Bridge+Cortafuegos Mini-COMO

Peter Breuer, *ptb@it.uc3m.es* v1.2, 19 Diciembre 1997

Configuración de un sistema en el que coexista un cortafuegos con *bridging* de interfaces de red

Índice General

1 Introducción **2**

2 ¿Qué y por qué (y cómo)? **2**

2.1 Qué 2

2.2 Por qué 2

2.3 ¿Cómo? 2

3 *Bridging* **3**

3.1 Software 3

3.2 Lecturas previas. 3

3.3 Configuración del arranque 3

3.4 Configuración del kernel 4

3.5 Direcciones de red 4

3.6 Rutado de red 5

3.7 Configuración de la tarjeta 6

3.8 Rutado adicional 6

3.9 Configuración del puente 7

3.10 Probarlo 7

3.11 Comprobaciones 7

4 Cortafuegos **8**

4.1 Software y lecturas 8

4.2 Comprobaciones preliminares 8

4.3 Reglas por defecto 9

4.4 Huecos por dirección 9

4.5 Huecos por protocolo 9

4.6 Comprobaciones 10

5 Anexo: El INSFLUG **10**

1 Introducción

Debería consultar el documento *Bridging mini-HOWTO*, *ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini/Bridg* por Chris Cole para obtener otra perspectiva de este TEMA. Su dirección es *chris@polymer.uakron.edu*. La versión de su COMO en la que he basado este documento es la 1.03, con fecha del 23 de Agosto de 1996.

2 ¿Qué y por qué (y cómo)?

2.1 Qué

Un puente es un cable inteligente que conecta dos tarjetas de red. Un cortafuegos es un aislante inteligente.

2.2 Por qué

Puede querer un puente cuando tenga varios ordenadores:

1. para ahorrar el precio de un nuevo concentrador si resulta tener una tarjeta ethernet de sobra.
2. para ahorrarse la molestia de aprender sobre reenvío IP y otros trucos similares cuando **ya tiene** dos tarjetas en su ordenador.
3. Para evitar el trabajo de mantenimiento cuando las cosas cambien en el futuro.

«Varios ordenadores» pueden ser tan pocos como tres si están rutando o puenteando o simplemente moviéndose por la habitación con frecuencia. También puede querer un puente sólo por la diversión de averiguar qué es lo que hace. De hecho esto (2 (2)) es para lo que yo lo quería.

Si realmente está interesado en el punto 1 (1), debe ser uno de los pocos. Lea los documentos Redes-En-Linux-Como, *http://www.insflug.org/documentos/Redes-En-Linux-Como/* y Serie-Como *http://www.insflug.org/documentos/Serie-Como/* en busca de trucos mejores.

Querrá un cortafuegos si

1. trata de proteger su red de accesos externos, o
2. quiere denegar el acceso al mundo exterior desde su red.

Curiosamente yo necesitaba el punto 2 (2) también aquí. La política de mi universidad es que no debemos actuar como proveedores de servicios internet a los pregraduados.

2.3 ¿Cómo?

Comencé haciendo puente entre las tarjetas de red de un cortafuegos, y acabé haciendo un cortafuegos sin quitar el puente. Parece funcionar y es más flexible que cualquiera de las configuraciones por sí solas. Puedo tirar el firewall y seguir haciendo puente o tirar el puente cuando quiero ser más prudente.

Supongo que el código del puente está justo encima del código de la capa física y que el código del cortafuegos está una capa más arriba, así que el puente y el cortafuegos actúan como si estuvieran ejecutando juntos, «secuencialmente» y no «en paralelo» (¡vaya!). diagrama:

```
-> Entrada-puente -> Entrada-cortafuegos -> Kernel -> Salida-cortafuegos -> Salida-puente ->
```

No hay otra manera de explicar cómo una máquina puede ser «conductor» y «aislante» a la vez. Hay varias advertencias sobre esto, pero las detallaré mas tarde. Básicamente deberá rutar los paquetes que quiera sean considerados por el firewall. De cualquier manera, parece funcionar bien de manera conjunta. Esto es lo que hará...

3 *Bridging*

3.1 Software

Obtenga la utilidad de configuración de puentes *ftp://shadow.cabi.net/pub/Linux/BRCFG.tgz* de las páginas personales de Alan Cox. Esta es la misma referencia que encuentra en el documento de Chris. No me había dado cuenta de que era una URL de un ftp y no de http ...

3.2 Lecturas previas.

Lea el *Multiple Ethernet HOWTO*, *ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/mini/Multiple-Ethernet* si quiere asesoramiento sobre cómo configurar más de una tarjeta de red en su máquina.

En el *BootPrompt HOWTO* *ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/BootPrompt-HOWTO* podrá encontrar aún más detalles de la magia involucrada en el proceso de arranque.

Puede escapar de la lectura del Redes-En-Linux-Como *http://www.insflug.org/documentos/Redes-En-Linux-Como/*. Es una lectura bien larga, y tendrá que seleccionar de ella los detalles que necesite.

3.3 Configuración del arranque

El material de lectura anterior le enseñará lo que necesita para preparar el kernel para reconocer un segundo dispositivo ethernet en el arranque, por ejemplo añadiendo los siguiente a `/etc/lilo.conf`, y volviendo a ejecutar `lilo`:

```
append = "ether=0,0,eth1"
```

Observe el "eth1". "eth0" es la primera tarjeta. "eth1" es la segunda tarjeta. Puede añadir los parámetros de arranque que quiera a la línea que lilo le ofrece. Esto es para tres tarjetas:

```
linux ether=0,0,eth1 ether=0,0,eth2
```

Yo uso `loadlin` para arrancar mi kernel desde DOS:

```
loadlin.exe c:\vmlinuz root=/dev/hda3 ro ether=0,0,eth1 ether=0,0,eth2
```

Fíjese que este truco obliga al kernel a sondear direcciones en el arranque. Esto no ocurrirá si carga los controladores ethernet como **módulos** (por seguridad, ya que la orden de sondeo no puede ser determinada) así que si usa módulos tendrá que añadir los parámetros de IRQ y puerto apropiados para el controlador específicamente en su fichero `/etc/conf.modules`. Yo por lo menos tengo

```
alias eth0 3c509
alias eth1 de620
options 3c509 irq=5 io=0x210
options de620 irq=7 bnc=1
```

Puede averiguar está usando módulos mediante `ps -aux` para ver si se está ejecutando `kerneld` y comprobando si hay archivos `.o` en algún subdirectorio del directorio `/lib/modules`. Necesita el que el directorio se llame como le diga `uname -r`. Si tiene `kerneld` y/o tiene algún archivo como `loquesea.o`, edite `/etc/conf.modules` y lea cuidadosamente la página del manual de `depmod`.

Tenga en cuenta también que hasta hace poco (kernel `2.0.25`) el controlador `3c509` no podía ser usado para más de una tarjeta si era usado como módulo. He visto un parche por ahí que soluciona esto. Puede que esté integrado en el kernel cuando lea este documento.

3.4 Configuración del kernel

Recompile el kernel con bridging activado.

```
CONFIG_BRIDGE=y
```

Yo compilé con el cortafuegos, reenvío IP, enmascaramiento y lo demás activado. Esto es sólo si quiere cortafuegos...

```
CONFIG_FIREWALL=y
CONFIG_NET_ALIAS=y
CONFIG_INET=y
CONFIG_IP_FORWARD=y
CONFIG_IP_MULTICAST=y
CONFIG_IP_FIREWALL=y
CONFIG_IP_FIREWALL_VERBOSE=y
CONFIG_IP_MASQUERADE=y
```

En realidad no necesita todo esto. Lo que sí necesita, además de esto, es la configuación normal de la red:

```
CONFIG_NET=y
```

y no creo que necesite preocuparse de ninguna de las demás opciones de red. Yo tengo opciones sin compilar dentro del kernel disponibles como módulos que puedo añadir más tarde.

Instale el nuevo kernel, vuelva a ejecutar `lilo` y rearranque con el kernel nuevo. ¡No debería haber cambios hasta ahora!

3.5 Direcciones de red

Chris dice que un puente no debería tener dirección IP, pero esta no es la configuración que describo aquí.

Seguro que querrá la máquina para conectarse a la red, así que va a necesitar una dirección y necesita asegurarse de que tiene el dispositivo de loopback activado de la manera normal, de tal forma que sus programas puedan hablar con los lugares que se supone deberían poder hablar. Si la dirección loopback no está activada, el servicio de resolución de nombres, y otros podrían no funcionar adecuadamente. Vea el Redes-En-Linux-Como (*http://www.insflug.org/documentos/Redes-En-Linux-Como/*), aunque la configuración estándard debería haber hecho esto:

```
ifconfig lo 127.0.0.1
route add -net 127.0.0.0
```

Va a necesitar dar direcciones a sus tarjetas de red. He modificado el archivo `/etc/rc.d/rc.inet1` de mi slackware (`3.x`) para configurar dos tarjetas y usted debería buscar en su archivo de configuración la manera de doblar o triplicar el número de instrucciones. Suponga que usted tiene direcciones en

```
192.168.2.100
```

(esto es en el espacio de direcciones reservado para redes privadas, pero no se preocupe, no va a hacerle daño a nadie si usa esta dirección por error) así que probablemente ya tenga una línea como

```
ifconfig eth0 192.168.2.100 netmask 255.255.255.0 metric 1
```

en su configuración. Lo primero que probablemente quiera hacer es limitar el espacio de direcciones que alcance esta tarjeta a la mitad, de tal forma que pueda en algún momento puentear o hacer cortafuegos con las dos mitades. Añada pues una línea que reduzca la máscara para direccionar un número menor de máquinas:

```
ifconfig eth0 netmask 255.255.255.128
```

Intente esto también. Limita a la tarjeta a, como mucho, las direcciones entre `.0` y `.127`.

Ahora puede configurar su segunda tarjeta en la otra mitad del espacio de direcciones local. Asegúrese que nadie está usando ya las direcciones. Por simetria, yo lo pongo en *228=128+100*. Cualquier dirección funcionará tan pronto como no esté en la máscara de la otra tarjeta. Bueno, seguramente. Evite direcciones especiales como `.0`, `.1`, `.128` etc. a no ser que de verdad sepa qué hace.

```
ifconfig eth1 192.168.2.228 netmask 255.255.255.128 metric 1
```

Esto evita que la segunda tarjeta direccione entre .128 and .255.

3.6 Rutado de red

Aquí es donde tengo que anunciar las salvedades en el esquema de puenteado y cortafuegos: no puede hacer cortafuegos con paquetes que no se ruten. No hay rutas, no hay cortafuegos. Al menos esto es verdad en el kernel `2.0.30` y en kernel más recientes. Los filtros para el cortafuegos están estrechamente relacionados con el código de reenvío IP.

Esto no significa que no pueda hacer puentes. Puede hacer un puente entre dos tarjetas y hacer cortafuegos con ellas desde una tercera. Puede tener dos tarjetas y hacer un cortafuegos con ellas contra una dirección IP externa como un router cercano, siempre y cuando el router sea rutado por usted hasta una de las tarjetas.

En otras palabras, ya que voy a hacer un cortafuegos quiero controlar con precisión el destino físico de algunos paquetes.

Tengo la pequeña red de máquinas conectadas a un concentrador que cuelga de eth0, por lo que configuro ahí una red:

```
route add -net 192.168.2.128 netmask 255.255.255.128 dev eth0
```

El `.128` sería `.0` si tuviera una clase C completa allí. No la tengo, por definición, ya que he partido a la mitad el espacio de direcciones. El `dev eth0` es innecesario porque las direcciones de las tarjetas caen en la máscara, pero podría ser necesario en su caso. Uno puede necesitar más de una tarjeta en esta subred (127 máquinas en un segmento) pero estas tarjetas serían puenteadas bajo la misma máscara de red, de tal forma que aparecen como una ante el código de rutado.

En la otra tarjeta tengo un cable directo a un router grande en el que confío.

```
                                                  cliente 129
          --                                         |    --
cliente 1   \       .0             .128              |   /   red 1
cliente 2 --- Hub - eth0 - Kernel - eth1 - Hub - Router ---  red 2
cliente 3 __/       .100           .228           .2 |   \__ red 3
                                                     |
                                                  cliente 254
```

Yo pongo la dirección del router a la tarjeta como una ruta fija («estática») porque si no caería entre la máscara de las primeras tarjetas y el kernel se confundiría al mandar paquetes al router grande. Voy a hacer cortafuegos con estos paquetes, y esta es otra razón por la que rutarlos específicamente.

```
route add 192.168.2.2 dev eth1
```

No los necesito, ya que no tengo más máquinas en esa mitad del espacio de direcciones, pero declaro una red también en la segunda tarjeta. Separar mis interfaces en dos grupos mediante el rutado me permitirá hacer unas reglas de cortafuegos muy estrictas si lo necesito, pero puede escapar con mucho menos rutado específico que el aquí expuesto.

```
route add -net 192.168.2.128 netmask 255.255.255.128 dev eth1
```

Necesito también enviar todos los paquetes no locales al mundo, así que le diré al kernel que se los mande al router grande.

```
route add default gw 192.168.2.2
```

3.7 Configuración de la tarjeta

La mayoría de lo que ha visto es configuración estándard de la red, pero estamos puenteando, así que también tenemos que escuchar paquetes en ambas tarjetas que no se dirijan a nosotros. Lo que sigue debe ir al fichero de configuración de red.

```
ifconfig promisc eth0
ifconfig promisc eth1
```

La página del manual dice que `allmulti` equivale a `promisc`, pero a mi no me funcionó.

3.8 Rutado adicional

Una cosa de la que me di cuenta era que tuve que poner al menos la segunda tarjeta en un modo en el que respondiera a las preguntas del router grande sobre qué máquinas escondía en mi red local..

```
ifconfig arp eth1
```

Por si acaso, le hice lo mismo a la otra tarjeta.

```
ifconfig arp eth0
```

3.9 Configuración del puente

Active el puenteo, también en su archivo de configuración de la red:

```
brcfg -enable
```

Debe haber probado esto extensivamente en pruebas reales, por supuesto. El configurador del puente mostrará algunos números. Puede experimentar con conectando y desconectando los puertos uno cada vez.

```
brcfg -port 0 -disable/-enable
brcfg -port 1 -disable/-enable
```

Para comprobar el estado en cualquier momento, ejecute

```
brcfg
```

sin parámetros. Verá cómo el puente escucha, aprende y hace el reenvío. (No entiendo por qué el código repite las mismas direcciones físicas en mis dos tarjetas, pero no importa, el HOWTO de Chris dice que es así)

3.10 Probarlo

Si todo funciona como es debido, pruebe su propio archivo de comandos de configuración tirando abajo ambas tarjetas y luego ejecutándolo:

```
ifconfig eth0 down ifconfig eth1 down
/etc/rc.d/rc.inet1
```

Con un poco de suerte los varios subsistemas (NFS, `ypbind`, etc.) ni se enterarán. **¡No intente esto a no ser que esté delante del teclado!**

Si quiere ser aún más cuidadoso, mate tantos demonios primero como pueda, y desmonte los directorios nfs. Lo peor que puede pasar es que tenga que resetear en modo monousuario (pasando el parámetro `single` a `lilo` o `loadlin`), y deshacer los cambios antes de rearrancar con las cosas como estaban antes de que empezara.

3.11 Comprobaciones

Verifique que hay tráfico distinto en cada interfaz:

```
tcpdump -i eth0
# (en una ventana)

tcpdump -i eth1

# (en otra ventana)
```

Debe acostumbrarse a usar `tcpdump` para buscar cosas que no deberían estar pasando o que no pasan y deberían.

Por ejemplo, busque los paquetes que pasan por el puente a la segunda tarjeta desde la red interna. Aquí busco paquetes de la máquina con dirección `.22`:

```
tcpdump -i eth1 -e host 192.168.2.22
```

Ahora le mando un `ping` desde la máquina `.22` al router. Debería ver el paquete según informe de `tcpdump`.

En esta fase tiene un puente listo que también tiene dos direcciones de red. Compruebe que puede hacer ping desde fuera y dentro de su red local, y que puede hacer telnet y ftp entre el exterior y el interior.

4 Cortafuegos

4.1 Software y lecturas

Lea el *Cortafuegos-Como*, *http://www.insflug.org/documentos/Cortafuegos-Como/*

Esto le dirá donde obtener `ipfwadm` si no lo tiene ya. Hay otras herramientas que puede obtener, pero no he hecho avances con ellas hasta que no probé `ipfwadm`. ¡Está muy bien, y es de bajo nivel! Puede ver exactamente lo que está pasando.

4.2 Comprobaciones preliminares

Ha compilado soporte de reenvío IP y enmascaramiento en el kernel, así que querrá comprobar que el cortafuegos está en su estado por defecto (aceptando) con

```
ipfwadm -I -l
ipfwadm -O -l
ipfwadm -F -l
```

Esto es respectivamente: mostrar las reglas que afecten entrantes o salientes o reenviando (enmascarando) hacia ambos lados del cortafuegos. El `-l` significa listar.

Si también ha compilado soporte de informes (accounting):

```
ipfwadm -A -l
```

Debería ver que no hay reglas definidas y que por defecto se aceptan todos los paquetes. Puede volver a este estado en cualquier momento usando

```
ipfwadm -I -f
ipfwadm -O -f
ipfwadm -F -f
```

El `-f` significa «vaciar». Puede que necesite usarlo.

4.3 Reglas por defecto

Quiero evitar a mi red llegar al mundo, y nada más, por lo que por lo menos daré una regla última (por defecto) que diga que el cortafuegos debería ignorar aquellos paquetes que vengan de la red interna dirigidos al exterior. Pongo las reglas (en este orden) en `/etc/rc.d/rc.firewall`y lo ejecuto desde `/etc/rc.d/rc.local` en el arranque.

```
ipfwadm -I -a reject -S 192.168.2.0/255.255.255.128 -D 0.0.0.0/0.0.0.0
```

El `-S` es la dirección/máscara origen. El `-D` es la dirección/máscara de destino.

Este formato es demasiado largo. **`ipfwadm`** es suficientemente inteligente sobre temas de red y entiende algunas abreviaturas. Lea las páginas del manual.

Posiblemente es más conveniente poner algunas o todas estas reglas en la parte saliente del cortafuegos usando `-O` en vez de `-I`, pero fijaré las reglas para la mitad entrante.

4.4 Huecos por dirección

Antes de la regla por defecto, tengo que poner algunas reglas que me sirvan como excepciones a esta denegación de servicios externos general a los clientes internos.

Quiero tratar las direcciones internas de los cortafuegos en especial. He de evitar que la gente entre en el firewall a no ser que tengan un permiso especial, pero una vez que entren deberían ser capaces de hablar con el resto del mundo.

```
ipfwadm -I -i accept -S 192.168.2.100/255.255.255.255 -D 0.0.0.0/0.0.0.0
```

También quiero que los clientes puedan hablar con el cortafuegos. ¡A lo mejor le convencen de que les deje salir!

```
ipfwadm -I -i accept -S 192.168.2.0/255.255.255.128 -D 192.168.2.100/255.255.255.255
```

Compruebe en este punto de que puede entrar en los clientes desde fuera del cortafuegos usando **`telnet`**, pero que no puede salir. Esto debería significar que puede hacer el primer contacto, pero los clientes no pueden enviarle la línea de comandos. Debe ser capaz de llegar hasta el final si usa la máquina cortafuegos como paso intermedio. Intente un **`rlogin`** y un **`ping`** también, con **`tcpdump`** ejecutándose en una tarjeta o en otra. Debe ser capaz de dar sentido a lo que ve.

4.5 Huecos por protocolo

Seguí relajando las reglas protocolo por protocolo. Quiero permitir los `ping` desde el exterior al interior para obtener un eco de respuesta, por ejemplo, así que inserté la regla:

```
ipfwadm -I -i accept -P icmp -S 192.168.2.0/255.255.255.128 -D 0.0.0.0/0.0.0.0
```

El parámetro `-P icmp` realiza la magia necesaria a nivel del protocolo.

Hasta que use un proxy `ftp` voy a permitir los ftp salientes con permisos específicos de puerto. Esto se dirige a los puertos `20 21` y `115` de las máquinas exteriores.

```
ipfwadm -I -i accept -P tcp -S 192.168.2.0/255.255.255.128 \
-D 0.0.0.0/0.0.0.0 20 21 115
```

No puedo hacer que `sendmail` funcione entre los clientes locales sin un servidor de nombres. En vez de instalar un servidor de nombres en el cortafuegos, lo levanto para las peticiones de servicio en el dominio tcp que se dirijan al servidor de nombres más cercano, poniendo esta dirección en los archivos `/etc/resolv.conf` de los clientes.(`nameserver 123.456.789.31` en una línea aparte).

```
ipfwadm -I -i accept -P tcp -S 192.168.2.0/255.255.255.128 \
 -D 123.456.789.31/255.255.255.255 54
```

Puede encontrar el puerto y protocolo usado por un servicio con `tcpdump`. Lance el servicio con un `ftp` o un `telnet` o lo que sea a o desde la máquina interna y mire lo que ocurre en los puertos de entrada y salida del cortafuegos con `tcpdump`:

```
tcpdump -i eth1 -e host client04
```

por ejemplo: El archivo `/etc/services` es otra fuente importante de pistas. Para permitir `telnet` y `ftp` entrantes al cortafuegos desde el exterior, debe permitir a los clientes locales llamadas salientes en un puerto específico. Entiendo por qué es necesario esto para para `ftp` (es el servidor el que establece el flujo de datos al final) pero no sé por qué `telnet` también lo necesita.

```
ipfwadm -I -i accept -P tcp -S 192.168.2.0/255.255.255.128 ftp telnet \
 -D 0.0.0.0/0.0.0.0
```

Hay un problema específico con algunos demonios que buscan el nombre del cortafuegos para buscar su dirección de red. `Rpc.yppasswdd` es uno con el que tuve problemas. Insiste en transmitir información que diga que está fuera del cortafuegos (en la segunda tarjeta). Esto significa que los clientes de dentro no pueden contactar con él.

En vez de empezar a hacer IP aliasing o cambiar el código del demonio, he correspondido el nombre a la dirección de la tarjeta interna en los clientes, en el archivo `/etc/hosts`.

4.6 Comprobaciones

Querrá comprobar que puede hacer `telnet`, `rlogin` y `ping` desde el exterior. Desde el interior debería poder hacer `ping` hacia fuera. Debería ser capaz también de hacer telnet a la máquina cortafuegos desde el interior, y la última debería ser incapaz de hacer nada.

Y ya está. En este momento probablemente quiera aprender cosas sobre `rpc` y NIS/NYS (*Páginas amarillas*) y la interacción con el archivo de contraseñas. La red con cortafuegos debe funcionar sin que los usuarios normales tengan capacidad de entrar al cortafuegos y, consiguientemente, salir al exterior. ¡A lo mejor esto es otro COMO!

5 Anexo: El INSFLUG

El *INSFLUG* forma parte del grupo internacional *Linux Documentation Project*, encargándose de las traducciones al castellano de los Howtos, así como de la producción de documentos originales en aquellos casos en los que no existe análogo en inglés, centrándose, preferentemente, en documentos breves, como los *COMOs* y *PUFs* (**P**reguntas de **U**so **F**recuente, las *FAQs*. `:)`), etc.

Diríjase a la sede del Insflug para más información al respecto.

En ella encontrará siempre las **últimas** versiones de las traducciones «oficiales»: ***http://www.insflug.org***. Asegúrese de comprobar cuál es la última versión disponible en el Insflug antes de bajar un documento de un servidor réplica.

Además, cuenta con un sistema interactivo de gestión de fe de erratas y sugerencias en línea, motor de búsqueda específico, y más servicios en los que estamos trabajando incesantemente.

Se proporciona también una lista de los servidores réplica (*mirror*) del Insflug más cercanos a Vd., e información relativa a otros recursos en castellano.

En ***http://www.insflug.org/insflug/creditos.php3*** cuenta con una detallada relación de las personas que hacen posible tanto esto como las traducciones.

¡Diríjase a ***http://www.insflug.org/colaboracion/index.php3*** si desea unirse a nosotros!.

«Cartel» Insflug, ***cartel@insflug.org***.