred Burke
Tim Sandquist
Rene Teinberg
Bernhard Rosenkraenzer
Gregory A Lundberg
Andre Gerhard
Matthias Zeichmann
Neil W Rickert
Mark.Andrews
Erik Loeth
David South Jr
John LeRoy Crain
Roberto Piola
Oliver Enzmann
Michael Brown
La-Roque
Colin Henry
Hong Sukbum
Brian Wellington
Colin Henry

APPENDIX C
Obtaining Requests for Comments (RFCs)

Obtaining Requests for Comments (RFCs)

Requests for Comments (RFCs) is an ongoing set of documents issued by the Internet Engineering Task Force (IETF) at the Network Information Center (NIC) that presents new protocols and establishes standards for the Internet protocol suite. Each such document defines an aspect of protocol regarding the Internet. We have listed below all the RFCs that pertain to this book, and various software described in this book. RFCs are available from the following site: http://www.cis.ohio-state.edu/rfc/

RFC706
On the Junk Mail Problem.

RFC733
Standard for the Format of ARPA Network Text Messages.

RFC768
User Datagram Protocol (UDP).

RFC791
Internet Protocol (IP).

RFC792
Internet Control Message Protocol (ICMP).

RFC793
Transmission Control Protocol (TCP).

RFC805
Computer Mail Meting Notes.

RFC821
Simple Mail Transfert Protocol (SMTP).

RFC822
Standard for the Format of ARPA Internet Text Massages.

RFC934
Proposed Standard for Message Encapsulation.

RFC950
IP Subnet Extention.

RFC959
File Transfer Protocol (FTP).

RFC976
UUCP Mail Interchange Format Standard.

RFC1034
Domain Names: Concepts and Facilities.

RFC1036
Standard for Interchange of USENET Message.

RFC1058
Routing Information Protocol (RIP).

RFC1112
Internet Group Multicast Protocol (IGMP).

RFC1122
Requirement for Internet Host—Communication Layers.

RFC1123
Requirements for Internet Host—Application and Support.

RFC1137
Mapping Between Full RFC 822 and RFC 822 with Restricted Encoding.

RFC1153
Digest Message Format.

RFC1155
Structure of Management Information (SMI).

RFC1157
Simple Network Management Protocol (SNMP).

RFC1176
Interactive Mail Access Protocol: Version 2.

RFC1274
The COSINE and Internet X.500 Schema.

RFC1275
Replication Requirements to provide an Internet Directory using X.500.

RFC1279
X.500 and Domains.

RFC1308
Executive Introduction to Directory Services Using the X.500 Protocol.

RFC1309
Technical Overview of Directory Services Using the X.500 Protocol.

RFC1310
The Internet Standards Process.

RFC1319
MD2 Message-Digest Algorithm.

RFC1320
MD4 Message-Digest Algorithm.

RFC1321
MD5 Message-Digest Algorithm.

RFC1343
User Agent Configuration Mechanism for Multimedia Mail Format Information.

RFC1344
Implications of MIME for Internet Mail Gateways.

RFC1345
Character Mnemonics and Character Sets.

RFC1421
Privacy Enhancement for Internet Electronic Mail: Part I—Message Encipherment and authentication Procedures.

RFC1422
Privacy Enhancement for Internet Electronic Mail: Part II—Certificate-based key Management.

RFC1423
Privacy Enhancement for Internet Electronic Mail: Part III—Algorithms, modes, and identifiers [Draft].

RFC1428
Transmition of Internet Mail from Just-Send-8 to 8bit-SMTP/MIME.

RFC1430
A Strategic Plan for Deploying an Internet X.500 Directory Service.

RFC1492
An Access Control Protocol, Sometimes Called TACACS.

RFC1495
Mapping Between X.400(1988)/ISO 10021 and RFC 822.

RFC1496
X.400 1988 to 1984 Downgrading.

RFC1505
Encoding Header Field for Internet Messages.

RFC1510
The Kerberos Network Authentication Service (V5).

RFC1519
Classless Inter-Domain Routing (CIDR) Assignment and Aggregation Strategy.

RFC1521
MIME (Multipurpose Internet Mail Extensions): Mechanisms for Specifying and Describing the Format of Internet Message Bodies (MIME).

RFC1522
Representation of Non-ASCII Text in Internet Message Headers.

RFC1558
A String Representation of LDAP Search Filters.

RFC1566
Mail Monitoring MIB.

RFC1579
Firewall-Friendly FTP.

RFC1583
Open Shortest Path First Routing V2 (OSPF2).

RFC1617
Naming and Structuring Guidelines for X.500 Directory Pilots.

RFC1625
WAIS over Z39.50-1988.

RFC1631
The IP Network Address Translator (NAT).

RFC1652
SMTP Service Extentions for 8bit-MIMEtransport.

RFC1661
Point-to-Point Protocol (PPP).

RFC1711
Classifications in E-mail Routing.

RFC1725
Post Office Protocol, Version 3 (POP)3.

RFC1738
Uniform Resource Locators (URL).

RFC1739
A Primer on Internet and TCP/IP Tools.

RFC1777
Lightweight Directory Access Protocol.

RFC1778
The String Representation of Standard Attribute Syntaxes.

RFC1779
A String Representation of Distinguished Names.

RFC1781
Using the OSI Directory to Achieve User Friendly Naming.

RFC1796
Not All RFCs are Standards.

RFC1798
Connection-less Lightweight Directory Access Protocol.

RFC1823
The LDAP Application Program Interface.

RFC1830
SMTP Services Extentions for Transmission of Large and Binary MIME Messages.

RFC1844
Multimedia E-mail (MIME) User Agent checklist.

RFC1845
SMTP Service Extension for Checkpoint/Restart.

RFC1846
SMTP 521 Reply Code.

RFC1854
SMTP Service Extention for command pipelining.

RFC1855
Netiquette Guidelines.

RFC1864
The content-MD5 Header.

RFC1866
Hypertext Markup Language - 2.0.

RFC1869
SMTP Service Extensions.

RFC1870
SMTP Service Extension for Message Size Declaration.

RFC1872
The MIME Multipart/Related Content-type.

RFC1873
Message/External-Body Content-ID Access-type.

RFC1883
Internet Protocol, Version 6 (Ipv6) Specification.

RFC1884
IP Version 6 Addressing Atchitecture.

RFC1886
DNS Extentions to support IP version 6.

RFC1891
SMTP Service Extension for Delivery Status Notifications.

RFC1892
The Multipart/Report Content Type for the Reporting of Mail System Administrative Messages.

RFC1893
Enhanced Mail System Status Codes.

RFC1894
An Extensible Message Format for Delivery Status Notifications.

RFC1918
Address Allocation for Private Internets.

RFC1928
SOCKS Protocol Version 5.

RFC1929
Username/Password Authentication for SOCKS V5.

RFC1959
An LDAP URL Format.

RFC1960
A String Representation of LDAP Search Filters.

RFC1961
GSS-API Authentication Method for SOCKS Version 5.

RFC2003
IP Encapsulation within IP.

RFC2028
The Organizations Involved in the IETF Standards Process.

RFC2044
UTF-8, a transformation format of Unicode and ISO 10646.

RFC2060
Internet Message Access Protocol – Version 4rev1 (IMAP4).

RFC2104
HMAC: Keyed-Hashing for Message Authentication.

RFC2138
Remote Authentication Dial In User Service (RADIUS).

RFC2164
Use of an X.500/LDAP directory to support MIXER address mapping.

RFC2200
Internet Official Protocol Standards.

RFC2218
A Common Schema for the Internet White Pages Service.

RFC2247
Using Domains in LDAP/X.500 Distinguished Names.

RFC2251
Lightweight Directory Access Protocol (v3).

RFC2252
Lightweight Directory Access Protocol (v3): Attribute Syntax Definitions.

RFC2253
Lightweight Directory Access Protocol (v3): UTF-8 String Representation of Distinguished Names

RFC2254
The String Representation of LDAP Search Filters.

RFC2255
The LDAP URL Format.

RFC2256
A Summary of the X.500(96) User Schema for use with LDAPv3.

RFC2279
UTF-8, a transformation format of ISO 10646.

RFC2293
Representing Tables and Subtrees in the X.500 Directory.

RFC2294
Representing the O/R Address hierarchy in the X.500 Directory Information Tree.

RFC2305
A Simple Mode of Facsimile Using Internet Mail.

RFC2307
An Approach for Using LDAP as a Network Information Service.

RFC2313
PKCS 1: RSA Encryption Version 1-5.

RFC2314
PKCS 10: Certification Request Syntax Version 1-5.

RFC2315
PKCS 7: Cryptographic Message Syntax Version 1-5.

RFC2377
Naming Plan for Internet Directory-Enabled Applications.

APPENDIX D
Port list

Port list

The port numbers are divided into three ranges: the Well Known Ports, the Registered Ports, and the Dynamic and/or Private Ports. There are two series of ports, using two different protocols: TCP and UDP. They are different, although they can have the same port number. UDP ports can't be telneted. This appendix also includes a list of ports commonly used by Trojan horses. All open ports have a service or daemon running on it. A service or a daemon is nothing but the software running on these ports, which provide a certain service to the users who connect to it.

You can find out the corresponding services running on them, referring to the table below or to the RFC 1700 (http://www.cis.ohio-state.edu/rfc/), which contains the complete and updated list of Port Numbers and the corresponding popularly running services.

Well Known Ports:

The Well Known Ports are those from 0 through 1023 and are assigned by IANA (**I**nternet **A**ssigned **N**umbers **A**uthority). For the latest status, please check at: http://www.iana.org/

Keyword	Decimal	Description
	0/tcp	Reserved
	0/udp	Reserved
tcpmux	1/tcp	TCP Port Service
tcpmux	1/udp	TCP Port Service
compressnet	2/tcp	Management Utility
compressnet	2/udp	Management Utility
compressnet	3/tcp	Compression Process
compressnet	3/udp	Compression Process
#	4/tcp	Unassigned
#	4/udp	Unassigned
rje	5/tcp	Remote Job Entry
rje	5/udp	Remote Job Entry
#	6/tcp	Unassigned
#	6/udp	Unassigned
echo	7/tcp	Echo
echo	7/udp	Echo
#	8/tcp	Unassigned
#	8/udp	Unassigned
discard	9/tcp	Discard
discard	9/udp	Discard
#	10/tcp	Unassigned
#	10/udp	Unassigned
systat	11/tcp	Active Users
systat	11/udp	Active Users
#	12/tcp	Unassigned
#	12/udp	Unassigned
daytime	13/tcp	Daytime (RFC 867)
daytime	13/udp	Daytime (RFC 867)
#	14/tcp	Unassigned
#	14/udp	Unassigned
#	15/tcp	Unassigned
#	15/udp	Unassigned
#	16/tcp	Unassigned
#	16/udp	Unassigned
qotd	17/tcp	Quote of the Day
qotd	17/udp	Quote of the Day
msp	18/tcp	Message Send
msp	18/udp	Message Send
chargen	19/tcp	Character Generator
chargen	19/udp	Character Generator
ftp-data	20/tcp	File Transfer
ftp-data	20/udp	File Transfer
ftp	21/tcp	File Transfer
ftp	21/udp	File Transfer

Keyword	Decimal	Description
opc-job-track	424/tcp	IBM Operations
opc-job-track	424/udp	IBM Operations
icad-el	425/tcp	ICAD
icad-el	425/udp	ICAD
smartsdp	426/tcp	smartsdp
smartsdp	426/udp	smartsdp
svrloc	427/tcp	Server Location
svrloc	427/udp	Server Location
ocs_cmu	428/tcp	OCS_CMU
ocs_cmu	428/udp	OCS_CMU
ocs_amu	429/tcp	OCS_AMU
ocs_amu	429/udp	OCS_AMU
utmpsd	430/tcp	UTMPSD
utmpsd	430/udp	UTMPSD
utmpcd	431/tcp	UTMPCD
utmpcd	431/udp	UTMPCD
iasd	432/tcp	IASD
iasd	432/udp	IASD
nnsp	433/tcp	NNSP
nnsp	433/udp	NNSP
mobileip-agent	434/tcp	MobileIP-Agent
mobileip-agent	434/udp	MobileIP-Agent
mobilip-mn	435/tcp	MobilIP-MN
mobilip-mn	435/udp	MobilIP-MN
dna-cml	436/tcp	DNA-CML
dna-cml	436/udp	DNA-CML
comscm	437/tcp	comscm
comscm	437/udp	comscm
dsfgw	438/tcp	dsfgw
dsfgw	438/udp	dsfgw
dasp	439/tcp	dasp
dasp	439/udp	dasp
sgcp	440/tcp	sgcp
sgcp	440/udp	sgcp
decvms-sysmgt	441/tcp	decvms-sysmgt
decvms-sysmgt	441/udp	decvms-sysmgt
cvc_hostd	442/tcp	cvc_hostd
cvc_hostd	442/udp	cvc_hostd
https	443/tcp	http proto TLS/SSL
https	443/udp	http proto TLS/SSL
snpp	444/tcp	Simple Network
snpp	444/udp	Simple Network
microsoft-ds	445/tcp	Microsoft-DS
microsoft-ds	445/udp	Microsoft-DS

```
ssh             22/tcp    SSH Remote Login
ssh             22/udp    SSH Remote Login
telnet          23/tcp    Telnet
telnet          23/udp    Telnet
#               24/tcp    any private mail sys
#               24/udp    any private mail sys
smtp            25/tcp    Simple Mail Transfer
smtp            25/udp    Simple Mail Transfer
#               26/tcp    Unassigned
#               26/udp    Unassigned
nsw-fe          27/tcp    NSW User System FE
nsw-fe          27/udp    NSW User System FE
#               28/tcp    Unassigned
#               28/udp    Unassigned
msg-icp         29/tcp    MSG ICP
msg-icp         29/udp    MSG ICP
#               30/tcp    Unassigned
#               30/udp    Unassigned
msg-auth        31/tcp    MSG Authentication
msg-auth        31/udp    MSG Authentication
#               32/tcp    Unassigned
#               32/udp    Unassigned
dsp             33/tcp    Display Support
dsp             33/udp    Display Support
#               34/tcp    Unassigned
#               34/udp    Unassigned
#               35/tcp    any private printer
#               35/udp    any private printer
#               36/tcp    Unassigned
#               36/udp    Unassigned
time            37/tcp    Time
time            37/udp    Time
rap             38/tcp    Route Access
rap             38/udp    Route Access
rlp             39/tcp    Resource Location
rlp             39/udp    Resource Location
#               40/tcp    Unassigned
#               40/udp    Unassigned
graphics        41/tcp    Graphics
graphics        41/udp    Graphics
nameserver      42/tcp    Host Name Server
nameserver      42/udp    Host Name Server
nicname         43/tcp    Who Is
nicname         43/udp    Who Is
mpm-flags       44/tcp    MPM FLAGS Protocol
mpm-flags       44/udp    MPM FLAGS Protocol
mpm             45/tcp    MPM [recv]
mpm             45/udp    MPM [recv]
mpm-snd         46/tcp    MPM [default send]
mpm-snd         46/udp    MPM [default send]
ni-ftp          47/tcp    NI FTP
ni-ftp          47/udp    NI FTP
auditd          48/tcp    Digital Audit Daemon
auditd          48/udp    Digital Audit Daemon
tacacs          49/tcp    Login Host Protocol
tacacs          49/udp    Login Host Protocol
re-mail-ck      50/tcp    Remote Mail Checking
re-mail-ck      50/udp    Remote Mail Checking
la-maint        51/tcp    IMP
la-maint        51/udp    IMP
xns-time        52/tcp    XNS Time Protocol
xns-time        52/udp    XNS Time Protocol
domain          53/tcp    Domain Name Server
domain          53/udp    Domain Name Server
xns-ch          54/tcp    XNS Clearinghouse
xns-ch          54/udp    XNS Clearinghouse
isi-gl          55/tcp    ISI Graphics Lang
isi-gl          55/udp    ISI Graphics Lang
xns-auth        56/tcp    XNS Authentication
xns-auth        56/udp    XNS Authentication
#               57/tcp    Private term access
ddm-rdb         446/tcp    DDM-RDB
ddm-rdb         446/udp    DDM-RDB
ddm-dfm         447/tcp    DDM-RFM
ddm-dfm         447/udp    DDM-RFM
ddm-ssl         448/tcp    DDM-SSL
ddm-ssl         448/udp    DDM-SSL
as-servermap    449/tcp    AS Server Mapper
as-servermap    449/udp    AS Server Mapper
tserver         450/tcp    TServer
tserver         450/udp    TServer
sfs-smp-net     451/tcp    Cray Network
sfs-smp-net     451/udp    Cray Network
sfs-config      452/tcp    Cray SFS config
sfs-config      452/udp    Cray SFS config
creativeserver  453/tcp    CreativeServer
creativeserver  453/udp    CreativeServer
contentserver   454/tcp    ContentServer
contentserver   454/udp    ContentServer
creativepartnr  455/tcp    CreativePartnr
creativepartnr  455/udp    CreativePartnr
macon-tcp       456/tcp    macon-tcp
macon-udp       456/udp    macon-udp
scohelp         457/tcp    scohelp
scohelp         457/udp    scohelp
appleqtc        458/tcp    apple quick time
appleqtc        458/udp    apple quick time
ampr-rcmd       459/tcp    ampr-rcmd
ampr-rcmd       459/udp    ampr-rcmd
skronk          460/tcp    skronk
skronk          460/udp    skronk
datasurfsrv     461/tcp    DataRampSrv
datasurfsrv     461/udp    DataRampSrv
datasurfsrvsec  462/tcp    DataRampSrvSec
datasurfsrvsec  462/udp    DataRampSrvSec
alpes           463/tcp    alpes
alpes           463/udp    alpes
kpasswd         464/tcp    kpasswd
kpasswd         464/udp    kpasswd
#               465        Unassigned
digital-vrc     466/tcp    digital-vrc
digital-vrc     466/udp    digital-vrc
mylex-mapd      467/tcp    mylex-mapd
mylex-mapd      467/udp    mylex-mapd
photuris        468/tcp    proturis
photuris        468/udp    proturis
rcp             469/tcp    Radio Control Proto
rcp             469/udp    Radio Control Proto
scx-proxy       470/tcp    scx-proxy
mondex          471/tcp    Mondex
mondex          471/udp    Mondex
ljk-login       472/tcp    ljk-login
ljk-login       472/udp    ljk-login
hybrid-pop      473/tcp    hybrid-pop
hybrid-pop      473/udp    hybrid-pop
tn-tl-w1        474/tcp    tn-tl-w1
tn-tl-w2        474/udp    tn-tl-w2
tcpnethaspsrv   475/tcp    tcpnethaspsrv
tcpnethaspsrv   475/udp    tcpnethaspsrv
tn-tl-fd1       476/tcp    tn-tl-fd1
tn-tl-fd1       476/udp    tn-tl-fd1
ss7ns           477/tcp    ss7ns
ss7ns           477/udp    ss7ns
spsc            478/tcp    spsc
spsc            478/udp    spsc
iafserver       479/tcp    iafserver
iafserver       479/udp    iafserver
iafdbase        480/tcp    iafdbase
iafdbase        480/udp    iafdbase
ph              481/tcp    Ph service
ph              481/udp    Ph service
bgs-nsi         482/tcp    bgs-nsi
```

```
#             57/udp    Private term access
xns-mail      58/tcp    XNS Mail
xns-mail      58/udp    XNS Mail
#             59/tcp    Private file service
#             59/udp    Private file service
#             60/tcp    Unassigned
#             60/udp    Unassigned
ni-mail       61/tcp    NI MAIL
ni-mail       61/udp    NI MAIL
acas          62/tcp    ACA Services
acas          62/udp    ACA Services
whois++       63/tcp    whois++
whois++       63/udp    whois++
covia         64/tcp    Com Integrator (CI)
covia         64/udp    Com Integrator (CI)
tacacs-ds     65/tcp    TACACS-Database Serv
tacacs-ds     65/udp    TACACS-Database Serv
sql*net       66/tcp    Oracle SQL*NET
sql*net       66/udp    Oracle SQL*NET
bootps        67/tcp    Bootstrap Server
bootps        67/udp    Bootstrap Server
bootpc        68/tcp    Bootstrap Client
bootpc        68/udp    Bootstrap Client
tftp          69/tcp    Trivial File Trans
tftp          69/udp    Trivial File Trans
gopher        70/tcp    Gopher
gopher        70/udp    Gopher
netrjs-1      71/tcp    Remote Job Service
netrjs-1      71/udp    Remote Job Service
netrjs-2      72/tcp    Remote Job Service
netrjs-2      72/udp    Remote Job Service
netrjs-3      73/tcp    Remote Job Service
netrjs-3      73/udp    Remote Job Service
netrjs-4      74/tcp    Remote Job Service
netrjs-4      74/udp    Remote Job Service
#             75/tcp    Private dial out
#             75/udp    Private dial out
deos          76/tcp    ExternalObject Store
deos          76/udp    ExternalObject Store
#             77/tcp    Private RJE service
#             77/udp    Private RJE service
vettcp        78/tcp    vettcp
vettcp        78/udp    vettcp
finger        79/tcp    Finger
finger        79/udp    Finger
http          80/tcp    World Wide Web HTTP
http          80/udp    World Wide Web HTTP
hosts2-ns     81/tcp    HOSTS2 Name Server
hosts2-ns     81/udp    HOSTS2 Name Server
xfer          82/tcp    XFER Utility
xfer          82/udp    XFER Utility
mit-ml-dev    83/tcp    MIT ML Device
mit-ml-dev    83/udp    MIT ML Device
ctf           84/tcp    CommonTrace Facility
ctf           84/udp    CommonTrace Facility
mit-ml-dev    85/tcp    MIT ML Device
mit-ml-dev    85/udp    MIT ML Device
mfcobol       86/tcp    Micro Focus Cobol
mfcobol       86/udp    Micro Focus Cobol
#             87/tcp    Private term link
#             87/udp    Private term link
kerberos      88/tcp    Kerberos
kerberos      88/udp    Kerberos
su-mit-tg     89/tcp    SU/MITTelnet Gateway
su-mit-tg     89/udp    SU/MITTelnet Gateway
dnsix         90/tcp    DNSIX
dnsix         90/udp    DNSIX
mit-dov       91/tcp    MIT Dover Spooler
mit-dov       91/udp    MIT Dover Spooler
npp           92/tcp    Network Printing
npp           92/udp    Network Printing
bgs-nsi       482/udp   bgs-nsi
ulpnet        483/tcp   ulpnet
ulpnet        483/udp   ulpnet
integra-sme   484/tcp   Integra Software
integra-sme   484/udp   Integra Software
powerburst    485/tcp   Air Soft Power Burst
powerburst    485/udp   Air Soft Power Burst
avian         486/tcp   avian
avian         486/udp   avian
saft          487/tcp   saft
saft          487/udp   saft
gss-http      488/tcp   gss-http
gss-http      488/udp   gss-http
nest-protocol   489/tcp   nest-protocol
nest-protocol   489/udp   nest-protocol
micom-pfs     490/tcp   micom-pfs
micom-pfs     490/udp   micom-pfs
go-login      491/tcp   go-login
go-login      491/udp   go-login
ticf-1        492/tcp   Transport for FNA
ticf-1        492/udp   Transport for FNA
ticf-2        493/tcp   Transport for FNA
ticf-2        493/udp   Transport for FNA
pov-ray       494/tcp   POV-Ray
pov-ray       494/udp   POV-Ray
intecourier   495/tcp   intecourier
intecourier   495/udp   intecourier
pim-rp-disc   496/tcp   PIM-RP-DISC
pim-rp-disc   496/udp   PIM-RP-DIS
dantz         497/tcp   dantz
dantz         497/udp   dantz
siam          498/tcp   siam
siam          498/udp   siam
iso-ill       499/tcp   ISO ILL Protocol
iso-ill       499/udp   ISO ILL Protocol
isakmp        500/tcp   isakmp
isakmp        500/udp   isakmp
stmf          501/tcp   STMF
stmf          501/udp   STMF
asa-appl-proto  502/tcp   asa-appl-proto
asa-appl-proto  502/udp   asa-appl-proto
intrinsa      503/tcp   Intrinsa
intrinsa      503/udp   Intrinsa
citadel       504/tcp   citadel
citadel       504/udp   citadel
mailbox-lm    505/tcp   mailbox-lm
mailbox-lm    505/udp   mailbox-lm
ohimsrv       506/tcp   ohimsrv
ohimsrv       506/udp   ohimsrv
crs           507/tcp   crs
crs           507/udp   crs
xvttp         508/tcp   xvttp
xvttp         508/udp   xvttp
snare         509/tcp   snare
snare         509/udp   snare
fcp           510/tcp   FirstClass Protocol
fcp           510/udp   FirstClass Protocol
passgo        511/tcp   PassGo
passgo        511/udp   PassGo
exec          512/tcp   remote process exec
comsat        512/udp
biff          512/udp   used by mail system
login         513/tcp   remote login
who           513/udp   maintains data bases
shell         514/tcp   cmd
syslog        514/udp
printer       515/tcp   spooler
printer       515/udp   spooler
videotex      516/tcp   videotex
videotex      516/udp   videotex
talk          517/tcp   like tenex
```

dcp	93/tcp	Device Control
dcp	93/udp	Device Control
objcall	94/tcp	Tivoli Object
objcall	94/udp	Tivoli Object
supdup	95/tcp	SUPDUP
supdup	95/udp	SUPDUP
dixie	96/tcp	DIXIE Specification
dixie	96/udp	DIXIE Specification
swift-rvf	97/tcp	Swift Remote
swift-rvf	97/udp	Swift Remote
tacnews	98/tcp	TAC News
tacnews	98/udp	TAC News
metagram	99/tcp	Metagram Relay
metagram	99/udp	Metagram Relay
newacct	100/tcp	[unauthorized use]
hostname	101/tcp	NIC Host Name Server
hostname	101/udp	NIC Host Name Server
iso-tsap	102/tcp	ISO-TSAP Class 0
iso-tsap	102/udp	ISO-TSAP Class 0
gppitnp	103/tcp	Genesis Trans Net
gppitnp	103/udp	Genesis Trans Net
acr-nema	104/tcp	ACR-NEMA Digital
acr-nema	104/udp	ACR-NEMA Digital
cso	105/tcp	CCSO name server
cso	105/udp	CCSO name server
csnet-ns	105/tcp	Mailbox Nameserver
csnet-ns	105/udp	Mailbox Nameserver
3com-tsmux	106/tcp	3COM-TSMUX
3com-tsmux	106/udp	3COM-TSMUX
rtelnet	107/tcp	Remote Telnet
rtelnet	107/udp	Remote Telnet
snagas	108/tcp	SNA
snagas	108/udp	SNA
pop2	109/tcp	Post Office - V2
pop2	109/udp	Post Office - V2
pop3	110/tcp	Post Office - V3
pop3	110/udp	Post Office - V3
sunrpc	111/tcp	SUN Remote Proc Call
sunrpc	111/udp	SUN Remote Proc Call
mcidas	112/tcp	McIDAS
mcidas	112/udp	McIDAS
ident	113/tcp	
auth	113/tcp	Auth Service
auth	113/udp	Auth Service
audionews	114/tcp	Audio News Multicast
audionews	114/udp	Audio News Multicast
sftp	115/tcp	Simple FTP
sftp	115/udp	Simple FTP
ansanotify	116/tcp	ANSA REX Notify
ansanotify	116/udp	ANSA REX Notify
uucp-path	117/tcp	UUCP Path Service
uucp-path	117/udp	UUCP Path Service
sqlserv	118/tcp	SQL Services
sqlserv	118/udp	SQL Services
nntp	119/tcp	NNTP
nntp	119/udp	NNTP
cfdptkt	120/tcp	CFDPTKT
cfdptkt	120/udp	CFDPTKT
erpc	121/tcp	Remote Pro.Call
erpc	121/udp	Remote Pro.Call
smakynet	122/tcp	SMAKYNET
smakynet	122/udp	SMAKYNET
ntp	123/tcp	Network Time Proto
ntp	123/udp	Network Time Proto
ansatrader	124/tcp	ANSA REX Trader
ansatrader	124/udp	ANSA REX Trader
locus-map	125/tcp	Locus Net Map Ser
locus-map	125/udp	Locus Net Map Ser
nxedit	126/tcp	NXEdit
nxedit	126/udp	NXEdit
#unitary	126/tcp	Unisys Unitary Login
talk	517/udp	like tenex
ntalk	518/tcp	
ntalk	518/udp	
utime	519/tcp	unixtime
utime	519/udp	unixtime
efs	520/tcp	extended file name
router	520/udp	routing process
ripng	521/tcp	ripng
ripng	521/udp	ripng
ulp	522/tcp	ULP
ulp	522/udp	ULP
ibm-db2	523/tcp	IBM-DB2
ibm-db2	523/udp	IBM-DB2
ncp	524/tcp	NCP
ncp	524/udp	NCP
timed	525/tcp	Timeserver
timed	525/udp	Timeserver
tempo	526/tcp	Newdate
tempo	526/udp	Newdate
stx	527/tcp	Stock IXChange
stx	527/udp	Stock IXChange
custix	528/tcp	Customer IXChange
custix	528/udp	Customer IXChange
irc-serv	529/tcp	IRC-SERV
irc-serv	529/udp	IRC-SERV
courier	530/tcp	rpc
courier	530/udp	rpc
conference	531/tcp	chat
conference	531/udp	chat
netnews	532/tcp	readnews
netnews	532/udp	readnews
netwall	533/tcp	Emergency broadcasts
netwall	533/udp	Emergency broadcasts
mm-admin	534/tcp	MegaMedia Admin
mm-admin	534/udp	MegaMedia Admin
iiop	535/tcp	iiop
iiop	535/udp	iiop
opalis-rdv	536/tcp	opalis-rdv
opalis-rdv	536/udp	opalis-rdv
nmsp	537/tcp	Media Streaming
nmsp	537/udp	Media Streaming
gdomap	538/tcp	gdomap
gdomap	538/udp	gdomap
apertus-ldp	539/tcp	Apertus Technologies
apertus-ldp	539/udp	Apertus Technologies
uucp	540/tcp	uucpd
uucp	540/udp	uucpd
uucp-rlogin	541/tcp	uucp-rlogin
uucp-rlogin	541/udp	uucp-rlogin
commerce	542/tcp	commerce
commerce	542/udp	commerce
klogin	543/tcp	
klogin	543/udp	
kshell	544/tcp	krcmd
kshell	544/udp	krcmd
appleqtcsrvr	545/tcp	appleqtcsrvr
appleqtcsrvr	545/udp	appleqtcsrvr
dhcpv6-client	546/tcp	DHCPv6 Client
dhcpv6-client	546/udp	DHCPv6 Client
dhcpv6-server	547/tcp	DHCPv6 Server
dhcpv6-server	547/udp	DHCPv6 Server
afpovertcp	548/tcp	AFP over TCP
afpovertcp	548/udp	AFP over TCP
idfp	549/tcp	IDFP
idfp	549/udp	IDFP
new-rwho	550/tcp	new-who
new-rwho	550/udp	new-who
cybercash	551/tcp	cybercash
cybercash	551/udp	cybercash
deviceshare	552/tcp	deviceshare
deviceshare	552/udp	deviceshare

#unitary	126/udp	Unisys Unitary Login
locus-con	127/tcp	Locus Conn Server
locus-con	127/udp	Locus Conn Server
gss-xlicen	128/tcp	GSS X Verification
gss-xlicen	128/udp	GSS X Verification
pwdgen	129/tcp	Password Generator
pwdgen	129/udp	Password Generator
cisco-fna	130/tcp	cisco FNATIVE
cisco-fna	130/udp	cisco FNATIVE
cisco-tna	131/tcp	cisco TNATIVE
cisco-tna	131/udp	cisco TNATIVE
cisco-sys	132/tcp	cisco SYSMAINT
cisco-sys	132/udp	cisco SYSMAINT
statsrv	133/tcp	Statistics Service
statsrv	133/udp	Statistics Service
ingres-net	134/tcp	INGRES-NET Service
ingres-net	134/udp	INGRES-NET Service
epmap	135/tcp	DCE
epmap	135/udp	DCE
profile	136/tcp	PROFILE Naming Sys
profile	136/udp	PROFILE Naming Sys
netbios-ns	137/tcp	NETBIOS Name Serv
netbios-ns	137/udp	NETBIOS Name Serv
netbios-dgm	138/tcp	NETBIOS Data Serv
netbios-dgm	138/udp	NETBIOS Data Serv
netbios-ssn	139/tcp	NETBIOS Session Serv
netbios-ssn	139/udp	NETBIOS Session Serv
emfis-data	140/tcp	EMFIS Data Serv
emfis-data	140/udp	EMFIS Data Serv
emfis-cntl	141/tcp	EMFIS Control Serv
emfis-cntl	141/udp	EMFIS Control Serv
bl-idm	142/tcp	Britton-Lee IDM
bl-idm	142/udp	Britton-Lee IDM
imap	143/tcp	IMAP Protocol
imap	143/udp	IMAP Protocol
uma	144/tcp	UMA Protocol
uma	144/udp	UMA Protocol
uaac	145/tcp	UAAC Protocol
uaac	145/udp	UAAC Protocol
iso-tp0	146/tcp	ISO-IP0
iso-tp0	146/udp	ISO-IP0
iso-ip	147/tcp	ISO-IP
iso-ip	147/udp	ISO-IP
jargon	148/tcp	Jargon
jargon	148/udp	Jargon
aed-512	149/tcp	AED 512 Emulation
aed-512	149/udp	AED 512 Emulation
sql-net	150/tcp	SQL-NET
sql-net	150/udp	SQL-NET
hems	151/tcp	HEMS
hems	151/udp	HEMS
bftp	152/tcp	Background FTP
bftp	152/udp	Background FTP
sgmp	153/tcp	SGMP
sgmp	153/udp	SGMP
netsc-prod	154/tcp	NETSC
netsc-prod	154/udp	NETSC
netsc-dev	155/tcp	NETSC
netsc-dev	155/udp	NETSC
sqlsrv	156/tcp	SQL Service
sqlsrv	156/udp	SQL Service
knet-cmp	157/tcp	KNET/VM Protocol
knet-cmp	157/udp	KNET/VM Protocol
pcmail-srv	158/tcp	PCMail Server
pcmail-srv	158/udp	PCMail Server
nss-routing	159/tcp	NSS-Routing
nss-routing	159/udp	NSS-Routing
sgmp-traps	160/tcp	SGMP-TRAPS
sgmp-traps	160/udp	SGMP-TRAPS
snmp	161/tcp	SNMP
snmp	161/udp	SNMP
pirp	553/tcp	pirp
pirp	553/udp	pirp
rtsp	554/tcp	Real Time Stream
rtsp	554/udp	Real Time Strea
dsf	555/tcp	
dsf	555/udp	
remotefs	556/tcp	rfs server
remotefs	556/udp	rfs server
openvms-sysipc	557/tcp	openvms-sysipc
openvms-sysipc	557/udp	openvms-sysipc
sdnskmp	558/tcp	SDNSKMP
sdnskmp	558/udp	SDNSKMP
teedtap	559/tcp	TEEDTAP
teedtap	559/udp	TEEDTAP
rmonitor	560/tcp	rmonitord
rmonitor	560/udp	rmonitord
monitor	561/tcp	
monitor	561/udp	
chshell	562/tcp	chcmd
chshell	562/udp	chcmd
nntps	563/tcp	nntp over TLS/SSL
nntps	563/udp	nntp over TLS/SSL
9pfs	564/tcp	plan 9 file service
9pfs	564/udp	plan 9 file service
whoami	565/tcp	whoami
whoami	565/udp	whoami
streettalk	566/tcp	streettalk
streettalk	566/udp	streettalk
banyan-rpc	567/tcp	banyan-rpc
banyan-rpc	567/udp	banyan-rpc
ms-shuttle	568/tcp	microsoft shuttle
ms-shuttle	568/udp	microsoft shuttle
ms-rome	569/tcp	microsoft rome
ms-rome	569/udp	microsoft rome
meter	570/tcp	demon
meter	570/udp	demon
meter	571/tcp	udemon
meter	571/udp	udemon
sonar	572/tcp	sonar
sonar	572/udp	sonar
banyan-vip	573/tcp	banyan-vip
banyan-vip	573/udp	banyan-vip
ftp-agent	574/tcp	FTP Software Agent
ftp-agent	574/udp	FTP Software Agent
vemmi	575/tcp	VEMMI
vemmi	575/udp	VEMMI
ipcd	576/tcp	ipcd
ipcd	576/udp	ipcd
vnas	577/tcp	vnas
vnas	577/udp	vnas
ipdd	578/tcp	ipdd
ipdd	578/udp	ipdd
decbsrv	579/tcp	decbsrv
decbsrv	579/udp	decbsrv
sntp-heartbeat	580/tcp	SNTP HEARTBEAT
sntp-heartbeat	580/udp	SNTP HEARTBEAT
bdp	581/tcp	Bundle Discovery
bdp	581/udp	Bundle Discovery
scc-security	582/tcp	SCC Security
scc-security	582/udp	SCC Security
philips-vc	583/tcp	Philips Video
philips-vc	583/udp	Philips Video
keyserver	584/tcp	Key Server
keyserver	584/udp	Key Server
imap4-ssl	585/tcp	IMAP4+SSL
imap4-ssl	585/udp	IMAP4+SSL
password-chg	586/tcp	Password Change
password-chg	586/udp	Password Change
submission	587/tcp	Submission
submission	587/udp	Submission
cal	588/tcp	CAL

```
snmptrap      162/tcp   SNMPTRAP
snmptrap      162/udp   SNMPTRAP
cmip-man      163/tcp   CMIP/TCP Manager
cmip-man      163/udp   CMIP/TCP Manager
cmip-agent    164/tcp   CMIP/TCP Agent
smip-agent    164/udp   CMIP/TCP Agent
xns-courier   165/tcp   Xerox
xns-courier   165/udp   Xerox
s-net         166/tcp   Sirius Systems
s-net         166/udp   Sirius Systems
namp          167/tcp   NAMP
namp          167/udp   NAMP
rsvd          168/tcp   RSVD
rsvd          168/udp   RSVD
send          169/tcp   SEND
send          169/udp   SEND
print-srv     170/tcp   Network PostScript
print-srv     170/udp   Network PostScript
multiplex     171/tcp   Network Innovations
multiplex     171/udp   Network Innovations
cl/1          172/tcp   Network Innovations
cl/1          172/udp   Network Innovations
xyplex-mux    173/tcp   Xyplex
xyplex-mux    173/udp   Xyplex
mailq         174/tcp   MAILQ
mailq         174/udp   MAILQ
vmnet         175/tcp   VMNET
vmnet         175/udp   VMNET
genrad-mux    176/tcp   GENRAD-MUX
genrad-mux    176/udp   GENRAD-MUX
xdmcp         177/tcp   X Display Manager
xdmcp         177/udp   X Display Manager
nextstep      178/tcp   NextStep Win Server
nextstep      178/udp   NextStep Win Server
bgp           179/tcp   Border Gateway
bgp           179/udp   Border Gateway
ris           180/tcp   Intergraph
ris           180/udp   Intergraph
unify         181/tcp   Unify
unify         181/udp   Unify
audit         182/tcp   Unisys Audit SITP
audit         182/udp   Unisys Audit SITP
ocbinder      183/tcp   OCBinder
ocbinder      183/udp   OCBinder
ocserver      184/tcp   OCServer
ocserver      184/udp   OCServer
remote-kis    185/tcp   Remote-KIS
remote-kis    185/udp   Remote-KIS
kis           186/tcp   KIS Protocol
kis           186/udp   KIS Protocol
aci           187/tcp   App Communication
aci           187/udp   App Communication
mumps         188/tcp   Plus Five's MUMPS
mumps         188/udp   Plus Five's MUMPS
qft           189/tcp   Queued File Trans
qft           189/udp   Queued File Trans
gacp          190/tcp   Gateway Acc Control
gacp          190/udp   Gateway Acc Control
prospero      191/tcp   Prospero Directory
prospero      191/udp   Prospero Directory
osu-nms       192/tcp   Net Monitoring Sys
osu-nms       192/udp   Net Monitoring Sys
srmp          193/tcp   Spider Monitoring
srmp          193/udp   Spider Monitoring
irc           194/tcp   Internet Relay Chat
irc           194/udp   Internet Relay Chat
dn6-nlm-aud   195/tcp   DNSIX Module Audit
dn6-nlm-aud   195/udp   DNSIX Module Audit
dn6-smm-red   196/tcp   DNSIX Session Mgt
dn6-smm-red   196/udp   DNSIX Session Mgt
dls           197/tcp   Directory Location
cal           588/udp    CAL
eyelink       589/tcp    EyeLink
eyelink       589/udp    EyeLink
tns-cml       590/tcp    TNS CML
tns-cml       590/udp    TNS CML
http-alt      591/tcp    FileMaker
http-alt      591/udp    FileMaker
eudora-set    592/tcp    Eudora Set
eudora-set    592/udp    Eudora Set
http-rpc-epmap  593/tcp    HTTP RPC Ep Map
http-rpc-epmap  593/udp    HTTP RPC Ep Map
tpip          594/tcp    TPIP
tpip          594/udp    TPIP
cab-protocol    595/tcp    CAB Protocol
cab-protocol    595/udp    CAB Protocol
smsd          596/tcp    SMSD
smsd          596/udp    SMSD
ptcnameservice  597/tcp    PTC Name Service
ptcnameservice  597/udp    PTC Name Service
sco-websrvrmg3  598/tcp    SCO Web Server
sco-websrvrmg3  598/udp    SCO Web Server
acp           599/tcp    Aeolon Core Protocol
acp           599/udp    Aeolon Core Protocol
ipcserver     600/tcp    Sun IPC server
ipcserver     600/udp    Sun IPC server
#             601-605    Unassigned
urm           606/tcp    Cray Unified
urm           606/udp    Cray Unified
nqs           607/tcp    nqs
nqs           607/udp    nqs
sift-uft      608/tcp    sender Init/Unsolici
sift-uft      608/udp    Sender-Init/Unsolici
npmp-trap     609/tcp    npmp-trap
npmp-trap     609/udp    npmp-trap
npmp-local    610/tcp    npmp-local
npmp-local    610/udp    npmp-local
npmp-gui      611/tcp    npmp-gui
npmp-gui      611/udp    npmp-gui
hmmp-ind      612/tcp    HMMP Indication
hmmp-ind      612/udp    HMMP Indication
hmmp-op       613/tcp    HMMP Operation
hmmp-op       613/udp    HMMP Operation
sshell        614/tcp    SSLshell
sshell        614/udp    SSLshell
sco-inetmgr   615/tcp    Internet Config Man
sco-inetmgr   615/udp    Internet Config Man
sco-sysmgr    616/tcp    SCO System Admin
sco-sysmgr    616/udp    SCO System Admin
sco-dtmgr     617/tcp    SCO Desktop Admin
sco-dtmgr     617/udp    SCO Desktop Admin
dei-icda      618/tcp    DEI-ICDA
dei-icda      618/udp    DEI-ICDA
digital-evm   619/tcp    Digital EVM
digital-evm   619/udp    Digital EVM
sco-websrvrmgr  620/tcp    SCO WebServer
sco-websrvrmgr  620/udp    SCO WebServer
escp-ip       621/tcp    ESCP
escp-ip       621/udp    ESCP
collaborator    622/tcp    Collaborator
collaborator    622/udp    Collaborator
aux_bus_shunt   623/tcp    Aux Bus Shunt
aux_bus_shunt   623/udp    Aux Bus Shunt
cryptoadmin     624/tcp    Crypto Admin
cryptoadmin     624/udp    Crypto Admin
dec_dlm     625/tcp    DEC DLM
dec_dlm     625/udp    DEC DLM
asia        626/tcp    ASIA
asia        626/udp    ASIA
passgo-tivoli   627/tcp    PassGo Tivoli
passgo-tivoli   627/udp    PassGo Tivoli
qmqp        628/tcp    QMQP
```

```
dls             197/udp   Directory Location
dls-mon         198/tcp   Directory Location
dls-mon         198/udp   Directory Location
smux            199/tcp   SMUX
smux            199/udp   SMUX
src             200/tcp   IBM System Resource
src             200/udp   IBM System Resource
at-rtmp         201/tcp   AppleTalk Routing
at-rtmp         201/udp   AppleTalk Routing
at-nbp          202/tcp   AppleTalk Name
at-nbp          202/udp   AppleTalk Name
at-3            203/tcp   AppleTalk Unused
at-3            203/udp   AppleTalk Unused
at-echo         204/tcp   AppleTalk Echo
at-echo         204/udp   AppleTalk Echo
at-5            205/tcp   AppleTalk Unused
at-5            205/udp   AppleTalk Unused
at-zis          206/tcp   AppleTalk Zone
at-zis          206/udp   AppleTalk Zone
at-7            207/tcp   AppleTalk Unused
at-7            207/udp   AppleTalk Unused
at-8            208/tcp   AppleTalk Unused
at-8            208/udp   AppleTalk Unused
qmtp            209/tcp   Quick Mail Transfer
qmtp            209/udp   Quick Mail Transfer
z39.50          210/tcp   ANSI Z39.50
z39.50          210/udp   ANSI Z39.50
914c/g          211/tcp   Texas Instruments
914c/g          211/udp   Texas Instruments
anet            212/tcp   ATEXSSTR
anet            212/udp   ATEXSSTR
ipx             213/tcp   IPX
ipx             213/udp   IPX
vmpwscs         214/tcp   VM PWSCS
vmpwscs         214/udp   VM PWSCS
softpc          215/tcp   Insignia Solutions
softpc          215/udp   Insignia Solutions
CAIlic          216/tcp   Computer Associates
CAIlic          216/udp   Computer Associates
dbase           217/tcp   dBASE Unix
dbase           217/udp   dBASE Unix
mpp             218/tcp   Netix Message Post
mpp             218/udp   Netix Message Post
uarps           219/tcp   Unisys ARPs
uarps           219/udp   Unisys ARPs
imap3           220/tcp   IMAP v3
imap3           220/udp   IMAP v3
fln-spx         221/tcp   Berkeley rlogind
fln-spx         221/udp   Berkeley rlogind
rsh-spx         222/tcp   Berkeley rshd
rsh-spx         222/udp   Berkeley rshd
cdc             223/tcp   Certificate Distrib
cdc             223/udp   Certificate Distrib
masqdialer      224/tcp   masqdialer
masqdialer      224/udp   masqdialer
#               225-241   Reserved
direct          242/tcp   Direct
direct          242/udp   Direct
sur-meas        243/tcp   Survey Measurement
sur-meas        243/udp   Survey Measurement
inbusiness      244/tcp   inbusiness
inbusiness      244/udp   inbusiness
link            245/tcp   LINK
link            245/udp   LINK
dsp3270         246/tcp   Display Systems
dsp3270         246/udp   Display Systems
subntbcst_tftp  247/tcp    SUBNTBCST_TFTP
subntbcst_tftp  247/udp    SUBNTBCST_TFTP
bhfhs           248/tcp   bhfhs
bhfhs           248/udp   bhfhs
#               249-255   Reserved
qmqp            628/udp    QMQP
3com-amp3       629/tcp    3Com AMP3
3com-amp3       629/udp    3Com AMP3
rda             630/tcp    RDA
rda             630/udp    RDA
ipp             631/tcp    Internet Printing
ipp             631/udp    Internet Printing
bmpp            632/tcp    bmpp
bmpp            632/udp    bmpp
servstat        633/tcp    Service Status update
servstat        633/udp    Service Status update
ginad           634/tcp    ginad
ginad           634/udp    ginad
rlzdbase        635/tcp    RLZ DBase
rlzdbase        635/udp    RLZ DBase
ldaps           636/tcp    ldap protocol TLS/SSL
ldaps           636/udp    ldap protocol TLS/SSL
lanserver       637/tcp    lanserver
lanserver       637/udp    lanserver
mcns-sec        638/tcp    mcns-sec
mcns-sec        638/udp    mcns-sec
msdp            639/tcp    MSDP
msdp            639/udp    MSDP
entrust-sps     640/tcp    entrust-sps
entrust-sps     640/udp    entrust-sps
repcmd          641/tcp    repcmd
repcmd          641/udp    repcmd
esro-emsdp      642/tcp    ESRO-EMSDP V1.3
esro-emsdp      642/udp    ESRO-EMSDP V1.3
sanity          643/tcp    SANity
sanity          643/udp    SANity
dwr             644/tcp    dwr
dwr             644/udp    dwr
pssc            645/tcp    PSSC
pssc            645/udp    PSSC
ldp             646/tcp    LDP
ldp             646/udp    LDP
dhcp-failover   647/tcp    DHCP Failover
dhcp-failover   647/udp    DHCP Failover
rrp             648/tcp    Registry Registrar
rrp             648/udp    Registry Registrar
aminet          649/tcp    Aminet
aminet          649/udp    Aminet
obex            650/tcp    OBEX
obex            650/udp    OBEX
ieee-mms        651/tcp    IEEE MMS
ieee-mms        651/udp    IEEE MMS
udlr-dtcp       652/tcp    UDLR_DTCP
udlr-dtcp       652/udp    UDLR_DTCP
repscmd         653/tcp    RepCmd
repscmd         653/udp    RepCmd
aodv            654/tcp    AODV
aodv            654/udp    AODV
tinc            655/tcp    TINC
tinc            655/udp    TINC
spmp            656/tcp    SPMP
spmp            656/udp    SPMP
rmc             657/tcp    RMC
rmc             657/udp    RMC
tenfold         658/tcp    TenFold
tenfold         658/udp    TenFold
url-rendezvous  659/tcp    URL Rendezvous
url-rendezvous  659/udp    URL Rendezvous
mac-srvr-admin  660/tcp    MacOS Serv Admin
mac-srvr-admin  660/udp    MacOS Ser Admin
hap             661/tcp    HAP
hap             661/udp    HAP
pftp            662/tcp    PFTP
pftp            662/udp    PFTP
purenoise       663/tcp    PureNoise
purenoise       663/udp    PureNoise
```

```
rap           256/tcp   RAP
rap           256/udp   RAP
set           257/tcp   Secure Elect Trans
set           257/udp   Secure Elect Trans
yak-chat      258/tcp   Yak Personal Chat
yak-chat      258/udp   Yak Personal Chat
esro-gen      259/tcp   Efficient Short
esro-gen      259/udp   Efficient Short
openport      260/tcp   Openport
openport      260/udp   Openport
nsiiops       261/tcp   IIOP over TLS/SSL
nsiiops       261/udp   IIOP over TLS/SSL
arcisdms      262/tcp   Arcisdms
arcisdms      262/udp   Arcisdms
hdap          263/tcp   HDAP
hdap          263/udp   HDAP
bgmp          264/tcp   BGMP
bgmp          264/udp   BGMP
x-bone-ctl    265/tcp   X-Bone CTL
x-bone-ctl    265/udp   X-Bone CTL
sst           266/tcp   SCSI on ST
sst           266/udp   SCSI on ST
td-service    267/tcp   Tobit David Layer
td-service    267/udp   Tobit David Layer
td-replica    268/tcp   Tobit David Replica
td-replica    268/udp   Tobit David Replica
#             269-279   Unassigned
http-mgmt     280/tcp   http-mgmt
http-mgmt     280/udp   http-mgmt
personal-link 281/tcp   Personal Link
personal-link 281/udp   Personal Link
cableport-ax  282/tcp   Cable Port A/X
cableport-ax  282/udp   Cable Port A/X
rescap        283/tcp   rescap
rescap        283/udp   rescap
corerjd       284/tcp   corerjd
corerjd       284/udp   corerjd
#             285       Unassigned
fxp-1         286/tcp   FXP-1
fxp-1         286/udp   FXP-1
k-block       287/tcp   K-BLOCK
k-block       287/udp   K-BLOCK
#             288-307   Unassigned
novastorbakcup  308/tcp    Novastor Backup
novastorbakcup  308/udp    Novastor Backup
entrusttime   309/tcp   EntrustTime
entrusttime   309/udp   EntrustTime
bhmds         310/tcp   bhmds
bhmds         310/udp   bhmds
asip-webadmin 311/tcp   AppleShare WebAdmin
asip-webadmin 311/udp   AppleShare WebAdmin
vslmp         312/tcp   VSLMP
vslmp         312/udp   VSLMP
magenta-logic 313/tcp   Magenta Logic
magenta-logic 313/udp   Magenta Logic
opalis-robot  314/tcp   Opalis Robot
opalis-robot  314/udp   Opalis Robot
dpsi          315/tcp   DPSI
dpsi          315/udp   DPSI
decauth       316/tcp   decAuth
decauth       316/udp   decAuth
zannet        317/tcp   Zannet
zannet        317/udp   Zannet
pkix-timestamp  318/tcp    PKIX TimeStamp
pkix-timestamp  318/udp    PKIX TimeStamp
ptp-event     319/tcp   PTP Event
ptp-event     319/udp   PTP Event
ptp-general   320/tcp   PTP General
ptp-general   320/udp   PTP General
pip           321/tcp   PIP
pip           321/udp   PIP
secure-aux-bus  664/tcp    Secure Aux Bus
secure-aux-bus  664/udp    Secure Aux Bus
sun-dr        665/tcp    Sun DR
sun-dr        665/udp    Sun DR
mdqs          666/tcp
mdqs          666/udp
doom          666/tcp    doom Id Software
doom          666/udp    doom Id Software
disclose      667/tcp    SDR Technologies
disclose      667/udp    SDR Technologies
mecomm        668/tcp    MeComm
mecomm        668/udp    MeComm
meregister    669/tcp    MeRegister
meregister    669/udp    MeRegister
vacdsm-sws    670/tcp    VACDSM-SWS
vacdsm-sws    670/udp    VACDSM-SWS
vacdsm-app    671/tcp    VACDSM-APP
vacdsm-app    671/udp    VACDSM-APP
vpps-qua      672/tcp    VPPS-QUA
vpps-qua      672/udp    VPPS-QUA
cimplex       673/tcp    CIMPLEX
cimplex       673/udp    CIMPLEX
acap          674/tcp    ACAP
acap          674/udp    ACAP
dctp          675/tcp    DCTP
dctp          675/udp    DCTP
vpps-via      676/tcp    VPPS Via
vpps-via      676/udp    VPPS Via
vpp           677/tcp    Virtual Presence
vpp           677/udp    Virtual Presence
ggf-ncp       678/tcp    GNU NCP
ggf-ncp       678/udp    GNU NCP
mrm           679/tcp    MRM
mrm           679/udp    MRM
entrust-aaas    680/tcp    entrust-aaas
entrust-aaas    680/udp    entrust-aaas
entrust-aams    681/tcp    entrust-aams
entrust-aams    681/udp    entrust-aams
xfr           682/tcp    XFR
xfr           682/udp    XFR
corba-iiop    683/tcp    CORBA IIOP
corba-iiop    683/udp    CORBA IIOP
corba-iiop-ssl  684/tcp    CORBA IIOP SSL
corba-iiop-ssl  684/udp    CORBA IIOP SSL
mdc-portmapper  685/tcp    MDC Port Mapper
mdc-portmapper  685/udp    MDC Port Mapper
hcp-wismar    686/tcp    Hardware Control
hcp-wismar    686/udp    Hardware Control
asipregistry    687/tcp    asipregistry
asipregistry    687/udp    asipregistry
realm-rusd    688/tcp    REALM-RUSD
realm-rusd    688/udp    REALM-RUSD
nmap          689/tcp    NMAP
nmap          689/udp    NMAP
vatp          690/tcp    VATP
vatp          690/udp    VATP
msexch-routing  691/tcp    MS Exchange
msexch-routing  691/udp    MS Exchange
hyperwave-isp   692/tcp    Hyperwave-ISP
hyperwave-isp   692/udp    Hyperwave-ISP
connendp      693/tcp    connendp
connendp      693/udp    connendp
ha-cluster    694/tcp    ha-cluster
ha-cluster    694/udp    ha-cluster
ieee-mms-ssl    695/tcp    IEEE-MMS-SSL
ieee-mms-ssl    695/udp    IEEE-MMS-SSL
rushd         696/tcp    RUSHD
rushd         696/udp    RUSHD
uuidgen       697/tcp    UUIDGEN
uuidgen       697/udp    UUIDGEN
olsr          698/tcp    OLSR
```

```
rtsps           322/tcp   RTSPS
rtsps           322/udp   RTSPS
#               323-332   Unassigned
texar           333/tcp   Texar Security Port
texar           333/udp   Texar Security Port
#               334-343   Unassigned
pdap            344/tcp   Prospero Data Access
pdap            344/udp   Prospero Data Access
pawserv         345/tcp   Perf Analysis Bench
pawserv         345/udp   Perf Analysis Bench
zserv           346/tcp   Zebra server
zserv           346/udp   Zebra server
fatserv         347/tcp   Fatmen Server
fatserv         347/udp   Fatmen Server
csi-sgwp        348/tcp   Cabletron Management
csi-sgwp        348/udp   Cabletron Management
mftp            349/tcp   mftp
mftp            349/udp   mftp
matip-type-a    350/tcp   MATIP Type A
matip-type-a    350/udp   MATIP Type A
matip-type-b    351/tcp   MATIP Type B
matip-type-b    351/udp   MATIP Type B
bhoetty         351/tcp   bhoetty
bhoetty         351/udp   bhoetty
dtag-ste-sb     352/tcp   DTAG
dtag-ste-sb     352/udp   DTAG
bhoedap4        352/tcp   bhoedap4
bhoedap4        352/udp   bhoedap4
ndsauth         353/tcp   NDSAUTH
ndsauth         353/udp   NDSAUTH
bh611           354/tcp   bh611
bh611           354/udp   bh611
datex-asn       355/tcp   DATEX-ASN
datex-asn       355/udp   DATEX-ASN
cloanto-net-1   356/tcp   Cloanto Net 1
cloanto-net-1   356/udp   Cloanto Net 1
bhevent         357/tcp   bhevent
bhevent         357/udp   bhevent
shrinkwrap      358/tcp   Shrinkwrap
shrinkwrap      358/udp   Shrinkwrap
tenebris_nts    359/tcp   Tenebris Network
tenebris_nts    359/udp   Tenebris Network
scoi2odialog    360/tcp   scoi2odialog
scoi2odialog    360/udp   scoi2odialog
semantix        361/tcp   Semantix
semantix        361/udp   Semantix
srssend         362/tcp   SRS Send
srssend         362/udp   SRS Send
rsvp_tunnel     363/tcp   RSVP Tunnel
rsvp_tunnel     363/udp   RSVP Tunnel
aurora-cmgr     364/tcp   Aurora CMGR
aurora-cmgr     364/udp   Aurora CMGR
dtk             365/tcp   DTK
dtk             365/udp   DTK
odmr            366/tcp   ODMR
odmr            366/udp   ODMR
mortgageware    367/tcp   MortgageWare
mortgageware    367/udp   MortgageWare
qbikgdp         368/tcp   QbikGDP
qbikgdp         368/udp   QbikGDP
rpc2portmap     369/tcp   rpc2portmap
rpc2portmap     369/udp   rpc2portmap
codaauth2       370/tcp   codaauth2
codaauth2       370/udp   codaauth2
clearcase       371/tcp   Clearcase
clearcase       371/udp   Clearcase
ulistproc       372/tcp   ListProcessor
ulistproc       372/udp   ListProcessor
legent-1        373/tcp   Legent Corporation
legent-1        373/udp   Legent Corporation
legent-2        374/tcp   Legent Corporation
olsr            698/udp   OLSR
#               699-703   Unassigned
elcsd           704/tcp   errlog copy
elcsd           704/udp   errlog copy
agentx          705/tcp   AgentX
agentx          705/udp   AgentX
silc            706/tcp   SILC
silc            706/udp   SILC
borland-dsj     707/tcp   Borland DSJ
borland-dsj     707/udp   Borland DSJ
#               708       Unassigned
entrust-kmsh    709/tcp   Entrust Key
entrust-kmsh    709/udp   Entrust Key
entrust-ash     710/tcp   Entrust Admin
entrust-ash     710/udp   Entrust Admin
cisco-tdp       711/tcp   Cisco TDP
cisco-tdp       711/udp   Cisco TDP
#               712-728   Unassigned
netviewdm1      729/tcp   IBM NetView serv/cli
netviewdm1      729/udp   IBM NetView serv/cli
netviewdm2      730/tcp   IBM NetView send/tcp
netviewdm2      730/udp   IBM NetView send/tcp
netviewdm3      731/tcp   IBM NetView recv/tcp
netviewdm3      731/udp   IBM NetView recv/tcp
#               732-740   Unassigned
netgw           741/tcp   netGW
netgw           741/udp   netGW
netrcs          742/tcp   Net Rev. Cont. Sys.
netrcs          742/udp   Net Rev. Cont. Sys.
#               743       Unassigned
flexlm          744/tcp   Flexible License Man
flexlm          744/udp   Flexible License Man
#               745-746   Unassigned
fujitsu-dev     747/tcp   Fujitsu Dev Ctl
fujitsu-dev     747/udp   Fujitsu Dev Ctl
ris-cm          748/tcp   Russell Info Sci
ris-cm          748/udp   Russell Info Sci
kerberos-adm    749/tcp   kerberos admin
kerberos-adm    749/udp   kerberos admin
rfile           750/tcp
loadav          750/udp
kerberos-iv     750/udp   kerberos iv
pump            751/tcp
pump            751/udp
qrh             752/tcp
qrh             752/udp
rrh             753/tcp
rrh             753/udp
tell            754/tcp   send
tell            754/udp   send
#               755-756   Unassigned
nlogin          758/tcp
nlogin          758/udp
con             759/tcp
con             759/udp
ns              760/tcp
ns              760/udp
rxe             761/tcp
rxe             761/udp
quotad          762/tcp
quotad          762/udp
cycleserv       763/tcp
cycleserv       763/udp
omserv          764/tcp
omserv          764/udp
webster         765/tcp
webster         765/udp
#               766       Unassigned
phonebook       767/tcp   phone
phonebook       767/udp   phone
#               768       Unassigned
```

```
legent-2      374/udp   Legent Corporation
hassle        375/tcp   Hassle
hassle        375/udp   Hassle
nip           376/tcp   Amiga Envoy Network
nip           376/udp   Amiga Envoy Network
tnETOS        377/tcp   NEC Corporation
tnETOS        377/udp   NEC Corporation
dsETOS        378/tcp   NEC Corporation
dsETOS        378/udp   NEC Corporation
is99c         379/tcp   TIA/EIA/IS-99 client
is99c         379/udp   TIA/EIA/IS-99 client
is99s         380/tcp   TIA/EIA/IS-99 server
is99s         380/udp   TIA/EIA/IS-99 server
hp-collector 381/tcp   hp performance data
hp-collector 381/udp   hp performance data
hp-managed-node 382/tcp    hp managed node
hp-managed-node 382/udp    hp managed node
hp-alarm-mgr    383/tcp    hp alarm manager
hp-alarm-mgr    383/udp    hp alarm manager
arns          384/tcp   Remote Net Server
arns          384/udp   Remote Net Server
ibm-app       385/tcp   IBM Application
ibm-app       385/udp   IBM Application
asa           386/tcp   ASA Message Router
asa           386/udp   ASA Message Router
aurp          387/tcp   Appletalk
aurp          387/udp   Appletalk
unidata-ldm   388/tcp   Unidata LDM
unidata-ldm   388/udp   Unidata LDM
ldap          389/tcp   LDAP
ldap          389/udp   LDAP
uis           390/tcp   UIS
uis           390/udp   UIS
synotics-relay  391/tcp    SynOptics SNMP
synotics-relay  391/udp    SynOptics SNMP
synotics-broker 392/tcp    SynOptics Port
synotics-broker 392/udp    SynOptics Port
dis           393/tcp   Data Interpretation
dis           393/udp   Data Interpretation
embl-ndt      394/tcp   EMBL Nucleic Data
embl-ndt      394/udp   MBL Nucleic Data
netcp         395/tcp   NETscout Control
netcp         395/udp   NETscout Control
netware-ip    396/tcp   Novell Netware IP
netware-ip    396/udp   Novell Netware IP
mptn          397/tcp   Multi Trans. Net.
mptn          397/udp   Multi Trans. Net.
kryptolan     398/tcp   Kryptolan
kryptolan     398/udp   Kryptolan
iso-tsap-c2   399/tcp   ISO Transport Class
iso-tsap-c2   399/udp   ISO Transport Class
work-sol      400/tcp   Workstation Sol
work-sol      400/udp   Workstation Sol
ups           401/tcp   UPS
ups           401/udp   UPS
genie         402/tcp   Genie Protocol
genie         402/udp   Genie Protocol
decap         403/tcp   decap
decap         403/udp   decap
nced          404/tcp   nced
nced          404/udp   nced
ncld          405/tcp   ncld
ncld          405/udp   ncld
imsp          406/tcp   Interactive Mail Sup
imsp          406/udp   Interactive Mail Sup
timbuktu      407/tcp   Timbuktu
timbuktu      407/udp   Timbuktu
prm-sm        408/tcp   Prospero Resource
prm-sm        408/udp   Prospero Resource
prm-nm        409/tcp   Prospero Resource
prm-nm        409/udp   Prospero Resource
vid         769/tcp
vid         769/udp
cadlock     770/tcp
cadlock     770/udp
rtip        771/tcp
rtip        771/udp
cycleserv2  772/tcp
cycleserv2  772/udp
submit      773/tcp
notify      773/udp
rpasswd     774/tcp
acmaint_dbd     774/udp
entomb      775/tcp
acmaint_transd  775/udp
wpages      776/tcp
wpages      776/udp
multiling-http  777/tcp    Multiling HTTP
multiling-http  777/udp    Multiling HTTP
#           778-779    Unassigned
wpgs        780/tcp
wpgs        780/udp
#           781-785    Unassigned
concert     786/tcp    Concert
concert     786/udp    Concert
qsc         787/tcp    QSC
qsc         787/udp    QSC
#           788-799    Unassigned
mdbs_daemon     800/tcp
mdbs_daemon     800/udp
device      801/tcp
device      801/udp
#           802-809    Unassigned
fcp-udp     810/tcp    FCP
fcp-udp     810/udp    FCP Datagram
#           811-827    Unassigned
itm-mcell-s     828/tcp    itm-mcell-s
itm-mcell-s     828/udp    itm-mcell-s
pkix-3-ca-ra    829/tcp    PKIX-3 CA/RA
pkix-3-ca-ra    829/udp    PKIX-3 CA/RA
#           830-872    Unassigned
rsync       873/tcp    rsync
rsync       873/udp    rsync
#           874-885    Unassigned
iclcnet-locate  886/tcp    ICL coNETion
iclcnet-locate  886/udp    ICL coNETion
iclcnet_svinfo  887/tcp    ICL coNETion
iclcnet_svinfo  887/udp    ICL coNETion
accessbuilder   888/tcp    AccessBuilder
accessbuilder   888/udp    AccessBuilder
cddbp           888/tcp    CD Database
#           889-899    Unassigned
omginitialrefs  900/tcp    OMG Initial Refs
omginitialrefs  900/udp    OMG Initial Refs
smpnameres      901/tcp    SMPNAMERES
smpnameres      901/udp    SMPNAMERES
ideafarm-chat   902/tcp    IDEAFARM-CHAT
ideafarm-chat   902/udp    IDEAFARM-CHAT
ideafarm-catch  903/tcp    IDEAFARM-CATCH
ideafarm-catch  903/udp    IDEAFARM-CATCH
#           904-910    Unassigned
xact-backup     911/tcp    xact-backup
xact-backup     911/udp    xact-backup
#           912-988    Unassigned
ftps-data   989/tcp    ftp protocol TLS/SSL
ftps-data   989/udp    ftp protocol TLS/SSL
ftps        990/tcp    ftp protocol TLS/SSL
ftps        990/udp    ftp protocol TLS/SSL
nas         991/tcp    Netnews Admin System
nas         991/udp    Netnews Admin System
telnets     992/tcp    telnet TLS/SSL
telnets     992/udp    telnet TLS/SSL
```

```
decladebug   410/tcp   DECLadebug
decladebug   410/udp   DECLadebug
rmt          411/tcp   Remote MT Protocol
rmt          411/udp   Remote MT Protocol
synoptics-trap  412/tcp    Trap Convention
synoptics-trap  412/udp    Trap Convention
smsp         413/tcp   SMSP
smsp         413/udp   SMSP
infoseek     414/tcp   InfoSeek
infoseek     414/udp   InfoSeek
bnet         415/tcp   BNet
bnet         415/udp   BNet
silverplatter416/tcp   Silverplatter
silverplatter416/udp   Silverplatter
onmux        417/tcp   Onmux
onmux        417/udp   Onmux
hyper-g      418/tcp   Hyper-G
hyper-g      418/udp   Hyper-G
ariel1       419/tcp   Ariel
ariel1       419/udp   Ariel
smpte        420/tcp   SMPTE
smpte        420/udp   SMPTE
ariel2       421/tcp   Ariel
ariel2       421/udp   Ariel
ariel3       422/tcp   Ariel
ariel3       422/udp   Ariel
opc-job-start423/tcp   IBM Operations
opc-job-start423/udp   IBM Operations
imaps        993/tcp    imap4 TLS/SSL
imaps        993/udp    imap4 TLS/SSL
ircs         994/tcp    irc TLS/SSL
ircs         994/udp    irc TLS/SSL
pop3s        995/tcp    pop3 TLS/SSL
pop3s        995/udp    pop3 TLS/SSL
vsinet       996/tcp    vsinet
vsinet       996/udp    vsinet
maitrd       997/tcp
maitrd       997/udp
busboy       998/tcp
puparp       998/udp
garcon       999/tcp
applix       999/udp    Applix ac
puprouter    999/tcp
puprouter    999/udp
cadlock2     1000/tcp
cadlock2     1000/udp
#            1001-1009  Unassigned
#            1008/udp   Possibly used by Sun
surf         1010/tcp   surf
surf         1010/udp   surf
#            1011-1022  Reserved
#            1023/tcp   Reserved
             1023/udp   Reserved
```

Registered / Dynamic and/or Private Ports:

Below is the list of registered as well as Dynamic and/or Private Ports. The Registered Ports are those from 1024 through 49151 and the Dynamic and/or Private Ports are those from 49152 through 65535.

Keyword	Decimal	Description
#	1024/tcp	Reserved
#	1024/udp	Reserved
blackjack	1025/tcp	network blackjack
blackjack	1025/udp	network blackjack
#	1026-1029	Unassigned
iad1	1030/tcp	BBN IAD
iad1	1030/udp	BBN IAD
iad2	1031/tcp	BBN IAD
iad2	1031/udp	BBN IAD
iad3	1032/tcp	BBN IAD
iad3	1032/udp	BBN IAD
#	1033-1046	Unassigned
neod1	1047/tcp	Sun's NEO Object
neod1	1047/udp	Sun's NEO Object
neod2	1048/tcp	Sun's NEO Object
neod2	1048/udp	Sun's NEO Object
td-postman	1049/tcp	Tobit David Postman
td-postman	1049/udp	Tobit David Postman
cma	1050/tcp	CORBA Manag Agent
cma	1050/udp	CORBA Manag Agent
optima-vnet	1051/tcp	Optima VNET
optima-vnet	1051/udp	Optima VNET
ddt	1052/tcp	Dynamic DNS Tools
ddt	1052/udp	Dynamic DNS Tools
remote-as	1053/tcp	Remote Assistant (RA)
remote-as	1053/udp	Remote Assistant (RA)
brvread	1054/tcp	BRVREAD
brvread	1054/udp	BRVREAD
ansyslmd	1055/tcp	ANSYS-License Manager
ansyslmd	1055/udp	ANSYS-License Manager
vfo	1056/tcp	VFO
vfo	1056/udp	VFO
startron	1057/tcp	STARTRON
startron	1057/udp	STARTRON
nim	1058/tcp	nim
nim	1058/udp	nim
nimreg	1059/tcp	nimreg
nimreg	1059/udp	nimreg
polestar	1060/tcp	POLESTAR
polestar	1060/udp	POLESTAR
kiosk	1061/tcp	KIOSK
kiosk	1061/udp	KIOSK
veracity	1062/tcp	Veracity
veracity	1062/udp	Veracity
kyoceranetdev	1063/tcp	KyoceraNetDev
kyoceranetdev	1063/udp	KyoceraNetDev
jstel	1064/tcp	JSTEL
jstel	1064/udp	JSTE
syscomlan	1065/tcp	SYSCOMLAN
syscomlan	1065/udp	SYSCOMLAN
fpo-fns	1066/tcp	FPO-FNS
fpo-fns	1066/udp	FPO-FNS
instl_boots	1067/tcp	Bootstrap Proto.
instl_boots	1067/udp	Bootstrap Proto.
instl_bootc	1068/tcp	Bootstrap Proto.
instl_bootc	1068/udp	Bootstrap Proto.
cognex-insight	1069/tcp	COGNEX-INSIGHT
cognex-insight	1069/udp	COGNEX-INSIGHT
gmrupdateserv	1070/tcp	GMRUpdateSERV
gmrupdateserv	1070/udp	GMRUpdateSERV
bsquare-voip	1071/tcp	BSQUARE-VOIP
bsquare-voip	1071/udp	BSQUARE-VOIP

Keyword	Decimal	Description
alarm-clock-s	2667/tcp	Alarm Clock Serv
alarm-clock-s	2667/udp	Alarm Clock Serv
alarm-clock-c	2668/tcp	Alarm Clock Clt
alarm-clock-c	2668/udp	Alarm Clock Clt
toad	2669/tcp	TOAD
toad	2669/udp	TOAD
tve-announce	2670/tcp	TVE Announce
tve-announce	2670/udp	TVE Announce
newlixreg	2671/tcp	newlixreg
newlixreg	2671/udp	newlixreg
nhserver	2672/tcp	nhserver
nhserver	2672/udp	nhserver
firstcall42	2673/tcp	First Call 42
firstcall42	2673/udp	First Call 42
ewnn	2674/tcp	ewnn
ewnn	2674/udp	ewnn
ttc-etap	2675/tcp	TTC ETAP
ttc-etap	2675/udp	TTC ETAP
simslink	2676/tcp	SIMSLink
simslink	2676/udp	SIMSLink
gadgetgate1way	2677/tcp	Gadget Gate1 Way
gadgetgate1way	2677/udp	Gadget Gate1 Way
gadgetgate2way	2678/tcp	Gadget Gate2 Way
gadgetgate2way	2678/udp	Gadget Gate2 Way
syncserverssl	2679/tcp	Sync Server SSL
syncserverssl	2679/udp	Sync Server SSL
pxc-sapxom	2680/tcp	pxc-sapxom
pxc-sapxom	2680/udp	pxc-sapxom
mpnjsomb	2681/tcp	mpnjsomb
mpnjsomb	2681/udp	mpnjsomb
srsp	2682/tcp	SRSP
srsp	2682/udp	SRSP
ncdloadbalance	2683/tcp	NCDLoadBalance
ncdloadbalance	2683/udp	NCDLoadBalance
mpnjsosv	2684/tcp	mpnjsosv
mpnjsosv	2684/udp	mpnjsosv
mpnjsocl	2685/tcp	mpnjsocl
mpnjsocl	2685/udp	mpnjsocl
mpnjsomg	2686/tcp	mpnjsomg
mpnjsomg	2686/udp	mpnjsomg
pq-lic-mgmt	2687/tcp	pq-lic-mgmt
pq-lic-mgmt	2687/udp	pq-lic-mgmt
md-cg-http	2688/tcp	md-cf-http
md-cg-http	2688/udp	md-cf-http
fastlynx	2689/tcp	FastLynx
fastlynx	2689/udp	FastLynx
hp-nnm-data	2690/tcp	HP NNM Embedded
hp-nnm-data	2690/udp	HP NNM Embedded
itinternet	2691/tcp	IT Internet
itinternet	2691/udp	IT Internet
admins-lms	2692/tcp	Admins LMS
admins-lms	2692/udp	Admins LMS
belarc-http	2693/tcp	belarc-http
belarc-http	2693/udp	belarc-http
pwrsevent	2694/tcp	pwrsevent
pwrsevent	2694/udp	pwrsevent
vspread	2695/tcp	VSPREAD
vspread	2695/udp	VSPREAD
unifyadmin	2696/tcp	Unify Admin
unifyadmin	2696/udp	Unify Admin
oce-snmp-trap	2697/tcp	Oce SNMP Trap
oce-snmp-trap	2697/udp	Oce SNMP Trap

cardax	1072/tcp	CARDAX
cardax	1072/udp	CARDA
bridgecontrol	1073/tcp	BridgeControl
bridgecontrol	1073/udp	BridgeContro
fastechnologlm	1074/tcp	FASTechnologies
fastechnologlm	1074/udp	FASTechnologie
rdrmshc	1075/tcp	RDRMSHC
rdrmshc	1075/udp	RDRMSHC
dab-sti-c	1076/tcp	DAB STI-C
dab-sti-c	1076/udp	DAB STI-C
imgames	1077/tcp	IMGames
imgames	1077/udp	IMGames
emanagecstp	1078/tcp	eManageCstp
emanagecstp	1078/udp	eManageCst
asprovatalk	1079/tcp	ASPROVATalk
asprovatalk	1079/udp	ASPROVATalk
socks	1080/tcp	Socks
socks	1080/udp	Socks
amt-esd-prot	1082/tcp	AMT-ESD-PROT
amt-esd-prot	1082/udp	AMT-ESD-PROT
ansoft-lm-1	1083/tcp	Anasoft
ansoft-lm-1	1083/udp	Anasoft
ansoft-lm-2	1084/tcp	Anasoft
ansoft-lm-2	1084/udp	Anasoft
webobjects	1085/tcp	Web Objects
webobjects	1085/udp	Web Objects
cplscrambler-lg	1086/tcp	CPL Scramble
cplscrambler-lg	1086/udp	CPL Scrambler
cplscrambler-in	1087/tcp	CPL Scrambler
cplscrambler-in	1087/udp	CPL Scramble
cplscrambler-al	1088/tcp	CPL Scrambler
cplscrambler-al	1088/udp	CPL Scramble
ff-annunc	1089/tcp	FF Annunciation
ff-annunc	1089/udp	FF Annunciation
ff-fms	1090/tcp	FF Fieldbus
ff-fms	1090/udp	FF Fieldbus
ff-sm	1091/tcp	FF System Manag
ff-sm	1091/udp	FF System Manag
obrpd	1092/tcp	OBRPD
obrpd	1092/udp	OBRPD
proofd	1093/tcp	PROOFD
proofd	1093/udp	PROOFD
rootd	1094/tcp	ROOTD
rootd	1094/udp	ROOTD
nicelink	1095/tcp	NICELink
nicelink	1095/udp	NICELink
cnrprotocol	1096/tcp	Common Name Resl
cnrprotocol	1096/udp	Common Name Resl
sunclustermgr	1097/tcp	Sun Cluster Man
sunclustermgr	1097/udp	Sun Cluster Man
rmiactivation	1098/tcp	RMI Activation
rmiactivation	1098/udp	RMI Activation
rmiregistry	1099/tcp	RMI Registry
rmiregistry	1099/udp	RMI Registry
mctp	1100/tcp	MCTP
mctp	1100/udp	MCTP
pt2-discover	1101/tcp	PT2-DISCOVER
pt2-discover	1101/udp	PT2-DISCOVER
adobeserver-1	1102/tcp	ADOBE SERVER 1
adobeserver-1	1102/udp	ADOBE SERVER 1
adobeserver-2	1103/tcp	ADOBE SERVER 2
adobeserver-2	1103/udp	ADOBE SERVER 2
xrl	1104/tcp	XRL
xrl	1104/udp	XRL
ftranhc	1105/tcp	FTRANHC
ftranhc	1105/udp	FTRANHC
isoipsigport-1	1106/tcp	ISOIPSIGPORT-1
isoipsigport-1	1106/udp	ISOIPSIGPORT-1
isoipsigport-2	1107/tcp	ISOIPSIGPORT-2
isoipsigport-2	1107/udp	ISOIPSIGPORT-2
ratio-adp	1108/tcp	ratio-adp
mck-ivpip	2698/tcp	MCK-IVPIP
mck-ivpip	2698/udp	MCK-IVPIP
csoft-plusclnt	2699/tcp	Csoft Plus Clt
csoft-plusclnt	2699/udp	Csoft Plus Clt
tqdata	2700/tcp	tqdata
tqdata	2700/udp	tqdata
sms-rcinfo	2701/tcp	SMS RCINFO
sms-rcinfo	2701/udp	SMS RCINFO
sms-xfer	2702/tcp	SMS XFER
sms-xfer	2702/udp	SMS XFER
sms-chat	2703/tcp	SMS CHAT
sms-chat	2703/udp	SMS CHAT
sms-remctrl	2704/tcp	SMS REMCTRL
sms-remctrl	2704/udp	SMS REMCTRL
sds-admin	2705/tcp	SDS Admin
sds-admin	2705/udp	SDS Admin
ncdmirroring	2706/tcp	NCD Mirroring
ncdmirroring	2706/udp	NCD Mirroring
emcsymapiport	2707/tcp	EMCSYMAPIPORT
emcsymapiport	2707/udp	EMCSYMAPIPORT
banyan-net	2708/tcp	Banyan-Net
banyan-net	2708/udp	Banyan-Net
supermon	2709/tcp	Supermon
supermon	2709/udp	Supermon
sso-service	2710/tcp	SSO Service
sso-service	2710/udp	SSO Service
sso-control	2711/tcp	SSO Control
sso-control	2711/udp	SSO Control
aocp	2712/tcp	Axapta Object
aocp	2712/udp	Axapta Object
raven1	2713/tcp	Raven1
raven1	2713/udp	Raven1
raven2	2714/tcp	Raven2
raven2	2714/udp	Raven2
hpstgmgr2	2715/tcp	HPSTGMGR2
hpstgmgr2	2715/udp	HPSTGMGR2
inova-ip-disco	2716/tcp	Inova IP Disco
inova-ip-disco	2716/udp	Inova IP Disco
pn-requester	2717/tcp	PN REQUESTER
pn-requester	2717/udp	PN REQUESTER
pn-requester2	2718/tcp	PN REQUESTER 2
pn-requester2	2718/udp	PN REQUESTER 2
scan-change	2719/tcp	Scan & Change
scan-change	2719/udp	Scan & Change
wkars	2720/tcp	wkars
wkars	2720/udp	wkars
smart-diagnose	2721/tcp	Smart Diagnose
smart-diagnose	2721/udp	Smart Diagnose
proactivesrvr	2722/tcp	Proactive Server
proactivesrvr	2722/udp	Proactive Server
watchdognt	2723/tcp	WatchDog NT
watchdognt	2723/udp	WatchDog NT
qotps	2724/tcp	qotps
qotps	2724/udp	qotps
msolap-ptp2	2725/tcp	MSOLAP PTP2
msolap-ptp2	2725/udp	MSOLAP PTP2
tams	2726/tcp	TAMS
tams	2726/udp	TAMS
mgcp-callagent	2727/tcp	Media Gateway
mgcp-callagent	2727/udp	Media Gateway
sqdr	2728/tcp	SQDR
sqdr	2728/udp	SQDR
tcim-control	2729/tcp	TCIM Control
tcim-control	2729/udp	TCIM Control
nec-raidplus	2730/tcp	NEC RaidPlus
nec-raidplus	2730/udp	NEC RaidPlus
netdragon-msngr	2731/tcp	NetDragon Mes
netdragon-msngr	2731/udp	NetDragon Mes
g5m	2732/tcp	G5M
g5m	2732/udp	G5M
signet-ctf	2733/tcp	Signet CTF

ratio-adp	1108/udp	ratio-adp
#	1109	Unassigned
nfsd-status	1110/tcp	Cluster status
nfsd-keepalive	1110/udp	Client status
lmsocialserver	1111/tcp	LM Social Server
lmsocialserver	1111/udp	LM Social Server
icp	1112/tcp	Intelligent Com
icp	1112/udp	Intelligent Com
#	1113	Unassigned
mini-sql	1114/tcp	Mini SQL
mini-sql	1114/udp	Mini SQL
ardus-trns	1115/tcp	ARDUS Transfer
ardus-trns	1115/udp	ARDUS Transfer
ardus-cntl	1116/tcp	ARDUS Control
ardus-cntl	1116/udp	ARDUS Control
ardus-mtrns	1117/tcp	ARDUS Multicast
ardus-mtrns	1117/udp	ARDUS Multicast
#	1118-1122	Unassigned
murray	1123/tcp	Murray
murray	1123/udp	Murray
#	1124-1154	Unassigned
nfa	1155/tcp	Network File Acs
nfa	1155/udp	Network File Acs
#	1156-1160	Unassigned
health-polling	1161/tcp	Health Polling
health-polling	1161/udp	Health Polling
health-trap	1162/tcp	Health Trap
health-trap	1162/udp	Health Trap
#	1163-1168	Unassigned
tripwire	1169/tcp	TRIPWIRE
tripwire	1169/udp	TRIPWIRE
#	1170-1179	Unassigned
mc-client	1180/tcp	Millicent Proxy
mc-client	1180/udp	Millicent Proxy
#	1181-1187	Unassigned
hp-webadmin	1188/tcp	HP Web Admin
hp-webadmin	1188/udp	HP Web Admin
#	1189-1199	Unassigned
scol	1200/tcp	SCOL
scol	1200/udp	SCOL
nucleus-sand	1201/tcp	Nucleus Sand
nucleus-sand	1201/udp	Nucleus Sand
caiccipc	1202/tcp	caiccipc
caiccipc	1202/udp	caiccipc
ssslic-mgr	1203/tcp	License Valid
ssslic-mgr	1203/udp	License Valid
ssslog-mgr	1204/tcp	Log Request
ssslog-mgr	1204/udp	Log Request
accord-mgc	1205/tcp	Accord-MGC
accord-mgc	1205/udp	Accord-MGC
anthony-data	1206/tcp	Anthony Data
anthony-data	1206/udp	Anthony Data
metasage	1207/tcp	MetaSage
metasage	1207/udp	MetaSage
seagull-ais	1208/tcp	SEAGULL AIS
seagull-ais	1208/udp	SEAGULL AIS
ipcd3	1209/tcp	IPCD3
ipcd3	1209/udp	IPCD3
eoss	1210/tcp	EOSS
eoss	1210/udp	EOSS
groove-dpp	1211/tcp	Groove DPP
groove-dpp	1211/udp	Groove DPP
lupa	1212/tcp	lupa
lupa	1212/udp	lupa
mpc-lifenet	1213/tcp	MPC LIFENET
mpc-lifenet	1213/udp	MPC LIFENET
kazaa	1214/tcp	KAZAA
kazaa	1214/udp	KAZAA
scanstat-1	1215/tcp	scanSTAT 1.0
scanstat-1	1215/udp	scanSTAT 1.0
etebac5	1216/tcp	ETEBAC 5
signet-ctf	2733/udp	Signet CTF
ccs-software	2734/tcp	CCS Software
ccs-software	2734/udp	CCS Software
monitorconsole	2735/tcp	Monitor Console
monitorconsole	2735/udp	Monitor Console
radwiz-nms-srv	2736/tcp	RADWIZ NMS SRV
radwiz-nms-srv	2736/udp	RADWIZ NMS SRV
srp-feedback	2737/tcp	SRP Feedback
srp-feedback	2737/udp	SRP Feedback
ndl-tcp-ois-gw	2738/tcp	NDL TCP-OSI Gty
ndl-tcp-ois-gw	2738/udp	NDL TCP-OSI Gty
tn-timing	2739/tcp	TN Timing
tn-timing	2739/udp	TN Timing
alarm	2740/tcp	Alarm
alarm	2740/udp	Alarm
tsb	2741/tcp	TSB
tsb	2741/udp	TSB
tsb2	2742/tcp	TSB2
tsb2	2742/udp	TSB2
murx	2743/tcp	murx
murx	2743/udp	murx
honyaku	2744/tcp	honyaku
honyaku	2744/udp	honyaku
urbisnet	2745/tcp	URBISNET
urbisnet	2745/udp	URBISNET
cpudpencap	2746/tcp	CPUDPENCAP
cpudpencap	2746/udp	CPUDPENCAP
fjippol-swrly	2747/tcp	
fjippol-swrly	2747/udp	
fjippol-polsvr	2748/tcp	
fjippol-polsvr	2748/udp	
fjippol-cnsl	2749/tcp	
fjippol-cnsl	2749/udp	
fjippol-port1	2750/tcp	
fjippol-port1	2750/udp	
fjippol-port2	2751/tcp	
fjippol-port2	2751/udp	
rsisysaccess	2752/tcp	RSISYS ACCESS
rsisysaccess	2752/udp	RSISYS ACCESS
de-spot	2753/tcp	de-spot
de-spot	2753/udp	de-spot
apollo-cc	2754/tcp	APOLLO CC
apollo-cc	2754/udp	APOLLO CC
expresspay	2755/tcp	Express Pay
expresspay	2755/udp	Express Pay
simplement-tie	2756/tcp	simplement-tie
simplement-tie	2756/udp	simplement-tie
cnrp	2757/tcp	CNRP
cnrp	2757/udp	CNRP
apollo-status	2758/tcp	APOLLO Status
apollo-status	2758/udp	APOLLO Status
apollo-gms	2759/tcp	APOLLO GMS
apollo-gms	2759/udp	APOLLO GMS
sabams	2760/tcp	Saba MS
sabams	2760/udp	Saba MS
dicom-iscl	2761/tcp	DICOM ISCL
dicom-iscl	2761/udp	DICOM ISCL
dicom-tls	2762/tcp	DICOM TLS
dicom-tls	2762/udp	DICOM TLS
desktop-dna	2763/tcp	Desktop DNA
desktop-dna	2763/udp	Desktop DNA
data-insurance	2764/tcp	Data Insurance
data-insurance	2764/udp	Data Insurance
qip-audup	2765/tcp	qip-audup
qip-audup	2765/udp	qip-audup
compaq-scp	2766/tcp	Compaq SCP
compaq-scp	2766/udp	Compaq SCP
uadtc	2767/tcp	UADTC
uadtc	2767/udp	UADTC
uacs	2768/tcp	UACS
uacs	2768/udp	UACS

etebac5	1216/udp	ETEBAC 5
hpss-ndapi	1217/tcp	HPSS-NDAPI
hpss-ndapi	1217/udp	HPSS-NDAPI
aeroflight-ads	1218/tcp	AeroFlight-ADs
aeroflight-ads	1218/udp	AeroFlight-ADs
aeroflight-ret	1219/tcp	AeroFlight-Ret
aeroflight-ret	1219/udp	AeroFlight-Ret
qt-serveradmin	1220/tcp	QT SERVER ADMIN
qt-serveradmin	1220/udp	QT SERVER ADMIN
sweetware-apps	1221/tcp	SweetWARE Apps
sweetware-apps	1221/udp	SweetWARE Apps
nerv	1222/tcp	SNI R&D network
nerv	1222/udp	SNI R&D network
tgp	1223/tcp	TGP
tgp	1223/udp	TGP
vpnz	1224/tcp	VPNz
vpnz	1224/udp	VPNz
slinkysearch	1225/tcp	SLINKYSEARCH
slinkysearch	1225/udp	SLINKYSEARCH
stgxfws	1226/tcp	STGXFWS
stgxfws	1226/udp	STGXFWS
dns2go	1227/tcp	DNS2Go
dns2go	1227/udp	DNS2Go
florence	1228/tcp	FLORENCE
florence	1228/udp	FLORENCE
novell-zfs	1229/tcp	Novell ZFS
novell-zfs	1229/udp	Novell ZFS
periscope	1230/tcp	Periscope
periscope	1230/udp	Periscope
menandmice-lpm	1231/tcp	menandmice-lpm
menandmice-lpm	1231/udp	menandmice-lpm
mtrgtrans	1232/tcp	mtrgtrans
mtrgtrans	1232/udp	mtrgtrans
univ-appserver	1233/tcp	Universal App
univ-appserver	1233/udp	Universal App
search-agent	1234/tcp	Infoseek Search
search-agent	1234/udp	Infoseek Search
#	1235-1238	Unassigned
nmsd	1239/tcp	NMSD
nmsd	1239/udp	NMSD
#	1240-1247	Unassigned
hermes	1248/tcp	
hermes	1248/udp	
#	1249-1299	Unassigned
h323hostcallsc	1300/tcp	H323 Host Call
h323hostcallsc	1300/udp	H323 Host Call
#	1301-1309	Unassigned
husky	1310/tcp	Husky
husky	1310/udp	Husky
rxmon	1311/tcp	RxMon
rxmon	1311/udp	RxMon
sti-envision	1312/tcp	STI Envision
sti-envision	1312/udp	STI Envision
bmc_patroldb	1313/tcp	BMC_PATROLDB
bmc-patroldb	1313/udp	BMC_PATROLDB
pdps	1314/tcp	Photoscript
pdps	1314/udp	Photoscript
#	1315-1318	Unassigned
panja-icsp	1319/tcp	Panja-ICSP
panja-icsp	1319/udp	Panja-ICSP
panja-axbnet	1320/tcp	Panja-AXBNET
panja-axbnet	1320/udp	Panja-AXBNET
pip	1321/tcp	PIP
pip	1321/udp	PIP
#	1322-1334	Unassigned
digital-notary	1335/tcp	Digital Notary
digital-notary	1335/udp	Digital Notary
#	1336-1344	Unassigned
vpjp	1345/tcp	VPJP
vpjp	1345/udp	VPJP
alta-ana-lm	1346/tcp	Alta Analytics
singlept-mvs	2769/tcp	Single Point MVS
singlept-mvs	2769/udp	Single Point MV
veronica	2770/tcp	Veronica
veronica	2770/udp	Veronica
vergencecm	2771/tcp	Vergence CM
vergencecm	2771/udp	Vergence C
auris	2772/tcp	auris
auris	2772/udp	auris
pcbakcup1	2773/tcp	PC Backup
pcbakcup1	2773/udp	PC Backup
pcbakcup2	2774/tcp	PC Backup
pcbakcup2	2774/udp	PC Backup
smpp	2775/tcp	SMMP
smpp	2775/udp	SMMP
ridgeway1	2776/tcp	Ridgeway
ridgeway1	2776/udp	Ridgeway
ridgeway2	2777/tcp	Ridgeway
ridgeway2	2777/udp	Ridgeway
gwen-sonya	2778/tcp	Gwen-Sonya
gwen-sonya	2778/udp	Gwen-Sonya
lbc-sync	2779/tcp	LBC Sync
lbc-sync	2779/udp	LBC Sync
lbc-control	2780/tcp	LBC Control
lbc-control	2780/udp	LBC Control
whosells	2781/tcp	whosells
whosells	2781/udp	whosells
everydayrc	2782/tcp	everydayrc
everydayrc	2782/udp	everydayrc
aises	2783/tcp	AISES
aises	2783/udp	AISES
www-dev	2784/tcp	world wide web
www-dev	2784/udp	world wide web
aic-np	2785/tcp	aic-np
aic-np	2785/udp	aic-np
aic-oncrpc	2786/tcp	aic-oncrpc
aic-oncrpc	2786/udp	aic-oncrpc
piccolo	2787/tcp	piccolo
piccolo	2787/udp	piccolo
fryeserv	2788/tcp	NetWare Loadable
fryeserv	2788/udp	NetWare Loadable
media-agent	2789/tcp	Media Agent
media-agent	2789/udp	Media Agent
plgproxy	2790/tcp	PLG Proxy
plgproxy	2790/udp	PLG Proxy
mtport-regist	2791/tcp	MT Port Regist
mtport-regist	2791/udp	MT Port Regist
f5-globalsite	2792/tcp	f5-globalsite
f5-globalsite	2792/udp	f5-globalsite
initlsmsad	2793/tcp	initlsmsad
initlsmsad	2793/udp	initlsmsad
aaftp	2794/tcp	aaftp
aaftp	2794/udp	aaftp
livestats	2795/tcp	LiveStats
livestats	2795/udp	LiveStats
ac-tech	2796/tcp	ac-tech
ac-tech	2796/udp	ac-tech
esp-encap	2797/tcp	esp-encap
esp-encap	2797/udp	esp-encap
tmesis-upshot	2798/tcp	TMESIS-UPShot
tmesis-upshot	2798/udp	TMESIS-UPShot
icon-discover	2799/tcp	ICON Discover
icon-discover	2799/udp	ICON Discover
acc-raid	2800/tcp	ACC RAID
acc-raid	2800/udp	ACC RAID
igcp	2801/tcp	IGCP
igcp	2801/udp	IGCP
veritas-tcp1	2802/tcp	Veritas TCP1
veritas-udp1	2802/udp	Veritas UDP1
btprjctrl	2803/tcp	btprjctrl
btprjctrl	2803/udp	btprjctrl
telexis-vtu	2804/tcp	Telexis VTU

alta-ana-lm	1346/udp	Alta Analytics
bbn-mmc	1347/tcp	multi media conf
bbn-mmc	1347/udp	multi media conf
bbn-mmx	1348/tcp	multi media conf
bbn-mmx	1348/udp	multi media conf
sbook	1349/tcp	Registration Net
sbook	1349/udp	Registration Net
editbench	1350/tcp	Registration Net
editbench	1350/udp	Registration Net
equationbuilder	1351/tcp	Digital Works
equationbuilder	1351/udp	Digital Works
lotusnote	1352/tcp	Lotus Note
lotusnote	1352/udp	Lotus Note
relief	1353/tcp	Relief Consult
relief	1353/udp	Relief Consult
rightbrain	1354/tcp	RightBrain Soft
rightbrain	1354/udp	RightBrain Soft
intuitive-edge	1355/tcp	Intuitive Edge
intuitive-edge	1355/udp	Intuitive Edge
cuillamartin	1356/tcp	CuillaMartin
cuillamartin	1356/udp	CuillaMartin
pegboard	1357/tcp	Elect PegBoard
pegboard	1357/udp	Elect PegBoard
connlcli	1358/tcp	CONNLCLI
connlcli	1358/udp	CONNLCLI
ftsrv	1359/tcp	FTSRV
ftsrv	1359/udp	FTSRV
mimer	1360/tcp	MIMER
mimer	1360/udp	MIMER
linx	1361/tcp	LinX
linx	1361/udp	LinX
timeflies	1362/tcp	TimeFlies
timeflies	1362/udp	TimeFlies
ndm-requester	1363/tcp	DataMover Req
ndm-requester	1363/udp	DataMover Req
ndm-server	1364/tcp	DataMover Server
ndm-server	1364/udp	DataMover Server
adapt-sna	1365/tcp	Software Ass
adapt-sna	1365/udp	Software Ass
netware-csp	1366/tcp	Novell NetWare
netware-csp	1366/udp	Novell NetWare
dcs	1367/tcp	DCS
dcs	1367/udp	DCS
screencast	1368/tcp	ScreenCast
screencast	1368/udp	ScreenCast
gv-us	1369/tcp	GV to Unix Shell
gv-us	1369/udp	GV to Unix Shell
us-gv	1370/tcp	Unix Shell to GV
us-gv	1370/udp	Unix Shell to GV
fc-cli	1371/tcp	Fujitsu Config
fc-cli	1371/udp	Fujitsu Config
fc-ser	1372/tcp	Fujitsu Config
fc-ser	1372/udp	Fujitsu Config
chromagrafx	1373/tcp	Chromagrafx
chromagrafx	1373/udp	Chromagrafx
molly	1374/tcp	EPI Software Sys
molly	1374/udp	EPI Software Sys
bytex	1375/tcp	Bytex
bytex	1375/udp	Bytex
ibm-pps	1376/tcp	IBM Pers to Pers
ibm-pps	1376/udp	IBM Pers to Pers
cichlid	1377/tcp	Cichlid
cichlid	1377/udp	Cichlid
elan	1378/tcp	Elan
elan	1378/udp	Elan
dbreporter	1379/tcp	Integrity Sol
dbreporter	1379/udp	Integrity Sol
telesis-licman	1380/tcp	Telesis Network
telesis-licman	1380/udp	Telesis Network
apple-licman	1381/tcp	Apple Network
apple-licman	1381/udp	Apple Network
telexis-vtu	2804/udp	Telexis VTU
wta-wsp-s	2805/tcp	WTA WSP-S
wta-wsp-s	2805/udp	WTA WSP-S
cspuni	2806/tcp	cspuni
cspuni	2806/udp	cspuni
cspmulti	2807/tcp	cspmulti
cspmulti	2807/udp	cspmulti
j-lan-p	2808/tcp	J-LAN-P
j-lan-p	2808/udp	J-LAN-P
corbaloc	2809/tcp	CORBA LOC
corbaloc	2809/udp	CORBA LOC
netsteward	2810/tcp	Active Net
netsteward	2810/udp	Active Net
gsiftp	2811/tcp	GSI FTP
gsiftp	2811/udp	GSI FTP
atmtcp	2812/tcp	atmtcp
atmtcp	2812/udp	atmtcp
llm-pass	2813/tcp	llm-pass
llm-pass	2813/udp	llm-pass
llm-csv	2814/tcp	llm-csv
llm-csv	2814/udp	llm-csv
lbc-measure	2815/tcp	LBC Measurement
lbc-measure	2815/udp	LBC Measurement
lbc-watchdog	2816/tcp	LBC Watchdog
lbc-watchdog	2816/udp	LBC Watchdog
nmsigport	2817/tcp	NMSig Port
nmsigport	2817/udp	NMSig Port
rmlnk	2818/tcp	rmlnk
rmlnk	2818/udp	rmlnk
fc-faultnotify	2819/tcp	FC Fault Notif
fc-faultnotify	2819/udp	FC Fault Notif
univision	2820/tcp	UniVision
univision	2820/udp	UniVision
vml-dms	2821/tcp	vml_dms
vml-dms	2821/udp	vml_dms
ka0wuc	2822/tcp	ka0wuc
ka0wuc	2822/udp	ka0wuc
cqg-netlan	2823/tcp	CQG Net/LAN
cqg-netlan	2823/udp	CQG Net/LAN
slc-systemlog	2826/tcp	slc systemlog
slc-systemlog	2826/udp	slc systemlog
slc-ctrlrloops	2827/tcp	slc ctrlrloops
slc-ctrlrloops	2827/udp	slc ctrlrloops
itm-lm	2828/tcp	ITM License Mgr
itm-lm	2828/udp	ITM License Mgr
silkp1	2829/tcp	silkp1
silkp1	2829/udp	silkp1
silkp2	2830/tcp	silkp2
silkp2	2830/udp	silkp2
silkp3	2831/tcp	silkp3
silkp3	2831/udp	silkp3
silkp4	2832/tcp	silkp4
silkp4	2832/udp	silkp4
glishd	2833/tcp	glishd
glishd	2833/udp	glishd
evtp	2834/tcp	EVTP
evtp	2834/udp	EVTP
evtp-data	2835/tcp	EVTP-DATA
evtp-data	2835/udp	EVTP-DATA
catalyst	2836/tcp	catalyst
catalyst	2836/udp	catalyst
repliweb	2837/tcp	Repliweb
repliweb	2837/udp	Repliweb
starbot	2838/tcp	Starbot
starbot	2838/udp	Starbot
nmsigport	2839/tcp	NMSigPort
nmsigport	2839/udp	NMSigPort
l3-exprt	2840/tcp	l3-exprt
l3-exprt	2840/udp	l3-exprt
l3-ranger	2841/tcp	l3-ranger
l3-ranger	2841/udp	l3-ranger

udt_os	1382/tcp	
udt_os	1382/udp	
gwha	1383/tcp	GW Hannaway
gwha	1383/udp	GW Hannaway
os-licman	1384/tcp	Objective Sol
os-licman	1384/udp	Objective Sol
atex_elmd	1385/tcp	Atex Publishing
atex_elmd	1385/udp	Atex Publishing
checksum	1386/tcp	CheckSum
checksum	1386/udp	CheckSum
cadsi-lm	1387/tcp	Computer Aided
cadsi-lm	1387/udp	Computer Aided
objective-dbc	1388/tcp	Objective Sol
objective-dbc	1388/udp	Objective Sol
iclpv-dm	1389/tcp	Document Manager
iclpv-dm	1389/udp	Document Manager
iclpv-sc	1390/tcp	Storage Ctl
iclpv-sc	1390/udp	Storage Ctl
iclpv-sas	1391/tcp	Storage Access
iclpv-sas	1391/udp	Storage Access
iclpv-pm	1392/tcp	Print Manager
iclpv-pm	1392/udp	Print Manager
iclpv-nls	1393/tcp	Network Log Serv
iclpv-nls	1393/udp	Network Log Serv
iclpv-nlc	1394/tcp	Network Log Clt
iclpv-nlc	1394/udp	Network Log Clt
iclpv-wsm	1395/tcp	PC Workstation
iclpv-wsm	1395/udp	PC Workstation
dvl-activemail	1396/tcp	DVL Active Mail
dvl-activemail	1396/udp	DVL Active Mail
audio-activmail	1397/tcp	Audio Act Mail
audio-activmail	1397/udp	Audio Act Mail
video-activmail	1398/tcp	Video Act Mail
video-activmail	1398/udp	Video Act Mail
cadkey-licman	1399/tcp	Cadkey
cadkey-licman	1399/udp	Cadkey
cadkey-tablet	1400/tcp	Cadkey
cadkey-tablet	1400/udp	Cadkey
goldleaf-licman	1401/tcp	Goldleaf
goldleaf-licman	1401/udp	Goldleaf
prm-sm-np	1402/tcp	Prospero Res Man
prm-sm-np	1402/udp	Prospero Res Man
prm-nm-np	1403/tcp	Prospero Res Man
prm-nm-np	1403/udp	Prospero Res Man
igi-lm	1404/tcp	Infinite Graph
igi-lm	1404/udp	Infinite Graph
ibm-res	1405/tcp	IBM Remote Exec
ibm-res	1405/udp	IBM Remote Exec
netlabs-lm	1406/tcp	NetLabs
netlabs-lm	1406/udp	NetLabs
dbsa-lm	1407/tcp	DBSA
dbsa-lm	1407/udp	DBSA
sophia-lm	1408/tcp	Sophia
sophia-lm	1408/udp	Sophia
here-lm	1409/tcp	Here License Man
here-lm	1409/udp	Here License Man
hiq	1410/tcp	HiQ License Man
hiq	1410/udp	HiQ License Mana
af	1411/tcp	AudioFile
af	1411/udp	AudioFile
innosys	1412/tcp	InnoSys
innosys	1412/udp	InnoSys
innosys-acl	1413/tcp	Innosys-ACL
innosys-acl	1413/udp	Innosys-ACL
ibm-mqseries	1414/tcp	IBM MQSeries
ibm-mqseries	1414/udp	IBM MQSeries
dbstar	1415/tcp	DBStar
dbstar	1415/udp	DBStar
novell-lu6.2	1416/tcp	Novell LU6.2
novell-lu6.2	1416/udp	Novell LU6.2
timbuktu-srv1	1417/tcp	Timbuktu Serv 1
l3-hawk	2842/tcp	l3-hawk
l3-hawk	2842/udp	l3-hawk
pdnet	2843/tcp	PDnet
pdnet	2843/udp	PDnet
bpcp-poll	2844/tcp	BPCP POLL
bpcp-poll	2844/udp	BPCP POLL
bpcp-trap	2845/tcp	BPCP TRAP
bpcp-trap	2845/udp	BPCP TRAP
aimpp-hello	2846/tcp	AIMPP Hello
aimpp-hello	2846/udp	AIMPP Hello
aimpp-port-req	2847/tcp	AIMPP Port Req
aimpp-port-req	2847/udp	AIMPP Port Req
amt-blc-port	2848/tcp	AMT-BLC-PORT
amt-blc-port	2848/udp	AMT-BLC-PORT
fxp	2849/tcp	FXP
fxp	2849/udp	FXP
metaconsole	2850/tcp	MetaConsole
metaconsole	2850/udp	MetaConsole
webemshttp	2851/tcp	webemshttp
webemshttp	2851/udp	webemshttp
bears-01	2852/tcp	bears-01
bears-01	2852/udp	bears-01
ispipes	2853/tcp	ISPipes
ispipes	2853/udp	ISPipes
infomover	2854/tcp	InfoMover
infomover	2854/udp	InfoMover
cesdinv	2856/tcp	cesdinv
cesdinv	2856/udp	cesdinv
simctlp	2857/tcp	SimCtIP
simctlp	2857/udp	SimCtIP
ecnp	2858/tcp	ECNP
ecnp	2858/udp	ECNP
activememory	2859/tcp	Active Memory
activememory	2859/udp	Active Memory
dialpad-voice1	2860/tcp	Dialpad Voice 1
dialpad-voice1	2860/udp	Dialpad Voice 1
dialpad-voice2	2861/tcp	Dialpad Voice 2
dialpad-voice2	2861/udp	Dialpad Voice 2
ttg-protocol	2862/tcp	TTG Protocol
ttg-protocol	2862/udp	TTG Protocol
sonardata	2863/tcp	Sonar Data
sonardata	2863/udp	Sonar Data
astromed-main	2864/tcp	main 5001 cmd
astromed-main	2864/udp	main 5001 cmd
pit-vpn	2865/tcp	pit-vpn
pit-vpn	2865/udp	pit-vpn
lwlistener	2866/tcp	lwlistener
lwlistener	2866/udp	lwlistener
esps-portal	2867/tcp	esps-portal
esps-portal	2867/udp	esps-portal
npep-messaging	2868/tcp	NPEP Messaging
npep-messaging	2868/udp	NPEP Messaging
icslap	2869/tcp	ICSLAP
icslap	2869/udp	ICSLAP
daishi	2870/tcp	daishi
daishi	2870/udp	daishi
msi-selectplay	2871/tcp	MSI Select Play
msi-selectplay	2871/udp	MSI Select Play
contract	2872/tcp	CONTRACT
contract	2872/udp	CONTRACT
paspar2-zoomin	2873/tcp	PASPAR2 ZoomIn
paspar2-zoomin	2873/udp	PASPAR2 ZoomIn
dxmessagebase1	2874/tcp	dxmessagebase1
dxmessagebase1	2874/udp	dxmessagebase1
dxmessagebase2	2875/tcp	dxmessagebase2
dxmessagebase2	2875/udp	dxmessagebase2
sps-tunnel	2876/tcp	SPS Tunnel
sps-tunnel	2876/udp	SPS Tunnel
bluelance	2877/tcp	BLUELANCE
bluelance	2877/udp	BLUELANCE
aap	2878/tcp	AAP

timbuktu-srv1	1417/udp	Timbuktu Serv 1
timbuktu-srv2	1418/tcp	Timbuktu Serv 2
timbuktu-srv2	1418/udp	Timbuktu Serv 2
timbuktu-srv3	1419/tcp	Timbuktu Serv 3
timbuktu-srv3	1419/udp	Timbuktu Serv 3
timbuktu-srv4	1420/tcp	Timbuktu Serv 4
timbuktu-srv4	1420/udp	Timbuktu Serv 4
gandalf-lm	1421/tcp	Gandalf
gandalf-lm	1421/udp	Gandalf
autodesk-lm	1422/tcp	Autodesk
autodesk-lm	1422/udp	Autodesk
essbase	1423/tcp	Essbase Arbor
essbase	1423/udp	Essbase Arbor
hybrid	1424/tcp	Hybrid Encrypt
hybrid	1424/udp	Hybrid Encrypt
zion-lm	1425/tcp	Zion Software
zion-lm	1425/udp	Zion Software
sais	1426/tcp	Satellite-data 1
sais	1426/udp	Satellite-data 1
mloadd	1427/tcp	mloadd
mloadd	1427/udp	mloadd
informatik-lm	1428/tcp	Informatik
informatik-lm	1428/udp	Informatik
nms	1429/tcp	Hypercom NMS
nms	1429/udp	Hypercom NMS
tpdu	1430/tcp	Hypercom TPDU
tpdu	1430/udp	Hypercom TPDU
rgtp	1431/tcp	Reverse Gossip
rgtp	1431/udp	Reverse Gossip
blueberry-lm	1432/tcp	Blueberry Soft
blueberry-lm	1432/udp	Blueberry Soft
ms-sql-s	1433/tcp	Microsoft-SQL
ms-sql-s	1433/udp	Microsoft-SQL
ms-sql-m	1434/tcp	Microsoft-SQL
ms-sql-m	1434/udp	Microsoft-SQL
ibm-cics	1435/tcp	IBM CICS
ibm-cics	1435/udp	IBM CICS
saism	1436/tcp	Satellite-data 2
saism	1436/udp	Satellite-data 2
tabula	1437/tcp	Tabula
tabula	1437/udp	Tabul
eicon-server	1438/tcp	Eicon Security
eicon-server	1438/udp	Eicon Security
eicon-x25	1439/tcp	Eicon X25/SNA
eicon-x25	1439/udp	Eicon X25/SNA
eicon-slp	1440/tcp	Eicon Service
eicon-slp	1440/udp	Eicon Service
cadis-1	1441/tcp	Cadis
cadis-1	1441/udp	Cadis
cadis-2	1442/tcp	Cadis
cadis-2	1442/udp	Cadis
ies-lm	1443/tcp	Int Eng Soft
ies-lm	1443/udp	Int Eng Soft
marcam-lm	1444/tcp	Marcam
marcam-lm	1444/udp	Marcam
proxima-lm	1445/tcp	Proxima
proxima-lm	1445/udp	Proxima
ora-lm	1446/tcp	Optical Research
ora-lm	1446/udp	Optical Research
apri-lm	1447/tcp	Applied Parallel
apri-lm	1447/udp	Applied Parallel
oc-lm	1448/tcp	OpenConnect
oc-lm	1448/udp	OpenConnect
peport	1449/tcp	PEport
peport	1449/udp	PEport
dwf	1450/tcp	Tandem
dwf	1450/udp	Tandem
infoman	1451/tcp	IBM Information
infoman	1451/udp	IBM Information
gtegsc-lm	1452/tcp	GTE Government
gtegsc-lm	1452/udp	GTE Government
aap	2878/udp	AAP
ucentric-ds	2879/tcp	ucentric-ds
ucentric-ds	2879/udp	ucentric-ds
synapse	2880/tcp	synapse
synapse	2880/udp	synapse
ndsp	2881/tcp	NDSP
ndsp	2881/udp	NDSP
ndtp	2882/tcp	NDTP
ndtp	2882/udp	NDTP
ndnp	2883/tcp	NDNP
ndnp	2883/udp	NDNP
flashmsg	2884/tcp	Flash Msg
flashmsg	2884/udp	Flash Msg
topflow	2885/tcp	TopFlow
topflow	2885/udp	TopFlow
responselogic	2886/tcp	RESPONSELOGIC
responselogic	2886/udp	RESPONSELOGIC
aironetddp	2887/tcp	aironet
aironetddp	2887/udp	aironet
spcsdlobby	2888/tcp	SPCSDLOBBY
spcsdlobby	2888/udp	SPCSDLOBBY
rsom	2889/tcp	RSOM
rsom	2889/udp	RSOM
cspclmulti	2890/tcp	CSPCLMULTI
cspclmulti	2890/udp	CSPCLMULTI
cinegrfx-elmd	2891/tcp	CINEGRFX-ELMD
cinegrfx-elmd	2891/udp	CINEGRFX-ELMD
snifferdata	2892/tcp	SNIFFERDATA
snifferdata	2892/udp	SNIFFERDATA
vseconnector	2893/tcp	VSECONNECTOR
vseconnector	2893/udp	VSECONNECTOR
abacus-remote	2894/tcp	ABACUS-REMOTE
abacus-remote	2894/udp	ABACUS-REMOTE
natuslink	2895/tcp	NATUS LINK
natuslink	2895/udp	NATUS LINK
ecovisiong6-1	2896/tcp	ECOVISIONG6-1
ecovisiong6-1	2896/udp	ECOVISIONG6-1
citrix-rtmp	2897/tcp	Citrix RTMP
citrix-rtmp	2897/udp	Citrix RTMP
appliance-cfg	2898/tcp	APPLIANCE-CFG
appliance-cfg	2898/udp	APPLIANCE-CFG
powergemplus	2899/tcp	POWERGEMPLUS
powergemplus	2899/udp	POWERGEMPLUS
quicksuite	2900/tcp	QUICKSUITE
quicksuite	2900/udp	QUICKSUITE
allstorcns	2901/tcp	ALLSTORCNS
allstorcns	2901/udp	ALLSTORCNS
netaspi	2902/tcp	NET ASPI
netaspi	2902/udp	NET ASPI
suitcase	2903/tcp	SUITCASE
suitcase	2903/udp	SUITCASE
m2ua	2904/tcp	M2UA
m2ua	2904/udp	M2UA
m3ua	2905/tcp	M3UA
m3ua	2905/udp	M3UA
caller9	2906/tcp	CALLER9
caller9	2906/udp	CALLER9
webmethods-b2b	2907/tcp	WEBMETHODS B2B
webmethods-b2b	2907/udp	WEBMETHODS B2B
mao	2908/tcp	mao
mao	2908/udp	mao
funk-dialout	2909/tcp	Funk Dialout
funk-dialout	2909/udp	Funk Dialout
tdaccess	2910/tcp	TDAccess
tdaccess	2910/udp	TDAccess
blockade	2911/tcp	Blockade
blockade	2911/udp	Blockade
epicon	2912/tcp	Epicon
epicon	2912/udp	Epicon
boosterware	2913/tcp	Booster Ware
boosterware	2913/udp	Booster Ware

genie-lm	1453/tcp	Genie
genie-lm	1453/udp	Genie
interhdl_elmd	1454/tcp	interHDL
interhdl_elmd	1454/udp	interHDL
esl-lm	1455/tcp	ESL
esl-lm	1455/udp	ESL
dca	1456/tcp	DCA
dca	1456/udp	DCA
valisys-lm	1457/tcp	Valisys
valisys-lm	1457/udp	Valisys
nrcabq-lm	1458/tcp	Nichols Research
nrcabq-lm	1458/udp	Nichols Research
proshare1	1459/tcp	Proshare App
proshare1	1459/udp	Proshare App
proshare2	1460/tcp	Proshare App
proshare2	1460/udp	Proshare App
ibm_wrless_lan	1461/tcp	IBM Wireless LAN
ibm_wrless_lan	1461/udp	IBM Wireless LAN
world-lm	1462/tcp	World
world-lm	1462/udp	World
nucleus	1463/tcp	Nucleus
nucleus	1463/udp	Nucleus
msl_lmd	1464/tcp	MSL License Man
msl_lmd	1464/udp	MSL License Man
pipes	1465/tcp	Pipes Platform
pipes	1465/udp	Pipes Platform
oceansoft-lm	1466/tcp	Ocean Software
oceansoft-lm	1466/udp	Ocean Software
csdmbase	1467/tcp	CSDMBASE
csdmbase	1467/udp	CSDMBASE
csdm	1468/tcp	CSDM
csdm	1468/udp	CSDM
aal-lm	1469/tcp	Active Analysis
aal-lm	1469/udp	Active Analysis
uaiact	1470/tcp	Univ Analytics
uaiact	1470/udp	Univ Analytics
csdmbase	1471/tcp	csdmbase
csdmbase	1471/udp	csdmbase
csdm	1472/tcp	csdm
csdm	1472/udp	csdm
openmath	1473/tcp	OpenMath
openmath	1473/udp	OpenMath
telefinder	1474/tcp	Telefinder
telefinder	1474/udp	Telefinder
taligent-lm	1475/tcp	Taligent
taligent-lm	1475/udp	Taligent
clvm-cfg	1476/tcp	clvm-cfg
clvm-cfg	1476/udp	clvm-cfg
ms-sna-server	1477/tcp	ms-sna-server
ms-sna-server	1477/udp	ms-sna-server
ms-sna-base	1478/tcp	ms-sna-base
ms-sna-base	1478/udp	ms-sna-base
dberegister	1479/tcp	dberegister
dberegister	1479/udp	dberegister
pacerforum	1480/tcp	PacerForum
pacerforum	1480/udp	PacerForum
airs	1481/tcp	AIRS
airs	1481/udp	AIRS
miteksys-lm	1482/tcp	Miteksys
miteksys-lm	1482/udp	Miteksys
afs	1483/tcp	AFS
afs	1483/udp	AFS
confluent	1484/tcp	Confluent
confluent	1484/udp	Confluent
lansource	1485/tcp	LANSource
lansource	1485/udp	LANSource
nms_topo_serv	1486/tcp	nms_topo_serv
nms_topo_serv	1486/udp	nms_topo_serv
localinfosrvr	1487/tcp	LocalInfoSrvr
localinfosrvr	1487/udp	LocalInfoSrvr
docstor	1488/tcp	DocStor
gamelobby	2914/tcp	Game Lobby
gamelobby	2914/udp	Game Lobby
tksocket	2915/tcp	TK Socket
tksocket	2915/udp	TK Socket
elvin_server	2916/tcp	Elvin Server
elvin_server	2916/udp	Elvin Server
elvin_client	2917/tcp	Elvin Client
elvin_client	2917/udp	Elvin Client
kastenchasepad	2918/tcp	Kasten Chase Pad
kastenchasepad	2918/udp	Kasten Chase Pad
roboer	2919/tcp	ROBOER
roboer	2919/udp	ROBOER
roboeda	2920/tcp	ROBOEDA
roboeda	2920/udp	ROBOEDA
cesdcdman	2921/tcp	CESD Contents
cesdcdman	2921/udp	CESD Contents
cesdcdtrn	2922/tcp	CESD Contents
cesdcdtrn	2922/udp	CESD Contents
wta-wsp-wtp-s	2923/tcp	WTA-WSP-WTP-S
wta-wsp-wtp-s	2923/udp	WTA-WSP-WTP-S
precise-vip	2924/tcp	PRECISE-VIP
precise-vip	2924/udp	PRECISE-VIP
frp	2925/tcp	Firewall Redund
frp	2925/udp	Firewall Redund
mobile-file-dl	2926/tcp	MOBILE-FILE-DL
mobile-file-dl	2926/udp	MOBILE-FILE-DL
unimobilectrl	2927/tcp	UNIMOBILECTRL
unimobilectrl	2927/udp	UNIMOBILECTRL
redstone-cpss	2928/tcp	REDSTONE-CPSS
redstone-cpss	2928/udp	REDSONTE-CPSS
panja-webadmin	2929/tcp	PANJA-WEBADMIN
panja-webadmin	2929/udp	PANJA-WEBADMIN
panja-weblinx	2930/tcp	PANJA-WEBLINX
panja-weblinx	2930/udp	PANJA-WEBLINX
circle-x	2931/tcp	Circle-X
circle-x	2931/udp	Circle-X
incp	2932/tcp	INCP
incp	2932/udp	INCP
4-tieropmgw	2933/tcp	4-TIER OPM GW
4-tieropmgw	2933/udp	4-TIER OPM GW
4-tieropmcli	2934/tcp	4-TIER OPM CLI
4-tieropmcli	2934/udp	4-TIER OPM CLI
qtp	2935/tcp	QTP
qtp	2935/udp	QTP
otpatch	2936/tcp	OTPatch
otpatch	2936/udp	OTPatch
pnaconsult-lm	2937/tcp	PNACONSULT-LM
pnaconsult-lm	2937/udp	PNACONSULT-LM
sm-pas-1	2938/tcp	SM-PAS-1
sm-pas-1	2938/udp	SM-PAS-1
sm-pas-2	2939/tcp	SM-PAS-2
sm-pas-2	2939/udp	SM-PAS-2
sm-pas-3	2940/tcp	SM-PAS-3
sm-pas-3	2940/udp	SM-PAS-3
sm-pas-4	2941/tcp	SM-PAS-4
sm-pas-4	2941/udp	SM-PAS-4
sm-pas-5	2942/tcp	SM-PAS-5
sm-pas-5	2942/udp	SM-PAS-5
ttnrepository	2943/tcp	TTNRepository
ttnrepository	2943/udp	TTNRepository
megaco-h248	2944/tcp	Megaco H-248
megaco-h248	2944/udp	Megaco H-248
h248-binary	2945/tcp	H248 Binary
h248-binary	2945/udp	H248 Binary
fjsvmpor	2946/tcp	FJSVmpor
fjsvmpor	2946/udp	FJSVmpor
gpsd	2947/tcp	GPSD
gpsd	2947/udp	GPSD
wap-push	2948/tcp	WAP PUSH
wap-push	2948/udp	WAP PUSH
wap-pushsecure	2949/tcp	WAP PUSH SECURE

docstor	1488/udp	DocStor
dmdocbroker	1489/tcp	dmdocbroker
dmdocbroker	1489/udp	dmdocbroker
insitu-conf	1490/tcp	insitu-conf
insitu-conf	1490/udp	insitu-conf
anynetgateway	1491/tcp	anynetgateway
anynetgateway	1491/udp	anynetgateway
stone-design-1	1492/tcp	stone-design-1
stone-design-1	1492/udp	stone-design-1
netmap_lm	1493/tcp	netmap_lm
netmap_lm	1493/udp	netmap_lm
ica	1494/tcp	ica
ica	1494/udp	ica
cvc	1495/tcp	cvc
cvc	1495/udp	cvc
liberty-lm	1496/tcp	liberty-lm
liberty-lm	1496/udp	liberty-lm
rfx-lm	1497/tcp	rfx-lm
rfx-lm	1497/udp	rfx-lm
sybase-sqlany	1498/tcp	Sybase SQL Any
sybase-sqlany	1498/udp	Sybase SQL Any
fhc	1499/tcp	Federico Heinz
fhc	1499/udp	Federico Heinz
vlsi-lm	1500/tcp	VLSI
vlsi-lm	1500/udp	VLSI
saiscm	1501/tcp	Satellite-data 3
saiscm	1501/udp	Satellite-data 3
shivadiscovery	1502/tcp	Shiva
shivadiscovery	1502/udp	Shiva
imtc-mcs	1503/tcp	Databeam
imtc-mcs	1503/udp	Databeam
evb-elm	1504/tcp	EVB Software
evb-elm	1504/udp	EVB Software
funkproxy	1505/tcp	Funk Software
funkproxy	1505/udp	Funk Software
utcd	1506/tcp	Universal Time
utcd	1506/udp	Universal Time
symplex	1507/tcp	Symplex
symplex	1507/udp	Symplex
diagmond	1508/tcp	diagmond
diagmond	1508/udp	diagmond
robcad-lm	1509/tcp	Robcad, Ltd.
robcad-lm	1509/udp	Robcad, Ltd.
mvx-lm	1510/tcp	Midland Valley
mvx-lm	1510/udp	Midland Valley
3l-l1	1511/tcp	3l-l1
3l-l1	1511/udp	3l-l1
wins	1512/tcp	Name Service
wins	1512/udp	Name Service
fujitsu-dtc	1513/tcp	Fujitsu Systems
fujitsu-dtc	1513/udp	Fujitsu Systems
fujitsu-dtcns	1514/tcp	Fujitsu Systems
fujitsu-dtcns	1514/udp	Fujitsu Systems
ifor-protocol	1515/tcp	ifor-protocol
ifor-protocol	1515/udp	ifor-protocol
vpad	1516/tcp	Virtual Places
vpad	1516/udp	Virtual Places
vpac	1517/tcp	Virtual Places
vpac	1517/udp	Virtual Places
vpvd	1518/tcp	Virtual Places
vpvd	1518/udp	Virtual Places
vpvc	1519/tcp	Virtual Places
vpvc	1519/udp	Virtual Places
atm-zip-office	1520/tcp	atm zip office
atm-zip-office	1520/udp	atm zip office
ncube-lm	1521/tcp	nCube
ncube-lm	1521/udp	nCube
ricardo-lm	1522/tcp	Ricardo North
ricardo-lm	1522/udp	Ricardo North
cichild-lm	1523/tcp	cichild
cichild-lm	1523/udp	cichild
wap-pushsecure	2949/udp	WAP PUSH SECURE
esip	2950/tcp	ESIP
esip	2950/udp	ESIP
ottp	2951/tcp	OTTP
ottp	2951/udp	OTTP
mpfwsas	2952/tcp	MPFWSAS
mpfwsas	2952/udp	MPFWSAS
ovalarmsrv	2953/tcp	OVALARMSRV
ovalarmsrv	2953/udp	OVALARMSRV
ovalarmsrv-cmd	2954/tcp	OVALARMSRV-CMD
ovalarmsrv-cmd	2954/udp	OVALARMSRV-CMD
csnotify	2955/tcp	CSNOTIFY
csnotify	2955/udp	CSNOTIFY
ovrimosdbman	2956/tcp	OVRIMOSDBMAN
ovrimosdbman	2956/udp	OVRIMOSDBMAN
jmact5	2957/tcp	JAMCT5
jmact5	2957/udp	JAMCT5
jmact6	2958/tcp	JAMCT6
jmact6	2958/udp	JAMCT6
rmopagt	2959/tcp	RMOPAGT
rmopagt	2959/udp	RMOPAGT
dfoxserver	2960/tcp	DFOXSERVER
dfoxserver	2960/udp	DFOXSERVER
boldsoft-lm	2961/tcp	BOLDSOFT-LM
boldsoft-lm	2961/udp	BOLDSOFT-LM
iph-policy-cli	2962/tcp	IPH-POLICY-CLI
iph-policy-cli	2962/udp	IPH-POLICY-CLI
iph-policy-adm	2963/tcp	IPH-POLICY-ADM
iph-policy-adm	2963/udp	IPH-POLICY-ADM
bullant-srap	2964/tcp	BULLANT SRAP
bullant-srap	2964/udp	BULLANT SRAP
bullant-rap	2965/tcp	BULLANT RAP
bullant-rap	2965/udp	BULLANT RAP
idp-infotrieve	2966/tcp	IDP-INFOTRIEVE
idp-infotrieve	2966/udp	IDP-INFOTRIEVE
ssc-agent	2967/tcp	SSC-AGENT
ssc-agent	2967/udp	SSC-AGENT
enpp	2968/tcp	ENPP
enpp	2968/udp	ENPP
essp	2969/tcp	ESSP
essp	2969/udp	ESSP
index-net	2970/tcp	INDEX-NET
index-net	2970/udp	INDEX-NET
netclip	2971/tcp	Net Clip
netclip	2971/udp	Net Clip
pmsm-webrctl	2972/tcp	PMSM Webrctl
pmsm-webrctl	2972/udp	PMSM Webrctl
svnetworks	2973/tcp	SV Networks
svnetworks	2973/udp	SV Networks
signal	2974/tcp	Signal
signal	2974/udp	Signal
fjmpcm	2975/tcp	Fujitsu
fjmpcm	2975/udp	Fujitsu
cns-srv-port	2976/tcp	CNS Server Port
cns-srv-port	2976/udp	CNS Server Port
ttc-etap-ns	2977/tcp	TTCs Enterprise
ttc-etap-ns	2977/udp	TTCs Enterprise
ttc-etap-ds	2978/tcp	TTCs Enterprise
ttc-etap-ds	2978/udp	TTCs Enterprise
h263-video	2979/tcp	H.263 Video
h263-video	2979/udp	H.263 Video
wimd	2980/tcp	Instant
wimd	2980/udp	Instant
mylxamport	2981/tcp	MYLXAMPORT
mylxamport	2981/udp	MYLXAMPORT
iwb-whiteboard	2982/tcp	IWB-WHITEBOARD
iwb-whiteboard	2982/udp	IWB-WHITEBOARD
netplan	2983/tcp	NETPLAN
netplan	2983/udp	NETPLAN
hpidsadmin	2984/tcp	HPIDSADMIN
hpidsadmin	2984/udp	HPIDSADMIN

ingreslock	1524/tcp	ingres
ingreslock	1524/udp	ingres
orasrv	1525/tcp	oracle
orasrv	1525/udp	oracle
prospero-np	1525/tcp	Prospero
prospero-np	1525/udp	Prospero
pdap-np	1526/tcp	Prospero
pdap-np	1526/udp	Prospero
tlisrv	1527/tcp	oracle
tlisrv	1527/udp	oracle
mciautoreg	1528/tcp	micautoreg
mciautoreg	1528/udp	micautoreg
coauthor	1529/tcp	oracle
coauthor	1529/udp	oracle
rap-service	1530/tcp	rap-service
rap-service	1530/udp	rap-service
rap-listen	1531/tcp	rap-listen
rap-listen	1531/udp	rap-listen
miroconnect	1532/tcp	miroconnect
miroconnect	1532/udp	miroconnect
virtual-places	1533/tcp	Virtual Places
virtual-places	1533/udp	Virtual Places
micromuse-lm	1534/tcp	micromuse-lm
micromuse-lm	1534/udp	micromuse-lm
ampr-info	1535/tcp	ampr-info
ampr-info	1535/udp	ampr-info
ampr-inter	1536/tcp	ampr-inter
ampr-inter	1536/udp	ampr-inter
sdsc-lm	1537/tcp	isi-lm
sdsc-lm	1537/udp	isi-lm
3ds-lm	1538/tcp	3ds-lm
3ds-lm	1538/udp	3ds-lm
intellistor-lm	1539/tcp	Intellistor
intellistor-lm	1539/udp	Intellistor
rds	1540/tcp	rds
rds	1540/udp	rds
rds2	1541/tcp	rds2
rds2	1541/udp	rds2
gridgen-elmd	1542/tcp	gridgen-elmd
gridgen-elmd	1542/udp	gridgen-elmd
simba-cs	1543/tcp	simba-cs
simba-cs	1543/udp	simba-cs
aspeclmd	1544/tcp	aspeclmd
aspeclmd	1544/udp	aspeclmd
vistium-share	1545/tcp	vistium-share
vistium-share	1545/udp	vistium-share
abbaccuray	1546/tcp	abbaccuray
abbaccuray	1546/udp	abbaccuray
laplink	1547/tcp	laplink
laplink	1547/udp	laplink
axon-lm	1548/tcp	Axon
axon-lm	1548/udp	Axon
shivahose	1549/tcp	Shiva Hose
shivasound	1549/udp	Shiva Sound
3m-image-lm	1550/tcp	Image 3M
3m-image-lm	1550/udp	Image 3M
hecmtl-db	1551/tcp	HECMTL-DB
hecmtl-db	1551/udp	HECMTL-DB
pciarray	1552/tcp	pciarray
pciarray	1552/udp	pciarray
sna-cs	1553/tcp	sna-cs
sna-cs	1553/udp	sna-cs
caci-lm	1554/tcp	CACI Products
caci-lm	1554/udp	CACI Products
livelan	1555/tcp	livelan
livelan	1555/udp	livelan
ashwin	1556/tcp	AshWin CI
ashwin	1556/udp	AshWin CI
arbortext-lm	1557/tcp	ArborText
arbortext-lm	1557/udp	ArborText
xingmpeg	1558/tcp	xingmpeg
hpidsagent	2985/tcp	HPIDSAGENT
hpidsagnet	2985/udp	HPIDSAGENT
stonefalls	2986/tcp	STONEFALLS
stonefalls	2986/udp	STONEFALLS
identify	2987/tcp	IDENTIFY
identify	2987/udp	IDENTIFY
classify	2988/tcp	CLASSIFY
classify	2988/udp	CLASSIFY
zarkov	2989/tcp	ZARKOV
zarkov	2989/udp	ZARKOV
boscap	2990/tcp	BOSCAP
boscap	2990/udp	BOSCAP
wkstn-mon	2991/tcp	WKSTN-MON
wkstn-mon	2991/udp	WKSTN-MON
itb301	2992/tcp	ITB301
itb301	2992/udp	ITB301
veritas-vis1	2993/tcp	VERITAS VIS1
veritas-vis1	2993/udp	VERITAS VIS1
veritas-vis2	2994/tcp	VERITAS VIS2
veritas-vis2	2994/udp	VERITAS VIS2
idrs	2995/tcp	IDRS
idrs	2995/udp	IDRS
vsixml	2996/tcp	vsixml
vsixml	2996/udp	vsixml
rebol	2997/tcp	REBOL
rebol	2997/udp	REBOL
realsecure	2998/tcp	Real Secure
realsecure	2998/udp	Real Secure
remoteware-un	2999/tcp	RemoteWare
remoteware-un	2999/udp	RemoteWare
hbci	3000/tcp	HBCI
hbci	3000/udp	HBCI
remoteware-cl	3000/tcp	RemoteWare Clt
remoteware-cl	3000/udp	RemoteWare Clt
redwood-broker	3001/tcp	Redwood Broker
redwood-broker	3001/udp	Redwood Broker
exlm-agent	3002/tcp	EXLM Agent
exlm-agent	3002/udp	EXLM Agent
remoteware-srv	3002/tcp	RemoteWare Serv
remoteware-srv	3002/udp	RemoteWare Serv
cgms	3003/tcp	CGMS
cgms	3003/udp	CGMS
csoftragent	3004/tcp	Csoft Agent
csoftragent	3004/udp	Csoft Agent
geniuslm	3005/tcp	Genius
geniuslm	3005/udp	Genius
ii-admin	3006/tcp	Instant Internet
ii-admin	3006/udp	Instant Internet
lotusmtap	3007/tcp	Lotus Mail
lotusmtap	3007/udp	Lotus Mail
midnight-tech	3008/tcp	Midnight Tech
midnight-tech	3008/udp	Midnight Techn
pxc-ntfy	3009/tcp	PXC-NTFY
pxc-ntfy	3009/udp	PXC-NTFY
gw	3010/tcp	Telerate Workst
ping-pong	3010/udp	Telerate Workst
trusted-web	3011/tcp	Trusted Web
trusted-web	3011/udp	Trusted Web
twsdss	3012/tcp	Trusted Web Clt
twsdss	3012/udp	Trusted Web Clt
gilatskysurfer	3013/tcp	Gilat Sky Surfer
gilatskysurfer	3013/udp	Gilat Sky Surfer
broker_service	3014/tcp	Broker Service
broker_service	3014/udp	Broker Service
nati-dstp	3015/tcp	NATI DSTP
nati-dstp	3015/udp	NATI DSTP
notify_srvr	3016/tcp	Notify Server
notify_srvr	3016/udp	Notify Server
event_listener	3017/tcp	Event Listener
event_listener	3017/udp	Event Listener
srvc_registry	3018/tcp	Service Registry

xingmpeg	1558/udp	xingmpeg
web2host	1559/tcp	web2host
web2host	1559/udp	web2host
asci-val	1560/tcp	asci-val
asci-val	1560/udp	asci-val
facilityview	1561/tcp	facilityview
facilityview	1561/udp	facilityview
pconnectmgr	1562/tcp	pconnectmgr
pconnectmgr	1562/udp	pconnectmgr
cadabra-lm	1563/tcp	Cadabra
cadabra-lm	1563/udp	Cadabra
pay-per-view	1564/tcp	Pay-Per-View
pay-per-view	1564/udp	Pay-Per-View
winddlb	1565/tcp	WinDD
winddlb	1565/udp	WinDD
corelvideo	1566/tcp	CORELVIDEO
corelvideo	1566/udp	CORELVIDEO
jlicelmd	1567/tcp	jlicelmd
jlicelmd	1567/udp	jlicelmd
tsspmap	1568/tcp	tsspmap
tsspmap	1568/udp	tsspmap
ets	1569/tcp	ets
ets	1569/udp	ets
orbixd	1570/tcp	orbixd
orbixd	1570/udp	orbixd
rdb-dbs-disp	1571/tcp	Oracle Rem DB
rdb-dbs-disp	1571/udp	Oracle Rem DB
chip-lm	1572/tcp	Chipcom License
chip-lm	1572/udp	Chipcom License
itscomm-ns	1573/tcp	itscomm-ns
itscomm-ns	1573/udp	itscomm-ns
mvel-lm	1574/tcp	mvel-lm
mvel-lm	1574/udp	mvel-lm
oraclenames	1575/tcp	oraclenames
oraclenames	1575/udp	oraclenames
moldflow-lm	1576/tcp	moldflow-lm
moldflow-lm	1576/udp	moldflow-lm
hypercube-lm	1577/tcp	hypercube-lm
hypercube-lm	1577/udp	hypercube-lm
jacobus-lm	1578/tcp	Jacobus
jacobus-lm	1578/udp	Jacobus
ioc-sea-lm	1579/tcp	ioc-sea-lm
ioc-sea-lm	1579/udp	ioc-sea-lm
tn-tl-r1	1580/tcp	tn-tl-r1
tn-tl-r2	1580/udp	tn-tl-r2
mil-2045-47001	1581/tcp	MIL-2045-47001
mil-2045-47001	1581/udp	MIL-2045-47001
msims	1582/tcp	MSIMS
msims	1582/udp	MSIMS
simbaexpress	1583/tcp	simbaexpress
simbaexpress	1583/udp	simbaexpress
tn-tl-fd2	1584/tcp	tn-tl-fd2
tn-tl-fd2	1584/udp	tn-tl-fd2
intv	1585/tcp	intv
intv	1585/udp	intv
ibm-abtact	1586/tcp	ibm-abtact
ibm-abtact	1586/udp	ibm-abtact
pra_elmd	1587/tcp	pra_elmd
pra_elmd	1587/udp	pra_elmd
triquest-lm	1588/tcp	triquest-lm
triquest-lm	1588/udp	triquest-lm
vqp	1589/tcp	VQP
vqp	1589/udp	VQPMcCloghrie
gemini-lm	1590/tcp	gemini-lm
gemini-lm	1590/udp	gemini-lm
ncpm-pm	1591/tcp	ncpm-pm
ncpm-pm	1591/udp	ncpm-pm
commonspace	1592/tcp	commonspace
commonspace	1592/udp	commonspace
mainsoft-lm	1593/tcp	mainsoft-lm
mainsoft-lm	1593/udp	mainsoft-lm
srvc_registry	3018/udp	Service Registry
resource_mgr	3019/tcp	Resource Manager
resource_mgr	3019/udp	Resource Manager
cifs	3020/tcp	CIFS
cifs	3020/udp	CIFS
agriserver	3021/tcp	AGRI Server
agriserver	3021/udp	AGRI Server
csregagent	3022/tcp	CSREGAGENT
csregagent	3022/udp	CSREGAGENT
magicnotes	3023/tcp	magicnotes
magicnotes	3023/udp	magicnotes
nds_sso	3024/tcp	NDS_SSO
nds_sso	3024/udp	NDS_SSO
arepa-raft	3025/tcp	Arepa Raft
arepa-raft	3025/udp	Arepa Raft
agri-gateway	3026/tcp	AGRI Gateway
agri-gateway	3026/udp	AGRI Gateway
LiebDevMgmt_C	3027/tcp	LiebDevMgmt_C
LiebDevMgmt_C	3027/udp	LiebDevMgmt_C
LiebDevMgmt_DM	3028/tcp	LiebDevMgmt_DM
LiebDevMgmt_DM	3028/udp	LiebDevMgmt_DM
LiebDevMgmt_A	3029/tcp	LiebDevMgmt_A
LiebDevMgmt_A	3029/udp	LiebDevMgmt_A
arepa-cas	3030/tcp	Arepa Cas
arepa-cas	3030/udp	Arepa Cas
agentvu	3031/tcp	AgentVU
agentvu	3031/udp	AgentVU
redwood-chat	3032/tcp	Redwood Chat
redwood-chat	3032/udp	Redwood Chat
pdb	3033/tcp	PDB
pdb	3033/udp	PDB
osmosis-aeea	3034/tcp	Osmosis AEEA
osmosis-aeea	3034/udp	Osmosis AEEA
fjsv-gssagt	3035/tcp	FJSV gssagt
fjsv-gssagt	3035/udp	FJSV gssagt
hagel-dump	3036/tcp	Hagel DUMP
hagel-dump	3036/udp	Hagel DUMP
hp-san-mgmt	3037/tcp	HP SAN Mgmt
hp-san-mgmt	3037/udp	HP SAN Mgmt
santak-ups	3038/tcp	Santak UPS
santak-ups	3038/udp	Santak UPS
cogitate	3039/tcp	Cogitate, Inc.
cogitate	3039/udp	Cogitate, Inc.
tomato-springs	3040/tcp	Tomato Springs
tomato-springs	3040/udp	Tomato Springs
di-traceware	3041/tcp	di-traceware
di-traceware	3041/udp	di-traceware
journee	3042/tcp	journee
journee	3042/udp	journee
brp	3043/tcp	BRP
brp	3043/udp	BRP
responsenet	3045/tcp	ResponseNet
responsenet	3045/udp	ResponseNet
di-ase	3046/tcp	di-ase
di-ase	3046/udp	di-ase
hlserver	3047/tcp	Fast Security HL
hlserver	3047/udp	Fast Security HL
pctrader	3048/tcp	Sierra Net PC
pctrader	3048/udp	Sierra Net PC
nsws	3049/tcp	NSWS
nsws	3049/udp	NSWS
gds_db	3050/tcp	gds_db
gds_db	3050/udp	gds_db
galaxy-server	3051/tcp	Galaxy Server
galaxy-server	3051/udp	Galaxy Server
apcpcns	3052/tcp	APCPCNS
apcpcns	3052/udp	APCPCNS
dsom-server	3053/tcp	dsom-server
dsom-server	3053/udp	dsom-server
amt-cnf-prot	3054/tcp	AMT CNF PROT
amt-cnf-prot	3054/udp	AMT CNF PROT

sixtrak	1594/tcp	sixtrak
sixtrak	1594/udp	sixtrak
radio	1595/tcp	radio
radio	1595/udp	radio
radio-sm	1596/tcp	radio-sm
radio-bc	1596/udp	radio-bc
orbplus-iiop	1597/tcp	orbplus-iiop
orbplus-iiop	1597/udp	orbplus-iiop
picknfs	1598/tcp	picknfs
picknfs	1598/udp	picknfs
simbaservices	1599/tcp	simbaservices
simbaservices	1599/udp	simbaservices
issd	1600/tcp	
issd	1600/udp	
aas	1601/tcp	aas
aas	1601/udp	aas
inspect	1602/tcp	inspect
inspect	1602/udp	inspect
picodbc	1603/tcp	pickodbc
picodbc	1603/udp	pickodbc
icabrowser	1604/tcp	icabrowser
icabrowser	1604/udp	icabrowser
slp	1605/tcp	Salutation
slp	1605/udp	Salutation
slm-api	1606/tcp	Salutation
slm-api	1606/udp	Salutation
stt	1607/tcp	stt
stt	1607/udp	stt
smart-lm	1608/tcp	Smart Corp.
smart-lm	1608/udp	Smart Corp.
isysg-lm	1609/tcp	isysg-lm
isysg-lm	1609/udp	isysg-lm
taurus-wh	1610/tcp	taurus-wh
taurus-wh	1610/udp	taurus-wh
ill	1611/tcp	Inter Library
ill	1611/udp	Inter Library
netbill-trans	1612/tcp	NetBill
netbill-trans	1612/udp	NetBill
netbill-keyrep	1613/tcp	NetBill Key
netbill-keyrep	1613/udp	NetBill Key
netbill-cred	1614/tcp	NetBill
netbill-cred	1614/udp	NetBill
netbill-auth	1615/tcp	NetBill
netbill-auth	1615/udp	NetBill
netbill-prod	1616/tcp	NetBill
netbill-prod	1616/udp	NetBill
nimrod-agent	1617/tcp	Nimrod
nimrod-agent	1617/udp	Nimrod
skytelnet	1618/tcp	skytelnet
skytelnet	1618/udp	skytelnet
xs-openstorage	1619/tcp	xs-openstorage
xs-openstorage	1619/udp	xs-openstorage
faxportwinport	1620/tcp	faxportwinport
faxportwinport	1620/udp	faxportwinport
softdataphone	1621/tcp	softdataphone
softdataphone	1621/udp	softdataphone
ontime	1622/tcp	ontime
ontime	1622/udp	ontime
jaleosnd	1623/tcp	jaleosnd
jaleosnd	1623/udp	jaleosnd
udp-sr-port	1624/tcp	udp-sr-port
udp-sr-port	1624/udp	udp-sr-port
svs-omagent	1625/tcp	svs-omagent
svs-omagent	1625/udp	svs-omagent
shockwave	1626/tcp	Shockwave
shockwave	1626/udp	Shockwave
t128-gateway	1627/tcp	T.128 Gateway
t128-gateway	1627/udp	T.128 Gateway
lontalk-norm	1628/tcp	LonTalk normal
lontalk-norm	1628/udp	LonTalk normal
lontalk-urgnt	1629/tcp	LonTalk urgent
policyserver	3055/tcp	Policy Server
policyserver	3055/udp	Policy Server
cdl-server	3056/tcp	CDL Server
cdl-server	3056/udp	CDL Server
goahead-fldup	3057/tcp	GoAhead FldUp
goahead-fldup	3057/udp	GoAhead FldUp
videobeans	3058/tcp	videobeans
videobeans	3058/udp	videobeans
qsoft	3059/tcp	qsoft
qsoft	3059/tcp	qsoft
interserver	3060/tcp	interserver
interserver	3060/udp	interserver
cautcpd	3061/tcp	cautcpd
cautcpd	3061/udp	cautcpd
ncacn-ip-tcp	3062/tcp	ncacn-ip-tcp
ncacn-ip-tcp	3062/udp	ncacn-ip-tcp
ncadg-ip-udp	3063/tcp	ncadg-ip-udp
ncadg-ip-udp	3063/udp	ncadg-ip-udp
slinterbase	3065/tcp	slinterbase
slinterbase	3065/udp	slinterbase
netattachsdmp	3066/tcp	NETATTACHSDMP
netattachsdmp	3066/udp	NETATTACHSDMP
fjhpjp	3067/tcp	FJHPJP
fjhpjp	3067/udp	FJHPJP
ls3bcast	3068/tcp	ls3 Broadcast
ls3bcast	3068/udp	ls3 Broadcast
ls3	3069/tcp	ls3
ls3	3069/udp	ls3
mgxswitch	3070/tcp	MGXSWITCH
mgxswitch	3070/udp	MGXSWITCH
#	3071-3074	Unassigned
orbix-locator	3075/tcp	Orbix 2000
orbix-locator	3075/udp	Orbix 2000
orbix-config	3076/tcp	Orbix 2000
orbix-config	3076/udp	Orbix 2000
orbix-loc-ssl	3077/tcp	Orbix 2000 SSL
orbix-loc-ssl	3077/udp	Orbix 2000 SSL
orbix-cfg-ssl	3078/tcp	Orbix 2000 SSL
orbix-cfg-ssl	3078/udp	Orbix 2000 SSL
lv-frontpanel	3079/tcp	LV Front Panel
lv-frontpanel	3079/udp	LV Front Panel
stm_pproc	3080/tcp	stm_pproc
stm_pproc	3080/udp	stm_pproc
tl1-lv	3081/tcp	TL1-LV
tl1-lv	3081/udp	TL1-LV
tl1-raw	3082/tcp	TL1-RAW
tl1-raw	3082/udp	TL1-RAW
tl1-telnet	3083/tcp	TL1-TELNET
tl1-telnet	3083/udp	TL1-TELNET
itm-mccs	3084/tcp	ITM-MCCS
itm-mccs	3084/udp	ITM-MCCS
pcihreq	3085/tcp	PCIHReq
pcihreq	3085/udp	PCIHReq
jdl-dbkitchen	3086/tcp	JDL-DBKitchen
jdl-dbkitchen	3086/udp	JDL-DBKitchen
#	3084-3104	Unassigned
cardbox	3105/tcp	Cardbox
cardbox	3105/udp	Cardbox
cardbox-http	3106/tcp	Cardbox HTTP
cardbox-http	3106/udp	Cardbox HTTP
#	3107-3129	Unassigned
icpv2	3130/tcp	ICPv2
icpv2	3130/udp	ICPv2
netbookmark	3131/tcp	Net Book Mark
netbookmark	3131/udp	Net Book Mark
#	3132-3140	Unassigned
vmodem	3141/tcp	VMODEM
vmodem	3141/udp	VMODEM
rdc-wh-eos	3142/tcp	RDC WH EOS
rdc-wh-eos	3142/udp	RDC WH EOS
seaview	3143/tcp	Sea View

lontalk-urgnt	1629/udp	LonTalk urgent
oraclenet8cman	1630/tcp	Oracle Net8 Cman
oraclenet8cman	1630/udp	Oracle Net8 Cman
visitview	1631/tcp	Visit view
visitview	1631/udp	Visit view
pammratc	1632/tcp	PAMMRATC
pammratc	1632/udp	PAMMRATC
pammrpc	1633/tcp	PAMMRPC
pammrpc	1633/udp	PAMMRPC
loaprobe	1634/tcp	America Probe
loaprobe	1634/udp	America Probe
edb-server1	1635/tcp	EDB Server 1
edb-server1	1635/udp	EDB Server 1
cncp	1636/tcp	CableNet
cncp	1636/udp	CableNet
cnap	1637/tcp	CableNet Admin
cnap	1637/udp	CableNet Admin
cnip	1638/tcp	CableNet Info
cnip	1638/udp	CableNet Info
cert-initiator	1639/tcp	cert-initiator
cert-initiator	1639/udp	cert-initiator
cert-responder	1640/tcp	cert-responder
cert-responder	1640/udp	cert-responder
invision	1641/tcp	InVision
invision	1641/udp	InVision
isis-am	1642/tcp	isis-am
isis-am	1642/udp	isis-am
isis-ambc	1643/tcp	isis-ambc
isis-ambc	1643/udp	isis-ambc
saiseh	1644/tcp	Satellite-data 4
datametrics	1645/tcp	datametrics
datametrics	1645/udp	datametrics
sa-msg-port	1646/tcp	sa-msg-port
sa-msg-port	1646/udp	sa-msg-port
rsap	1647/tcp	rsap
rsap	1647/udp	rsap
concurrent-lm	1648/tcp	concurrent-lm
concurrent-lm	1648/udp	concurrent-lm
kermit	1649/tcp	kermit
kermit	1649/udp	kermit
nkd	1650/tcp	nkd
nkd	1650/udp	nkd
shiva_confsrvr	1651/tcp	shiva_confsrvr
shiva_confsrvr	1651/udp	shiva_confsrvr
xnmp	1652/tcp	xnmp
xnmp	1652/udp	xnm
alphatech-lm	1653/tcp	alphatech-lm
alphatech-lm	1653/udp	alphatech-lm
stargatealerts	1654/tcp	stargatealerts
stargatealerts	1654/udp	stargatealerts
dec-mbadmin	1655/tcp	dec-mbadmin
dec-mbadmin	1655/udp	dec-mbadmin
dec-mbadmin-h	1656/tcp	dec-mbadmin-h
dec-mbadmin-h	1656/udp	dec-mbadmin-h
fujitsu-mmpdc	1657/tcp	fujitsu-mmpdc
fujitsu-mmpdc	1657/udp	fujitsu-mmpdc
sixnetudr	1658/tcp	sixnetudr
sixnetudr	1658/udp	sixnetudr
sg-lm	1659/tcp	Silicon Grail
sg-lm	1659/udp	Silicon Grail
skip-mc-gikreq	1660/tcp	skip-mc-gikreq
skip-mc-gikreq	1660/udp	skip-mc-gikreq
netview-aix-1	1661/tcp	netview-aix-1
netview-aix-1	1661/udp	netview-aix-1
netview-aix-2	1662/tcp	netview-aix-2
netview-aix-2	1662/udp	netview-aix-2
netview-aix-3	1663/tcp	netview-aix-3
netview-aix-3	1663/udp	netview-aix-3
netview-aix-4	1664/tcp	netview-aix-4
netview-aix-4	1664/udp	netview-aix-4
netview-aix-5	1665/tcp	netview-aix-5
seaview	3143/udp	Sea View
tarantella	3144/tcp	Tarantella
tarantella	3144/udp	Tarantella
csi-lfap	3145/tcp	CSI-LFAP
csi-lfap	3145/udp	CSI-LFAP
#	3146	Unassigned
rfio	3147/tcp	RFIO
rfio	3147/udp	RFIO
nm-game-admin	3148/tcp	NetMike Game
nm-game-admin	3148/udp	NetMike Game
nm-game-server	3149/tcp	NetMike Game
nm-game-server	3149/udp	NetMike Game
nm-asses-admin	3150/tcp	NetMike Assessor
nm-asses-admin	3150/udp	NetMike Assessor
nm-assessor	3151/tcp	NetMike
nm-assessor	3151/udp	NetMike
#	3152-3179	Unassigned
mc-brk-srv	3180/tcp	Millicent Broker
mc-brk-srv	3180/udp	Millicent Broker
bmcpatrolagent	3181/tcp	BMC Patrol Agent
bmcpatrolagent	3181/udp	BMC Patrol Agent
bmcpatrolrnvu	3182/tcp	BMC Patrol
bmcpatrolrnvu	3182/udp	BMC Patrol
#	3183-3261	Unassigned
necp	3262/tcp	NECP
necp	3262/udp	NECP
#	3263	Unassigned
ccmail	3264/tcp	cc:mail/lotus
ccmail	3264/udp	cc:mail/lotus
altav-tunnel	3265/tcp	Altav Tunnel
altav-tunnel	3265/udp	Altav Tunnel
ns-cfg-server	3266/tcp	NS CFG Server
ns-cfg-server	3266/udp	NS CFG Server
ibm-dial-out	3267/tcp	IBM Dial Out
ibm-dial-out	3267/udp	IBM Dial Out
msft-gc	3268/tcp	Microsoft Global
msft-gc	3268/udp	Microsoft Global
msft-gc-ssl	3269/tcp	Microsoft Global
msft-gc-ssl	3269/udp	Microsoft Global
verismart	3270/tcp	Verismart
verismart	3270/udp	Verismart
csoft-prev	3271/tcp	CSoft Prev Port
csoft-prev	3271/udp	CSoft Prev Port
user-manager	3272/tcp	Fujitsu User Mgr
user-manager	3272/udp	Fujitsu User Mgr
sxmp	3273/tcp	SXMP
sxmp	3273/udp	SXMP
ordinox-server	3274/tcp	Ordinox Server
ordinox-server	3274/udp	Ordinox Server
samd	3275/tcp	SAMD
samd	3275/udp	SAMD
maxim-asics	3276/tcp	Maxim ASICs
maxim-asics	3276/udp	Maxim ASICs
awg-proxy	3277/tcp	AWG Proxy
awg-proxy	3277/udp	AWG Proxy
lkcmserver	3278/tcp	LKCM Server
lkcmserver	3278/udp	LKCM Server
admind	3279/tcp	admind
admind	3279/udp	admind
vs-server	3280/tcp	VS Server
vs-server	3280/udp	VS Server
sysopt	3281/tcp	SYSOPT
sysopt	3281/udp	SYSOPT
datusorb	3282/tcp	Datusorb
datusorb	3282/udp	Datusorb
net-assistant	3283/tcp	Net Assistant
net-assistant	3283/udp	Net Assistant
4talk	3284/tcp	4Talk
4talk	3284/udp	4Talk
plato	3285/tcp	Plato
plato	3285/udp	Plato

netview-aix-5	1665/udp	netview-aix-5
netview-aix-6	1666/tcp	netview-aix-6
netview-aix-6	1666/udp	netview-aix-6
netview-aix-7	1667/tcp	netview-aix-7
netview-aix-7	1667/udp	netview-aix-7
netview-aix-8	1668/tcp	netview-aix-8
netview-aix-8	1668/udp	netview-aix-8
netview-aix-9	1669/tcp	netview-aix-9
netview-aix-9	1669/udp	netview-aix-9
netview-aix-10	1670/tcp	netview-aix-10
netview-aix-10	1670/udp	netview-aix-10
netview-aix-11	1671/tcp	netview-aix-11
netview-aix-11	1671/udp	netview-aix-11
netview-aix-12	1672/tcp	netview-aix-12
netview-aix-12	1672/udp	netview-aix-12
proshare-mc-1	1673/tcp	Intel Proshare
proshare-mc-1	1673/udp	Intel Proshare
proshare-mc-2	1674/tcp	Intel Proshare
proshare-mc-2	1674/udp	Intel Proshare
pdp	1675/tcp	Pacific Data
pdp	1675/udp	Pacific Data
netcomm1	1676/tcp	netcomm1
netcomm2	1676/udp	netcomm2
groupwise	1677/tcp	groupwise
groupwise	1677/udp	groupwise
prolink	1678/tcp	prolink
prolink	1678/udp	prolink
darcorp-lm	1679/tcp	darcorp-lm
darcorp-lm	1679/udp	darcorp-lm
microcom-sbp	1680/tcp	microcom-sbp
microcom-sbp	1680/udp	microcom-sbp
sd-elmd	1681/tcp	sd-elmd
sd-elmd	1681/udp	sd-elmd
lanyon-lantern	1682/tcp	lanyon-lantern
lanyon-lantern	1682/udp	lanyon-lantern
ncpm-hip	1683/tcp	ncpm-hip
ncpm-hip	1683/udp	ncpm-hip
snaresecure	1684/tcp	SnareSecure
snaresecure	1684/udp	SnareSecure
n2nremote	1685/tcp	n2nremote
n2nremote	1685/udp	n2nremote
cvmon	1686/tcp	cvmon
cvmon	1686/udp	cvmon
nsjtp-ctrl	1687/tcp	nsjtp-ctrl
nsjtp-ctrl	1687/udp	nsjtp-ctrl
nsjtp-data	1688/tcp	nsjtp-data
nsjtp-data	1688/udp	nsjtp-data
firefox	1689/tcp	firefox
firefox	1689/udp	firefox
ng-umds	1690/tcp	ng-umds
ng-umds	1690/udp	ng-umds
empire-empuma	1691/tcp	empire-empuma
empire-empuma	1691/udp	empire-empuma
sstsys-lm	1692/tcp	sstsys-lm
sstsys-lm	1692/udp	sstsys-lm
rrirtr	1693/tcp	rrirtr
rrirtr	1693/udp	rrirtr
rrimwm	1694/tcp	rrimwm
rrimwm	1694/udp	rrimwm
rrilwm	1695/tcp	rrilwm
rrilwm	1695/udp	rrilwm
rrifmm	1696/tcp	rrifmm
rrifmm	1696/udp	rrifmm
rrisat	1697/tcp	rrisat
rrisat	1697/udp	rrisat
rsvp-encap-1	1698/tcp	ENCAPSULATION-1
rsvp-encap-1	1698/udp	ENCAPSULATION-1
rsvp-encap-2	1699/tcp	ENCAPSULATION-2
rsvp-encap-2	1699/udp	ENCAPSULATION-2
mps-raft	1700/tcp	mps-raft
mps-raft	1700/udp	mps-raft
e-net	3286/tcp	E-Net
e-net	3286/udp	E-Net
directvdata	3287/tcp	DIRECTVDATA
directvdata	3287/udp	DIRECTVDATA
cops	3288/tcp	COPS
cops	3288/udp	COPS
enpc	3289/tcp	ENPC
enpc	3289/udp	ENPC
caps-lm	3290/tcp	CAPS LOGISTICS
caps-lm	3290/udp	CAPS LOGISTICS
sah-lm	3291/tcp	S A Holditch &
sah-lm	3291/udp	S A Holditch &
cart-o-rama	3292/tcp	Cart O Rama
cart-o-rama	3292/udp	Cart O Rama
fg-fps	3293/tcp	fg-fps
fg-fps	3293/udp	fg-fps
fg-gip	3294/tcp	fg-gip
fg-gip	3294/udp	fg-gip
dyniplookup	3295/tcp	Dynamic IP
dyniplookup	3295/udp	Dynamic IP
rib-slm	3296/tcp	Rib License Mgr
rib-slm	3296/udp	Rib License Mgr
cytel-lm	3297/tcp	Cytel Mgr
cytel-lm	3297/udp	Cytel Mgr
transview	3298/tcp	Transview
transview	3298/udp	Transview
pdrncs	3299/tcp	pdrncs
pdrncs	3299/udp	pdrncs
mcs-fastmail	3302/tcp	MCS Fastmail
mcs-fastmail	3302/udp	MCS Fastmail
opsession-clnt	3303/tcp	OP Session Clt
opsession-clnt	3303/udp	OP Session Clt
opsession-srvr	3304/tcp	OP Session Serv
opsession-srvr	3304/udp	OP Session Serv
odette-ftp	3305/tcp	ODETTE-FTP
odette-ftp	3305/udp	ODETTE-FTP
mysql	3306/tcp	MySQL
mysql	3306/udp	MySQL
opsession-prxy	3307/tcp	OP Session Proxy
opsession-prxy	3307/udp	OP Session Proxy
tns-server	3308/tcp	TNS Server
tns-server	3308/udp	TNS Server
tns-adv	3309/tcp	TNS ADV
tns-adv	3309/udp	TND ADV
dyna-access	3310/tcp	Dyna Access
dyna-access	3310/udp	Dyna Access
mcns-tel-ret	3311/tcp	MCNS Tel Ret
mcns-tel-ret	3311/udp	MCNS Tel Ret
appman-server	3312/tcp	Application
appman-server	3312/udp	Application
uorb	3313/tcp	Unify Object
uorb	3313/udp	Unify Object
uohost	3314/tcp	Unify Object
uohost	3314/udp	Unify Object
cdid	3315/tcp	CDID
cdid	3315/udp	CDID
aicc-cmi	3316/tcp	AICC/CMI
aicc-cmi	3316/udp	AICC/CMI
vsaiport	3317/tcp	VSAI PORT
vsaiport	3317/udp	VSAI PORT
ssrip	3318/tcp	Swith to Swith
ssrip	3318/udp	Swith to Swith
sdt-lmd	3319/tcp	SDT License Mgr
sdt-lmd	3319/udp	SDT License Mgr
officelink2000	3320/tcp	Office Link 2000
officelink2000	3320/udp	Office Link 2000
vnsstr	3321/tcp	VNSSTR
vnsstr	3321/udp	VNSSTR
active-net	3322-3325	Active Networks
sftu	3326/tcp	SFTU
sftu	3326/udp	SFTU

l2f	1701/tcp	l2f
l2f	1701/udp	l2f
l2tp	1701/tcp	l2tp
l2tp	1701/udp	l2tp
deskshare	1702/tcp	deskshare
deskshare	1702/udp	deskshare
bcs-broker	1704/tcp	bcs-broker
bcs-broker	1704/udp	bcs-broker
slingshot	1705/tcp	slingshot
slingshot	1705/udp	slingshot
jetform	1706/tcp	jetform
jetform	1706/udp	jetform
vdmplay	1707/tcp	vdmplay
vdmplay	1707/udp	vdmplay
gat-lmd	1708/tcp	gat-lmd
gat-lmd	1708/udp	gat-lmd
centra	1709/tcp	centra
centra	1709/udp	centra
impera	1710/tcp	impera
impera	1710/udp	impera
pptconference	1711/tcp	pptconference
pptconference	1711/udp	pptconference
registrar	1712/tcp	resource mon
registrar	1712/udp	resource mon
conferencetalk	1713/tcp	ConferenceTalk
conferencetalk	1713/udp	ConferenceTalk
sesi-lm	1714/tcp	sesi-lm
sesi-lm	1714/udp	sesi-lm
houdini-lm	1715/tcp	houdini-lm
houdini-lm	1715/udp	houdini-lm
xmsg	1716/tcp	xmsg
xmsg	1716/udp	xmsg
fj-hdnet	1717/tcp	fj-hdnet
fj-hdnet	1717/udp	fj-hdnet
h323gatedisc	1718/tcp	h323gatedisc
h323gatedisc	1718/udp	h323gatedisc
h323gatestat	1719/tcp	h323gatestat
h323gatestat	1719/udp	h323gatestat
h323hostcall	1720/tcp	h323hostcall
h323hostcall	1720/udp	h323hostcall
caicci	1721/tcp	caicci
caicci	1721/udp	caicci
hks-lm	1722/tcp	HKS
hks-lm	1722/udp	HKS
pptp	1723/tcp	pptp
pptp	1723/udp	pptp
csbphonemaster	1724/tcp	csbphonemaster
csbphonemaster	1724/udp	csbphonemaster
iden-ralp	1725/tcp	iden-ralp
iden-ralp	1725/udp	iden-ralp
iberiagames	1726/tcp	IBERIAGAMES
iberiagames	1726/udp	IBERIAGAMES
winddx	1727/tcp	winddx
winddx	1727/udp	winddx
telindus	1728/tcp	TELINDUS
telindus	1728/udp	TELINDUS
roketz	1730/tcp	roketz
roketz	1730/udp	roketz
msiccp	1731/tcp	MSICCP
msiccp	1731/udp	MSICCP
proxim	1732/tcp	proxim
proxim	1732/udp	proxim
siipat	1733/tcp	SIMS
siipat	1733/udp	SIMS
cambertx-lm	1734/tcp	Camber
cambertx-lm	1734/udp	Camber
privatechat	1735/tcp	PrivateChat
privatechat	1735/udp	PrivateChat
street-stream	1736/tcp	street-stream
street-stream	1736/udp	street-stream
ultimad	1737/tcp	ultimad
bbars	3327/tcp	BBARS
bbars	3327/udp	BBARS
egptlm	3328/tcp	Eaglepoint
egptlm	3328/udp	Eaglepoint
hp-device-disc	3329/tcp	HP Device Disc
hp-device-disc	3329/udp	HP Device Disc
mcs-calypsoicf	3330/tcp	MCS Calypso ICF
mcs-calypsoicf	3330/udp	MCS Calypso ICF
mcs-messaging	3331/tcp	MCS Messaging
mcs-messaging	3331/udp	MCS Messaging
mcs-mailsvr	3332/tcp	MCS Mail Server
mcs-mailsvr	3332/udp	MCS Mail Server
dec-notes	3333/tcp	DEC Notes
dec-notes	3333/udp	DEC Notes
directv-web	3334/tcp	Direct TV
directv-web	3334/udp	Direct TV
directv-soft	3335/tcp	Direct TV
directv-soft	3335/udp	Direct TV
directv-tick	3336/tcp	Direct TV
directv-tick	3336/udp	Direct TV
directv-catlg	3337/tcp	Direct TV Data
directv-catlg	3337/udp	Direct TV Data
anet-b	3338/tcp	OMF data b
anet-b	3338/udp	OMF data b
anet-l	3339/tcp	OMF data l
anet-l	3339/udp	OMF data l
anet-m	3340/tcp	OMF data m
anet-m	3340/udp	OMF data m
anet-h	3341/tcp	OMF data h
anet-h	3341/udp	OMF data h
webtie	3342/tcp	WebTIE
webtie	3342/udp	WebTIE
ms-cluster-net	3343/tcp	MS Cluster Net
ms-cluster-net	3343/udp	MS Cluster Net
bnt-manager	3344/tcp	BNT Manager
bnt-manager	3344/udp	BNT Manager
influence	3345/tcp	Influence
influence	3345/udp	Influence
trnsprntproxy	3346/tcp	Trnsprnt Proxy
trnsprntproxy	3346/udp	Trnsprnt Proxy
phoenix-rpc	3347/tcp	Phoenix RPC
phoenix-rpc	3347/udp	Phoenix RPC
pangolin-laser	3348/tcp	Pangolin Laser
pangolin-laser	3348/udp	Pangolin Laser
chevinservices	3349/tcp	Chevin Services
chevinservices	3349/udp	Chevin Services
findviatv	3350/tcp	FINDVIATV
findviatv	3350/udp	FINDVIATV
btrieve	3351/tcp	BTRIEVE
btrieve	3351/udp	BTRIEVE
ssql	3352/tcp	SSQL
ssql	3352/udp	SSQL
fatpipe	3353/tcp	FATPIPE
fatpipe	3353/udp	FATPIPE
suitjd	3354/tcp	SUITJD
suitjd	3354/udp	SUITJD
ordinox-dbase	3355/tcp	Ordinox Dbase
ordinox-dbase	3355/udp	Ordinox Dbase
upnotifyps	3356/tcp	UPNOTIFYPS
upnotifyps	3356/udp	UPNOTIFYPS
adtech-test	3357/tcp	Adtech Test IP
adtech-test	3357/udp	Adtech Test IP
mpsysrmsvr	3358/tcp	Mp Sys Rmsvr
mpsysrmsvr	3358/udp	Mp Sys Rmsvr
wg-netforce	3359/tcp	WG NetForce
wg-netforce	3359/udp	WG NetForce
kv-server	3360/tcp	KV Server
kv-server	3360/udp	KV Server
kv-agent	3361/tcp	KV Agent
kv-agent	3361/udp	KV Agent
dj-ilm	3362/tcp	DJ ILM

ultimad	1737/udp	ultimad
gamegen1	1738/tcp	GameGen1
gamegen1	1738/udp	GameGen1
webaccess	1739/tcp	webaccess
webaccess	1739/udp	webaccess
encore	1740/tcp	encore
encore	1740/udp	encore
cisco-net-mgmt	1741/tcp	cisco-net-mgmt
cisco-net-mgmt	1741/udp	cisco-net-mgmt
3Com-nsd	1742/tcp	3Com-nsd
3Com-nsd	1742/udp	3Com-nsd
cinegrfx-lm	1743/tcp	Cinema Graphics
cinegrfx-lm	1743/udp	Cinema Graphics
ncpm-ft	1744/tcp	ncpm-ft
ncpm-ft	1744/udp	ncpm-ft
remote-winsock	1745/tcp	remote-winsock
remote-winsock	1745/udp	remote-winsock
ftrapid-1	1746/tcp	ftrapid-1
ftrapid-1	1746/udp	ftrapid-1
ftrapid-2	1747/tcp	ftrapid-2
ftrapid-2	1747/udp	ftrapid-2
oracle-em1	1748/tcp	oracle-em1
oracle-em1	1748/udp	oracle-em1
aspen-services	1749/tcp	aspen-services
aspen-services	1749/udp	aspen-services
sslp	1750/tcp	Simple Socket
sslp	1750/udp	Simple Socket
swiftnet	1751/tcp	SwiftNet
swiftnet	1751/udp	SwiftNet
lofr-lm	1752/tcp	Leap of Faith
lofr-lm	1752/udp	Leap of Faith
translogic-lm	1753/tcp	Translogic
translogic-lm	1753/udp	Translogic
oracle-em2	1754/tcp	oracle-em2
oracle-em2	1754/udp	oracle-em2
ms-streaming	1755/tcp	ms-streaming
ms-streaming	1755/udp	ms-streaming
capfast-lmd	1756/tcp	capfast-lmd
capfast-lmd	1756/udp	capfast-lmd
cnhrp	1757/tcp	cnhrp
cnhrp	1757/udp	cnhrp
tftp-mcast	1758/tcp	tftp-mcast
tftp-mcast	1758/udp	tftp-mcast
spss-lm	1759/tcp	SPSS
spss-lm	1759/udp	SPSS
www-ldap-gw	1760/tcp	www-ldap-gw
www-ldap-gw	1760/udp	www-ldap-gw
cft-0	1761/tcp	cft-0
cft-0	1761/udp	cft-0
cft-1	1762/tcp	cft-1
cft-1	1762/udp	cft-1
cft-2	1763/tcp	cft-2
cft-2	1763/udp	cft-2
cft-3	1764/tcp	cft-3
cft-3	1764/udp	cft-3
cft-4	1765/tcp	cft-4
cft-4	1765/udp	cft-4
cft-5	1766/tcp	cft-5
cft-5	1766/udp	cft-5
cft-6	1767/tcp	cft-6
cft-6	1767/udp	cft-6
cft-7	1768/tcp	cft-7
cft-7	1768/udp	cft-7
bmc-net-adm	1769/tcp	bmc-net-adm
bmc-net-adm	1769/udp	bmc-net-adm
bmc-net-svc	1770/tcp	bmc-net-svc
bmc-net-svc	1770/udp	bmc-net-svc
vaultbase	1771/tcp	vaultbase
vaultbase	1771/udp	vaultbase
essweb-gw	1772/tcp	EssWeb Gateway
essweb-gw	1772/udp	EssWeb Gateway
dj-ilm	3362/udp	DJ ILM
nati-vi-server	3363/tcp	NATI Vi Server
nati-vi-server	3363/udp	NATI Vi Server
creativeserver	3364/tcp	Creative Server
creativeserver	3364/udp	Creative Server
contentserver	3365/tcp	Content Server
contentserver	3365/udp	Content Server
creativepartnr	3366/tcp	Creative Partner
creativepartnr	3366/udp	Creative Partner
satvid-datalnk	3367-3371	Satellite Video
tip2	3372/tcp	TIP 2
tip2	3372/udp	TIP 2
lavenir-lm	3373/tcp	Lavenir
lavenir-lm	3373/udp	Lavenir
cluster-disc	3374/tcp	Cluster Disc
cluster-disc	3374/udp	Cluster Disc
vsnm-agent	3375/tcp	VSNM Agent
vsnm-agent	3375/udp	VSNM Agent
cdborker	3376/tcp	CD Broker
cdbroker	3376/udp	CD Broker
cogsys-lm	3377/tcp	Cogsys Network
cogsys-lm	3377/udp	Cogsys Network
wsicopy	3378/tcp	WSICOPY
wsicopy	3378/udp	WSICOPY
socorfs	3379/tcp	SOCORFS
socorfs	3379/udp	SOCORFS
sns-channels	3380/tcp	SNS Channels
sns-channels	3380/udp	SNS Channels
geneous	3381/tcp	Geneous
geneous	3381/udp	Geneous
fujitsu-neat	3382/tcp	Fujitsu Network
fujitsu-neat	3382/udp	Fujitsu Network
esp-lm	3383/tcp	Enterprise
esp-lm	3383/udp	Enterprise
hp-clic	3384/tcp	Cluster
hp-clic	3384/udp	Hardware
qnxnetman	3385/tcp	qnxnetman
qnxnetman	3385/udp	qnxnetman
gprs-data	3386/tcp	GPRS Data
gprs-sig	3386/udp	GPRS SIG
backroomnet	3387/tcp	Back Room Net
backroomnet	3387/udp	Back Room Net
cbserver	3388/tcp	CB Server
cbserver	3388/udp	CB Server
ms-wbt-server	3389/tcp	MS WBT Server
ms-wbt-server	3389/udp	MS WBT Server
dsc	3390/tcp	Distributed
dsc	3390/udp	Distributed
savant	3391/tcp	SAVANT
savant	3391/udp	SAVANT
efi-lm	3392/tcp	EFI License
efi-lm	3392/udp	EFI License
d2k-tapestry1	3393/tcp	D2K Tapestry
d2k-tapestry1	3393/udp	D2K Tapestry
d2k-tapestry2	3394/tcp	D2K Tapestry
d2k-tapestry2	3394/udp	D2K Tapestry
dyna-lm	3395/tcp	Dyna (Elam)
dyna-lm	3395/udp	Dyna (Elam)
printer_agent	3396/tcp	Printer Agent
printer_agent	3396/udp	Printer Agent
cloanto-lm	3397/tcp	Cloanto
cloanto-lm	3397/udp	Cloanto
mercantile	3398/tcp	Mercantile
mercantile	3398/udp	Mercantile
csms	3399/tcp	CSMS
csms	3399/udp	CSMS
csms2	3400/tcp	CSMS2
csms2	3400/udp	CSMS2
filecast	3401/tcp	filecast
filecast	3401/udp	filecast
#	3402-3420	Unassigned

kmscontrol	1773/tcp	KMSControl
kmscontrol	1773/udp	KMSControl
global-dtserv	1774/tcp	global-dtserv
global-dtserv	1774/udp	global-dtserv
#	1775/tcp	
femis	1776/tcp	F E M I S
femis	1776/udp	F E M I S
powerguardian	1777/tcp	powerguardian
powerguardian	1777/udp	powerguardian
prodigy-intrnet	1778/tcp	prodigy
prodigy-intrnet	1778/udp	prodigy
pharmasoft	1779/tcp	pharmasoft
pharmasoft	1779/udp	pharmasoft
dpkeyserv	1780/tcp	dpkeyserv
dpkeyserv	1780/udp	dpkeyserv
answersoft-lm	1781/tcp	answersoft-lm
answersoft-lm	1781/udp	answersoft-lm
hp-hcip	1782/tcp	hp-hcip
hp-hcip	1782/udp	hp-hcip
#	1783	Decomissioned P
finle-lm	1784/tcp	Finle
finle-lm	1784/udp	Finle
windlm	1785/tcp	Wind River
windlm	1785/udp	Wind River
funk-logger	1786/tcp	funk-logger
funk-logger	1786/udp	funk-logger
funk-license	1787/tcp	funk-license
funk-license	1787/udp	funk-license
psmond	1788/tcp	psmond
psmond	1788/udp	psmond
hello	1789/tcp	hello
hello	1789/udp	hello
nmsp	1790/tcp	Narrative Media
nmsp	1790/udp	Narrative Media
ea1	1791/tcp	EA1
ea1	1791/udp	EA1
ibm-dt-2	1792/tcp	ibm-dt-2
ibm-dt-2	1792/udp	ibm-dt-2
rsc-robot	1793/tcp	rsc-robot
rsc-robot	1793/udp	rsc-robot
cera-bcm	1794/tcp	cera-bcm
cera-bcm	1794/udp	cera-bcm
dpi-proxy	1795/tcp	dpi-proxy
dpi-proxy	1795/udp	dpi-proxy
vocaltec-admin	1796/tcp	Vocaltec Server
vocaltec-admin	1796/udp	Vocaltec Server
uma	1797/tcp	UMA
uma	1797/udp	UMA
etp	1798/tcp	Event Transfer
etp	1798/udp	Event Transfer
netrisk	1799/tcp	NETRISK
netrisk	1799/udp	NETRISK
ansys-lm	1800/tcp	ANSYS
ansys-lm	1800/udp	ANSYS
msmq	1801/tcp	MS Message Que
msmq	1801/udp	MS Message Que
concomp1	1802/tcp	ConComp1
concomp1	1802/udp	ConComp1
hp-hcip-gwy	1803/tcp	HP-HCIP-GWY
hp-hcip-gwy	1803/udp	HP-HCIP-GWY
enl	1804/tcp	ENL
enl	1804/udp	ENL
enl-name	1805/tcp	ENL-Name
enl-name	1805/udp	ENL-Name
musiconline	1806/tcp	Musiconline
musiconline	1806/udp	Musiconline
fhsp	1807/tcp	Fujitsu Hot
fhsp	1807/udp	Fujitsu Hot
oracle-vp2	1808/tcp	Oracle-VP2
oracle-vp2	1808/udp	Oracle-VP2
oracle-vp1	1809/tcp	Oracle-VP1
bmap	3421/tcp	Bull Apprise
bmap	3421/udp	Bull Apprise
#	3422-3453	Unassigned
mira	3454/tcp	Apple Remote
prsvp	3455/tcp	RSVP Port
prsvp	3455/udp	RSVP Port
vat	3456/tcp	VAT default data
vat	3456/udp	VAT default data
vat-control	3457/tcp	VAT default Ctrl
vat-control	3457/udp	VAT default Ctrl
d3winosfi	3458/tcp	D3WinOsfi
d3winosfi	3458/udp	DsWinOSFI
integral	3459/tcp	TIP Integral
integral	3459/udp	TIP Integral
edm-manager	3460/tcp	EDM Manger
edm-manager	3460/udp	EDM Manger
edm-stager	3461/tcp	EDM Stager
edm-stager	3461/udp	EDM Stager
edm-std-notify	3462/tcp	EDM STD Notify
edm-std-notify	3462/udp	EDM STD Notify
edm-adm-notify	3463/tcp	EDM ADM Notify
edm-adm-notify	3463/udp	EDM ADM Notify
edm-mgr-sync	3464/tcp	EDM MGR Sync
edm-mgr-sync	3464/udp	EDM MGR Sync
edm-mgr-cntrl	3465/tcp	EDM MGR Cntrl
edm-mgr-cntrl	3465/udp	EDM MGR Cntrl
workflow	3466/tcp	WORKFLOW
workflow	3466/udp	WORKFLOW
rcst	3467/tcp	RCST
rcst	3467/udp	RCST
ttcmremotectrl	3468/tcp	TTCM Remote Ctrl
ttcmremotectrl	3468/udp	TTCM Remote Ctrl
pluribus	3469/tcp	Pluribus
pluribus	3469/udp	Pluribus
jt400	3470/tcp	jt400
jt400	3470/udp	jt400
jt400-ssl	3471/tcp	jt400-ssl
jt400-ssl	3471/udp	jt400-ssl
#	3472-3534	Unassigned
ms-la	3535/tcp	MS-LA
ms-la	3535/udp	MS-LA
#	3536-3562	Unassigned
watcomdebug	3563/tcp	Watcom Debug
watcomdebug	3563/udp	Watcom Debug
#	3564-3671	Unassigned
harlequinorb	3672/tcp	harlequinorb
harlequinorb	3672/udp	harlequinorb
#	3673-3801	Unassigned
vhd	3802/tcp	VHD
vhd	3802/udp	VHD
#	3803-3844	Unassigned
v-one-spp	3845/tcp	V-ONE Single
v-one-spp	3845/udp	V-ONE Single
#	3846-3861	Unassigned
giga-pocket	3862/tcp	GIGA-POCKET
giga-pocket	3862/udp	GIGA-POCKET
#	3863-3874	Unassigned
pnbscada	3875/tcp	PNBSCADA
pnbscada	3875/udp	PNBSCADA
#	3876-3899	Unassigned
udt_os	3900/tcp	Unidata UDT OS
udt_os	3900/udp	Unidata UDT OS
#	3901-3983	Unassigned
mapper-nodemgr	3984/tcp	MAPPER network
mapper-nodemgr	3984/udp	MAPPER network
mapper-mapethd	3985/tcp	MAPPER TCP/IP
mapper-mapethd	3985/udp	MAPPER TCP/IP
mapper-ws_ethd	3986/tcp	MAPPER
mapper-ws_ethd	3986/udp	MAPPER
centerline	3987/tcp	Centerline
centerline	3987/udp	Centerline

oracle-vp1	1809/udp	Oracle-VP1
jerand-lm	1810/tcp	Jerand
jerand-lm	1810/udp	Jerand
scientia-sdb	1811/tcp	Scientia-SDB
scientia-sdb	1811/udp	Scientia-SDB
radius	1812/tcp	RADIUS
radius	1812/udp	RADIUS
radius-acct	1813/tcp	RADIUS Acc
radius-acct	1813/udp	RADIUS Acc
tdp-suite	1814/tcp	TDP Suite
tdp-suite	1814/udp	TDP Suite
mmpft	1815/tcp	MMPFT
mmpft	1815/udp	MMPFT
harp	1816/tcp	HARP
harp	1816/udp	HARP
rkb-oscs	1817/tcp	RKB-OSCS
rkb-oscs	1817/udp	RKB-OSCS
etftp	1818/tcp	Enhanced TFTP
etftp	1818/udp	Enhanced TFTP
plato-lm	1819/tcp	Plato
plato-lm	1819/udp	Plato
mcagent	1820/tcp	mcagent
mcagent	1820/udp	mcagent
donnyworld	1821/tcp	donnyworld
donnyworld	1821/udp	donnyworld
es-elmd	1822/tcp	es-elmd
es-elmd	1822/udp	es-elmd
unisys-lm	1823/tcp	Unisys
unisys-lm	1823/udp	Unisys
metrics-pas	1824/tcp	metrics-pas
metrics-pas	1824/udp	metrics-pas
direcpc-video	1825/tcp	DirecPC Video
direcpc-video	1825/udp	DirecPC Video
ardt	1826/tcp	ARDT
ardt	1826/udp	ARDT
asi	1827/tcp	ASI
asi	1827/udp	ASI
itm-mcell-u	1828/tcp	itm-mcell-u
itm-mcell-u	1828/udp	itm-mcell-u
optika-emedia	1829/tcp	Optika eMedia
optika-emedia	1829/udp	Optika eMedia
net8-cman	1830/tcp	Oracle Net8
net8-cman	1830/udp	Oracle Net8
myrtle	1831/tcp	Myrtle
myrtle	1831/udp	Myrtle
tht-treasure	1832/tcp	ThoughtTreasure
tht-treasure	1832/udp	ThoughtTreasure
udpradio	1833/tcp	udpradio
udpradio	1833/udp	udpradio
ardusuni	1834/tcp	ARDUS Unicast
ardusuni	1834/udp	ARDUS Unicast
ardusmul	1835/tcp	ARDUS Multicast
ardusmul	1835/udp	ARDUS Multicast
ste-smsc	1836/tcp	ste-smsc
ste-smsc	1836/udp	ste-smsc
csoft1	1837/tcp	csoft1
csoft1	1837/udp	csoft1
talnet	1838/tcp	TALNET
talnet	1838/udp	TALNET
netopia-vo1	1839/tcp	netopia-vo1
netopia-vo1	1839/udp	netopia-vo1
netopia-vo2	1840/tcp	netopia-vo2
netopia-vo2	1840/udp	netopia-vo2
netopia-vo3	1841/tcp	netopia-vo3
netopia-vo3	1841/udp	netopia-vo3
netopia-vo4	1842/tcp	netopia-vo4
netopia-vo4	1842/udp	netopia-vo4
netopia-vo5	1843/tcp	netopia-vo5
netopia-vo5	1843/udp	netopia-vo5
direcpc-dll	1844/tcp	DirecPC-DLL
direcpc-dll	1844/udp	DirecPC-DLL
#	3988-3999	Unassigned
terabase	4000/tcp	Terabase
terabase	4000/udp	Terabase
newoak	4001/tcp	NewOak
newoak	4001/udp	NewOak
pxc-spvr-ft	4002/tcp	pxc-spvr-ft
pxc-spvr-ft	4002/udp	pxc-spvr-ft
pxc-splr-ft	4003/tcp	pxc-splr-ft
pxc-splr-ft	4003/udp	pxc-splr-ft
pxc-roid	4004/tcp	pxc-roid
pxc-roid	4004/udp	pxc-roid
pxc-pin	4005/tcp	pxc-pin
pxc-pin	4005/udp	pxc-pin
pxc-spvr	4006/tcp	pxc-spvr
pxc-spvr	4006/udp	pxc-spvr
pxc-splr	4007/tcp	pxc-splr
pxc-splr	4007/udp	pxc-splr
netcheque	4008/tcp	NetCheque acc
netcheque	4008/udp	NetCheque acc
chimera-hwm	4009/tcp	Chimera HWM
chimera-hwm	4009/udp	Chimera HWM
samsung-unidex	4010/tcp	Samsung Unidex
samsung-unidex	4010/udp	Samsung Unidex
altserviceboot	4011/tcp	Alternate Boot
altserviceboot	4011/udp	Alternate Boot
pda-gate	4012/tcp	PDA Gate
pda-gate	4012/udp	PDA Gate
acl-manager	4013/tcp	ACL Manager
acl-manager	4013/udp	ACL Manager
taiclock	4014/tcp	TAICLOCK
taiclock	4014/udp	TAICLOCK
talarian-mcast1	4015/tcp	Talarian Mcast
talarian-mcast1	4015/udp	Talarian Mcast
talarian-mcast2	4016/tcp	Talarian Mcast
talarian-mcast2	4016/udp	Talarian Mcast
talarian-mcast3	4017/tcp	Talarian Mcast
talarian-mcast3	4017/udp	Talarian Mcast
talarian-mcast4	4018/tcp	Talarian Mcast
talarian-mcast4	4018/udp	Talarian Mcast
talarian-mcast5	4019/tcp	Talarian Mcast
talarian-mcast5	4019/udp	Talarian Mcast
#	4020-4095	Unassigned
bre	4096/tcp	BRE
bre	4096/udp	BRE
patrolview	4097/tcp	Patrol View
patrolview	4097/udp	Patrol View
drmsfsd	4098/tcp	drmsfsd
drmsfsd	4098/udp	drmsfsd
dpcp	4099/tcp	DPCP
dpcp	4099/udp	DPCP
#	4100-4131	Unassigned
nuts_dem	4132/tcp	NUTS Daemon
nuts_dem	4132/udp	NUTS Daemon
nuts_bootp	4133/tcp	NUTS Bootp Serv
nuts_bootp	4133/udp	NUTS Bootp Serv
nifty-hmi	4134/tcp	NIFTY-Serve HMI
nifty-hmi	4134/udp	NIFTY-Serve HMI
oirtgsvc	4141/tcp	Workflow Server
oirtgsvc	4141/udp	Workflow Server
oidocsvc	4142/tcp	Document Server
oidocsvc	4142/udp	Document Server
oidsr	4143/tcp	Document Replic
oidsr	4143/udp	Document Replic
#	4144-4159	Unassigned
jini-discovery	4160/tcp	Jini Discovery
jini-discovery	4160/udp	Jini Discovery
#	4161-4198	Unassigned
eims-admin	4199/tcp	EIMS ADMIN
eims-admin	4199/udp	EIMS ADMIN
vrml-multi-use	4200-4299	VRML Multi
corelccam	4300/tcp	Corel CCam

#	1845-1849	Unassigned
gsi	1850/tcp	GSI
gsi	1850/udp	GSI
ctcd	1851/tcp	ctcd
ctcd	1851/udp	ctcd
#	1852-1859	Unassigned
sunscalar-svc	1860/tcp	SunSCALAR
sunscalar-svc	1860/udp	SunSCALAR
lecroy-vicp	1861/tcp	LeCroy VICP
lecroy-vicp	1861/udp	LeCroy VICP
techra-server	1862/tcp	techra-server
techra-server	1862/udp	techra-server
msnp	1863/tcp	MSNP
msnp	1863/udp	MSNP
paradym-31port	1864/tcp	Paradym 31 Port
paradym-31port	1864/udp	Paradym 31 Port
entp	1865/tcp	ENTP
entp	1865/udp	ENTP
#	1866-1869	Unassigned
sunscalar-dns	1870/tcp	SunSCALAR DNS
sunscalar-dns	1870/udp	SunSCALAR DNS
canocentral0	1871/tcp	Cano Central 0
canocentral0	1871/udp	Cano Central 0
canocentral1	1872/tcp	Cano Central 1
canocentral1	1872/udp	Cano Central 1
fjmpjps	1873/tcp	Fjmpjps
fjmpjps	1873/udp	Fjmpjps
fjswapsnp	1874/tcp	Fjswapsnp
fjswapsnp	1874/udp	Fjswapsnp
#	1875-1880	Unassigned
ibm-mqseries2	1881/tcp	IBM MQSeries
ibm-mqseries2	1881/udp	IBM MQSeries
#	1882-1894	Unassigned
vista-4gl	1895/tcp	Vista 4GL
vista-4gl	1895/udp	Vista 4GL
#	1896-1898	Unassigned
mc2studios	1899/tcp	MC2Studios
mc2studios	1899/udp	MC2Studio
ssdp	1900/tcp	SSDP
ssdp	1900/udp	SSDP
fjicl-tep-a	1901/tcp	Fujitsu ICL A
fjicl-tep-a	1901/udp	Fujitsu ICL A
fjicl-tep-b	1902/tcp	Fujitsu ICL B
fjicl-tep-b	1902/udp	Fujitsu ICL B
linkname	1903/tcp	Local Link Name
linkname	1903/udp	Local Link Name
fjicl-tep-c	1904/tcp	Fujitsu ICL C
fjicl-tep-c	1904/udp	Fujitsu ICL C
sugp	1905/tcp	Secure UP.Link
sugp	1905/udp	Secure UP.Link
tpmd	1906/tcp	TPortMapperReq
tpmd	1906/udp	TPortMapperReq
intrastar	1907/tcp	IntraSTAR
intrastar	1907/udp	IntraSTAR
dawn	1908/tcp	Dawn
dawn	1908/udp	Dawn
global-wlink	1909/tcp	Global World
global-wlink	1909/udp	Global World
ultrabac	1910/tcp	ultrabac
ultrabac	1910/udp	ultrabac
mtp	1911/tcp	Starlight
mtp	1911/udp	Starlight
rhp-iibp	1912/tcp	rhp-iibp
rhp-iibp	1912/udp	rhp-iibp
armadp	1913/tcp	armadp
armadp	1913/udp	armadp
elm-momentum	1914/tcp	Elm-Momentum
elm-momentum	1914/udp	Elm-Momentum
facelink	1915/tcp	FACELINK
facelink	1915/udp	FACELINK
persona	1916/tcp	Persoft Persona
corelccam	4300/udp	Corel CCam
#	4301-4320	Unassigned
rwhois	4321/tcp	Remote Who Is
rwhois	4321/udp	Remote Who Is
unicall	4343/tcp	UNICALL
unicall	4343/udp	UNICALL
vinainstall	4344/tcp	VinaInstall
vinainstall	4344/udp	VinaInstall
m4-network-as	4345/tcp	Macro 4 Network
m4-network-as	4345/udp	Macro 4 Network
elanlm	4346/tcp	ELAN LM
elanlm	4346/udp	ELAN LM
lansurveyor	4347/tcp	LAN Surveyor
lansurveyor	4347/udp	LAN Surveyor
itose	4348/tcp	ITOSE
itose	4348/udp	ITOSE
fsportmap	4349/tcp	FileSys Port Map
fsportmap	4349/udp	FileSys Port Map
net-device	4350/tcp	Net Device
net-device	4350/udp	Net Device
plcy-net-svcs	4351/tcp	PLCY Net Serv
plcy-net-svcs	4351/udp	PLCY Net Serv
#	4352	Unassigned
f5-iquery	4353/tcp	F5 iQuery
f5-iquery	4353/udp	F5 iQuery
#	4354-4443	Unassigned
saris	4442/tcp	Saris
saris	4442/udp	Saris
pharos	4443/tcp	Pharos
pharos	4443/udp	Pharos
krb524	4444/tcp	KRB524
krb524	4444/udp	KRB524
nv-video	4444/tcp	NV Video default
nv-video	4444/udp	NV Video default
upnotifyp	4445/tcp	UPNOTIFYP
upnotifyp	4445/udp	UPNOTIFYP
n1-fwp	4446/tcp	N1-FWP
n1-fwp	4446/udp	N1-FWP
n1-rmgmt	4447/tcp	N1-RMGMT
n1-rmgmt	4447/udp	N1-RMGMT
asc-slmd	4448/tcp	ASC Licence Mgr
asc-slmd	4448/udp	ASC Licence Mgr
privatewire	4449/tcp	PrivateWire
privatewire	4449/udp	PrivateWire
camp	4450/tcp	Camp
camp	4450/udp	Camp
ctisystemmsg	4451/tcp	CTI System Msg
ctisystemmsg	4451/udp	CTI System Msg
ctiprogramload	4452/tcp	CTI Program Load
ctiprogramload	4452/udp	CTI Program Load
nssalertmgr	4453/tcp	NSS Alert Mgr
nssalertmgr	4453/udp	NSS Alert Mgr
nssagentmgr	4454/tcp	NSS Agent Mgr
nssagentmgr	4454/udp	NSS Agent Mgr
prchat-user	4455/tcp	PR Chat User
prchat-user	4455/udp	PR Chat User
prchat-server	4456/tcp	PR Chat Server
prchat-server	4456/udp	PR Chat Server
prRegister	4457/tcp	PR Register
prRegister	4457/udp	PR Register
#	4458-4499	Unassigned
sae-urn	4500/tcp	sae-urn
sae-urn	4500/udp	sae-urn
urn-x-cdchoice	4501/tcp	urn-x-cdchoice
urn-x-cdchoice	4501/udp	urn-x-cdchoice
worldscores	4545/tcp	WorldScores
worldscores	4545/udp	WorldScores
sf-lm	4546/tcp	SF (Sentinel)
sf-lm	4546/udp	SF (Sentinel)
lanner-lm	4547/tcp	Lanner
lanner-lm	4547/udp	Lanner

persona	1916/udp	Persoft Persona
noagent	1917/tcp	nOAgent
noagent	1917/udp	nOAgent
can-nds	1918/tcp	Candle NDS
can-nds	1918/udp	Candle NDS
can-dch	1919/tcp	Candle DCH
can-dch	1919/udp	Candle DCH
can-ferret	1920/tcp	Candle FERRET
can-ferret	1920/udp	Candle FERRET
noadmin	1921/tcp	NoAdmin
noadmin	1921/udp	NoAdmin
tapestry	1922/tcp	Tapestry
tapestry	1922/udp	Tapestry
spice	1923/tcp	SPICE
spice	1923/udp	SPICE
xiip	1924/tcp	XIIP
xiip	1924/udp	XIIP
#	1925-1929	Unassigned
driveappserver	1930/tcp	Drive AppServer
driveappserver	1930/udp	Drive AppServer
amdsched	1931/tcp	AMD SCHED
amdsched	1931/udp	AMD SCHED
#	1932-1943	Unassigned
close-combat	1944/tcp	close-combat
close-combat	1944/udp	close-combat
dialogic-elmd	1945/tcp	dialogic-elmd
dialogic-elmd	1945/udp	dialogic-elmd
tekpls	1946/tcp	tekpls
tekpls	1946/udp	tekpls
hlserver	1947/tcp	hlserver
hlserver	1947/udp	hlserver
eye2eye	1948/tcp	eye2eye
eye2eye	1948/udp	eye2eye
ismaeasdaqlive	1949/tcp	ISMA Easdaq Live
ismaeasdaqlive	1949/udp	ISMA Easdaq Live
ismaeasdaqtest	1950/tcp	ISMA Easdaq Test
ismaeasdaqtest	1950/udp	ISMA Easdaq Test
bcs-lmserver	1951/tcp	bcs-lmserver
bcs-lmserver	1951/udp	bcs-lmserver
mpnjsc	1952/tcp	mpnjsc
mpnjsc	1952/udp	mpnjsc
rapidbase	1953/tcp	Rapid Base
rapidbase	1953/udp	Rapid Base
#	1954-1960	Unassigned
bts-appserver	1961/tcp	BTS APPSERVER
bts-appserver	1961/udp	BTS APPSERVER
biap-mp	1962/tcp	BIAP-MP
biap-mp	1962/udp	BIAP-MP
webmachine	1963/tcp	WebMachine
webmachine	1963/udp	WebMachine
solid-e-engine	1964/tcp	SOLID E ENGINE
solid-e-engine	1964/udp	SOLID E ENGINE
tivoli-npm	1965/tcp	Tivoli NPM
tivoli-npm	1965/udp	Tivoli NPM
slush	1966/tcp	Slush
slush	1966/udp	Slush
sns-quote	1967/tcp	SNS Quote
sns-quote	1967/udp	SNS Quote
#	1968-1971	Unassigned
intersys-cache	1972/tcp	Cache
intersys-cache	1972/udp	Cache
dlsrap	1973/tcp	Data Link
dlsrap	1973/udp	Data Link
drp	1974/tcp	DRP
drp	1974/udp	DRP
tcoflashagent	1975/tcp	TCO Flash Agent
tcoflashagent	1975/udp	TCO Flash Agent
tcoregagent	1976/tcp	TCO Reg Agent
tcoregagent	1976/udp	TCO Reg Agent
tcoaddressbook	1977/tcp	TCO Address Book
tcoaddressbook	1977/udp	TCO Address Book
#	4548-4566	Unassigned
tram	4567/tcp	TRAM
tram	4567/udp	TRAM
bmc-reporting	4568/tcp	BMC Reporting
bmc-reporting	4568/udp	BMC Reporting
#	4569-4599	Unassigned
piranha1	4600/tcp	Piranha1
piranha1	4600/udp	Piranha1
piranha2	4601/tcp	Piranha2
piranha2	4601/udp	Piranha2
#	4602-4671	Unassigned
rfa	4672/tcp	remote file acc
rfa	4672/udp	remote file acc
#	4673-4799	Unassigned
iims	4800/tcp	Icona Instant
iims	4800/udp	Icona Instant
iwec	4801/tcp	Icona Web
iwec	4801/udp	Icona Web
ilss	4802/tcp	Icona
ilss	4802/udp	Icona
#	4803-4826	Unassigned
htcp	4827/tcp	HTCP
htcp	4827/udp	HTCP
#	4828-4836	Unassigned
varadero-0	4837/tcp	Varadero-0
varadero-0	4837/udp	Varadero-0
varadero-1	4838/tcp	Varadero-1
varadero-1	4838/udp	Varadero-1
varadero-2	4839/udp	Varadero-2
varadero-2	4839/udp	Varadero-2
#	4840-4867	Unassigned
phrelay	4868/tcp	Photon Relay
phrelay	4868/udp	Photon Relay
phrelaydbg	4869/tcp	Photon Relay
phrelaydbg	4869/udp	Photon Relay
#	4870-4884	Unassigned
abbs	4885/tcp	ABBS
abbs	4885/udp	ABBS
#	4886-4982	Unassigned
att-intercom	4983/tcp	AT&T Intercom
att-intercom	4983/udp	AT&T Intercom
#	4984-4999	Unassigned
commplex-main	5000/tcp	
commplex-main	5000/udp	
commplex-link	5001/tcp	
commplex-link	5001/udp	
rfe	5002/tcp	radio free eth
rfe	5002/udp	radio free eth
fmpro-internal	5003/tcp	FileMaker, Inc.
fmpro-internal	5003/udp	FileMaker, Inc.
avt-profile-1	5004/tcp	avt-profile-1
avt-profile-1	5004/udp	avt-profile-1
avt-profile-2	5005/tcp	avt-profile-2
avt-profile-2	5005/udp	avt-profile-2
wsm-server	5006/tcp	wsm server
wsm-server	5006/udp	wsm server
wsm-server-ssl	5007/tcp	wsm server ssl
wsm-server-ssl	5007/udp	wsm server ssl
#	5008-5009	Unassigned
telelpathstart	5010/tcp	TelepathStart
telelpathstart	5010/udp	TelepathStart
telelpathattack	5011/tcp	TelepathAttack
telelpathattack	5011/udp	TelepathAttack
#	5012-5019	Unassigned
zenginkyo-1	5020/tcp	zenginkyo-1
zenginkyo-1	5020/udp	zenginkyo-1
zenginkyo-2	5021/tcp	zenginkyo-2
zenginkyo-2	5021/udp	zenginkyo-2
#	5022-5041	Unassigned
asnaacceler8db	5042/tcp	asnaacceler8db
asnaacceler8db	5042/udp	asnaacceler8db

```
unisql          1978/tcp   UniSQL
unisql          1978/udp   UniSQL
unisql-java     1979/tcp   UniSQL Java
unisql-java     1979/udp   UniSQL Java
#               1980-1983  Unassigned
bb              1984/tcp   BB
bb              1984/udp   BB
hsrp            1985/tcp   Hot Standby
hsrp            1985/udp   Hot Standby
licensedaemon   1986/tcp   cisco
licensedaemon   1986/udp   cisco
tr-rsrb-p1      1987/tcp   cisco RSRB
tr-rsrb-p1      1987/udp   cisco RSRB
tr-rsrb-p2      1988/tcp   cisco RSRB
tr-rsrb-p2      1988/udp   cisco RSRB
tr-rsrb-p3      1989/tcp   cisco RSRB
tr-rsrb-p3      1989/udp   cisco RSRB
mshnet          1989/tcp   MHSnet system
mshnet          1989/udp   MHSnet system
stun-p1         1990/tcp   cisco STUN 1
stun-p1         1990/udp   cisco STUN 1
stun-p2         1991/tcp   cisco STUN 2
stun-p2         1991/udp   cisco STUN 2
stun-p3         1992/tcp   cisco STUN 3
stun-p3         1992/udp   cisco STUN 3
ipsendmsg       1992/tcp   IPsendmsg
ipsendmsg       1992/udp   IPsendmsg
snmp-tcp-port   1993/tcp   cisco SNMP TCP
snmp-tcp-port   1993/udp   cisco SNMP TCP
stun-port       1994/tcp   cisco serial
stun-port       1994/udp   cisco serial
perf-port       1995/tcp   cisco perf port
perf-port       1995/udp   cisco perf port
tr-rsrb-port    1996/tcp   cisco Remote SRB
tr-rsrb-port    1996/udp   cisco Remote SRB
gdp-port        1997/tcp   cisco Gateway
gdp-port        1997/udp   cisco Gateway
x25-svc-port    1998/tcp   cisco X.25 (XOT)
x25-svc-port    1998/udp   cisco X.25 (XOT)
tcp-id-port     1999/tcp   cisco ident port
tcp-id-port     1999/udp   cisco ident port
callbook        2000/tcp
callbook        2000/udp
dc              2001/tcp
wizard          2001/udp   curry
globe           2002/tcp
globe           2002/udp
mailbox         2004/tcp
emce            2004/udp    CCWS mm conf
berknet         2005/tcp
oracle          2005/udp
invokator       2006/tcp
raid-cc         2006/udp    raid
dectalk         2007/tcp
raid-am         2007/udp
conf            2008/tcp
terminaldb      2008/udp
news            2009/tcp
whosockami      2009/udp
search          2010/tcp
pipe_server     2010/udp
raid-cc         2011/tcp    raid
servserv        2011/udp
ttyinfo         2012/tcp
raid-ac         2012/udp
raid-am         2013/tcp
raid-cd         2013/udp
troff           2014/tcp
raid-sf         2014/udp
cypress         2015/tcp
raid-cs         2015/udp
#               5043-5049  Unassigned
mmcc            5050/tcp   multimedia
mmcc            5050/udp   multimedia
ita-agent       5051/tcp   ITA Agent
ita-agent       5051/udp   ITA Agent
ita-manager     5052/tcp   ITA Manager
ita-manager     5052/udp   ITA Manager
#               5053-5054  Unassigned
unot            5055/tcp   UNOT
unot            5055/udp   UNOT
#               5056-5059  Unassigned
sip             5060/tcp   SIP
sip             5060/udp   SIP
#               5061-5068  Unassigned
i-net-2000-npr  5069/tcp   I/Net 2000-NPR
i-net-2000-npr  5069/udp   I/Net 2000-NPR
#               5070       Unassigned
powerschool     5071/tcp   PowerSchool
powerschool     5071/udp   PowerSchool
#               5072-5092  Unassigned
sentinel-lm     5093/tcp   Sentinel LM
sentinel-lm     5093/udp   Sentinel LM
#               5094-5098  Unassigned
sentlm-srv2srv  5099/tcp   SentLM Srv2Srv
sentlm-srv2srv  5099/udp   SentLM Srv2Srv
#               5100-5144  Unassigned
rmonitor_secure 5145/tcp   RMONITOR SECURE
rmonitor_secure 5145/udp   RMONITOR SECURE
#               5146-5149  Unassigned
atmp            5150/tcp   Ascend Tunnel
atmp            5150/udp   Ascend Tunnel
esri_sde        5151/tcp   ESRI SDE
esri_sde        5151/udp   ESRI SDE
sde-discovery   5152/tcp   ESRI SDE
sde-discovery   5152/udp   ESRI SDE
#               5153-5164  Unassigned
ife_icorp       5165/tcp   ife_1corp
ife_icorp       5165/udp   ife_1corp
#               5166-5189  Unassigned
aol             5190/tcp   America-Online
aol             5190/udp   America-Online
aol-1           5191/tcp   AmericaOnline1
aol-1           5191/udp   AmericaOnline1
aol-2           5192/tcp   AmericaOnline2
aol-2           5192/udp   AmericaOnline2
aol-3           5193/tcp   AmericaOnline3
aol-3           5193/udp   AmericaOnline3
#               5194-5199  Unassigned
targus-aib1     5200/tcp   Targus AIB 1
targus-aib1     5200/udp   Targus AIB 1
targus-aib2     5201/tcp   Targus AIB 2
targus-aib2     5201/udp   Targus AIB 2
targus-tnts1    5202/tcp   Targus TNTS 1
targus-tnts1    5202/udp   Targus TNTS 1
targus-tnts2    5203/tcp   Targus TNTS 2
targus-tnts2    5203/udp   Targus TNTS 2
#               5204-5235  Unassigned
padl2sim        5236/tcp
padl2sim        5236/udp
#               5237-5271  Unassigned
pk              5272/tcp   PK
pk              5272/udp   PK
#               5273-5299  Unassigned
hacl-hb         5300/tcp
hacl-hb         5300/udp
hacl-gs         5301/tcp
hacl-gs         5301/udp
hacl-cfg        5302/tcp
hacl-cfg        5302/udp
hacl-probe      5303/tcp
hacl-probe      5303/udp
```

bootserver	2016/tcp	
bootserver	2016/udp	
cypress-stat	2017/tcp	
bootclient	2017/udp	
terminaldb	2018/tcp	
rellpack	2018/udp	
whosockami	2019/tcp	
about	2019/udp	
xinupageserver	2020/tcp	
xinupageserver	2020/udp	
servexec	2021/tcp	
xinuexpansion1	2021/udp	
down	2022/tcp	
xinuexpansion2	2022/udp	
xinuexpansion3	2023/tcp	
xinuexpansion3	2023/udp	
xinuexpansion4	2024/tcp	
xinuexpansion4	2024/udp	
ellpack	2025/tcp	
xribs	2025/udp	
scrabble	2026/tcp	
scrabble	2026/udp	
shadowserver	2027/tcp	
shadowserver	2027/udp	
submitserver	2028/tcp	
submitserver	2028/udp	
device2	2030/tcp	
device2	2030/udp	
blackboard	2032/tcp	
blackboard	2032/udp	
glogger	2033/tcp	
glogger	2033/udp	
scoremgr	2034/tcp	
scoremgr	2034/udp	
imsldoc	2035/tcp	
imsldoc	2035/udp	
objectmanager	2038/tcp	
objectmanager	2038/udp	
lam	2040/tcp	
lam	2040/udp	
interbase	2041/tcp	
interbase	2041/udp	
isis	2042/tcp	isis
isis	2042/udp	isis
isis-bcast	2043/tcp	isis-bcast
isis-bcast	2043/udp	isis-bcast
rimsl	2044/tcp	
rimsl	2044/udp	
cdfunc	2045/tcp	
cdfunc	2045/udp	
sdfunc	2046/tcp	
sdfunc	2046/udp	
dls	2047/tcp	
dls	2047/udp	
dls-monitor	2048/tcp	
dls-monitor	2048/udp	
shilp	2049/tcp	
shilp	2049/udp	
nfs	2049/tcp	Network File Sys
nfs	2049/udp	Network File Sys
dlsrpn	2065/tcp	Data Link Switch
dlsrpn	2065/udp	Data Link Switch
dlswpn	2067/tcp	Data Link Switch
dlswpn	2067/udp	Data Link Switch
lrp	2090/tcp	Load Report
lrp	2090/udp	Load Report
prp	2091/tcp	PRP
prp	2091/udp	PRP
descent3	2092/tcp	Descent 3
descent3	2092/udp	Descent 3
nbx-cc	2093/tcp	NBX CC
hacl-local	5304/tcp	
hacl-local	5304/udp	
hacl-test	5305/tcp	
hacl-test	5305/udp	
sun-mc-grp	5306/tcp	Sun MC Group
sun-mc-grp	5306/udp	Sun MC Group
sco-aip	5307/tcp	SCO AIP
sco-aip	5307/udp	SCO AIP
cfengine	5308/tcp	CFengine
cfengine	5308/udp	CFengine
jprinter	5309/tcp	J Printer
jprinter	5309/udp	J Printer
outlaws	5310/tcp	Outlaws
outlaws	5310/udp	Outlaws
tmlogin	5311/tcp	TM Login
tmlogin	5311/udp	TM Login
#	5312-5399	Unassigned
excerpt	5400/tcp	Excerpt Search
excerpt	5400/udp	Excerpt Search
excerpts	5401/tcp	Excerpt Search
excerpts	5401/udp	Excerpt Search
mftp	5402/tcp	MFTP
mftp	5402/udp	MFTP
hpoms-ci-lstn	5403/tcp	HPOMS-CI-LSTN
hpoms-ci-lstn	5403/udp	HPOMS-CI-LSTN
hpoms-dps-lstn	5404/tcp	HPOMS-DPS-LSTN
hpoms-dps-lstn	5404/udp	HPOMS-DPS-LSTN
netsupport	5405/tcp	NetSupport
netsupport	5405/udp	NetSupport
systemics-sox	5406/tcp	Systemics Sox
systemics-sox	5406/udp	Systemics Sox
foresyte-clear	5407/tcp	Foresyte-Clear
foresyte-clear	5407/udp	Foresyte-Clear
foresyte-sec	5408/tcp	Foresyte-Sec
foresyte-sec	5408/udp	Foresyte-Sec
salient-dtasrv	5409/tcp	Salient Data
salient-dtasrv	5409/udp	Salient Data
salient-usrmgr	5410/tcp	Salient User Mgr
salient-usrmgr	5410/udp	Salient User Mgr
actnet	5411/tcp	ActNet
actnet	5411/udp	ActNet
continuus	5412/tcp	Continuus
continuus	5412/udp	Continuus
wwiotalk	5413/tcp	WWIOTALK
wwiotalk	5413/udp	WWIOTALK
statusd	5414/tcp	StatusD
statusd	5414/udp	StatusD
ns-server	5415/tcp	NS Server
ns-server	5415/udp	NS Server
sns-gateway	5416/tcp	SNS Gateway
sns-gateway	5416/udp	SNS Gateway
sns-agent	5417/tcp	SNS Agent
sns-agent	5417/udp	SNS Agent
mcntp	5418/tcp	MCNTP
mcntp	5418/udp	MCNTP
dj-ice	5419/tcp	DJ-ICE
dj-ice	5419/udp	DJ-ICE
cylink-c	5420/tcp	Cylink-C
cylink-c	5420/udp	Cylink-C
netsupport2	5421/tcp	Net Support 2
netsupport2	5421/udp	Net Support 2
salient-mux	5422/tcp	Salient MUX
salient-mux	5422/udp	Salient MUX
virtualuser	5423/tcp	VIRTUALUSER
virtualuser	5423/udp	VIRTUALUSER
#	5424-5425	Unassigned
devbasic	5426/tcp	DEVBASIC
devbasic	5426/udp	DEVBASIC
sco-peer-tta	5427/tcp	SCO-PEER-TTA
sco-peer-tta	5427/udp	SCO-PEER-TTA
telaconsole	5428/tcp	TELACONSOLE

nbx-cc	2093/udp	NBX CC
nbx-au	2094/tcp	NBX AU
nbx-au	2094/udp	NBX AU
nbx-ser	2095/tcp	NBX SER
nbx-ser	2095/udp	NBX SER
nbx-dir	2096/tcp	NBX DIR
nbx-dir	2096/udp	NBX DIR
jetformpreview	2097/tcp	Jet Form Preview
jetformpreview	2097/udp	Jet Form Preview
dialog-port	2098/tcp	Dialog Port
dialog-port	2098/udp	Dialog Port
h2250-annex-g	2099/tcp	H.225.0 Annex G
h2250-annex-g	2099/udp	H.225.0 Annex G
amiganetfs	2100/tcp	amiganetfs
amiganetfs	2100/udp	amiganetfs
rtcm-sc104	2101/tcp	rtcm-sc104
rtcm-sc104	2101/udp	rtcm-sc104
zephyr-srv	2102/tcp	Zephyr server
zephyr-srv	2102/udp	Zephyr server
zephyr-clt	2103/tcp	Zephyr serv-hm
zephyr-clt	2103/udp	Zephyr serv-hm
zephyr-hm	2104/tcp	Zephyr hostman
zephyr-hm	2104/udp	Zephyr hostman
minipay	2105/tcp	MiniPay
minipay	2105/udp	MiniPay
mzap	2106/tcp	MZAP
mzap	2106/udp	MZAP
bintec-admin	2107/tcp	BinTec Admin
bintec-admin	2107/udp	BinTec Admin
comcam	2108/tcp	Comcam
comcam	2108/udp	Comcam
ergolight	2109/tcp	Ergolight
ergolight	2109/udp	Ergolight
umsp	2110/tcp	UMSP
umsp	2110/udp	UMSP
dsatp	2111/tcp	DSATP
dsatp	2111/udp	DSATP
idonix-metanet	2112/tcp	Idonix MetaNet
idonix-metanet	2112/udp	Idonix MetaNet
hsl-storm	2113/tcp	HSL StoRM
hsl-storm	2113/udp	HSL StoRM
newheights	2114/tcp	NEWHEIGHTS
newheights	2114/udp	NEWHEIGHTS
kdm	2115/tcp	KDM
kdm	2115/udp	KDM
ccowcmr	2116/tcp	CCOWCMR
ccowcmr	2116/udp	CCOWCMR
mentaclient	2117/tcp	MENTACLIENT
mentaclient	2117/udp	MENTACLIENT
mentaserver	2118/tcp	MENTASERVER
mentaserver	2118/udp	MENTASERVER
gsigatekeeper	2119/tcp	GSIGATEKEEPER
gsigatekeeper	2119/udp	GSIGATEKEEPER
qencp	2120/tcp	Quick Eagle CP
qencp	2120/udp	Quick Eagle CP
scientia-ssdb	2121/tcp	SCIENTIA-SSDB
scientia-ssdb	2121/udp	SCIENTIA-SSDB
caupc-remote	2122/tcp	CauPC Remote Ctl
caupc-remote	2122/udp	CauPC Remote Ctl
gtp-control	2123/tcp	GTP-Control 3GPP
gtp-control	2123/udp	GTP-Control 3GPP
elatelink	2124/tcp	ELATELINK
elatelink	2124/udp	ELATELINK
lockstep	2125/tcp	LOCKSTEP
lockstep	2125/udp	LOCKSTEP
pktcable-cops	2126/tcp	PktCable-COPS
pktcable-cops	2126/udp	PktCable-COPS
index-pc-wb	2127/tcp	INDEX-PC-WB
index-pc-wb	2127/udp	INDEX-PC-WB
net-steward	2128/tcp	Net Steward Ctl
net-steward	2128/udp	Net Steward Ctl
telaconsole	5428/udp	TELACONSOLE
base	5429/tcp	Billing and Acc
base	5429/udp	Billing and Acc
radec-corp	5430/tcp	RADEC CORP
radec-corp	5430/udp	RADEC CORP
park-agent	5431/tcp	PARK AGENT
park-agnet	5431/udp	PARK AGENT
#	5432-5434	Unassigned
dttl	5435/tcp	Data (DTTL)
dttl	5435/udp	Data (DTTL)
#	5436-5453	Unassigned
apc-tcp-udp-4	5454/tcp	apc-tcp-udp-4
apc-tcp-udp-4	5454/udp	apc-tcp-udp-4
apc-tcp-udp-5	5455/tcp	apc-tcp-udp-5
apc-tcp-udp-5	5455/udp	apc-tcp-udp-5
apc-tcp-udp-6	5456/tcp	apc-tcp-udp-6
apc-tcp-udp-6	5456/udp	apc-tcp-udp-6
#	5457-5460	Unassigned
silkmeter	5461/tcp	SILKMETER
silkmeter	5461/udp	SILKMETER
ttl-publisher	5462/tcp	TTL Publisher
ttl-publisher	5462/udp	TTL Publisher
#	5463-5464	Unassigned
netops-broker	5465/tcp	NETOPS-BROKER
netops-broker	5465/udp	NETOPS-BROKER
#	5466-5499	Unassigned
fcp-addr-srvr1	5500/tcp	fcp-addr-srvr1
fcp-addr-srvr1	5500/udp	fcp-addr-srvr1
fcp-addr-srvr2	5501/tcp	fcp-addr-srvr2
fcp-addr-srvr2	5501/udp	fcp-addr-srvr2
fcp-srvr-inst1	5502/tcp	fcp-srvr-inst1
fcp-srvr-inst1	5502/udp	fcp-srvr-inst1
fcp-srvr-inst2	5503/tcp	fcp-srvr-inst2
fcp-srvr-inst2	5503/udp	fcp-srvr-inst2
fcp-cics-gw1	5504/tcp	fcp-cics-gw1
fcp-cics-gw1	5504/udp	fcp-cics-gw1
#	5504-5553	Unassigned
sgi-esphttp	5554/tcp	SGI ESP HTTP
sgi-esphttp	5554/udp	SGI ESP HTTP
personal-agent	5555/tcp	Personal Agent
personal-agent	5555/udp	Personal Agent
#	5556-5598	Unassigned
esinstall	5599/tcp	Enterprise
esinstall	5599/udp	Enterprise
esmmanager	5600/tcp	Enterprise
esmmanager	5600/udp	Enterprise
esmagent	5601/tcp	Enterprise
esmagent	5601/udp	Enterprise
a1-msc	5602/tcp	A1-MSC
a1-msc	5602/udp	A1-MSC
a1-bs	5603/tcp	A1-BS
a1-bs	5603/udp	A1-BS
a3-sdunode	5604/tcp	A3-SDUNode
a3-sdunode	5604/udp	A3-SDUNode
a4-sdunode	5605/tcp	A4-SDUNode
a4-sdunode	5605/udp	A4-SDUNode
#	5606-5630	Unassigned
pcanywheredata	5631/tcp	pcANYWHEREdata
pcanywheredata	5631/udp	pcANYWHEREdata
pcanywherestat	5632/tcp	pcANYWHEREstat
pcanywherestat	5632/udp	pcANYWHEREstat
#	5633-5677	Unassigned
rrac	5678/tcp	Remote RAC
rrac	5678/udp	Remote RAC
dccm	5679/tcp	Direct Cable Mgr
dccm	5679/udp	Direct Cable Mgr
#	5780-5712	Unassigned
proshareaudio	5713/tcp	proshare audio
proshareaudio	5713/udp	proshare audio
prosharevideo	5714/tcp	proshare video
prosharevideo	5714/udp	proshare video

cs-live	2129/tcp	cs-live.com
cs-live	2129/udp	cs-live.com
swc-xds	2130/tcp	SWC-XDS
swc-xds	2130/udp	SWC-XDS
avantageb2b	2131/tcp	Avantageb2b
avantageb2b	2131/udp	Avantageb2b
avail-epmap	2132/tcp	AVAIL-EPMAP
avail-epmap	2132/udp	AVAIL-EPMAP
zymed-zpp	2133/tcp	ZYMED-ZPP
zymed-zpp	2133/udp	ZYMED-ZPP
avenue	2134/tcp	AVENUE
avenue	2134/udp	AVENUE
gris	2135/tcp	Grid Resource
gris	2135/udp	Grid Resource
appworxsrv	2136/tcp	APPWORXSRV
appworxsrv	2136/udp	APPWORXSRV
connect	2137/tcp	CONNECT
connect	2137/udp	CONNECT
unbind-cluster	2138/tcp	UNBIND-CLUSTER
unbind-cluster	2138/udp	UNBIND-CLUSTER
ias-auth	2139/tcp	IAS-AUTH
ias-auth	2139/udp	IAS-AUTH
ias-reg	2140/tcp	IAS-REG
ias-reg	2140/udp	IAS-REG
ias-admind	2141/tcp	IAS-ADMIND
ias-admind	2141/udp	IAS-ADMIND
tdm-over-ip	2142/tcp	TDM-OVER-IP
tdm-over-ip	2142/udp	TDM-OVER-IP
lv-jc	2143/tcp	Live Vault
lv-jc	2143/udp	Live Vault
lv-ffx	2144/tcp	Live Vault
lv-ffx	2144/udp	Live Vault
lv-pici	2145/tcp	Live Vault
lv-pici	2145/udp	Live Vault
lv-not	2146/tcp	Live Vault
lv-not	2146/udp	Live Vault
lv-auth	2147/tcp	Live Vault
lv-auth	2147/udp	Live Vault
veritas-ucl	2148/tcp	VERITAS
veritas-ucl	2148/udp	VERITAS
acptsys	2149/tcp	ACPTSYS
acptsys	2149/udp	ACPTSYS
dynamic3d	2150/tcp	DYNAMIC3D
dynamic3d	2150/udp	DYNAMIC3D
docent	2151/tcp	DOCENT
docent	2151/udp	DOCENT
gtp-user	2152/tcp	GTP-User (3GPP)
gtp-user	2152/udp	GTP-User (3GPP)
#	2153-2164	Unassigned
x-bone-api	2165/tcp	X-Bone API
x-bone-api	2165/udp	X-Bone API
iwserver	2166/tcp	IWSERVER
iwserver	2166/udp	IWSERVER
#	2167-2179	Unassigned
mc-gt-srv	2180/tcp	MVGS
mc-gt-srv	2180/udp	MVGS
eforward	2181/tcp	eforward
eforward	2181/udp	eforward
ici	2200/tcp	ICI
ici	2200/udp	ICI
ats	2201/tcp	ATSP
ats	2201/udp	ATSP
imtc-map	2202/tcp	Int. Multimedia
imtc-map	2202/udp	Int. Multimedia
kali	2213/tcp	Kali
kali	2213/udp	Kali
ganymede	2220/tcp	Ganymede
ganymede	2220/udp	Ganymede
rockwell-csp1	2221/tcp	Rockwell CSP1
rockwell-csp1	2221/udp	Rockwell CSP1
rockwell-csp2	2222/tcp	Rockwell CSP2
prosharedata	5715/tcp	proshare data
prosharedata	5715/udp	proshare data
prosharerequest	5716/tcp	proshare request
prosharerequest	5716/udp	proshare request
prosharenotify	5717/tcp	proshare notify
prosharenotify	5717/udp	proshare notify
#	5718-5728	Unassigned
openmail	5729/tcp	Openmail
openmail	5729/udp	Openmail
#	5730-5740	Unassigned
ida-discover1	5741/tcp	IDA Disc Port 1
ida-discover1	5741/udp	IDA Disc Port 1
ida-discover2	5742/tcp	IDA Disc Port 2
ida-discover2	5742/udp	IDA Disc Port 2
#	5743-5744	Unassigned
fcopy-server	5745/tcp	fcopy-server
fcopy-server	5745/udp	fcopy-server
fcopys-server	5746/tcp	fcopys-server
fcopys-server	5746/udp	fcopys-server
#	5769-5770	Unassigned
netagent	5771/tcp	NetAgent
netagent	5771/udp	NetAgent
#	5772-5812	Unassigned
icmpd	5813/tcp	ICMPD
icmpd	5813/udp	ICMPD
#	5814-5858	Unassigned
wherehoo	5859/tcp	WHEREHOO
wherehoo	5859/udp	WHEREHOO
#	5860-5967	Unassigned
mppolicy-v5	5968/tcp	mppolicy-v5
mppolicy-v5	5968/udp	mppolicy-v5
mppolicy-mgr	5969/tcp	mppolicy-mgr
mppolicy-mgr	5969/udp	mppolicy-mgr
#	5970-5998	Unassigned
cvsup	5999/tcp	CVSup
cvsup	5999/udp	CVSup
x11	6000-6063/tcp	X Window
x11	6000-6063/udp	X Window
ndl-ahp-svc	6064/tcp	NDL-AHP-SVC
ndl-ahp-svc	6064/udp	NDL-AHP-SVC
winpharaoh	6065/tcp	WinPharaoh
winpharaoh	6065/udp	WinPharaoh
ewctsp	6066/tcp	EWCTSP
ewctsp	6066/udp	EWCTSP
srb	6067/tcp	SRB
srb	6067/udp	SRB
gsmp	6068/tcp	GSMP
gsmp	6068/udp	GSMP
trip	6069/tcp	TRIP
trip	6069/udp	TRIP
messageasap	6070/tcp	Messageasap
messageasap	6070/udp	Messageasap
ssdtp	6071/tcp	SSDTP
ssdtp	6071/udp	SSDTP
diagnose-proc	6072/tcp	DIAGNOSE-PROC
diagmose-proc	6072/udp	DIAGNOSE-PROC
directplay8	6073/tcp	DirectPlay8
directplay8	6073/udp	DirectPlay8
#	6074-6099	Unassigned
synchronet-db	6100/tcp	SynchroNet-db
synchronet-db	6100/udp	SynchroNet-db
synchronet-rtc	6101/tcp	SynchroNet-rtc
synchronet-rtc	6101/udp	SynchroNet-rtc
synchronet-upd	6102/tcp	SynchroNet-upd
synchronet-upd	6102/udp	SynchroNet-upd
rets	6103/tcp	RETS
rets	6103/udp	RETS
dbdb	6104/tcp	DBDB
dbdb	6104/udp	DBDB
primaserver	6105/tcp	Prima Server
primaserver	6105/udp	Prima Server

rockwell-csp2	2222/udp	Rockwell CSP2
rockwell-csp3	2223/tcp	Rockwell CSP3
rockwell-csp3	2223/udp	Rockwell CSP3
ivs-video	2232/tcp	IVS Video
ivs-video	2232/udp	IVS Video
infocrypt	2233/tcp	INFOCRYPT
infocrypt	2233/udp	INFOCRYPT
directplay	2234/tcp	DirectPlay
directplay	2234/udp	DirectPlay
sercomm-wlink	2235/tcp	Sercomm-WLink
sercomm-wlink	2235/udp	Sercomm-WLink
nani	2236/tcp	Nani
nani	2236/udp	Nani
optech-port1-lm	2237/tcp	Optech Port1
optech-port1-lm	2237/udp	Optech Port1
aviva-sna	2238/tcp	AVIVA SNA SERVER
aviva-sna	2238/udp	AVIVA SNA SERVER
imagequery	2239/tcp	Image Query
imagequery	2239/udp	Image Query
recipe	2240/tcp	RECIPe
recipe	2240/udp	RECIPe
ivsd	2241/tcp	IVS Daemon
ivsd	2241/udp	IVS Daemon
foliocorp	2242/tcp	Folio Remote
foliocorp	2242/udp	Folio Remote
magicom	2243/tcp	Magicom Protocol
magicom	2243/udp	Magicom Protocol
nmsserver	2244/tcp	NMS Server
nmsserver	2244/udp	NMS Server
hao	2245/tcp	HaO
hao	2245/udp	HaO
#	2245-2278	Unassigned
xmquery	2279/tcp	xmquery
xmquery	2279/udp	xmquery
lnvpoller	2280/tcp	LNVPOLLER
lnvpoller	2280/udp	LNVPOLLER
lnvconsole	2281/tcp	LNVCONSOLE
lnvconsole	2281/udp	LNVCONSOLE
lnvalarm	2282/tcp	LNVALARM
lnvalarm	2282/udp	LNVALARM
lnvstatus	2283/tcp	LNVSTATUS
lnvstatus	2283/udp	LNVSTATUS
lnvmaps	2284/tcp	LNVMAPS
lnvmaps	2284/udp	LNVMAPS
lnvmailmon	2285/tcp	LNVMAILMON
lnvmailmon	2285/udp	LNVMAILMON
nas-metering	2286/tcp	NAS-Metering
nas-metering	2286/udp	NAS-Metering
dna	2287/tcp	DNA
dna	2287/udp	DNA
netml	2288/tcp	NETML
netml	2288/udp	NETML
#	2289-2293	Unassigned
konshus-lm	2294/tcp	Konshus (FLEX)
konshus-lm	2294/udp	Konshus (FLEX)
advant-lm	2295/tcp	Advant
advant-lm	2295/udp	Advant
theta-lm	2296/tcp	Theta (Rainbow)
theta-lm	2296/udp	Theta (Rainbow)
d2k-datamover1	2297/tcp	D2K DataMover 1
d2k-datamover1	2297/udp	D2K DataMover 1
d2k-datamover2	2298/tcp	D2K DataMover 2
d2k-datamover2	2298/udp	D2K DataMover 2
pc-telecommute	2299/tcp	PC Telecommute
pc-telecommute	2299/udp	PC Telecommute
cvmmon	2300/tcp	CVMMON
cvmmon	2300/udp	CVMMON
cpq-wbem	2301/tcp	Compaq HTTP
cpq-wbem	2301/udp	Compaq HTTP
binderysupport	2302/tcp	Bindery Support
binderysupport	2302/udp	Bindery Support
mpsserver	6106/tcp	MPS Server
mpsserver	6106/udp	MPS Server
etc-control	6107/tcp	ETC Control
etc-control	6107/udp	ETC Control
sercomm-scadmin	6108/tcp	Sercomm-SCAdmin
sercomm-scadmin	6108/udp	Sercomm-SCAdmin
globecast-id	6109/tcp	GLOBECAST-ID
globecast-id	6109/udp	GLOBECAST-ID
softcm	6110/tcp	HP SoftBench CM
softcm	6110/udp	HP SoftBench CM
spc	6111/tcp	HP SoftBench
spc	6111/udp	HP SoftBench
dtspcd	6112/tcp	dtspcd
dtspcd	6112/udp	dtspcd
#	6113-6122	Unassigned
backup-express	6123/tcp	Backup Express
backup-express	6123/udp	Backup Express
#	6124-6140	Unassigned
meta-corp	6141/tcp	Meta Corporation
meta-corp	6141/udp	Meta Corporation
aspentec-lm	6142/tcp	Aspen Technology
aspentec-lm	6142/udp	Aspen Technology
watershed-lm	6143/tcp	Watershed
watershed-lm	6143/udp	Watershed
statsci1-lm	6144/tcp	StatSci - 1
statsci1-lm	6144/udp	StatSci - 1
statsci2-lm	6145/tcp	StatSci - 2
statsci2-lm	6145/udp	StatSci - 2
lonewolf-lm	6146/tcp	Lone Wolf
lonewolf-lm	6146/udp	Lone Wolf
montage-lm	6147/tcp	Montage
montage-lm	6147/udp	Montage
ricardo-lm	6148/tcp	Ricardo America
ricardo-lm	6148/udp	Ricardo America
tal-pod	6149/tcp	tal-pod
tal-pod	6149/udp	tal-pod
#	6150-6252	Unassigned
crip	6253/tcp	CRIP
crip	6253/udp	CRIP
#	6254-6320	Unassigned
emp-server1	6321/tcp	Empress Software
emp-server1	6321/udp	Empress Software
emp-server2	6322/tcp	Empress Software
emp-server2	6322/udp	Empress Software
#	6323-6388	Unassigned
clariion-evr01	6389/tcp	clariion-evr01
clariion-evr01	6389/udp	clariion-evr01
#	6390-6399	Unassigned
info-aps	6400	
info-was	6401	
info-eventsvr	6402	
info-cachesvr	6403	
info-filesvr	6404	
info-pagesvr	6405	
info-processvr	6406	
reserved1	6407	
reserved2	6408	
reserved3	6409	
reserved4	6410	
#	6411-6454	Unassigned
skip-cert-recv	6455/tcp	SKIP Certificate
skip-cert-send	6456/tcp	SKIP Certificate
#	6457-6470	Unassigned
lvision-lm	6471/tcp	LVision
lvision-lm	6471/udp	LVision
#	6472-6499	Unassigned
boks	6500/tcp	BoKS Master
boks	6500/udp	BoKS Master
boks_servc	6501/tcp	BoKS Servc
boks_servc	6501/udp	BoKS Servc
boks_servm	6502/tcp	BoKS Servm

proxy-gateway	2303/tcp	Proxy Gateway
proxy-gateway	2303/udp	Proxy Gateway
attachmate-uts	2304/tcp	Attachmate UTS
attachmate-uts	2304/udp	Attachmate UTS
mt-scaleserver	2305/tcp	MT ScaleServer
mt-scaleserver	2305/udp	MT ScaleServer
tappi-boxnet	2306/tcp	TAPPI BoxNet
tappi-boxnet	2306/udp	TAPPI BoxNet
pehelp	2307/tcp	pehelp
pehelp	2307/udp	pehelp
sdhelp	2308/tcp	sdhelp
sdhelp	2308/udp	sdhelp
sdserver	2309/tcp	SD Server
sdserver	2309/udp	SD Server
sdclient	2310/tcp	SD Client
sdclient	2310/udp	SD Client
messageservice	2311/tcp	Message Service
messageservice	2311/udp	Message Service
iapp	2313/tcp	IAPP
iapp	2313/udp	IAPP
cr-websystems	2314/tcp	CR WebSystems
cr-websystems	2314/udp	CR WebSystems
precise-sft	2315/tcp	Precise Sft.
precise-sft	2315/udp	Precise Sft.
sent-lm	2316/tcp	SENT
sent-lm	2316/udp	SENT
attachmate-g32	2317/tcp	Attachmate G32
attachmate-g32	2317/udp	Attachmate G32
cadencecontrol	2318/tcp	Cadence Control
cadencecontrol	2318/udp	Cadence Control
infolibria	2319/tcp	InfoLibria
infolibria	2319/udp	InfoLibria
siebel-ns	2320/tcp	Siebel NS
siebel-ns	2320/udp	Siebel NS
rdlap	2321/tcp	RDLAP over UDP
rdlap	2321/udp	RDLAP
ofsd	2322/tcp	ofsd
ofsd	2322/udp	ofsd
3d-nfsd	2323/tcp	3d-nfsd
3d-nfsd	2323/udp	3d-nfsd
cosmocall	2324/tcp	Cosmocall
cosmocall	2324/udp	Cosmocall
designspace-lm	2325/tcp	Design Space
designspace-lm	2325/udp	Design Space
idcp	2326/tcp	IDCP
idcp	2326/udp	IDCP
xingcsm	2327/tcp	xingcsm
xingcsm	2327/udp	xingcsm
netrix-sftm	2328/tcp	Netrix SFTM
netrix-sftm	2328/udp	Netrix SFTM
nvd	2329/tcp	NVD
nvd	2329/udp	NVD
tscchat	2330/tcp	TSCCHAT
tscchat	2330/udp	TSCCHAT
agentview	2331/tcp	AGENTVIEW
agentview	2331/udp	AGENTVIEW
rcc-host	2332/tcp	RCC Host
rcc-host	2332/udp	RCC Host
snapp	2333/tcp	SNAPP
snapp	2333/udp	SNAPP
ace-client	2334/tcp	ACE Client Auth
ace-client	2334/udp	ACE Client Auth
ace-proxy	2335/tcp	ACE Proxy
ace-proxy	2335/udp	ACE Proxy
appleugcontrol	2336/tcp	Apple UG Control
appleugcontrol	2336/udp	Apple UG Control
ideesrv	2337/tcp	ideesrv
ideesrv	2337/udp	ideesrv
norton-lambert	2338/tcp	Norton Lambert
norton-lambert	2338/udp	Norton Lambert
3com-webview	2339/tcp	3Com WebView
boks_servm	6502/udp	BoKS Servm
boks_clntd	6503/tcp	BoKS Clntd
boks_clntd	6503/udp	BoKS Clntd
#	6504	Unassigned
badm_priv	6505/tcp	BoKS Admin
badm_priv	6505/udp	BoKS Admin
badm_pub	6506/tcp	BoKS Admin
badm_pub	6506/udp	BoKS Admin
bdir_priv	6507/tcp	BoKS Dir Server
bdir_priv	6507/udp	BoKS Dir Server
bdir_pub	6508/tcp	BoKS Dir Server
bdir_pub	6508/udp	BoKS Dir Server
#	6509-6546	Unassigned
apc-tcp-udp-1	6547/tcp	apc-tcp-udp-1
apc-tcp-udp-1	6547/udp	apc-tcp-udp-1
apc-tcp-udp-2	6548/tcp	apc-tcp-udp-2
apc-tcp-udp-2	6548/udp	apc-tcp-udp-2
apc-tcp-udp-3	6549/tcp	apc-tcp-udp-3
apc-tcp-udp-3	6549/udp	apc-tcp-udp-3
fg-sysupdate	6550/tcp	fg-sysupdate
fg-sysupdate	6550/udp	fg-sysupdate
#	6551-6557	Unassigned
xdsxdm	6558/tcp	
xdsxdm	6558/udp	
ircu	6665-6669/tcp	IRCU
ircu	6665-6669/udp	IRCU
vocaltec-gold	6670/tcp	Vocaltec Global
vocaltec-gold	6670/udp	Vocaltec Global
vision_server	6672/tcp	vision_server
vision_server	6672/udp	vision_server
vision_elmd	6673/tcp	vision_elmd
vision_elmd	6673/udp	vision_elmd
kti-icad-srvr	6701/tcp	KTI/ICAD NS
kti-icad-srvr	6701/udp	KTI/ICAD NS
#	6702-6766	Unassigned
bmc-perf-agent	6767/tcp	BMC PERFORM
bmc-perf-agent	6767/udp	BMC PERFORM
bmc-perf-mgrd	6768/tcp	BMC PERFORM
bmc-perf-mgrd	6768/udp	BMC PERFORM
#	6769-6789	Unassigned
hnmp	6790/tcp	HNMP
hnmp	6790/udp	HNMP
ambit-lm	6831/tcp	ambit-lm
ambit-lm	6831/udp	ambit-lm
netmo-default	6841/tcp	Netmo Default
netmo-default	6841/udp	Netmo Default
netmo-http	6842/tcp	Netmo HTTP
netmo-http	6842/udp	Netmo HTTP
#	6843-6849	Unassigned
iccrushmore	6850/tcp	ICCRUSHMORE
iccrushmore	6850/udp	ICCRUSHMORE
#	6851-6887	Unassigned
muse	6888/tcp	MUSE
muse	6888/udp	MUSE
#	6889-6960	Unassigned
jmact3	6961/tcp	JMACT3
jmact3	6961/udp	JMACT3
jmevt2	6962/tcp	jmevt2
jmevt2	6962/udp	jmevt2
swismgr1	6963/tcp	swismgr1
swismgr1	6963/udp	swismgr1
swismgr2	6964/tcp	swismgr2
swismgr2	6964/udp	swismgr2
swistrap	6965/tcp	swistrap
swistrap	6965/udp	swistrap
swispol	6966/tcp	swispol
swispol	6966/udp	swispol
acmsoda	6969/tcp	acmsoda
acmsoda	6969/udp	acmsoda
iatp-highpri	6998/tcp	IATP-highPri
iatp-highpri	6998/udp	IATP-highPri

3com-webview	2339/udp	3Com WebView
wrs_registry	2340/tcp	WRS Registry
wrs_registry	2340/udp	WRS Registry
xiostatus	2341/tcp	XIO Status
xiostatus	2341/udp	XIO Status
manage-exec	2342/tcp	Seagate Manage
manage-exec	2342/udp	Seagate Manage
nati-logos	2343/tcp	nati logos
nati-logos	2343/udp	nati logos
fcmsys	2344/tcp	fcmsys
fcmsys	2344/udp	fcmsys
dbm	2345/tcp	dbm
dbm	2345/udp	dbm
redstorm_join	2346/tcp	Game Connection
redstorm_join	2346/udp	Game Connection
redstorm_find	2347/tcp	Game
redstorm_find	2347/udp	Game
redstorm_info	2348/tcp	Game status
redstorm_info	2348/udp	Game status
redstorm_diag	2349/tcp	Diagnostics Port
redstorm_diag	2349/udp	Disgnostics Port
psbserver	2350/tcp	psbserver
psbserver	2350/udp	psbserver
psrserver	2351/tcp	psrserver
psrserver	2351/udp	psrserver
pslserver	2352/tcp	pslserver
pslserver	2352/udp	pslserver
pspserver	2353/tcp	pspserver
pspserver	2353/udp	pspserver
psprserver	2354/tcp	psprserver
psprserver	2354/udp	psprserver
psdbserver	2355/tcp	psdbserver
psdbserver	2355/udp	psdbserver
gxtelmd	2356/tcp	GXT License Man
gxtelmd	2356/udp	GXT License Man
unihub-server	2357/tcp	UniHub Server
unihub-server	2357/udp	UniHub Server
futrix	2358/tcp	Futrix
futrix	2358/udp	Futrix
flukeserver	2359/tcp	FlukeServer
flukeserver	2359/udp	FlukeServer
nexstorindltd	2360/tcp	NexstorIndLtd
nexstorindltd	2360/udp	NexstorIndLtd
tl1	2361/tcp	TL1
tl1	2361/udp	TL1
digiman	2362/tcp	digiman
digiman	2362/udp	digiman
mediacntrlnfsd	2363/tcp	Media Cent NFSD
mediacntrlnfsd	2363/udp	Media Cent NFSD
oi-2000	2364/tcp	OI-2000
oi-2000	2364/udp	OI-2000
dbref	2365/tcp	dbref
dbref	2365/udp	dbref
qip-login	2366/tcp	qip-login
qip-login	2366/udp	qip-login
service-ctrl	2367/tcp	Service Control
service-ctrl	2367/udp	Service Control
opentable	2368/tcp	OpenTable
opentable	2368/udp	OpenTable
acs2000-dsp	2369/tcp	ACS2000 DSP
acs2000-dsp	2369/udp	ACS2000 DSP
l3-hbmon	2370/tcp	L3-HBMon
l3-hbmon	2370/udp	L3-HBMon
#	2371-2380	Unassigned
compaq-https	2381/tcp	Compaq HTTPS
compaq-https	2381/udp	Compaq HTTPS
ms-olap3	2382/tcp	Microsoft OLAP
ms-olap3	2382/udp	Microsoft OLAP
ms-olap4	2383/tcp	Microsoft OLAP
ms-olap4	2383/udp	Microsoft OLAP
sd-request	2384/tcp	SD-REQUEST
iatp-normalpri	6999/tcp	IATP-normalPri
iatp-normalpri	6999/udp	IATP-normalPri
afs3-fileserver	7000/tcp	file server
afs3-fileserver	7000/udp	file server
afs3-callback	7001/tcp	callbacks
afs3-callback	7001/udp	callbacks
afs3-prserver	7002/tcp	users & groups
afs3-prserver	7002/udp	users & groups
afs3-vlserver	7003/tcp	volume location
afs3-vlserver	7003/udp	volume location
afs3-kaserver	7004/tcp	AFS/Kerberos
afs3-kaserver	7004/udp	AFS/Kerberos
afs3-volser	7005/tcp	volume managment
afs3-volser	7005/udp	volume managment
afs3-errors	7006/tcp	error service
afs3-errors	7006/udp	error service
afs3-bos	7007/tcp	basic overseer
afs3-bos	7007/udp	basic overseer
afs3-update	7008/tcp	server-to-server
afs3-update	7008/udp	server-to-server
afs3-rmtsys	7009/tcp	remote cache
afs3-rmtsys	7009/udp	remote cache
ups-onlinet	7010/tcp	onlinet
ups-onlinet	7010/udp	onlinet
talon-disc	7011/tcp	Talon Discovery
talon-disc	7011/udp	Talon Discovery
talon-engine	7012/tcp	Talon Engine
talon-engine	7012/udp	Talon Engine
microtalon-dis	7013/tcp	Microtalon
microtalon-dis	7013/udp	Microtalon
microtalon-com	7014/tcp	Microtalon
microtalon-com	7014/udp	Microtalon
talon-webserver	7015/tcp	Talon Webserver
talon-webserver	7015/udp	Talon Webserver
#	7016-7019	Unassigned
dpserve	7020/tcp	DP Serve
dpserve	7020/udp	DP Serve
dpserveadmin	7021/tcp	DP Serve Admin
dpserveadmin	7021/udp	DP Serve Admin
#	7022-7069	Unassigned
arcp	7070/tcp	ARCP
arcp	7070/udp	ARCP
#	7071-7098	Unassigned
lazy-ptop	7099/tcp	lazy-ptop
lazy-ptop	7099/udp	lazy-ptop
font-service	7100/tcp	X Font Service
font-service	7100/udp	X Font Service
#	7101-7120	Unassigned
virprot-lm	7121/tcp	Virtual Proto
virprot-lm	7121/udp	Virtual Proto
#	7122-7173	Unassigned
clutild	7174/tcp	Clutild
clutild	7174/udp	Clutild
#	7175-7199	Unassigned
fodms	7200/tcp	FODMS FLIP
fodms	7200/udp	FODMS FLIP
dlip	7201/tcp	DLIP
dlip	7201/udp	DLIP
swx	7300-7390	The Swiss Exch
#	7391-7394	Unassigned
winqedit	7395/tcp	winqedit
winqedit	7395/udp	winqedit
#	7396-7425	Unassigned
pmdmgr	7426/tcp	OpenView DM Post
pmdmgr	7426/udp	OpenView DM Post
oveadmgr	7427/tcp	OpenView DM Even
oveadmgr	7427/udp	OpenView DM Even
ovladmgr	7428/tcp	OpenView DM Log
ovladmgr	7428/udp	OpenView DM Log
opi-sock	7429/tcp	OpenView DM rqt
opi-sock	7429/udp	OpenView DM rqt

sd-request	2384/udp	SD-REQUEST
#	2384-2388	Unassigned
ovsessionmgr	2389/tcp	OpenView Ses Mgr
ovsessionmgr	2389/udp	OpenView Ses Mgr
rsmtp	2390/tcp	RSMTP
rsmtp	2390/udp	RSMTP
3com-net-mgmt	2391/tcp	3COM Net Mgr
3com-net-mgmt	2391/udp	3COM Net Mgr
tacticalauth	2392/tcp	Tactical Auth
tacticalauth	2392/udp	Tactical Auth
ms-olap1	2393/tcp	MS OLAP 1
ms-olap1	2393/udp	MS OLAP 1
ms-olap2	2394/tcp	MS OLAP 2
ms-olap2	2394/udp	MA OLAP 2
lan900_remote	2395/tcp	LAN900 Remote
lan900_remote	2395/udp	LAN900 Remote
wusage	2396/tcp	Wusage
wusage	2396/udp	Wusage
ncl	2397/tcp	NCL
ncl	2397/udp	NCL
orbiter	2398/tcp	Orbiter
orbiter	2398/udp	Orbiter
fmpro-fdal	2399/tcp	FileMaker, Inc.
fmpro-fdal	2399/udp	FileMaker, Inc.
opequus-server	2400/tcp	OpEquus Server
opequus-server	2400/udp	OpEquus Server
cvspserver	2401/tcp	cvspserver
cvspserver	2401/udp	cvspserver
taskmaster2000	2402/tcp	TaskMaster 2000
taskmaster2000	2402/udp	TaskMaster 2000
taskmaster2000	2403/tcp	TaskMaster 2000
taskmaster2000	2403/udp	TaskMaster 2000
iec870-5-104	2404/tcp	IEC870-5-104
iec870-5-104	2404/udp	IEC870-5-104
trc-netpoll	2405/tcp	TRC Netpoll
trc-netpoll	2405/udp	TRC Netpoll
jediserver	2406/tcp	JediServer
jediserver	2406/udp	JediServer
orion	2407/tcp	Orion
orion	2407/udp	Orion
optimanet	2408/tcp	OptimaNet
optimanet	2408/udp	OptimaNet
sns-protocol	2409/tcp	SNS Protocol
sns-protocol	2409/udp	SNS Protocol
vrts-registry	2410/tcp	VRTS Registry
vrts-registry	2410/udp	VRTS Registry
netwave-ap-mgmt	2411/tcp	Netwave AP Mgr
netwave-ap-mgmt	2411/udp	Netwave AP Mgr
cdn	2412/tcp	CDN
cdn	2412/udp	CDN
orion-rmi-reg	2413/tcp	orion-rmi-reg
orion-rmi-reg	2413/udp	orion-rmi-reg
interlingua	2414/tcp	Interlingua
interlingua	2414/udp	Interlingua
comtest	2415/tcp	COMTEST
comtest	2415/udp	COMTEST
rmtserver	2416/tcp	RMT Server
rmtserver	2416/udp	RMT Server
composit-server	2417/tcp	Composit Server
composit-server	2417/udp	Composit Server
cas	2418/tcp	cas
cas	2418/udp	cas
attachmate-s2s	2419/tcp	Attachmate S2S
attachmate-s2s	2419/udp	Attachmate S2S
dslremote-mgmt	2420/tcp	DSL Remote Mgr
dslremote-mgmt	2420/udp	DSL Remote Mgr
g-talk	2421/tcp	G-Talk
g-talk	2421/udp	G-Talk
crmsbits	2422/tcp	CRMSBITS
crmsbits	2422/udp	CRMSBITS
rnrp	2423/tcp	RNRP
xmpv7	7430/tcp	OpenView DM
xmpv7	7430/udp	OpenView DM
pmd	7431/tcp	OpenView DM
pmd	7431/udp	OpenView DM
faximum	7437/tcp	Faximum
faximum	7437/udp	Faximum
telops-lmd	7491/tcp	telops-lmd
telops-lmd	7491/udp	telops-lmd
pafec-lm	7511/tcp	pafec-lm
pafec-lm	7511/udp	pafec-lm
nta-ds	7544/tcp	FlowAnalyzer
nta-ds	7544/udp	FlowAnalyzer
nta-us	7545/tcp	FlowAnalyzer
nta-us	7545/udp	FlowAnalyzer
vsi-omega	7566/tcp	VSI Omega
vsi-omega	7566/udp	VSI Omega
#	7567-7569	Unassigned
aries-kfinder	7570/tcp	Aries Kfinder
aries-kfinder	7570/udp	Aries Kfinder
#	7571-7587	Unassigned
sun-lm	7588/tcp	Sun License Mgr
sun-lm	7588/udp	Sun License Mgr
#	7589-7632	Unassigned
pmdfmgt	7633/tcp	PMDF Management
pmdfmgt	7633/udp	PMDF Management
#	7634-7776	Unassigned
cbt	7777/tcp	cbt
cbt	7777/udp	cbt
interwise	7778/tcp	Interwise
interwise	7778/udp	Interwise
#	7779-7780	Unassigned
accu-lmgr	7781/tcp	accu-lmgr
accu-lmgr	7781/udp	accu-lmgr
#	7782-7785	Unassigned
minivend	7786/tcp	MINIVEND
minivend	7786/udp	MINIVEND
#	7787-7931	Unassigned
t2-drm	7932/tcp	Tier 2 Data
t2-drm	7932/udp	Tier 2 Data
t2-brm	7933/tcp	Tier 2 Business
t2-brm	7933/udp	Tier 2 Business
supercell	7967/tcp	Supercell
supercell	7967/udp	Supercell
#	7968-7978	Unassigned
micromuse-ncps	7979/tcp	Micromuse-ncps
micromuse-ncps	7979/udp	Micromuse-ncps
quest-vista	7980/tcp	Quest Vista
quest-vista	7980/udp	Quest Vista
#	7981-7998	Unassigned
irdmi2	7999/tcp	iRDMI2
irdmi2	7999/udp	iRDMI2
irdmi	8000/tcp	iRDMI
irdmi	8000/udp	iRDMI
vcom-tunnel	8001/tcp	VCOM Tunnel
vcom-tunnel	8001/udp	VCOM Tunnel
teradataordbms	8002/tcp	Teradata ORDBMS
teradataordbms	8002/udp	Teradata ORDBMS
#	8003-8007	Unassigned
http-alt	8008/tcp	HTTP Alternate
http-alt	8008/udp	HTTP Alternate
#	8009-8031	Unassigned
pro-ed	8032/tcp	ProEd
pro-ed	8032/udp	ProEd
mindprint	8033/tcp	MindPrint
mindprint	8033/udp	MindPrint
#	8034-8079	Unassigned
http-alt	8080/tcp	HTTP Alternate
http-alt	8080/udp	HTTP Alternate
#	8081-8129	Unassigned
indigo-vrmi	8130/tcp	INDIGO-VRMI
indigo-vrmi	8130/udp	INDIGO-VRMI

rnrp	2423/udp	RNRP
kofax-svr	2424/tcp	KOFAX-SVR
kofax-svr	2424/udp	KOFAX-SVR
fjitsuappmgr	2425/tcp	Fujitsu App Mgr
fjitsuappmgr	2425/udp	Fujitsu App Mgr
applianttcp	2426/tcp	Appliant TCP
appliantudp	2426/udp	Appliant UDP
mgcp-gateway	2427/tcp	Media Gateway
mgcp-gateway	2427/udp	Media Gateway
ott	2428/tcp	1 Way Trip Time
ott	2428/udp	1 Way Trip Time
ft-role	2429/tcp	FT-ROLE
ft-role	2429/udp	FT-ROLE
venus	2430/tcp	venus
venus	2430/udp	venus
venus-se	2431/tcp	venus-se
venus-se	2431/udp	venus-se
codasrv	2432/tcp	codasrv
codasrv	2432/udp	codasrv
codasrv-se	2433/tcp	codasrv-se
codasrv-se	2433/udp	codasrv-se
pxc-epmap	2434/tcp	pxc-epmap
pxc-epmap	2434/udp	pxc-epmap
optilogic	2435/tcp	OptiLogic
optilogic	2435/udp	OptiLogic
topx	2436/tcp	TOP/X
topx	2436/udp	TOP/X
unicontrol	2437/tcp	UniControl
unicontrol	2437/udp	UniControl
msp	2438/tcp	MSP
msp	2438/udp	MSP
sybasedbsynch	2439/tcp	SybaseDBSynch
sybasedbsynch	2439/udp	SybaseDBSynch
spearway	2440/tcp	Spearway Lockers
spearway	2440/udp	Spearway Lockser
pvsw-inet	2441/tcp	pvsw-inet
pvsw-inet	2441/udp	pvsw-inet
netangel	2442/tcp	Netangel
netangel	2442/udp	Netangel
powerclientcsf	2443/tcp	PowerClient
powerclientcsf	2443/udp	PowerClient
btpp2sectrans	2444/tcp	BT PP2 Sectrans
btpp2sectrans	2444/udp	BT PP2 Sectrans
dtn1	2445/tcp	DTN1
dtn1	2445/udp	DTN1
bues_service	2446/tcp	bues_service
bues_service	2446/udp	bues_service
ovwdb	2447/tcp	OpenView NNM
ovwdb	2447/udp	OpenView NNM
hpppssvr	2448/tcp	hpppsvr
hpppssvr	2448/udp	hpppsvr
ratl	2449/tcp	RATL
ratl	2449/udp	RATL
netadmin	2450/tcp	netadmin
netadmin	2450/udp	netadmin
netchat	2451/tcp	netchat
netchat	2451/udp	netchat
snifferclient	2452/tcp	SnifferClient
snifferclient	2452/udp	SnifferClient
madge-om	2453/tcp	madge-om
madge-om	2453/udp	madge-om
indx-dds	2454/tcp	IndX-DDS
indx-dds	2454/udp	IndX-DDS
wago-io-system	2455/tcp	WAGO-IO-SYSTEM
wago-io-system	2455/udp	WAGO-IO-SYSTEM
altav-remmgt	2456/tcp	altav-remmgt
altav-remmgt	2456/udp	altav-remmgt
rapido-ip	2457/tcp	Rapido_IP
rapido-ip	2457/udp	Rapido_IP
griffin	2458/tcp	griffin
griffin	2458/udp	griffin
indigo-vbcp	8131/tcp	INDIGO-VBCP
indigo-vbcp	8131/udp	INDIGO-VBCP
#	8132-8159	Unassigned
patrol	8160/tcp	Patrol
patrol	8160/udp	Patrol
patrol-snmp	8161/tcp	Patrol SNMP
patrol-snmp	8161/udp	Patrol SNMP
#	8162-8199	Unassigned
trivnet1	8200/tcp	TRIVNET
trivnet1	8200/udp	TRIVNET
trivnet2	8201/tcp	TRIVNET
trivnet2	8201/udp	TRIVNET
#	8202-8203	Unassigned
lm-perfworks	8204/tcp	LM Perfworks
lm-perfworks	8204/udp	LM Perfworks
lm-instmgr	8205/tcp	LM Instmgr
lm-instmgr	8205/udp	LM Instmgr
lm-dta	8206/tcp	LM Dta
lm-dta	8206/udp	LM Dta
lm-sserver	8207/tcp	LM SServer
lm-sserver	8207/udp	LM SServer
lm-webwatcher	8208/tcp	LM Webwatcher
lm-webwatcher	8208/udp	LM Webwatcher
#	8209-8350	Unassigned
server-find	8351/tcp	Server Find
server-find	8351/udp	Server Find
#	8352-8375	Unassigned
cruise-enum	8376/tcp	Cruise ENUM
cruise-enum	8376/udp	Cruise ENUM
cruise-swroute	8377/tcp	Cruise SWROUTE
cruise-swroute	8377/udp	Cruise SWROUTE
cruise-config	8378/tcp	Cruise CONFIG
cruise-config	8378/udp	Cruise CONFIG
cruise-diags	8379/tcp	Cruise DIAGS
cruise-diags	8379/udp	Cruise DIAGS
cruise-update	8380/tcp	Cruise UPDATE
cruise-update	8380/udp	Cruise UPDATE
#	8381-8399	Unassigned
cvd	8400/tcp	cvd
cvd	8400/udp	cvd
sabarsd	8401/tcp	sabarsd
sabarsd	8401/udp	sabarsd
abarsd	8402/tcp	abarsd
abarsd	8402/udp	abarsd
admind	8403/tcp	admind
admind	8403/udp	admind
#	8404-8449	Unassigned
npmp	8450/tcp	npmp
npmp	8450/udp	npmp
#	8451-8472	Unassigned
vp2p	8473/tcp	Virtual P-to-P
vp2p	8473/udp	Virtual P-to-P
#	8474-8553	Unassigned
rtsp-alt	8554/tcp	RTSP Alternate
rtsp-alt	8554/udp	RTSP Alternate
#	8555-8732	Unassigned
ibus	8733/tcp	iBus
ibus	8733/udp	iBus
#	8734-8762	Unassigned
mc-appserver	8763/tcp	MC-APPSERVER
mc-appserver	8763/udp	MC-APPSERVER
openqueue	8764/tcp	OPENQUEUE
openqueue	8764/udp	OPENQUEUE
ultraseek-http	8765/tcp	Ultraseek HTTP
ultraseek-http	8765/udp	Ultraseek HTTP
#	8766-8803	Unassigned
truecm	8804/tcp	truecm
truecm	8804/udp	truecm
#	8805-8879	
cddbp-alt	8880/tcp	CDDBP
cddbp-alt	8880/udp	CDDBP

community	2459/tcp	Community
community	2459/udp	Community
ms-theater	2460/tcp	ms-theater
ms-theater	2460/udp	ms-theater
qadmifoper	2461/tcp	qadmifoper
qadmifoper	2461/udp	qadmifoper
qadmifevent	2462/tcp	qadmifevent
qadmifevent	2462/udp	qadmifevent
symbios-raid	2463/tcp	Symbios Raid
symbios-raid	2463/udp	Symbios Raid
direcpc-si	2464/tcp	DirecPC SI
direcpc-si	2464/udp	DirecPC SI
lbm	2465/tcp	Load Balance Mgr
lbm	2465/udp	Load Balance Mgr
lbf	2466/tcp	Load Balance Fwr
lbf	2466/udp	Load Balance Fwr
high-criteria	2467/tcp	High Criteria
high-criteria	2467/udp	High Criteria
qip-msgd	2468/tcp	qip_msgd
qip-msgd	2468/udp	qip_msgd
mti-tcs-comm	2469/tcp	MTI-TCS-COMM
mti-tcs-comm	2469/udp	MTI-TCS-COMM
taskman-port	2470/tcp	taskman port
taskman-port	2470/udp	taskman port
seaodbc	2471/tcp	SeaODBC
seaodbc	2471/udp	SeaODBC
c3	2472/tcp	C3
c3	2472/udp	C3
aker-cdp	2473/tcp	Aker-cdp
aker-cdp	2473/udp	Aker-cdp
vitalanalysis	2474/tcp	Vital Analysis
vitalanalysis	2474/udp	Vital Analysis
ace-server	2475/tcp	ACE Server
ace-server	2475/udp	ACE Server
ace-svr-prop	2476/tcp	ACE Server
ace-svr-prop	2476/udp	ACE Server
ssm-cvs	2477/tcp	SecurSight
ssm-cvs	2477/udp	SecurSight
ssm-cssps	2478/tcp	SecurSight (SLL)
ssm-cssps	2478/udp	SecurSight (SSL)
ssm-els	2479/tcp	SecurSight (SSL)
ssm-els	2479/udp	SecurSight (SSL)
lingwood	2480/tcp	Lingwood's
lingwood	2480/udp	Lingwood's
giop	2481/tcp	Oracle GIOP
giop	2481/udp	Oracle GIOP
giop-ssl	2482/tcp	Oracle GIOP SSL
giop-ssl	2482/udp	Oracle GIOP SSL
ttc	2483/tcp	Oracle TTC
ttc	2483/udp	Oracel TTC
ttc-ssl	2484/tcp	Oracle TTC SSL
ttc-ssl	2484/udp	Oracle TTC SSL
netobjects1	2485/tcp	Net Objects1
netobjects1	2485/udp	Net Objects1
netobjects2	2486/tcp	Net Objects2
netobjects2	2486/udp	Net Objects2
pns	2487/tcp	Policy Notice
pns	2487/udp	Policy Notice
moy-corp	2488/tcp	Moy Corporation
moy-corp	2488/udp	Moy Corporation
tsilb	2489/tcp	TSILB
tsilb	2489/udp	TSILB
qip-qdhcp	2490/tcp	qip_qdhcp
qip-qdhcp	2490/udp	qip_qdhcp
conclave-cpp	2491/tcp	Conclave CPP
conclave-cpp	2491/udp	Conclave CPP
groove	2492/tcp	GROOVE
groove	2492/udp	GROOVE
talarian-mqs	2493/tcp	Talarian MQS
talarian-mqs	2493/udp	Talarian MQS
bmc-ar	2494/tcp	BMC AR
#	8881-8887	Unassigned
ddi-tcp-1	8888/tcp	NewsEDGE TCP 1
ddi-udp-1	8888/udp	NewsEDGE UDP 1
ddi-tcp-2	8889/tcp	TCP 1
ddi-udp-2	8889/udp	NewsEDGE server
ddi-tcp-3	8890/tcp	TCP 2
ddi-udp-3	8890/udp	NewsEDGE
ddi-tcp-4	8891/tcp	NESS app
ddi-udp-4	8891/udp	NESS app
ddi-tcp-5	8892/tcp	FARM product
ddi-udp-5	8892/udp	FARM product
ddi-tcp-6	8893/tcp	NewsEDGE
ddi-udp-6	8893/udp	NewsEDGE
ddi-tcp-7	8894/tcp	COAL app
ddi-udp-7	8894/udp	COAL app
#	8895-8899	Unassigned
jmb-cds1	8900/tcp	JMB-CDS 1
jmb-cds1	8900/udp	JMB-CDS 1
jmb-cds2	8901/tcp	JMB-CDS 2
jmb-cds2	8901/udp	JMB-CDS 2
#	8902-8999	Unassigned
cslistener	9000/tcp	CSlistener
cslistener	9000/udp	CSlistener
#	9001-9005	Unassigned
#	9006	De-Commissioned
#	9007-9089	Unassigned
websm	9090/tcp	WebSM
websm	9090/udp	WebSM
#	9091-9159	Unassigned
netlock1	9160/tcp	NetLOCK1
netlock1	9160/udp	NetLOCK1
netlock2	9161/tcp	NetLOCK2
netlock2	9161/udp	NetLOCK2
netlock3	9162/tcp	NetLOCK3
netlock3	9162/udp	NetLOCK3
netlock4	9163/tcp	NetLOCK4
netlock4	9163/udp	NetLOCK4
netlock5	9164/tcp	NetLOCK5
netlock5	9164/udp	NetLOCK5
#	9165-9199	Unassigned
wap-wsp	9200/tcp	WAP
wap-wsp	9200/udp	WAP
wap-wsp-wtp	9201/tcp	WAP session
wap-wsp-wtp	9201/udp	WAP session
wap-wsp-s	9202/tcp	WAP secure
wap-wsp-s	9202/udp	WAP secure
wap-wsp-wtp-s	9203/tcp	WAP secure
wap-wsp-wtp-s	9203/udp	WAP secure
wap-vcard	9204/tcp	WAP vCard
wap-vcard	9204/udp	WAP vCard
wap-vcal	9205/tcp	WAP vCal
wap-vcal	9205/udp	WAP vCal
wap-vcard-s	9206/tcp	WAP vCard
wap-vcard-s	9206/udp	WAP vCard
wap-vcal-s	9207/tcp	WAP vCal Secure
wap-vcal-s	9207/udp	WAP vCal Secure
#	9208-9282	Unassigned
callwaveiam	9283/tcp	CallWaveIAM
callwaveiam	9283/udp	CallWaveIAM
#	9284-9320	Unassigned
guibase	9321/tcp	guibase
guibase	9321/udp	guibase
#	9322-9342	Unassigned
mpidcmgr	9343/tcp	MpIdcMgr
mpidcmgr	9343/udp	MpIdcMgr
mphlpdmc	9344/tcp	Mphlpdmc
mphlpdmc	9344/udp	Mphlpdmc
#	9345-9373	Unassigned
fjdmimgr	9374/tcp	fjdmimgr
fjdmimgr	9374/udp	fjdmimgr
#	9375-9395	Unassigned

```
bmc-ar          2494/udp   BMC AR
fast-rem-serv   2495/tcp   Fast Remote Serv
fast-rem-serv   2495/udp   Fast Remote Serv
dirgis          2496/tcp   DIRGIS
dirgis          2496/udp   DIRGIS
quaddb          2497/tcp   Quad DB
quaddb          2497/udp   Quad DB
odn-castraq     2498/tcp   ODN-CasTraq
odn-castraq     2498/udp   ODN-CasTraq
unicontrol      2499/tcp   UniControl
unicontrol      2499/udp   UniControl
rtsserv         2500/tcp   Resource Track
rtsserv         2500/udp   Resource Track
rtsclient       2501/tcp   Resource Track
rtsclient       2501/udp   Resource Track
kentrox-prot    2502/tcp   Kentrox Protocol
kentrox-prot    2502/udp   Kentrox Protocol
nms-dpnss       2503/tcp   NMS-DPNSS
nms-dpnss       2503/udp   NMS-DPNSS
wlbs            2504/tcp   WLBS
wlbs            2504/udp   WLBS
torque-traffic  2505/tcp   torque-traffic
torque-traffic  2505/udp   torque-traffic
jbroker         2506/tcp   jbroker
jbroker         2506/udp   jbroker
spock           2507/tcp   spock
spock           2507/udp   spock
jdatastore      2508/tcp   JDataStore
jdatastore      2508/udp   JDataStore
fjmpss          2509/tcp   fjmpss
fjmpss          2509/udp   fjmpss
fjappmgrbulk    2510/tcp   fjappmgrbulk
fjappmgrbulk    2510/udp   fjappmgrbulk
metastorm       2511/tcp   Metastorm
metastorm       2511/udp   Metastorm
citrixima       2512/tcp   Citrix IMA
citrixima       2512/udp   Citrix IMA
citrixadmin     2513/tcp   Citrix ADMIN
citrixadmin     2513/udp   Citrix ADMIN
facsys-ntp      2514/tcp   Facsys NTP
facsys-ntp      2514/udp   Facsys NTP
facsys-router   2515/tcp   Facsys Router
facsys-router   2515/udp   Facsys Router
maincontrol     2516/tcp   Main Control
maincontrol     2516/udp   Main Control
call-sig-trans  2517/tcp   H.323 Annex E
call-sig-trans  2517/udp   H.323 Annex E
willy           2518/tcp   Willy
willy           2518/udp   Willy
globmsgsvc      2519/tcp   globmsgsvc
globmsgsvc      2519/udp   globmsgsvc
pvsw            2520/tcp   pvsw
pvsw            2520/udp   pvsw
adaptecmgr      2521/tcp   Adaptec Manager
adaptecmgr      2521/udp   Adaptec Manager
windb           2522/tcp   WinDb
windb           2522/udp   WinDb
qke-llc-v3      2523/tcp   Qke LLC V.3
qke-llc-v3      2523/udp   Qke LLC V.3
optiwave-lm     2524/tcp   Optiwave
optiwave-lm     2524/udp   Optiwave
ms-v-worlds     2525/tcp   MS V-Worlds
ms-v-worlds     2525/udp   MS V-Worlds
ema-sent-lm     2526/tcp   EMA License Mgr
ema-sent-lm     2526/udp   EMA License Mgr
iqserver        2527/tcp   IQ Server
iqserver        2527/udp   IQ Server
ncr_ccl         2528/tcp   NCR CCL
ncr_ccl         2528/udp   NCR CCL
utsftp          2529/tcp   UTS FTP
utsftp          2529/udp   UTS FTP
fjinvmgr        9396/tcp   fjinvmgr
fjinvmgr        9396/udp   fjinvmgr
mpidcagt        9397/tcp   MpIdcAgt
mpidcagt        9397/udp   MpIdcAgt
#               9398-9499  Unassigned
ismserver       9500/tcp   ismserver
ismserver       9500/udp   ismserver
#               9501-9534  Unassigned
man             9535/tcp
man             9535/udp
#               9536-9593  Unassigned
msgsys          9594/tcp   Message System
msgsys          9594/udp   Message System
pds             9595/tcp   Ping Discovery
pds             9595/udp   Ping Discovery
#               9596-9599  Unassigned
micromuse-ncpw  9600/tcp   MICROMUSE-NCPW
micromuse-ncpw  9600/udp   MICROMUSE-NCPW
#               9601-9752  Unassigned
rasadv          9753/tcp   rasadv
rasadv          9753/udp   rasadv
#               9754-9875  Unassigned
sd              9876/tcp   Session Direct
sd              9876/udp   Session Direct
cyborg-systems  9888/tcp   CYBORG Systems
cyborg-systems  9888/udp   CYBORG Systems
monkeycom       9898/tcp   MonkeyCom
monkeycom       9898/udp   MonkeyCom
sctp-tunneling  9899/tcp   SCTP TUNNELING
sctp-tunneling  9899/udp   SCTP TUNNELING
iua             9900/tcp   IUA
iua             9900/udp   IUA
#               9901-9908  Unassigned
domaintime      9909/tcp   domaintime
domaintime      9909/udp   domaintime
#               9910-9949  Unassigned
apcpcpluswin1   9950/tcp   APCPCPLUSWIN1
apcpcpluswin1   9950/udp   APCPCPLUSWIN1
apcpcpluswin2   9951/tcp   APCPCPLUSWIN2
apcpcpluswin2   9951/udp   APCPCPLUSWIN2
apcpcpluswin3   9952/tcp   APCPCPLUSWIN3
apcpcpluswin3   9952/udp   APCPCPLUSWIN3
#               9953-9991  Unassigned
palace          9992/tcp   Palace
palace          9992/udp   Palace
palace          9993/tcp   Palace
palace          9993/udp   Palace
palace          9994/tcp   Palace
palace          9994/udp   Palace
palace          9995/tcp   Palace
palace          9995/udp   Palace
palace          9996/tcp   Palace
palace          9996/udp   Palace
palace          9997/tcp   Palace
palace          9997/udp   Palace
distinct32      9998/tcp   Distinct32
distinct32      9998/udp   Distinct32
distinct        9999/tcp   distinct
distinct        9999/udp   distinct
ndmp            10000/tcp  Network Data
ndmp            10000/udp  Network Data
#               10001-10006 Unassigned
mvs-capacity    10007/tcp  MVS Capacity
mvs-capacity    10007/udp  MVS Capacity
#               10008-10079 Unassigned
amanda          10080/tcp  Amanda
amanda          10080/udp  Amanda
#               10081-10112 Unassigned
netiq-endpoint  10113/tcp  NetIQ Endpoint
netiq-endpoint  10113/udp  NetIQ Endpoint
netiq-qcheck    10114/tcp  NetIQ Qcheck
```

```
vrcommerce      2530/tcp   VR Commerce
vrcommerce      2530/udp   VR Commerce
ito-e-gui       2531/tcp   ITO-E GUI
ito-e-gui       2531/udp   ITO-E GUI
ovtopmd         2532/tcp   OVTOPMD
ovtopmd         2532/udp   OVTOPMD
snifferserver   2533/tcp   SnifferServer
snifferserver   2533/udp   SnifferServer
combox-web-acc  2534/tcp   Combox Web Acc
combox-web-acc  2534/udp   Combox Web Acc
madcap          2535/tcp   MADCAP
madcap          2535/udp   MADCAP
btpp2audctr1    2536/tcp   btpp2audctr1
btpp2audctr1    2536/udp   btpp2audctr1
upgrade         2537/tcp   Upgrade Protocol
upgrade         2537/udp   Upgrade Protocol
vnwk-prapi      2538/tcp   vnwk-prapi
vnwk-prapi      2538/udp   vnwk-prapi
vsiadmin        2539/tcp   VSI Admin
vsiadmin        2539/udp   VSI Admin
lonworks        2540/tcp   LonWorks
lonworks        2540/udp   LonWorks
lonworks2       2541/tcp   LonWorks2
lonworks2       2541/udp   LonWorks2
davinci         2542/tcp   daVinci
davinci         2542/udp   daVinci
reftek          2543/tcp   REFTEK
reftek          2543/udp   REFTEK
novell-zen      2544/tcp   Novell ZEN
novell-zen      2544/tcp   Novell ZEN
sis-emt         2545/tcp   sis-emt
sis-emt         2545/udp   sis-emt
vytalvaultbrtp  2546/tcp   vytalvaultbrtp
vytalvaultbrtp  2546/udp   vytalvaultbrtp
vytalvaultvsmp  2547/tcp   vytalvaultvsmp
vytalvaultvsmp  2547/udp   vytalvaultvsmp
vytalvaultpipe  2548/tcp   vytalvaultpipe
vytalvaultpipe  2548/udp   vytalvaultpipe
ipass           2549/tcp   IPASS
ipass           2549/udp   IPASS
ads             2550/tcp   ADS
ads             2550/udp   ADS
isg-uda-server  2551/tcp   ISG UDA Server
isg-uda-server  2551/udp   ISG UDA Server
call-logging    2552/tcp   Call Logging
call-logging    2552/udp   Call Logging
efidiningport   2553/tcp   efidiningport
efidiningport   2553/udp   efidiningport
vcnet-link-v10  2554/tcp   VCnet-Link v10
vcnet-link-v10  2554/udp   VCnet-Link v10
compaq-wcp      2555/tcp   Compaq WCP
compaq-wcp      2555/udp   Compaq WCP
nicetec-nmsvc   2556/tcp   nicetec-nmsvc
nicetec-nmsvc   2556/udp   nicetec-nmsvc
nicetec-mgmt    2557/tcp   nicetec-mgmt
nicetec-mgmt    2557/udp   nicetec-mgmt
pclemultimedia  2558/tcp   PCLE Multi Media
pclemultimedia  2558/udp   PCLE Multi Media
lstp            2559/tcp   LSTP
lstp            2559/udp   LSTP
labrat          2560/tcp   labrat
labrat          2560/udp   labrat
mosaixcc        2561/tcp   MosaixCC
mosaixcc        2561/udp   MosaixCC
delibo          2562/tcp   Delibo
delibo          2562/udp   Delibo
cti-redwood     2563/tcp   CTI Redwood
cti-redwood     2563/udp   CTI Redwood
hp-3000-telnet  2564/tcp   HP 3000 NS/VT
coord-svr       2565/tcp   Coordinator Serv
coord-svr       2565/udp   Coordinator Serv
netiq-qcheck    10114/udp  NetIQ Qcheck
ganymede-endpt  10115/tcp  Ganymede
ganymede-endpt  10115/udp  Ganymede
#               10116-10127 Unassigned
bmc-perf-sd     10128/tcp  BMC-PERFORM
bmc-perf-sd     10128/udp  BMC-PERFORM
#               10129-10287 Unassigned
blocks          10288/tcp  Blocks
blocks          10288/udp  Blocks
#               10289-10999 Unassigned
irisa           11000/tcp  IRISA
irisa           11000/udp  IRISA
metasys         11001/tcp  Metasys
metasys         11001/udp  Metasys
#               11002-11110 Unassigned
vce             11111/tcp  Viral (VCE)
vce             11111/udp  Viral (VCE)
#               11112-11366 Unassigned
atm-uhas        11367/tcp  ATM UHAS
atm-uhas        11367/udp  ATM UHAS
#               11368-11719 Unassigned
h323callsigalt  11720/tcp  h323 Call Signal
h323callsigalt  11720/udp  h323 Call Signal
#               11721-11999 Unassigned
entextxid       12000/tcp  IBM Enterprise
entextxid       12000/udp  IBM Enterprise
entextnetwk     12001/tcp  IBM Enterprise
entextnetwk     12001/udp  IBM Enterprise
entexthigh      12002/tcp  IBM Enterprise
entexthigh      12002/udp  IBM Enterprise
entextmed       12003/tcp  IBM Enterprise
entextmed       12003/udp  IBM Enterprise
entextlow       12004/tcp  IBM Enterprise
entextlow       12004/udp  IBM Enterprise
#               12005-12171 Unassigned
hivep           12172/tcp  HiveP
hivep           12172/udp  HiveP
#               12173-12752 Unassigned
tsaf            12753/tcp  tsaf port
tsaf            12753/udp  tsaf port
#               12754-13159 Unassigned
i-zipqd         13160/tcp  I-ZIPQD
i-zipqd         13160/udp  I-ZIPQD
#               13161-13222 Unassigned
powwow-client   13223/tcp  PowWow Client
powwow-client   13223/udp  PowWow Client
powwow-server   13224/tcp  PowWow Server
powwow-server   13224/udp  PowWow Server
#               13225-13719 Unassigned
bprd            13720/tcp  BPRD Protocol
bprd            13720/udp  BPRD Protocol
bpbrm           13721/tcp  BPBRM Protocol
bpbrm           13721/udp  BPBRM Protocol
bpjava-msvc     13722/tcp  BP Java MSVC
bpjava-msvc     13722/udp  BP Java MSVC
#               13723-13781 Unassigned
bpcd            13782/tcp  VERITAS
bpcd            13782/udp  VERITAS
vopied          13783/tcp  VOPIED Protnocol
vopied          13783/udp  VOPIED Protocol
#               13784-13817 Unassigned
dsmcc-config    13818/tcp  DSMCC Config
dsmcc-config    13818/udp  DSMCC Config
dsmcc-session   13819/tcp  DSMCC Session
dsmcc-session   13819/udp  DSMCC Session
dsmcc-passthru  13820/tcp  DSMCC Pass-Thru
dsmcc-passthru  13820/udp  DSMCC Pass-Thru
dsmcc-download  13821/tcp  DSMCC Download
dsmcc-download  13821/udp  DSMCC Download
dsmcc-ccp       13822/tcp  DSMCC Channel
dsmcc-ccp       13822/udp  DSMCC Channel
```

```
pcs-pcw          2566/tcp   pcs-pcw
pcs-pcw          2566/udp   pcs-pcw
clp              2567/tcp   Cisco Line Proto
clp              2567/udp   Cisco Line Proto
spamtrap         2568/tcp   SPAM TRAP
spamtrap         2568/udp   SPAM TRAP
sonuscallsig     2569/tcp   Sonus Call Sign
sonuscallsig     2569/udp   Sonus Call Sign
hs-port          2570/tcp   HS Port
hs-port          2570/udp   HS Port
cecsvc           2571/tcp   CECSVC
cecsvc           2571/udp   CECSVC
ibp              2572/tcp   IBP
ibp              2572/udp   IBP
trustestablish   2573/tcp   Trust Establish
trustestablish   2573/udp   Trust Establish
blockade-bpsp    2574/tcp   Blockade BPSP
blockade-bpsp    2574/udp   Blockade BPSP
hl7              2575/tcp   HL7
hl7              2575/udp   HL7
tclprodebugger   2576/tcp   TCL Pro Debugger
tclprodebugger   2576/udp   TCL Pro Debugger
scipticslsrvr    2577/tcp   Scriptics Lsrvr
scipticslsrvr    2577/udp   Scriptics Lsrvr
rvs-isdn-dcp     2578/tcp   RVS ISDN DCP
rvs-isdn-dcp     2578/udp   RVS ISDN DCP
mpfoncl          2579/tcp   mpfoncl
mpfoncl          2579/udp   mpfoncl
tributary        2580/tcp   Tributary
tributary        2580/udp   Tributary
argis-te         2581/tcp   ARGIS TE
argis-te         2581/udp   ARGIS TE
argis-ds         2582/tcp   ARGIS DS
argis-ds         2582/udp   ARGIS DS
mon              2583/tcp   MON
mon              2583/udp   MON
cyaserv          2584/tcp   cyaserv
cyaserv          2584/udp   cyaserv
netx-server      2585/tcp   NETX Server
netx-server      2585/udp   NETX Server
netx-agent       2586/tcp   NETX Agent
netx-agent       2586/udp   NETX Agent
masc             2587/tcp   MASC
masc             2587/udp   MASC
privilege        2588/tcp   Privilege
privilege        2588/udp   Privilege
quartus-tcl      2589/tcp   quartus tcl
quartus-tcl      2589/udp   quartus tcl
idotdist         2590/tcp   idotdist
idotdist         2590/udp   idotdist
maytagshuffle    2591/tcp   Maytag Shuffle
maytagshuffle    2591/udp   Maytag Shuffle
netrek           2592/tcp   netrek
netrek           2592/udp   netrek
mns-mail         2593/tcp   MNS Mail Notice
mns-mail         2593/udp   MNS Mail Notice
dts              2594/tcp   Data Base Server
dts              2594/udp   Data Base Server
worldfusion1     2595/tcp   World Fusion 1
worldfusion1     2595/udp   World Fusion 1
worldfusion2     2596/tcp   World Fusion 2
worldfusion2     2596/udp   World Fusion 2
homesteadglory   2597/tcp   Homestead Glory
homesteadglory   2597/udp   Homestead Glory
citriximaclient  2598/tcp   Citrix MA Client
citriximaclient  2598/udp   Citrix MA Client
meridiandata     2599/tcp   Meridian Data
meridiandata     2599/udp   Meridian Data
hpstgmgr         2600/tcp   HPSTGMGR
hpstgmgr         2600/udp   HPSTGMGR
discp-client     2601/tcp   discp client
#                13823-14000 Unassigned
itu-sccp-ss7     14001/tcp  ITU SCCP (SS7)
itu-sccp-ss7     14001/udp  ITU SCCP (SS7)
#                14002-16359 Unassigned
netserialext1    16360/tcp  netserialext1
netserialext1    16360/udp  netserialext1
netserialext2    16361/tcp  netserialext2
netserialext2    16361/udp  netserialext2
#                16362-16366 Unassigned
netserialext3    16367/tcp  netserialext3
netserialext3    16367/udp  netserialext3
netserialext4    16368/tcp  netserialext4
netserialext4    16368/udp  netserialext4
#                16369-16990 Unassigned
intel-rci-mp     16991/tcp  INTEL-RCI-MP
intel-rci-mp     16991/udp  INTEL-RCI-MP
#                16992-17006 Unassigned
isode-dua        17007/tcp
isode-dua        17007/udp
#                17008-17218 Unassigned
chipper          17219/tcp  Chipper
chipper          17219/udp  Chipper
#                17220-17999 Unassigned
biimenu          18000/tcp  Beckman Inc.
biimenu          18000/udp  Beckman Inc.
#                18001-18180 Unassigned
opsec-cvp        18181/tcp  OPSEC CVP
opsec-cvp        18181/udp  OPSEC CVP
opsec-ufp        18182/tcp  OPSEC UFP
opsec-ufp        18182/udp  OPSEC UFP
opsec-sam        18183/tcp  OPSEC SAM
opsec-sam        18183/udp  OPSEC SAM
opsec-lea        18184/tcp  OPSEC LEA
opsec-lea        18184/udp  OPSEC LEA
opsec-omi        18185/tcp  OPSEC OMI
opsec-omi        18185/udp  OPSEC OMI
#                18186      Unassigned
opsec-ela        18187/tcp  OPSEC ELA
opsec-ela        18187/udp  OPSEC ELA
ac-cluster       18463/tcp  AC Cluster
ac-cluster       18463/udp  AC Cluster
#                18464-18887 Unassigned
apc-necmp        18888/tcp  APCNECMP
apc-necmp        18888/udp  APCNECMP
#                18889-19190 Unassigned
opsec-uaa        19191/tcp  opsec-uaa
opsec-uaa        19191/udp  opsec-uaa
#                19192-19282 Unassigned
keysrvr          19283/tcp  Key Server
keysrvr          19283/udp  Key Server
#                19284-19314 Unassigned
keyshadow        19315/tcp  Key Shadow
keyshadow        19315/udp  Key Shadow
#                19316-19409 Unassigned
hp-sco           19410/tcp  hp-sco
hp-sco           19410/udp  hp-sco
hp-sca           19411/tcp  hp-sca
hp-sca           19411/udp  hp-sca
hp-sessmon       19412/tcp  HP-SESSMON
hp-sessmon       19412/udp  HP-SESSMON
#                19413-19540 Unassigned
jcp              19541/tcp  JCP Client
#                19542-19999 Unassigned
dnp              20000/tcp  DNP
dnp              20000/udp  DNP
#                20001-20669 Unassigned
track            20670/tcp  Track
track            20670/udp  Track
#                20671-20998 Unassigned
athand-mmp       20999/tcp  At Hand MMP
athand-mmp       20999/udp  AT Hand MMP
```

```
discp-client    2601/udp   discp client
discp-server    2602/tcp   discp server
discp-server    2602/udp   discp server
servicemeter    2603/tcp   Service Meter
servicemeter    2603/udp   Service Meter
nsc-ccs         2604/tcp   NSC CCS
nsc-ccs         2604/udp   NSC CCS
nsc-posa        2605/tcp   NSC POSA
nsc-posa        2605/udp   NSC POSA
netmon          2606/tcp   Dell Netmon
netmon          2606/udp   Dell Netmon
connection      2607/tcp   Dell Connection
connection      2607/udp   Dell Connection
wag-service     2608/tcp   Wag Service
wag-service     2608/udp   Wag Service
system-monitor  2609/tcp   System Monitor
system-monitor  2609/udp   System Monitor
versa-tek       2610/tcp   VersaTek
versa-tek       2610/udp   VersaTek
lionhead        2611/tcp   LIONHEAD
lionhead        2611/udp   LIONHEAD
qpasa-agent     2612/tcp   Qpasa Agent
qpasa-agent     2612/udp   Qpasa Agent
smntubootstrap  2613/tcp   SMNTUBootstrap
smntubootstrap  2613/udp   SMNTUBootstrap
neveroffline    2614/tcp   Never Offline
neveroffline    2614/udp   Never Offline
firepower       2615/tcp   firepower
firepower       2615/udp   firepower
appswitch-emp   2616/tcp   appswitch-emp
appswitch-emp   2616/udp   appswitch-emp
cmadmin         2617/tcp   Clinical Context
cmadmin         2617/udp   Clinical Context
priority-e-com  2618/tcp   Priority E-Com
priority-e-com  2618/udp   Priority E-Com
bruce           2619/tcp   bruce
bruce           2619/udp   bruc
lpsrecommender  2620/tcp   LPSRecommender
lpsrecommender  2620/udp   LPSRecommender
miles-apart     2621/tcp   Miles Apart
miles-apart     2621/udp   Miles Apart
metricadbc      2622/tcp   MetricaDBC
metricadbc      2622/udp   MetricaDBC
lmdp            2623/tcp   LMDP
lmdp            2623/udp   LMDP
aria            2624/tcp   Aria
aria            2624/udp   Aria
blwnkl-port     2625/tcp   Blwnkl Port
blwnkl-port     2625/udp   Blwnkl Port
gbjd816         2626/tcp   gbjd816
gbjd816         2626/udp   gbjd816
moshebeeri      2627/tcp   Moshe Beeri
moshebeeri      2627/udp   Moshe Beeri
dict            2628/tcp   DICT
dict            2628/udp   DICT
sitaraserver    2629/tcp   Sitara Server
sitaraserver    2629/udp   Sitara Server
sitaramgmt      2630/tcp   Sitara Mgrt
sitaramgmt      2630/udp   Sitara Mgr
sitaradir       2631/tcp   Sitara Dir
sitaradir       2631/udp   Sitara Dir
irdg-post       2632/tcp   IRdg Post
irdg-post       2632/udp   IRdg Post
interintelli    2633/tcp   InterIntelli
interintelli    2633/udp   InterIntelli
pk-electronics  2634/tcp   PK Electronics
pk-electronics  2634/udp   PK Electronics
backburner      2635/tcp   Back Burner
backburner      2635/udp   Back Burner
solve           2636/tcp   Solve
solve           2636/udp   Solve
#               20300-21589 Unassigned
vofr-gateway    21590/tcp  VoFR Gateway
vofr-gateway    21590/udp  VoFR Gateway
#               21591-21844 Unassigned
webphone        21845/tcp webphone
webphone        21845/udp webphone
netspeak-is     21846/tcp NetSpeak
netspeak-is     21846/udp NetSpeak
netspeak-cs     21847/tcp NetSpeak
netspeak-cs     21847/udp NetSpeak
netspeak-acd    21848/tcp NetSpeak
netspeak-acd    21848/udp NetSpeak
netspeak-cps    21849/tcp NetSpeak
netspeak-cps    21849/udp NetSpeak
#               21850-21999 Unassigned
snapenetio      22000/tcp SNAPenetIO
snapenetio      22000/udp SNAPenetIO
optocontrol     22001/tcp OptoControl
optocontrol     22001/udp OptoControl
#               22002-22272 Unassigned
wnn6            22273/tcp wnn6
wnn6            22273/udp wnn6
#               22556-22799 Unassigned
aws-brf         22800/tcp Telerate LAN
aws-brf         22800/udp Telerate LAN
#               22801-22950 Unassigned
brf-gw          22951/tcp Telerate WAN
brf-gw          22951/udp Telerate WAN
#               22952-23999 Unassigned
med-ltp         24000/tcp med-ltp
med-ltp         24000/udp med-ltp
med-fsp-rx      24001/tcp med-fsp-rx
med-fsp-rx      24001/udp med-fsp-rx
med-fsp-tx      24002/tcp med-fsp-tx
med-fsp-tx      24002/udp med-fsp-tx
med-supp        24003/tcp med-supp
med-supp        24003/udp med-supp
med-ovw         24004/tcp med-ovw
med-ovw         24004/udp med-ovw
med-ci          24005/tcp med-ci
med-net-svc     24006/tcp med-net-svc
med-net-svc     24006/udp med-net-svc
#               24007-24385 Unassigned
intel_rci       24386/tcp Intel RCI
intel_rci       24386/udp Intel RCI
#               24387-24553 Unassigned
binkp           24554/tcp BINKP
binkp           24554/udp BINKP
#               24555-24999 Unassigned
icl-twobase1    25000/tcp icl-twobase1
icl-twobase1    25000/udp icl-twobase1
icl-twobase2    25001/tcp icl-twobase2
icl-twobase2    25001/udp icl-twobase2
icl-twobase3    25002/tcp icl-twobase3
icl-twobase3    25002/udp icl-twobase3
icl-twobase4    25003/tcp icl-twobase4
icl-twobase4    25003/udp icl-twobase4
icl-twobase5    25004/tcp icl-twobase5
icl-twobase5    25004/udp icl-twobase5
icl-twobase6    25005/tcp icl-twobase6
icl-twobase6    25005/udp icl-twobase6
icl-twobase7    25006/tcp icl-twobase7
icl-twobase7    25006/udp icl-twobase7
icl-twobase8    25007/tcp icl-twobase8
icl-twobase8    25007/udp icl-twobase8
icl-twobase9    25008/tcp icl-twobase9
icl-twobase9    25008/udp icl-twobase9
icl-twobase10   25009/tcp icl-twobase10
icl-twobase10   25009/udp icl-twobase10
#               25010-25792 Unassigned
vocaltec-hos    25793/tcp  Vocaltec
```

```
imdocsvc        2637/tcp   Import Document
imdocsvc        2637/udp   Import Document
sybaseanywhere  2638/tcp   Sybase Anywhere
sybaseanywhere  2638/udp   Sybase Anywhere
aminet          2639/tcp   AMInet
aminet          2639/udp   AMInet
sai_sentlm      2640/tcp   Sabbagh
sai_sentlm      2640/udp   Sabbagh
hdl-srv         2641/tcp   HDL Server
hdl-srv         2641/udp   HDL Server
tragic          2642/tcp   Tragic
tragic          2642/udp   Tragic
gte-samp        2643/tcp   GTE-SAMP
gte-samp        2643/udp   GTE-SAMP
travsoft-ipx-t  2644/tcp   Travsoft IPX
travsoft-ipx-t  2644/udp   Travsoft IPX
novell-ipx-cmd  2645/tcp   Novell IPX CMD
novell-ipx-cmd  2645/udp   Novell IPX CMD
and-lm          2646/tcp   AND Licence Mgr
and-lm          2646/udp   AND License Mgr
syncserver      2647/tcp   SyncServer
syncserver      2647/udp   SyncServer
upsnotifyprot   2648/tcp   Upsnotifyprot
upsnotifyprot   2648/udp   Upsnotifyprot
vpsipport       2649/tcp   VPSIPPORT
vpsipport       2649/udp   VPSIPPORT
eristwoguns     2650/tcp   eristwoguns
eristwoguns     2650/udp   eristwoguns
ebinsite        2651/tcp   EBInSite
ebinsite        2651/udp   EBInSite
interpathpanel  2652/tcp   InterPathPanel
interpathpanel  2652/udp   InterPathPanel
sonus           2653/tcp   Sonus
sonus           2653/udp   Sonus
corel_vncadmin  2654/tcp   Corel VNC Admin
corel_vncadmin  2654/udp   Corel VNC Admin
unglue          2655/tcp   UNIX Nt Glue
unglue          2655/udp   UNIX Nt Glue
kana            2656/tcp   Kana
kana            2656/udp   Kana
sns-dispatcher  2657/tcp   SNS Dispatcher
sns-dispatcher  2657/udp   SNS Dispatcher
sns-admin       2658/tcp   SNS Admin
sns-admin       2658/udp   SNS Admin
sns-query       2659/tcp   SNS Query
sns-query       2659/udp   SNS Query
gcmonitor       2660/tcp   GC Monitor
gcmonitor       2660/udp   GC Monitor
olhost          2661/tcp   OLHOST
olhost          2661/udp   OLHOST
bintec-capi     2662/tcp   BinTec-CAPI
bintec-capi     2662/udp   BinTec-CAPI
bintec-tapi     2663/tcp   BinTec-TAPI
bintec-tapi     2663/udp   BinTec-TAPI
command-mq-gm   2664/tcp   Command MQ GM
command-mq-gm   2664/udp   Command MQ GM
command-mq-pm   2665/tcp   Command MQ PM
command-mq-pm   2665/udp   Command MQ PM
extensis        2666/tcp   extensis
extensis        2666/udp   extensis
vocaltec-hos    25793/udp  Vocaltec
#               25794-25999 Unassigned
quake           26000/tcp  quake
quake           26000/udp  quake
#               26001-26207 Unassigned
wnn6-ds         26208/tcp  wnn6-ds
wnn6-ds         26208/udp  wnn6-ds
#               26209-26999 Unassigned
flex-lm         27000-27009 FLEX LM (1-10)
#               27008-27998 Unassigned
tw-auth-key     27999/tcp  TW
tw-auth-key     27999/udp  Attribute
#               28000-32767 Unassigned
filenet-tms     32768/tcp  Filenet TMS
filenet-tms     32768/udp  Filenet TMS
filenet-rpc     32769/tcp  Filenet RPC
filenet-rpc     32769/udp  Filenet RPC
filenet-nch     32770/tcp  Filenet NCH
filenet-nch     32770/udp  Filenet NCH
#               32771-33433 Unassigned
traceroute      33434/tcp  traceroute use
traceroute      33434/udp  traceroute use
#               33435-36864 Unassigned
kastenxpipe     36865/tcp  KastenX Pipe
kastenxpipe     36865/udp  KastenX Pipe
#               36866-40840 Unassigned
cscp            40841/tcp  CSCP
cscp            40841/udp  CSCP
#               40842-43187 Unassigned
rockwell-encap  44818/tcp  Rockwell Encaps
rockwell-encap  44818/udp  Rockwell Encaps
#               44819-45677 Unassigned
eba             45678/tcp  EBA PRISE
eba             45678/udp  EBA PRISE
#               45679-45965 Unassigned
ssr-servermgr   45966/tcp  SSRServerMgr
ssr-servermgr   45966/udp  SSRServerMgr
#               45967-47556 Unassigned
dbbrowse        47557/tcp  Databeam Corp
dbbrowse        47557/udp  Databeam Corpo
#               47558-47623 Unassigned
directplaysrvr  47624/tcp  Direct Play Serv
directplaysrvr  47624/udp  Direct Play Serv
#               47625-47805 Unassigned
ap              47806/tcp  ALC Protocol
ap              47806/udp  ALC Protocol
#               47807      Unassigned
bacnet          47808/tcp  Building Aut
bacnet          47808/udp  Building Aut
#               47809-47999 Unassigned
nimcontroller   48000/tcp  Nimbus Control
nimcontroller   48000/udp  Nimbus Control
nimspooler      48001/tcp  Nimbus Spooler
nimspooler      48001/udp  Nimbus Spooler
nimhub          48002/tcp  Nimbus Hub
nimhub          48002/udp  Nimbus Hub
nimgtw          48003/tcp  Nimbus Gateway
nimgtw          48003/udp  Nimbus Gateway
#               48004-49151 Unassigne
```

Trojan Ports:

This is a list of ports commonly used by **Trojan horses**. Please note that all ports are TCP unless UDP is stated.

```
Decimal Trojan(s)
------- ---------

    2  - Death
    21 - Back Construction, Blade Runner, Doly Trojan, Fore, FTP trojan, Invisible
    FTP, Larva, MBT, Motiv, Net Administrator, Senna Spy FTP Server, WebEx, WinCrash
    23 - Tiny Telnet Server, Truva Atl
    25 - Aji, Antigen, Email Password Sender, Gip, Happy 99, I Love You, Kuang 2,
    Magic Horse, Moscow Email Trojan, Naebi, NewApt, ProMail trojan, Shtrilitz,
    Stealth, Tapiras, Terminator, WinPC, WinSpy
    31 - Agent 31, Hackers Paradise, Masters Paradise
    41 - DeepThroat
    48 - DRAT
    50 - DRAT
    59 - DMSetup
    79 - Firehotcker
    80 - Back End, Executor, Hooker, RingZero
    99 - Hidden Port
    110 - ProMail trojan
    113 - Invisible Identd Deamon, Kazimas
    119 - Happy 99
    121 - JammerKillah
    123 - Net Controller
    133 - Farnaz, port 146 - Infector
    146 - Infector(UDP)
    170 - A-trojan
    421 - TCP Wrappers
    456 - Hackers Paradise
    531 - Rasmin
    555 - Ini-Killer, NeTAdministrator, Phase Zero, Stealth Spy
    606 - Secret Service
    666 - Attack FTP, Back Construction, NokNok, Cain & Abel, Satanz Backdoor,
    ServeU, Shadow Phyre
    667 - SniperNet
    669 - DP Trojan
    692 - GayOL
    777 - Aim Spy
    808 - WinHole
    911 - Dark Shadow
    999 - DeepThroat, WinSatan
    1000 - Der Spacher 3
    1001 - Der Spacher 3, Le Guardien, Silencer, WebEx
    1010 - Doly Trojan
    1011 - Doly Trojan
    1012 - Doly Trojan
    1015 - Doly Trojan
    1016 - Doly Trojan
    1020 - Vampire
    1024 - NetSpy
    1042 - Bla
    1045 - Rasmin
    1050 - MiniCommand
    1080 - WinHole
    1081 - WinHole
    1082 - WinHole
    1083 - WinHole
    1090 - Xtreme
    1095 - RAT
    1097 - RAT
    1098 - RAT
    1099 - BFevolution, RAT
    1170 - Psyber Stream Server, Streaming Audio trojan, Voice
    1200 - NoBackO (UDP)
    1201 - NoBackO (UDP)
    1207 - SoftWAR
    1212 - Kaos
```

1225 - Scarab
1234 - Ultors Trojan
1243 - BackDoor-G, SubSeven, SubSeven Apocalypse, Tiles
1245 - VooDoo Doll
1255 - Scarab
1256 - Project nEXT
1269 - Mavericks Matrix
1313 - NETrojan
1338 - Millenium Worm
1349 - BO DLL (UDP)
1492 - FTP99CMP
1509 - Psyber Streaming Server
1524 - Trinoo
1600 - Shivka-Burka
1777 - Scarab
1807 - SpySender
1966 - Fake FTP
1969 - OpC BO
1981 - Shockrave
1999 - BackDoor, TransScout
2000 - Der Spaeher 3, Insane Network, TransScout
2001 - Der Spaeher 3, TransScout, Trojan Cow
2002 - TransScout
2003 - TransScout
2004 - TransScout
2005 - TransScout
2023 - Ripper
2080 - WinHole
2115 - Bugs
2140 - Deep Throat, The Invasor
2155 - Illusion Mailer
2283 - HVL Rat5
2300 - Xplorer
2565 - Striker
2583 - WinCrash
2600 - Digital RootBeer
2716 - The Prayer
2773 - SubSeven
2801 - Phineas Phucker
3000 - Remote Shutdown
3024 - WinCrash
3128 - RingZero
3129 - Masters Paradise
3150 - Deep Throat, The Invasor
3456 - Teror Trojan
3459 - Eclipse 2000, Sanctuary
3700 - Portal of Doom
3791 - Eclypse
3801 - Eclypse (UDP)
4000 - Skydance
4092 - WinCrash
4242 - Virtual hacking Machine
4321 - BoBo
4444 - Prosiak, Swift remote
4567 - File Nail
4590 - ICQTrojan
5000 - Bubbel, Back Door Setup, Sockets de Troie
5001 - Back Door Setup, Sockets de Troie
5010 - Solo
5011 - One of the Last Trojans (OOTLT)
5031 - NetMetropolitan
5031 - NetMetropolitan
5321 - Firehotcker
5343 - wCrat
5400 - Blade Runner, Back Construction
5401 - Blade Runner, Back Construction
5402 - Blade Runner, Back Construction
5550 - Xtcp
5512 - Illusion Mailer
5555 - ServeMe
5556 - BO Facil

```
5557 - BO Facil
5569 - Robo-Hack
5637 - PC Crasher
5638 - PC Crasher
5742 - WinCrash
5882 - Y3K RAT (UDP)
5888 - Y3K RAT
6000 - The Thing
6006 - The Thing
6272 - Secret Service
6400 - The Thing
6667 - Schedule Agent
6669 - Host Control, Vampyre
6670 - DeepThroat, BackWeb Server, WinNuke eXtreame
6711 - SubSeven
6712 - Funny Trojan, SubSeven
6713 - SubSeven
6723 - Mstream
6771 - DeepThroat
6776 - 2000 Cracks, BackDoor-G, SubSeven
6838 - Mstream (UDP)
6912 - Shit Heep (not port 69123!)
6939 - Indoctrination
6969 - GateCrasher, Priority, IRC 3, NetController
6970 - GateCrasher
7000 - Remote Grab, Kazimas, SubSeven
7001 - Freak88
7215 - SubSeven
7300 - NetMonitor
7301 - NetMonitor
7306 - NetMonitor
7307 - NetMonitor
7308 - NetMonitor
7424 - Host Control
7424 - Host Control (UDP)
7789 - Back Door Setup, ICKiller
7983 - Mstream
8080 - RingZero
8787 - Back Orifice 2000
8897 - HackOffice
8988 - BacHack
8989 - Rcon
9000 - Netministrator
9325 - Mstream (UDP)
9400 - InCommand
9872 - Portal of Doom
9873 - Portal of Doom
9874 - Portal of Doom
9875 - Portal of Doom
9876 - Cyber Attacker, RUX
9878 - TransScout
9989 - iNi-Killer
9999 - The Prayer
10067 - Portal of Doom (UDP)
10085 - Syphillis
10086 - Syphillis
10101 - BrainSpy
10167 - Portal of Doom (UDP)
10528 - Host Control
10520 - Acid Shivers
10607 - Coma
10666 - Ambush (UDP)
11000 - Senna Spy
11050 - Host Control
11051 - Host Control
11223 - Progenic trojan, Secret Agent
12076 - Gjamer
12223 - Hack´99 KeyLogger
12345 - GabanBus, My Pics, NetBus, Pie Bill Gates, Whack Job, X-bill
12346 - GabanBus, NetBus, X-bill
12349 - BioNet
```

```
12361 - Whack-a-mole
12362 - Whack-a-mole
12623 - DUN Control (UDP)
12624 - Buttman
12631 - WhackJob
12754 - Mstream
13000 - Senna Spy
13010 - Hacker Brazil
15092 - Host Control
15104 - Mstream
16660 - Stacheldracht
16484 - Mosucker
16772 - ICQ Revenge
16969 - Priority
17166 - Mosaic
17300 - Kuang2 The Virus
17777 - Nephron
18753 - Shaft (UDP)
19864 - ICQ Revenge
20001 - Millennium
20002 - AcidkoR
20034 - NetBus 2 Pro, NetRex, Whack Job
20203 - Chupacabra
20331 - Bla
20432 - Shaft
20432 - Shaft (UDP)
21544 - GirlFriend, Kidterror, Schwindler, WinSp00fer
22222 - Prosiak
23023 - Logged
23432 - Asylum
23456 - Evil FTP, Ugly FTP, Whack Job
23476 - Donald Dick
23476 - Donald Dick (UDP)
23477 - Donald Dick
26274 - Delta Source (UDP)
26681 - Spy Voice
27374 - SubSeven
27444 - Trinoo (UDP)
27573 - SubSeven
27665 - Trinoo
29104 - Host Control
29891 - The Unexplained (UDP)
30001 - TerrOr32
30029 - AOL Trojan
30100 - NetSphere
30101 - NetSphere
30102 - NetSphere
30103 - NetSphere
30103 - NetSphere (UDP)
30133 - NetSphere
30303 - Sockets de Troie
30947 - Intruse
30999 - Kuang2
31335 - Trinoo (UDP)
31336 - Bo Whack, ButtFunnel
31337 - ["ELEET" port] - Baron Night, BO client, BO2, Bo Facil
31337 - ["ELEET" port] - BackFire, Back Orifice, DeepBO, Freak> (UDP)
31338 - NetSpy DK, ButtFunnel
31338 - Back Orifice, DeepBO (UDP)
31339 - NetSpy DK
31666 - BOWhack
31785 - Hack´a´Tack
31787 - Hack´a´Tack
31788 - Hack´a´Tack
31789 - Hack´a´Tack (UDP)
31791 - Hack´a´Tack (UDP)
31792 - Hack´a´Tack
32100 - Peanut Brittle, Project nEXT
32418 - Acid Battery
33333 - Blakharaz, Prosiak
33577 - PsychWard
```

```
33777 - PsychWard
33911 - Spirit 2001a
34324 - BigGluck, TN
34555 - Trinoo (Windows) (UDP)
35555 - Trinoo (Windows) (UDP)
37651 - YAT
40412 - The Spy
40421 - Agent 40421, Masters Paradise
40422 - Masters Paradise
40423 - Masters Paradise
40426 - Masters Paradise
41666 - Remote Boot
41666 - Remote Boot (UDP)
44444 - Prosiak
47262 - Delta Source (UDP)
50505 - Sockets de Troie
50766 - Fore, Schwindler
51996 - Cafeini
52317 - Acid Battery 2000
53001 - Remote Windows Shutdown
54283 - SubSeven
54320 - Back Orifice 2000
54321 - School Bus
54321 - Back Orifice 2000 (UDP)
57341 - NetRaider
58339 - ButtFunnel
60000 - Deep Throat
60068 - Xzip 6000068
60411 - Connection
61348 - Bunker-Hill
61466 - Telecommando
61603 - Bunker-Hill
63485 - Bunker-Hill
65000 - Devil, Stacheldracht
65432 - The Traitor
65432 - The Traitor (UDP)
65535 - RC
```