Como Levantar un Gateway en Linux Red-Hat 7.0

Luis Gerardo de la Fraga

Resumen

En este documento se describe como instalar un gateway específicamente en la versión 7.0 de RedHat. Un *gateway* es una máquina con dos interfaces de red y que sirve para conectar dos redes diferentes. Aquí se recomienda su uso para conectar intranets (o redes con direcciones IP no válidas) a Internet, con lo que no se requieren más que unas cuantas (dos, cuatro, ocho) IP válidas para conectar cientos de máquinas (clientes) a Internet. Este mismo esquema se usa para construir firewalls con Linux.

1 Introducción

La conección de laboratorios de cómputo, salas de cómputo, etc., en donde es necesario tener acceso a Internet, esto es, en un escenario donde necesitemos instalar de unos pocos a cientos de *clientes* de Internet, *no es necesario* asignar direcciones IP válidas a cada máquina, o instalar un servidor DHCP que ofrezca direcciones IP válidas. Con una máquina Linux podemos interconectar estos clientes formando una intranet y poner esta máquina como un *gateway* para que todos los clientes "vean" Internet.

Direcciones IP *no válidas* son las especificadas en el RFC1918 [1, 2, 3] para diseñar redes privadas o intranets. Estas son $10.*.*.*$, $172.16.*.*$ a $172.31.*.*$ y $192.168.*.*$. Todos los ruteadores actuales filtran estas direcciones por lo que no se pueden accesar estas direcciones desde Internet.

El esquema aquí presentado no funciona directamente si se quiere poner una máquina para ofrecer información a Internet. Esto es un *servidor* y utiliza un esquema diferente que puede revisarse en [4].

Para más información se recomienda leer los HOWTO's [1, 2] y los papers [3, 5, 6, 3, 7].

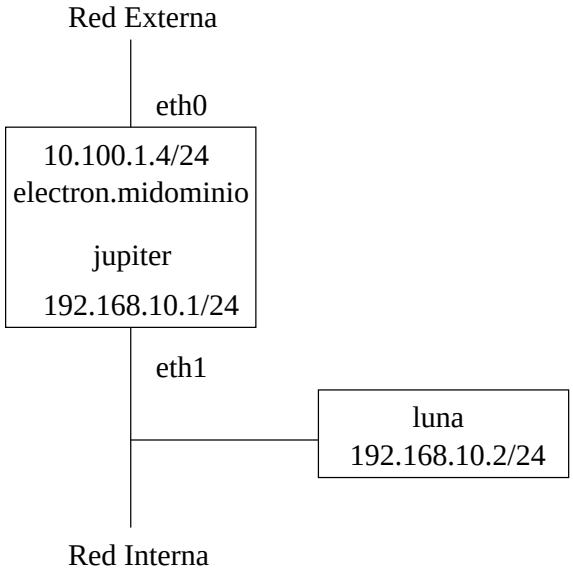

Figura 1: Esquema de uso de un gateway

2 Configuración de la Red

El esquema que se va a usar se presenta en la Fig. 1. Una máquina Linux tiene dos tarjetas ethernet, una, eth0, está conectada a Internet (aunque en la figura se muestra una dirección no válida para este dispositivo por razones obvias de seguridad, eth0 debe tener asignada una dirección IP válida) y la otra, eth1, se usa para conectar la máquina con la red privada.

En la Fig. 1, el gateway se conoce como "electron.midominio" desde Internet y como "júpiter" desde la red interna.

2.1 El gateway

Para acceder a la configuración de la red, se edita el archivo:

/etc/sysconfig/network

con el contenido:

```
NETWORKING=yes
HOSTNAME="electron.midominio"
```

```
GATEWAYDEV=eth0
# IP del gateway que se conecta a la red externa (Internet)
GATEWAY=10.100.1.254
```

A esta máquina se le instalaron 2 tarjetas de red ethernet (que corresponden a los dispositivos eth0 y eth1). La primera tarjeta se configura en el archivo

/etc/sysconfig/network-scripts/ifcfg-eth0

de la siguiente manera para que su conección sea hacia la red externa (Internet):

```
DEVICE="eth0" # Dispositivo electrónico a utilizarse (ethernet)
ONBOOT="yes"  # Comando habilitado para que se cargue en el sistema
BOOTPROTO= "none"
IPADDR=10.100.1.4     # Dirección IP asignada a la máquina
NETMASQ=255.255.255.0
```

La configuración para el dispositivo eth1, la segunda tarjeta ethernet del gateway, se guarda en el archivo:

/etc/sysconfig/network-scripts/ifcfg-eth1

que hay que crearlo. Una forma de hacerlo es copiando el archivo anterior de la forma:

```
cp ifcfg-eth0 ifcfg-eth1
```

y se edita el archivo ifcfg-eth1 con la nueva información:

```
DEVICE="eth1" # Dispositivo electrónico a utilizarse (ethernet)
ONBOOT="yes"  # Habilitado para que se cargue en el sistema
BOOTPROTO="none"
IPADDR=192.168.10.1   # Dirección IP asignada a la máquina jupiter
NETMASQ=255.255.255.0
```

Además es necesario habilitar el "IP forwarding" editando el archivo /etc/sysctl.conf :

```
net.ipv4.ip_forward=1
net.ipv4.conf.all.rp_filter=1
net.ipv4.ip_always.defrag=1
```

Finalmente, para que los cambios tomen efecto se tira y levanta la red:
`/etc/rc.d/init.d/network  restart`

2.2 Un cliente

A la máquina cliente "luna" (ver Fig. 1) se le instaló una tarjeta de red (eth0) y se editó el archivos de configuración de red /etc/sysconfig/network con algo como

```
NETWORKING=yes
HOSTNAME="luna"     # Nombre del Host
GATEWAY= 192.168.10.1 # Dirección IP del gateway de júpiter
```

Y el archivo /etc/sysconfig/network-scripts/ifcfg-eth0 con la información:

```
DEVICE="eth0"  # Dispositivo electrónico a utilizarse (ethernet)
IPADDR=192.168.10.2 # Dirección IP asignada a luna
NETMASQ=255.255.255.0 # Dirección IP-Masquerade asignada a luna
ONBOOT="yes" # Comando habilitado para que se cargue en el sistema
BOOTPROTO="none"
```

Y también, para que se reconozcan los cambios en la configuración, hay que tirar y levantar los servicios de la red con la siguiente instrucción.
`/etc/rc.d/init.d/network stop-start`

3 Levantar los Manejadores de Red

Generalmente kuzdu, el programa de redhat que reconoce el hardware nuevo, configura de forma efectiva los manejadores (drivers) de red. Sin embargo, en nuestro caso no fue así. Se tenian las tarjetas de red del gateway 3C509B de 3COM, una tuvo que configurarse con un programa en MSDOS para cambiar su interrupción y dirección de puerto (algo que no debe de preocuparse ahora con una tarjeta moderna para el bus PCI), y se editó el archivo de configuración:

/etc/conf.modules

con la información:

```
alias eth0 3c509
alias eth1 3c509
options 3c509 io=0x200,0x280 irq=7,10
```

y tuvo que editarse el script /etc/rc.d/init.d/network agregando la línea

/sbin/insmod 3c509

justo en el siguiente renglón bajo la etiqueta "start)" para que cargara los manejadores. Hay un bug en el manejador para estas tarjetas que impide que cargue automáticamente cuando la máquina bootea.

4 Levantar el Gateway

Se creo un script [2] (llamado "rc.firewall") con el objetivo de controlar las reglas que configuran propiamente al gateway. El script contiene:

```
PATH=/sbin
#################################
# Limpiamos las reglas actuales #
#################################
ipchains -F input
ipchains -F forward
ipchains -F output

#####################################
# Establecemos la política por defecto
#      Permitir entrada
#      Denegar IP Forward
#      Permitir salida
#-------------------------------------------
ipchains -P input   ACCEPT
ipchains -P forward DENY
ipchains -P output  ACCEPT
```

```
# Permitimos que todos los clientes atraviesen el gateway
ipchains -A forward -s 192.168.10.0/255.255.255.0 -j MASQ
```

Ejecutando el script se da de alta el gateway y la máquina luna debe acceder Internet.

El gateway deja pasar los protocolos TCP/IP y UDP, sin embargo para algunas aplicaciones es necesario levantar algunos módulos [5] como

ip_masq_user
ip_masq_ftp
ip_masq_raudio
ip_masq_irc

editando el archivo /etc/modules.con

4.1 Poniendo la configuración permanente

Para que el gateway se configure automáticamente cuando la máquina se reenciende, se ejecuta el comando

`/sbin/ipchains-save > /etc/sysconfig/ipchains`

Este comando crea el archivo /etc/sysconfig/ipchains con las reglas que creamos para hacer al gateway. Redhat automaticamente, si están habilitadas las ipchains, lee la configuración de ese archivo y habilita de esta forma el gateway.

Si las ipchains están deshabilitadas, se habilitan con el comando
`/sbin/chkconfig --add ipchains`
para que adicione el nuevo servicio al administrador de tareas.

Para revisar que se guardaron las reglas se ejecuta

`/etc/rc.d/init.d/ipchains stop /etc/rc.d/init.d/ipchains start`

Referencias

[1] www.linuxdoc.org. *IPCHAINS–HOWTO.*

[2] www.linuxdoc.org. *IP–Masquerade–HOWTO.*

[3] J.D. Blair and L. Grinzo. Connected to the net. *Linux Magazine*, 2(5):50–59, 2000. www.linux-mag.com.

[4] M. Bauer. Designing and using dmz networks to protect internet servers. *Linux Journal*, (83):27–36, March 2001.

[5] L. Teo. Setting up a linux gateway. *Linux Journal*, (72):86–88, April 2000. www.linuxjournal.com.

[6] P.F. Crow. The linux home network. *Linux Journal*, (72):80–84, April 2000.

[7] C. Easwaran. Linux apprentice: A heterogeneous linux/windows 95 home network. *Linux Journal*, (76):62–67, August 2000.