Redes de Computadoras y Cortafuegos con GNU/Linux

Dr. Luis Gerardo de la Fraga

Departamento de Computación
Cinvestav
Correo-e: fraga@cs.cinvestav.mx

21 de mayo de 2009

Contenido

Configuración del núcleo (1/5)

Para esta distribución

```
# uname -a
Linux sigma 2.4.20-8 #1 Thu Mar 13 17:18:24 EST 2003 \
    i686 athlon i386 GNU/Linux
# cat /etc/redhat-release
Red Hat Linux release 9 (Shrike)
```

La configuración con la que se compiló el núcleo de GNU/Linux que se está usando, se encuentra en:

```
/boot/config-2.4.20-8
```

Configuración del núcleo (2/5)

Se debe instalar los fuentes del núcleo, ya sea el que viene con la distribución

`kernel-source-2.4.20-8.i386.rpm`

o la última versión estable en www.kernel.org

`linux-2.4.34.1.tar.bz2`
`linux-2.4.34.1.tar.bz2.sign`

Configuración del núcleo (3/5)

Los fuentes quedan instalados en

```
# ls -1F /usr/src
debug/
linux-2.4@
linux-2.4.34.1/
redhat/
```

Configuración del núcleo (4/5)

Si necesitásemos recompilar el núcleo de Linux, para activar alguna de función que normalmente no está activada.

```
cd /usr/src/linux-2.4/
make mrproper
cp /boot/config-2.4.20-8  .config
# Si usamos esta configuracion en una nueva versi'on
# del n'ucleo se realiza
# make oldconfig
# que solo requisir'a las respuestas a las preguntas nuevas
make xconfig    # o make menuconfig
make dep
make bzImage
make modules
make modules_install
```

Configuración del núcleo (5/5)

El núcleo está compilado en:

`/usr/src/linux-2.4/arch/i386/boot/`

en el archivo `bzImage`.

Los módulos están en

`/lib/modules/2.4.20-8/`
`/lib/modules/2.4.20-8/kernel/net/ipv4/netfilter/`

Software disponible en la pág. del curso

```
linux-2.4.34.1.tar.bz2 (30967660 bytes) (n'ucleo 2.4)
busybox-1.5.0.tar.gz   ( 1844118 bytes) (utilerias)
dnsmasq-2.38.tar.gz    (  272953 bytes) (DNS, DHCP)
dropbear-0.48.1.tar.gz ( 1473114 bytes) (servidor SSH)
```

Pasos para crear un cortafuegos sin disco

1. Necesitamos un núcleo y algunos módulos
2. Crear el sistema de archivos
3. Poner los dos puntos anteriores en algún dispositivo e instalar un arrancador (grub, por ejemplo).
4. Arrancar

Crear el sistema de archivos (1/2)

- Necesitamos algún espacio en disco duro para crearlo
- Lo llenamos con la información necesaria:
  - Estructura de directorios
  - Configurar busybox, compilarlo e instalarlo en el sistema de archivos.
  - Poner la bibliotecas compartidas necesarias para ejecutar busybox.
- Configurar el sistema operativo (/etc/passwd, /etc/rc, agregar módulos, etc.).

Crear el sistema de archivos (2/2)

- En memoria
- En la compact flash
- En disco duro en un archivo
- En disco duro en una partición
- No podemos crearla en CDROM
- Tipo ext2 (y solo de lectura)

http://www.tldp.org/HOWTO/Bootdisk-HOWTO/index.html

Crear el sistema de archivos (3/3)

En disco duro en un archivo

```
dd if=/dev/zero of=archivo bs=1k count=4096
# crear'a un archivo de 4MB de tamanio

# Se crea el sistema de archivos
mkfs.ext2 -F archivo
# Para deshabilitar el chequeo autom'atico del sistema
# de archivos y los mensajes de que el sistema debe
# checarse para buscar errores
tune2fs -c0 -i0 archivo

# Se monta el sistema de archivos creados en "archivo"
mount -o loop /mnt/k  archivo
# ya debe de estar creado el directorio /mnt/k
```

Llenar el sistema de archivos (1/3)

Aquí está un conjunto mínimo de directorios para el sistema de archivos raíz:

- /dev – Archivos de dispositivos, requeridos para E/S
- /proc – Directorio requerido para el sistema de archivos /proc
- /etc – Archivos de configuración del sistema
- /sbin – Binarios críticos del sistema
- /bin – Binarios del sistema (busybox)
- /lib – Bibliotecas compartidas para tiempo de ejecución
- /mnt – Un punto de montaje para otros discos
- /usr – Utilerias adicionales y aplicaciones

Llenar el sistema de archivos (2/3)

Esto es opcional:
Los directorios /var (temporales) y /dev (dispositivos) deben de ser de lectura y escritura, se harán en memoria RAM

```
/sbin/mkfs.minix /dev/ram1
mount -n -t minix /dev/ram1 /mnt
cp -a /dev/* /mnt/
rm -rf /mnt/lost+found
umount /mnt
mount -t minix /dev/ram1 /dev
```

Llenar el sistema de archivos (3/3)

Podemos usar el mismo esquema para el directorio /var (mantemos una copia en el disco y) :

```
/sbin/mkfs.minix /dev/ram0
mount -n -t minix /dev/ram0 /mnt
cp -a /var/* /mnt/
rm -rf /mnt/lost+found
umount /mnt
mount -t minix /dev/ram0 /var
```

Para al arranque del sistema se creen /var y /dev en memoria, los dos scripts anteriores los podemos poner en el directorio /etc/ y llamarlos desde el script /etc/rc

Configuración de dropbear

Las llaves van el `/etc/dropbear`, y el siguiente archivo va en `/etc/rc.local`

```
$ cat dropbearServer
#!/bin/sh

touch /var/log/lastlog
echo -n "Setting dropbear server.."
/usr/sbin/dropbear -w
```

Configuración de dnsmasq (1/2)

En el mismo directorio /etc/rc.local

```
$ cat dns_fw
#!/bin/sh

echo "Setting dnsmasq server.."
mkdir -p /var/lib/dhcp
/sbin/dnsmasq
#
echo  "Turning on IP Forwarding & Masquerading"
echo 1 >/proc/sys/net/ipv4/ip_forward
#
echo "Turning on firewall..."
/etc/fw.rules > /dev/null 2>&1
```

Configuración de dnsmasq (2/2)

```
$ cat dnsmasq.conf
dhcp-range=192.168.10.100,192.168.10.200,24h
dhcp-host=00:B0:D0:CF:66:C1,compu02,192.168.10.91
#
dhcp-leasefile=/var/lib/dhcp/dnsmasq.leases
```