IP−MASQUERADING

En primer lugar, vamos a ver un poco de teoría al respecto de qué es el enmascaramiento.

Cuando te conectas a Internet sin tener un dominio ni una dirección IP fija, ésta es asignada automáticamente por tu proveedor. Una dirección IP identifica unívocamente a un único "hosts" (ordenador), que en el caso que nos proponemos va a ser tu máquina Linux.

Según esto, las máquinas Windows de la red no podrían acceder a Internet porque sólo disponemos de una dirección IP (la del servidor Linux), y no podemos asignarsela a todos los equipos a la vez.

Para solucionar este problema, existe el "enmascaramiento" (ip−masquerading).

El enmascaramiento de la dirección IP consiste en que el servidor identifica que uno de los equipos de la red intenta conectar a una dirección de fuera de ésta (http://www.pntic.mec.es, por ejemplo), y es el mismo servidor el que realiza la petición por sí mismo, en lugar de que la máquina cliente lo haga. Así, el servidor toma la dirección IP de la máquina que hace la solicitud, la "enmascara" con la dirección IP que le ha sido asignada, la envía, y cuando llega el paquete de respuesta lo "reenvía" a la máquina que realizó la petición.

¿Complicado...? Desde luego. Pero en esto, como en todo, a nosotros no nos interesa demasiado cómo funciona el asunto, siempre que tengamos una herramienta que nos solucione el trabajo simplemente con configurarla. La "herramienta" en este caso es una aplicación denominada "ipfwadm", que deberías tener instalada en tu sistema Linux si este es Red Hat 5.2 y elegiste la instalación para un sistema de red. Si no la tienes, instálala ahora desde el CD del que realizaste la instalación del sistema.

En realidad, la "teoría" es bastante más compleja, pero me voy a limitar a darte una "receta" que funciona y te va a dar resultado. Si quieres ajustar más concretamente la configuración para tu equipo, puedes leerte el "Mini−Como de IP−Masquerade" (en español) o el "IP−Masquerade HOWTO" (en inglés), además de las páginas "man" del "ipfwadm".

Nota: a partir del kernel 2.2.x, ya no se utiliza la herramienta "ipfwadm", sino "ipchains". Leete la documentación para estas distribuciones (Red Hat 6.0, por ejemplo).

Para nuestros propósitos, bastará con que incluyas las siguientes líneas en tu archivo "/etc/rc.d/rc.local" detrás de los comandos "route" que vimos anteriormente:

Kernel 2.0.x:

```
ipfwadm −F −p deny
ipfwadm −F −a m −S 192.168.1.0/255.255.255.0 −D 0.0.0.0/0
```

Kernel 2.2.x:

```
ipchains −P forward DENY
ipchains −A forward −i ppp0 −j MASQ
```

La primera línea sirve para denegar cualquier conexión (hacer que la norma por defecto sea no permitir nada), y la siguiente autoriza las conexiones para la red 192.168.1.0 (que es la que hemos tomado como ejemplo). También podrías permitir que determinados equipos conecten, y otros no, poniendo una línea para cada una de las direcciones IP que tendrán acceso al exterior. Para más detalles, te vuelvo a remitir a la documentación.

Ahora deberías comprobar que los equipos cliente acceden al exterior. Para ello, establece la conexión a Internet desde el servidor Linux, ponte delante de uno de los equipos cliente, e intenta acceder al exterior desde, por ejemplo, el navegador que utilices. Escribe una dirección "fiable" (conocida), y comprueba que accedes. Si todo ha ido bien, deberías tener delante la página a la que pretendías acceder.

Si tienes problemas, prueba a conectar a una dirección dando su IP en lugar de su nombre. Por ejemplo, si intentas acceder a:

http://www.pntic.mec.es

prueba en su lugar:

http://193.144.238.1

Si accedes bien a través de la dirección IP, pero no consigues acceder a través del nombre, puede que tengas mal configurado el "/etc/resolv.conf" o que el servidor de nombres elegido no funcione correctamente. Si has instalado tú un servidor de nombres en la máquina Linux, puede que haya algún fichero incorrectamente configurado o con datos erróneos. Eso lo veremos en la siguiente sección.

Nota: el enmascaramiento puede conllevar determinados problemas para establecer algunas conexiones que presuponen que existen direcciones IP fijas, etc. Consulta la documentación para conocer los detalles y los clientes de estas aplicaciones que funcionan, porque determinados servicios (videoconferencia, etc.) pueden no estar disponibles desde los clientes.

Lo primero es lo primero, y para tener una red es indispensable instalar y configurar las tarjetas en los equipos que vayas a tener en la red. Las tarjetas que yo utilizo son ethernet 10BaseT, compatibles NE2000, y puesto que tengo pocos equipos en red y están todos juntos los he conectado con cable RG58 y conector BNC. Eso sí, las tarjetas disponen de ambos conectores (BNC y RJ45), porque espero que algún día tengamos todos los equipos del Centro en la misma red y no están los presupuestos para cambiar las tarjetas cada dos por tres :−). Puesto que los equipos que están en red son todos Pentium 75, las tarjetas son PCI, pero casi todos los modelos están disponibles tanto en PCI como en ISA, así que no tendrás problemas sin tus equipos no tienen ranuras PCI.

Lo que te cuento a continuación es válido para una sola red local. Si vas a instalar más de una red, tendrás que establecer máquinas "pasarela" entre ellas. En los documentos mencionados en el apartado anterior te explica más o menos claramente cómo lo puedes hacer.

En nuestro caso, los ordenadores del aula de informática tenían Windows 3.11 para Trabajo en Grupo preinstalado, así que ya tenemos las capacidades de red casi a punto de funcionar... Si tienes Windows 3.1, me temo que será bastante más complicado (aquí no te puedo echar una mano...)

Si dispones de Windows 95, todavía más fácil, porque las utilidades de red están bastante mejoradas con respecto a su predecesor. Entre otras cosas, incluye de serie el soporte para TCP/IP, mientras que en Windows 3.11 tienes que instalar el "parche" para TCP/IP (que puedes encontrar en la web de Microsoft).

Instalación de las tarjetas en los equipos:

¿Que todavía no has abierto un equipo para instalar nada...? Me extraña. Seguro que has tenido que "entrar" en más de uno para hacer algún "apaño". En cualquier caso, es muy fácil:

- Abre la carcasa.
- Localiza una ranura libre que coincida con la "pestaña" de tu tarjeta de red.

- "Pincha" la tarjeta en la ranura. Ten cuidado con la placa base, que podría sufrir daños si presionas sin control sobre la ranura.

Ya está instalada la tarjeta en el ordenador. No cierres todavía la carcasa, por si tienes que hacer algún "retoque" en las IRQs u otras historias raras. (Consulta la documentación de la tarjeta de red para ver en qué IRQ, DMA, etc., está preconfigurada. No deberías tener problemas, pero si tienes otras tarjetas instaladas podrían existir conflictos.)

Cableado de la red

Si el cable es para BNC, lo único que tendrás que hacer es montar la "T" (que suele venir con la tarjeta) en el conector, y los cables en los laterales. Si encuentras cables hechos mejor, pero si los tienes que hacer tú mismo tampoco te preocupes mucho. Busca una tienda de electrónica donde vendan componentes informáticos, mide bien la distancia entre los equipos, deja un poco de cable sobrante "por si acaso", y "engancha" el cable en los conectores. Es tan fácil como el cableado de la antena de la televisión, así que tampoco tiene mucha complicación.

Para "cerrar" la red, tienes que montar "terminadores" en cada extremo de la misma, es decir, en la parte de la "T" que queda libre en cada uno de los ordenadores que "terminan" la red.

Si el cableado es para RJ45 el montaje es más complicado y necesitas más herramientas (y más equipamiento: hubs, etc.). En las páginas del PNTIC sobre el Proyecto "Echanos un cable" tienes todas las explicaciones para montar una red de este tipo, así que no me voy a extender al respecto.

Configuración de las tarjetas:

Si todo ha ido bien, los respectivos sistemas operativos "encontrarán" la tarjeta sin problemas. Es posible que en los equipos Windows necesites instalar los "drivers" que se incluyen normalmente en un disco junto a las tarjetas, así que tenlo a mano para poder configurarla. También debes tener a mano los discos o CD−ROM de instalación del sistema operativo, por si te los pidiera para instalar los protocolos de red adecuados. En cualquier caso, consulta las instrucciones del fabricante para instalar los "drivers".

Configuración de la tarjeta en Linux

Si pudiste configurar el acceso a Internet sin problemas es porque tienes soporte para red en tu núcleo (kernel) de la distribución Linux que estés utilizando. Como dije anteriormente, en mi caso utilizo Red Hat 6.0 con Kernel 2.2.5−15. El soporte para PPP está instalado como módulo, y funciona sin problemas. El soporte para la tarjeta también está instalado como módulo, y la red va bien, así que no me he metido todavía en recompilar el núcleo ni otras historias por el estilo. Si te animas a hacerlo, consulta el Kernel−HOWTO.

Para comprobar que Linux ha reconocido la tarjeta, escribe lo siguiente desde la línea de comandos (por supuesto, tienes que entrar en el sistema como "root"):

less /proc/pci

Debería aparecer tu tarjeta con los datos sobre IRQ, DMA, etc. Si está ahí, la instalación ha sido un éxito. Si no, tendrás que comprobar que tu tarjeta es compatible con Linux... y eso ya es otro cantar.

En algún lugar del sistema de arranque de Linux (yo lo tengo en /etc/rc.d/rc.local, pero seguramente podrías ponerlo en cualquier otro archivo de inicio del sistema), tendrás que cargar el soporte para NE2000 como módulo. Edita el fichero, y escribe:

modprobe ne

Suponiendo que en /modules/preferred tengas el módulo ne.o, Linux debería instalar correctamente la tarjeta identificándola como /dev/eth0.

Ahora podemos proceder a "identificar" nuestra tarjeta de red con su correspondiente dirección de red:

ifconfig eth0 192.168.1.1 netmask 255.255.255.0
route add −net 192.168.1.0

(Estos comandos pueden ser tecleados manualmente, pero lo normal es incluirlos en algún "script" de inicio del sistema. El IP−Masquerade MINI−HOWTO recomienda /etc/sysconfig/network−scripts/ifcfg−eth0 para sistemas Red Hat Linux, pero el archivo que te comenté antes funciona también a la perfección).

Reinicializa el sistema (o los servicios de red correspondientes) con /sbin/reboot, y, si todo ha ido bien, deberás ser capaz de hacer ping y telnet a tu propia máquina a través de la dirección de red asignada a la tarjeta. Pruebalo ahora:

ping 192.168.1.1
telnet 192.168.1.1

Si recibes los "ping" correctamente, y si entras en tu sistema a través de telnet, ¡enhorabuena!. Tienes correctamente instalada la tarjeta de red en Linux.

Después veremos el resto de la configuración de la máquina Linux. Vamos ahora a configurar los otros equipos.

Configuración de la tarjeta y el sistema bajo Windows:

(Las referencias están tomadas del IP−Masquerade Mini−HOWTO de Ambrose Au, y la traducción de Xosé Velazquez. Gracias a ambos por hacernos la vida más fácil en la instalación de nuestra red :−D)

- **Windows 95:**

Empezamos por lo "fácil". Si Windows 95 te reconoce la tarjeta automáticamente, no tendrás que hacer nada más. Si no, instala los "drivers" que te haya proporcionado el fabricante. Deberás ahora tener acceso a tu tarjeta de red desde "Panel de control −> Red" de Windows 95.

Añade entonces el protocolo TCP/IP si no lo tienes (Agregar protocolo −> Microsoft −> TCP/IP).

En "Red −> Propiedades −> TCP/IP", pincha en "Dirección IP", y escribe la dirección que hayas asignado a esa máquina (192.168.1.2 o 192.168.1.3, según el ejemplo que estamos siguiendo). La máscara de subred es 255.255.255.0 (red de clase C).

En "Puerta de enlace (Gateway)", escribe 192.168.1.1 (la dirección de tu máquina Linux).

Si no vas a configurar un DNS en la máquina Linux, escribe lo siguiente en "Configuración DNS −> Orden de búsqueda del servidor DNS": 193.144.238.1 (dirección IP del servidor de nombres del PNTIC). Además, puedes añadir como "Sufijo de búsqueda de dominio" el siguiente: "pntic.mec.es".

Si por el contrario vas a utilizar un servidor de nombres local, escribe como dirección IP 192.168.1.1, y como sufijo de búsqueda de dominio el sufijo que hayas asignado para tu red (en el ejemplo, "colegio.local").

Pulsa "Aceptar" en todas las cajas de diálogo que has ido abriendo, y reinicializa el sistema (si lo has hecho todo "bien", el propio Windows te dirá que tienes que reiniciar).

Una vez que arranque Windows, prueba a hacer un ping a la máquina Linux para comprobar que la red está en funcionamiento ("Inicio −> Ejecutar" y escribes "ping 192.168.1.1"). Si todo va bien, deberás recibir los "ping" en tiempos de respuesta de 0.x...

Nota: si no vas a instalar el DNS local y quieres identificar a las máquinas de la red por su nombre en lugar de su dirección IP, puedes hacerlo creando un fichero HOSTS en el directorio c:\windows. Mira el ejemplo que viene preinstalado como "c:\windows\hosts.sam" y modificalo para que se adapte a tus necesidades:

Archivo "c:\windows\hosts":
127.0.0.1 localhost 192.168.1.1 atenea1.colegio.local 192.168.1.2 atenea2.colegio.local 192.168.1.3 atenea3.colegio.local 192.168.1.4 atenea4.colegio.local 192.168.1.5 atenea5.colegio.local

- **Windows 3.11 para trabajo en grupo**

Instala en primer lugar el "parche" para TCP/IP (el fichero lo puedes encontrar en www.microsoft.com. Busca el fichero llamado WFWT32.EXE. Crea un directorio, por ejemplo "c:\tcp_ip", copialo en él y ejecutalo desde allí).

Entra en "Principal −> Configuración de Windows −> Configuración de la Red" (si no has instalado el "software" de red, hazlo lo primero de todo). Pulsa "Controladores".

Marca "Microsotf TCP/IP−32". En la sección "Controladores de Red", pulsa "Setup" o "Configuración".

Escribe como dirección IP 192.168.1.4 o 192.168.1.5 (según el ejemplo que hemos dado), y como máscara de subred escribe 255.255.255.0. Como "Default gateway" escribe 192.168.1.1

Pulsa "DNS", y añade la información que te indicaba en la configuración de Windows 95, dependiendo de si instalas o no el servidor DNS en la máquina Linux. Aquí también puedes tener tu fichero HOSTS como se indicaba antes.

Pulsa OK en todas las cajas de diálogo, y reinicia el sistema. Prueba a hacer "ping" a la máquina Linux, y también a hacer un telnet para confirmar el funcionamiento de todo.

En ambos casos (Windows 95 y 3.11), lo único que estaremos comprobando con esto es si la red funciona. Todavía no podemos "salir" a Internet con la configuración actual.

- **Otros sistemas operativos**

Para saber cómo configurar otros sistemas operativos, consulta el IP−Masquerade Mini−HOWTO. Puedes montar tu red con máquinas Mac, Unix, etc. Para configurar Windows 98, puedes seguir los pasos indicados en la configuración de Windows 95, puesto que no se diferencian prácticamente en nada.

4. Otras características de IP Masquerade y soporte de programas

4.1 Problemas con IP Masquerade

Algunos protocolos no trabajarán adecuadamente con masquerading porque presuponen cosas sobre números de puerto, o datos cifrados en sus cadenas de datos sobre direcciones y puertos − estos últimos protocolos necesitan un proxi específico en el código masquerading para que funcionen.

4.2 Servicios de entrada

Masquerading no puede manejar todos los servicios de entrada. Hay unos pocas formas de permitirlos, pero son completamente independientes de masquerading, y son realmente parte de la práctica estándar de cortafuegos.

Si no requiere altos niveles de seguridad simplemete puede redirecionar los puertos. Hay varias formas para hacer esto − Yo uso un programa modificado redir (espero que este disponible en sunsite y mirrors). Si desea tener algún nivel de autorización en conexiones entrantes puede usar TCP wrappers o Xinetd sobre redir (0.7 o superior) para permitir sólo direciones IP específicas, o usar alguna otra herramienta. El TIS Firewall Toolkit es un buen lugar para buscar herramientas e información.

Mas detalles pueden ser encontrados en IP Masquerade Resource.

4.3 Programas cliente soportados y otras notas de configuración

**** La siguiente lista no esta siendo mantenida durante mas tiempo. Porfavor remitase a esta pagina para saber que aplicaciones funionan atraves de un servidor Linux con IP masquerading y a IP Masquerade Resource para mas detalles. ****

Generalmente, las aplicaciones que usan TCP y UDP deberían de funcionar. Si tiene alguna sugerencia sobre aplicaciones que no son compatibles con IP Masquerade, por favor envíeme correo electrónico con el nombre del cliente y una breve descripción.

Clientes que funcionan

Clientes Generales

HTTP

todas las plataformas soportadas, navegadores web

POP & SMTP

todas las plataformas soportadas, clientes de correo electrónico

Telnet

todas las plataformas soportadas, sesión remota

FTP

todas las plataformas soportadas, con el modulo ip_masq_ftp.o (no todos los servidores funcionan con ciertos clientes; p.e. algun sitio puede no ser alcanzado usando ws_ftp32 pero funciona con Netscape)

Archie

todas las plataformas soportadas, buscador de archivos (no todos los clientes archie son soportados)

NNTP (USENET)

todas las plataformas soportadas, cliente de noticias USENET

VRML

Windows(posiblemente soportado en todas las plataformas), navegación con realidad virtual

traceroute

principalmente las plataformas basadas en UNIX, algunas variantes pueden no funcionar

ping

todas las plataformas, con el parche ICMP

todo lo basado en IRC

todas las plataformas soportadas, con el módulo ip_masq_irc.o

Clientes Gopher

todas las plataformas soportadas

clientes WAIS

todas las plataformas soportadas

Clientes Multimedia

Real Audio Player

Windows, envío de audio por red, con el módulo ip_masq_raudio cargado

True Speech Player 1.1b

Windows, envío de audio por red

Internet Wave Player

Windows, envío de audio por red

Worlds Chat 0.9a

Windows, programa Cliente−Servidor 3D de conversación

Alpha Worlds

Windows, programa Cliente−Servidor 3D de conversación

Internet Phone 3.2

Windows, comunicaciones de audio de Igual−a−igual(Peer−to−peer), la gente puede ponerse en contacto con vd. si inicia vd. la conexión, pero no le pueden llamar

Powwow

Windows, comunicaciones whiteboard de audio texto de Igual−a−igual, la gente puede ponerse en contacto con vd. si vd. inicia la llamada, pero ellos no pueden llamarle

CU−SeeMe

todas las plataforma soportadas, con el módulo cuseeme cargado, por favor mire IP Masquerade Resource para los detalles

VDOLive

Windows, con el parche vdolive

Nota: Algunos clientes como IPhone y Powwow pueden trabajar incluso si vd. no es el que inicia la llamada por usar el paquete *ipautofw* (mire la sección 4.6)

Otros Clientes

NCSA Telnet 2.3.08

DOS, un paquete que contiene telnet, ftp, ping, etc.

PC−anywhere para windows 2.0

MS−Windows, controla remotamante un PC sobre TCP/IP, sólo trabaja si es cliente pero no servidor

Socket Watch

usa ntp − protocolo de tiempo por red

Linux net−acct package

Linux, paquete de administración de cuentas de red

Clientes que NO funcionan

Intel Internet Phone Beta 2

Conecta pero la voz viaja en una dirección (sale)Trafico sólo

Intel Streaming Media Viewer Beta 1

No puede conectar con el servidor

Netscape CoolTalk

No puede conectar con el lado opuesto

talk,ntalk

no funcionará − requiere que sea escrito un proxy en el núcleo .

WebPhone

No puede trabajar por el momento(asume direcciones inválidas).

X

Sin probar,pero creo que no puede trabajar a menos que alguien construya un X proxy, el cual probablemente es un programa externo al código de masquerading. Una forma de hacer que esto funcione es usar un **ssh** como el enlace y usar un X proxy interno que haga las cosas funcionar!

Plataformas/SO testeados como las OTRAS máquinas

- Linux
- Solaris
- Windows 95
- Windows NT (ambos, workstation y server)
- Windows para TrabajoEnGrupo 3.11 (con el paquete TCP/IP)
- Windows 3.1 (con el paquete Chameleon)
- Novel 4.01 Server
- OS/2 (incluido Warp v3)
- Macintosh OS (con MacTCP o Open Transport)
- DOS (con el paquete NCSA Telnet, el DOS Trumpet trabaja parcialmente)
- Amiga (con AmiTCP o AS225−stack)
- VAX Stations 3520 y 3100 con UCX (TCP/IP stack para VMS)
- Alpha/AXP con Linux/Redhat
- SCO Openserver (v3.2.4.2 y 5)
- IBM RS/6000 con AIX
- (¿ Alguien lo ha intentado con otras plataformas ?)

4.4 Administración de IP cortafuegos con ipfwadm

Esta sección proporciona una guía más exhaustiva del uso de ipfwadm.

Esta es una configuración para un sistema de cortafuegos/masquerade detrás de un enlace PPP con la siguiente dirección PPP estática.El interfaz de confianza es 192.168.255.1, el interfaz PPP ha sido cambiado para proteger al culpable :). Listé cada interfaz de entrada y salida individualmente para capturar IP spoofing también como ruta relleno y/o masquerading. Además algo no explicitamente permitido esta prohibido !

#!/bin/sh

```
#
# /etc/rc.d/rc.firewall, define la configuración del cortafuegos, invocado desde
# rc.local.
#
PATH=/sbin:/bin:/usr/sbin:/usr/bin
# comprobación, espera un bit luego limpia toda norma del cortafuegos.
# descomente las siguientes lineas si quiere que el cortafuegos se
# desconecte automaticamente pasados 10 minutos
# (sleep 600; \
# ipfwadm −I −f; \
# ipfwadm −I −p accept; \
# ipfwadm −O −f; \
# ipfwadm −O −p accept; \
# ipfwadm −F −f; \
# ipfwadm −F −p accept; \
# ) &
# Entrante, purga y pone la política por defecto de denegar. La verdad es que la
# política por defecto es irrelevante porque hay un cierre de toda
# norma con denegar y anotar
ipfwadm −I −f
ipfwadm −I −p deny
# interfaz local, máquinas locales, van a cualquier sitio válido
ipfwadm −I −a accept −V 192.168.255.1 −S 192.168.0.0/16 −D 0.0.0.0/0
# interfaz remota, reclamando ser máquinas locales, IP spoofing, perdido
ipfwadm −I −a deny −V su.dirección.ppp.estática −S 192.168.0.0/16 −D 0.0.0.0/0 −o
# interfaz remota, cualquier origen, es válido ir a direcciones PPP permanentes
```

```
ipfwadm −I −a accept −V su.dirección.PPP.estática −S 0.0.0.0/0 −D
su.dirección.PPP.estática/32
# el interfaz loopback es válido.
ipfwadm −I −a accept −V 127.0.0.1 −S 0.0.0.0/0 −D 0.0.0.0/0
# cierra toda norma, todas las otras entradas son denegadas y registradas. Lástima
# que no hay un opción de registro en la política pero esto hace el trabajo en su
# lugar
ipfwadm −I −a deny −S 0.0.0.0/0 −D 0.0.0.0/0 −o
# Saliente, purga y pone la política por defecto de denegar. La verdad es que
# la política por defecto es irrelevante porque hay un cierre de toda norma con
# denegar y anotar
ipfwadm −O −f
ipfwadm −O −p deny
# interfaz local, cualquier origen con destino a la red local es válido
ipfwadm −O −a accept −V 192.168.255.1 −S 0.0.0.0/0 −D 192.168.0.0/16
# saliente para la red local sobre el interfaz remoto , stuffed routing, denegado
ipfwadm −O −a deny −V su.dirección.PPP.estática −S 0.0.0.0/0 −D 192.168.0.0/16 −o
# saliente desde la red local sobre el interfaz remoto, stuffed masquerading,
# denegado
ipfwadm −O −a deny −V su.dirección.PPP.estática −S 192.168.0.0/16 −D 0.0.0.0/0 −o
# saliente desde la red local sobre el interfaz remoto, stuffed masquerading,
# denegado
ipfwadm −O −a deny −V su.dirección.PPP.estática −S 0.0.0.0/0 −D 192.168.0.0/16 −o
# algo más saliente sobre la interfaz remota es válido.
ipfwadm −O −a accept −V su.dirección.PPP.estática −S su.dirección.PPP.estática/32 −D
0.0.0.0/0
```

la interfaz loopback es válido.

ipfwadm −O −a accept −V 127.0.0.1 −S 0.0.0.0/0 −D 0.0.0.0/0

norma de capturar todo, todo lo otro saliente es denegado y anotado. Lástima

que no hay una opción de registro en la política pero esto hace

el trabajo en su lugar.

ipfwadm −O −a deny −S 0.0.0.0/0 −D 0.0.0.0/0 −o

Envíos, purga y pone la política por defecto de denegar. La verdad es que

la política por defecto es irrelevante porque hay una cierre de toda norma con

denegar y anotar.

ipfwadm −F −f

ipfwadm −F −p deny

Enmascaramiento (Masquerade) desde la red local sobre el interfaz local

hacia cualquier sitio.

ipfwadm −F −a masquerade −W ppp0 −S 192.168.0.0/16 −D 0.0.0.0/0

norma de capturar todo, el resto de envíos son denegados y anotados. Lástima

que no hay una opción de registro en la política, pero esto hace el

trabajo en su lugar.

ipfwadm −F −a deny −S 0.0.0.0/0 −D 0.0.0.0/0 −o

Puede bloquear el tráfico para un sitio en particular usando −I, −O o −F. Recuerde que el conjunto de las normas son examinadas desde arriba hacia abajo y −a significa "añadir" a los valores existententes de la norma, así cualquier restricción necesita venir antes de las normas generales. Por ejemplo (sin probar) :−

Usando normas −I. Probablemente es la más rápida pero sólo detiene las máquinas locales, el cortafuegos puede todavía acceder a sitios "prohibidos". Por supuesto le puede interesar permitir esa combinación .

... inicio de la norma −I ...

rechaza y anota el interfaz local, las máquinas locales van a 204.50.10.13

ipfwadm −I −a reject −V 192.168.255.1 −S 192.168.0.0/16 −D 204.50.10.13/32 −o

interfaz local, máquinas locales, ir a cualquier sitio es válido

ipfwadm −I −a accept −V 192.168.255.1 −S 192.168.0.0/16 −D 0.0.0.0/0

... fin de la norma −I ...

Usando la norma −O. Más lento porque los paquetes van primero através de masquerading pero esta norma incluso detiene los accesos del cortafuegos a los sitios prohibidos.

... inicio de la norma −O ...

deniega y anota las salidas a 204.50.10.13

ipfwadm −O −a reject −V su.dirección.PPP.estática −S su.dirección.PPP.estática/32 −D

204.50.10.13/32 −o

cualquier otra salida sobre el interfaz remoto es válida

ipfwadm −O −a accept −V su.dirección.PPP.estática −S su.dirección.PPP.estática/32 −D

0.0.0.0/0

... fin de la norma −O ...

Usando la norma −F. Probablemente más lento que −I y esto todavía solo detiene a las máquinas enmascaradas (por ejemplo: las internas), el cortafuegos puede todavía acceder a sitios prohibidos

... inicio de la norma −F ...

deniega y anota desde la red local sobre el interfaz PPP hacia 204.50.10.13.

ipfwadm −F −a reject −W ppp0 −S 192.168.0.0/16 −D 204.50.10.13/32 −o

Enmascara(Masquerade) desde la red local sobre interfaces locales hacia

cualquier sitio.

ipfwadm −F −a masquerade −W ppp0 −S 192.168.0.0/16 −D 0.0.0.0/0

... fin de la norma −F ...

No es necesario una norma especial para permitir a 192.168.0.0/16 para ir a 204.50.11.0, está cubierto por la norma global

Hay más que una forma de codificar el interfaz en las normas de arriba. Por ejemplo en lugar de −V 192.168.255.1 puede poner −W eth0, en lugar de −V su.dirección.PPP.estática y puede usar −W ppp0. La eleción es más que otra cosa personal y de documentación.

4.5 IP Masquerade y llamada bajo demanda(Demand−Dial−Up)

- Si le gustaría activar su red automáticamente para llamar a Internet, el paquete de llamada bajo demanda *diald* le será de gran utilidad.
- Para configurar diald, por favor mire en Setting Up Diald for Linux Page
- Una vez que diald e IP masq hayan sido configurados, puede ir a alguna de las máquina clientes e iniciar una sesion web, telnet o ftp.

- Diald detectará peticiones entrantes, luego llama a su ISP y establece la conexión.
- Hay un tiempo de espera que ocurrirá con la primera conexión. Esto es inevitable si esta usando un módem analógico. El tiempo dado para establecer el enlace entre el módem y la conexión PPP causará que su programa cliente termine. Esto puede ser evitado si esta usando un conexión RDSI. Todo lo que necesita hacer es terminar el proceso corriente en el cliente y reiniciarlo.

4.6 Envío de paquete IPautofw

IPautofw es un transportador genérico de TCP y UDP para masquerading de Linux. Generalmente para utilizar un paquete el cual requiere UDP necesita ser cargado un módulo específico de ip_masq; ip_masq_raudio, ip_masq_cuseeme, ... Ipautofw actúa de una forma más genérica, enviará cualquier tipo de tráfico incluído aquel que los módulos especificos no envían. Esto puede crear un agujero de seguridad si no se administra corrrectamente.