



Cómo instalar, configurar y poner en marcha de una red con un firewall.

Por Sergio Vergara

svergara@notess.telscape.com.mx

Edición y formato por Joel Barrios Dueñas

admin@jjnet.prohosting.com

[[Contenido](#)] [[Regresar a página principal](#)]

Copyright.

© 2000 Sergio Vergara y [Linux Para Todos](#). Se permite la libre distribución de este documento por cualquier medio mientras se haga de acuerdo con los términos de la licencia pública general GNU. Linux® es una marca registrada de Linus Torvalds, LinuxPPP® es una marca registrada de José Neif Jury Fabre, RedHat® Linux es una marca registrada de RedHat Software, Unix® es marca registrada de X/Open. MS-DOS® y Windows® son marcas registradas de Microsoft Corporation. La información contenida en este documento se proporciona tal cual es y el autor no asumirá responsabilidad alguna si el usuario o lector hace mal uso de esta.

Contenido.

1. Introducción
2. Breve historia sobre Firewall y Enmascaramiento de direcciones
3. El filtrado de paquetes
4. Cadenas IP
 - Cadenas de Entrada
 - Cadenas de Salida
 - Cadenas de desviación (reenvío)
5. Reglas en el Firewall
 - Aceptar
 - Negar
 - Rechazar
 - Enmascarar
 - Redireccionar
 - Regresar
6. Enmascaramiento
7. Instalación del software necesario.
 - RPM en el disco LinuxPPP 6.2
 - Script para hacer funcionar el enmascaramiento
8. El Firewall
 - Aspectos generales
 - Script para funcionar el firewall
 - Aspectos generales de bash2
 - Como funciona este script
 - Soporte para el servicio proxy
 - Configuración del firewall

Introducción.

Introducción.

Este documento pretende ayudar al administrador a instalar y configurar un firewall dentro de una maquina con Linux, así mismo, como varias maquinas pueden salir a otras redes por medio de una sola dirección IP (Enmascaramiento).

Esta primera versión cubriría aspectos básicos de instalación y configuración, esperamos ir mejorando con el paso del tiempo este documento y pueda ser una valiosa guía.

Este documento a sido probado en una maquina con un procesador a 400Mhz y 64Mbytes en RAM, dos tarjetas de red 3Com y un disco duro de 4Gbytes. El equipo tiene instalado LinuxPPP 6.2 (Versión Mexicana que incluye algunos programas extras a la versión de RedHat) y la aplicación de Firewall fue instalado del mismo disco de LinuxPPP donde se localizan la mayor parte de los RPM's. La maquina en la que ha sido probada funciona como gateway y firewall entre dos redes (una red corporativa: secretarias, gerentes y personal en general y una red privada: con equipos de tienen bases de datos, facturación, servidores de WEB y equipo que no puede ser visto o accesado por cualquier persona), el cual protege a una de la otra de accesos indebidos a equipos de misión crítica.

Por otra parte, si tu deseas salir a internet desde una linea telefónica y también los equipos de la red interna primero debes de leer el manual de configuración PPP (Point to Point) para que tu Servidor Linux pueda tener acceso a internet; Una vez configurado tu salida a internet puedes proseguir con este documento.

Finalmente, toda colaboración, corrección y demás es bien recibido para poder mejorar este documento.

Breve historia sobre Firewall y Enmascaramiento de direcciones

No hace mucho tiempo una red de computo era algo que no muchas empresas o escuelas tenían en su centro de computo. La gran mayoría de estas computadoras se encontraban aisladas una de la otra aunque se encontraran en el mismo cuarto o salón de computo. Únicamente grandes empresas, Instituciones Gubernamentales o Universidades tenían este recurso. Eso a cambiado con el paso del tiempo y hoy en día es común encontrar computadoras conectadas a una red y obtener información de ella o brindar servicios (tal es el caso de internet o una intranet en una oficina u hogar), y es común el uso de ellas, como por ejemplo enviar y recibir correo electrónico, entre muchos servicios más.

El gran desarrollo de estas redes no ha sido del todo positivo en varios aspectos. Uno de ellos es la disponibilidad de Direcciones IP, que esta limitado a 4.300 millones de direcciones IP validas aproximadamente. Esta cantidad de direcciones puede ser a primera vista muchísimas direcciones, pero direcciones validas libres en internet son actualmente muy pocas, por lo que cada vez es más difícil poder obtener una dirección valida en internet. Con la llegada de la versión 6 del protocolo IP se espera poder extender este rango de direcciones en un par de millones más. Pero como esta nueva versión aun no se encuentra disponible debemos de trabajar con la actual (IPv4) y por ende debemos administrar mejor el uso de este tipo de direcciones. Una forma de administrar mejor esto, es escondiendo computadoras con direcciones no validas dentro de una red, detrás de una dirección IP valida. A esta técnica se le conoce como enmascaramiento de direcciones.

Existe otro problema que no es técnico sino social. Cada día existen más computadoras y personas que accesan a internet. La necesidad de proteger los sistemas conectados a una red de usuarios no deseados es cada vez más común y se vuelve más importante día a día.

Instalar un firewall es en buena medida una buena solución para protegerse de ataques a una red interna o de usuarios no deseados. Actualmente, el Kernel de Linux (Por ejemplo LinuxPPP 6.2, RedHat 6.2) soporta filtrado de paquetes, que pueden se utilizados para implementar un sencillo firewall.

Dentro de este documento se hablara del mecanismo para el filtrado de paquetes, como puede ser utilizado para el enmascaramiento de paquetes y construcción de un Firewall. El RPM que se encuentra dentro del disco de Linux contiene el software necesario para que el firewall y el enmascaramiento funcionen correctamente.

El filtrado de paquetes

Actualmente linux soporta el el filtrado de paquetes. La versión de Kernel 2.2 de Linux contiene cambios significativos en su estructura para brindar este servicio.

Existen cadenas o reglas que los paquetes IP deben de igualar para que este pueda ser aceptado. Si llega algún paquete al equipo una regla decide que hacer con el. Este paquete puede ser aceptad, negado, rechazado, enmascarado o enviado a otra regla.

Con este mecanismo es fácil construir reglas sencillas para el filtrado de paquetes con un firewall. Esto quiere decir que todos lo paquetes que lleguen a la maquina, independientemente si son TCP o UDP, serán primero filtrados antes de ser enviados a si destino.

Linux soporta una gran número de características para las reglas de un firewall y el enmascaramineto de paquetes.

Linux soporta una gran número de características para las reglas de un firewall y el enmascaramiento de paquetes.

Cadenas IP

Las cadenas de un firewall no son más que reglas que se utilizan para que el paquete cumpla con alguna de ellas y en un cierto orden. Esto quiere decir que el paquete debe de cumplir con alguna regla. La regla determina que es lo que va a suceder con el paquete que ha sido recibido. Si el paquete no coincide la próxima regla determinará que hacer con él. Si llega al final de esta regla se utilizará la política que se encuentra por omisión.

Existen tres tipos de reglas por omisión que se utilizan:

INPUT

Aceptación de paquetes de entrada. Todos los paquetes que vienen de una de las interfaces de la red local son revisados por la regla de entrada. Si el paquete no coincide con alguna de las reglas de entrada este los rechaza.

OUTPUT

Esta regla define los permisos para enviar paquetes IP. Todos los paquetes se encuentran listos para ser enviados a una de las interfaces de la red local y son revisados por la regla de salida. Si el paquete no coincide con alguna de las reglas el paquete es rechazado.

FORWARD

Esta regla define los permisos para el envío del paquete a otro sitio. Todos los paquetes se envían a un equipo remoto. Nuevamente, si el paquete no coincide con alguna de las reglas este paquete es rechazado.

Si observamos como la maquina puede funcionar como ruteador, observaremos que existen tres tipos de trafico. Los paquetes enviados a esta maquina, los paquetes originados por la maquina y los paquetes ruteados a travez de la maquina. La tabla 2.1 muestra como estos paquetes son ruteados a travez de las reglas.

Tabla 2.1: Paquetes y cadenas IP

		To local host	To remote host	
	<i>From local host</i>		<i>Output</i>	
	<i>From remote host</i>	<i>Input</i>	<i>Input -> forward -> Output</i>	

Como se muestra en la tabla 2.1 solo aquellos paquetes que provienen de un host y se dirigen a otros host tienen todo el acceso de las tres cadenas, los paquetes de entrada a la red local podrán solo entrar a travez de la cadena o regla de entrada y los paquetes originados desde una maquina local solo podrán salir a travez de la cadena o regla de salida. Con este esquema de niveles podemos tener una gran flexibilidad para instalar reglas para diferente tipo de trafico.

En la versión del kernel 2.2 hay un grupo de reglas predefinidas. Estas están integradas dentro de las reglas de entrada y salida y no son necesarias separar una de otra para que estas funcionen correctamente.

Reglas en el Firewall

Como se ha comentado, el kernel contiene cadenas de reglas para realizar el filtrado de paquetes. Se mostró también como una cadena filtra un tipo específico de trafico (entrada o salida). Ahora mostraremos cuales son estas reglas.

Las reglas en un Firewall se crean de igual forma como se a mencionado con anterioridad, tenemos una condición que debe de cumplirse para que el paquete de entrada o salida tenga los permisos para poder llegar a su destino. Los valores que debemos de utilizar para crear una regla se muestran a continuación:

ACCEPT

Este valor quiere decir que permite pasar a los paquetes que pasan a travez del Firewall. Todos aquellos paquetes que cumplan con la regla de entrada podrán tener acceso de entrada o salida.

DENY

o salida.

DENY

Este valor quiere decir que los paquetes no podrán ser aceptados. Aquellos paquetes que coincidan con la regla (DENY) no podrán llegar a su destino y es tirado a la basura.

REJECT

Es casi igual al valor DENY pero es mas fina la forma de negar el acceso de los paquetes. Por ejemplo los mensajes ICMP se envían de regreso al originador de este paquete, indicandole que este ha sido rechazado (Los valores DENY y REJECT son todos ellos paquetes ICMP).

MASQ

Este valor es únicamente utilizado para el envío y cadenas definidas por el usuario y puede ser utilizado únicamente si el kernel es compilado con el soporte de enmascaramiento. Con eso, los paquetes serán enmascarados como si se tratara del equipo maestro (tu maquina que tiene instalado el Firewall, para que entiendas). Desafortunadamente, los paquetes que regresen del equipo remoto al que se enviaron los paquetes enmascarados, deben de pasar por el equipo maestro y este debe desenmascarar el paquete para que puede ser recibido por su originador.

REDIRECT

Este valor indica que únicamente los paquetes serán redireccionados de la entrada las cadenas definidas por el usuario y pueden ser únicamente utilizados cuando el kernel es compilado con el soporte de "Transparent Proxy". Con esto, los paquetes puedes ser redireccionados al soquet local de la maquina maestro siempre y cuando estos sean enviados desde un host remoto.

RETURN

Este valor es definido por las colas, esto quiere decir que el procesamiento de paquetes continuara en la próxima regla de la siguiente cadena.

Las condiciones de las declaraciones se vuelven mas complejas a medida que se tienen diversos tipos de paquetes que filtrar. Los paquetes IP se agrupan por tipo de paquete, los cuales tienen características semejantes entre ellos, así con esto, podemos determinar mas fácilmente que paquetes coinciden con alguna regla o no. Las reglas contienen un conjunto de valores para cada uno de los parámetros. Estos parámetros se especifican en la regla de filtrado:

PROTOCOL

El protocolo de paquetes es revisado. El protocolo especificado puede ser uno de los siguientes: TCP, UDP, ICMP o todos ellos, de igual forma pueden ser valores numéricos que representan a cada uno de estos protocolos y los hace diferentes uno de otro. Los nombres y valores de estos protocolos se almacenan en el archivo `/etc/protocols`

SOURCE

De donde provienen los paquetes. La información fuente contiene la dirección IP que muestra la procedencia o un rango de direcciones al igual que la mascara de esas redes, estas también pueden incluir la especificación del puerto o ICMP. Este puede proporcionar el nombre del servicio que se solicita el numero del puerto, el valor numero de ICMP o el nombre del servicio ICMP que se solicita.

DESTINATION

Es el mismo valor como en el parámetro SOURCE pero esta vez se especifica a donde el paquete va a ser enviado.

INTERFACE

Los mismos valores como en el PARAMETRO SOURCE pero esta vez indica por que interfaces el paquete debe de ser enviado.

FRAGMENT

Esto significa que la regla únicamente observara fragmentos de un paquete completo.

SYN BIT SET

Únicamente coinciden paquetes del tipo TCP y si se encuentra habilitado y el SYN BIT a ACK y FIN se encuentran limpios. Estos pueden ser paquetes utilizados para la inicialización de conexión de una petición TCP; Por ejemplo, el bloqueo de paquetes de entrada hacia una interface que realiza conexiones del tipo TCP. Esta opción es útil cuando el tipo de protocolo es TCP.

Ahora... ¿Como instalar una regla en el Firewall y como monitorearlas? El comando para insertar, borrar y listar las reglas del firewall es `/sbin/ipchains`. La sintaxis para utilizar este comando la puedes encontrar en el manual `ipchains(8)`.

Ahora... ¿Como instalar una regla en el Firewall y como monitorearlas? El comando para insertar, borrar y listar las reglas del firewall es `/sbin/ipchains`. La sintaxis para utilizar este comando la puedes encontrar en el manual `ipchains(8)`.

Enmascaramiento

El enmascaramiento significa, que el router reemplaza la información que viene de un paquete, es decir, le pone su propia dirección IP y numero de puerto y lo envía a su destino. Y los paquetes de regreso llegan al router, este ve a que equipo deben de llegar, le quita el enmascaramiento y lo envía al host que envió la petición origen.

Esto significa que con el enmascaramiento de IP se puede esconder una red completa con computadoras con direcciones no validas, detrás de una dirección IP valida. Esto es totalmente transparente para la maquina que se encuentra dentro de la red protegida o con direcciones no validas. Cuando se establece una conexión entre un equipo de la red local con otro de la red externa , el equipo externo no necesita saber que se esta conectando con un equipo que contiene una dirección no valida. El equipo remoto pensara que se esta conectado nuestro equipo router (o maestro), pero en realidad únicamente los paquetes están pasando atravez del equipo remoto, los paquetes se reenvían al equipo que realizo la petición dentro de nuestra red local y cuando este equipo envía algún paquete, el equipo maestro enmascarara este paquete, como si el fuera el que originara la información al servidor remoto.

En el ejemplo que se menciona anteriormente puede ser utilizado cuando desde tu servidor maestro te conectas a internet y detrás de tu equipo maestro tienes conectado una red local varias computadoras conectadas a el mediante una tarjeta Ethernet. Como es sabido el equipo maestro tendría solamente la dirección IP valida que le da permiso de acceder a internet, esta dirección IP comúnmente es proporcionada por tu ISP (Internet Service Provider= Proveedor de Servicios de Internet) y cambia cada vez que tu accedas a internet mediante tu ISP, ahora, para que las demás computadoras que se encuentran conectadas en red puedan tener acceso de salida a internet, tu servidor maestro debe de tener el enmascaramiento instalado y funcionando, de lo contrario estas no podrán tener salida. Ahora cuando tu equipo maestro ya tiene acceso a internet y esta haciendo la función de ruteador de todas tus maquinas internas, todo el trafico generado por las maquinas internas podrán tener acceso a internet atravez de tu maquina maestra sin problemas.

Otra característica importante, es de que tu LAN interna se encuentra escondida del mundo externo. Nadie podrá observar que tienes mas de una maquina que esta saliendo a internet atravez de tu equipo maestro y como esta hecha tu LAN interna o cuantas computadoras están conectadas a ella. Si utilizas direcciones no registradas (o validas) en tu red interna, nadie tendrá acceso a conectarse a ellas (únicamente se podrán ver las maquinas locales en la LAN) sin que pasen por el equipo maestro (o firewall). La tabla 2.2 muestra el rango de direcciones IP privadas o no validas

Tabla 2.2: Rango de IP's privadas

Class A net	10.0.0.0	->	10.255.255.255	
Class B net	127.16.0.0	->	172.31.255.255	
Class C net	192.168.0.0	->	192168255255	

Para enmascarar una red no es necesario hacer diferencias entre un tipo de equipo y otro dentro de tu LAN. Mientras tus equipos utilicen el protocolo TCP/IP todos ellos podrán tener acceso a internet mediante tu equipo maestro. El enmascaramiento toma efecto en el equipo maestro (Tu firewall que también es un gateway entre una red y otra) y no realiza alguna excepción en los equipos conectados dentro de tu LAN interna.

Instalación del software necesario.

La figura 2.1 muestra una red sencilla conectada a tu equipo maestro (Firewall/Gateway). Una maquina (tu equipo maestro) es utilizado como Gateway hacia internet. La red interna (LAN) consta de un conjunto de maquinas conectadas entre si mediante una tarjeta de red Ethernet.

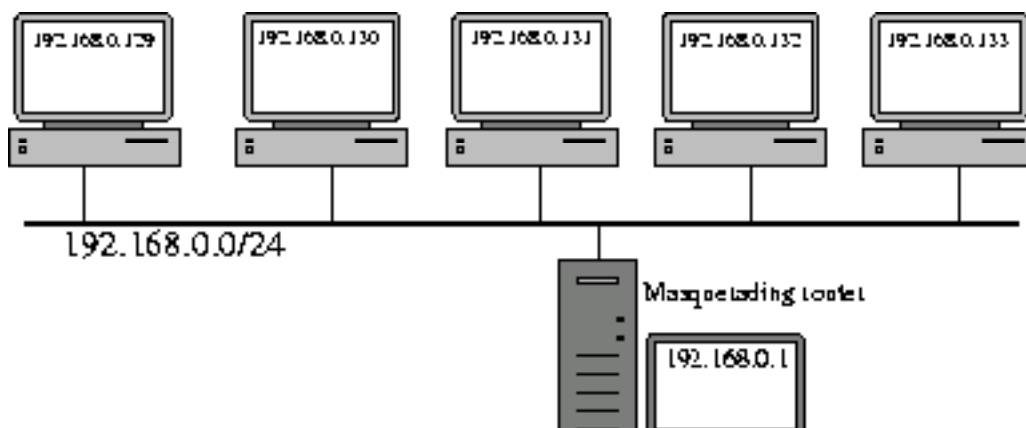




Figura 2.1: Una red sencilla con enmascaramiento

Todas las maquinas conectadas en la LAN interna tienen una dirección Clase C 192.168.0.0. El único equipo con una dirección válida hacia internet es el router (equipo maestro que sirve como gateway entre nuestra red local e internet, así mismo funcionara como firewall) con una dirección IP 192.141.17.53. Esta es la típica configuración de una pequeña red de oficina o red casera.

Ahora vamos a instalar el enmascaramiento para la maquina maestro (gateway). Primero hay que estar seguros de que el kernel este configurado para poder enmascarar direcciones IP. La configuración del Kernel la puedes ver en el manual de manejo del sistema. Recuerda que debe estar habilitada la opción de enmascaramiento de IP's para su correcto funcionamiento (LinuxPPP 6.2 ya cuenta con esta opción).

Ahora debemos de instalar algunas reglas del firewall para poder habilitar el enmascaramiento de todo nuestro trafico de salida, originado por nuestras maquinas en la red local.

Primeramente hay que crear una nueva cola llamada `user_msq` y enmascarar todo el trafico ruteado a través de este equipo con el comando:

```
# /sbin/ipchains -N user_msq
# /sbin/ipchains -A user_msq -s 0/0 -d 0/0 -j MASQ
```

El parámetro `-N` se utiliza para crear una nueva cadena. Para agregar nuevas reglas al final de la cadena se utiliza la opción `-A`. Para especificar las direcciones origen se realiza con la opción `-s` y las de salida con la opción `-d`.

Estas opciones toman direcciones IP como parámetros, con una máscara opcional y número de puerto. En el ejemplo se ha utilizado `0/0` que con esto cubrimos todas las posibles direcciones IP. El formato general podría utilizarse de esta forma: `ip/network address/netmask`. La máscara puede escribirse en el formato usual que es: `255.255.255.0` para una red clase C o únicamente proporcionar el número de bits que utilizara la máscara de esa red, esto quiere decir que `/24` es igual a escribir `/255.255.255.0`.

El origen de la regla se toma por la opción `-j`. En este caso, se tomo `MASQ` como el origen para realizar el enmascaramiento.

Ahora, se necesita enviar los paquetes originados por la LAN a esta cadena. Como se a mostrado, con `ipchains` todos los paquetes ruteados serán procesados para ser enviados a la cadena, y también hay que agregar esta regla a esta cadena, la cual enviara todos los paquetes de nuestra red `192.168.0.0/24` hacia la nueva cadena llamada `user_msq`. Existe un punto más a considerar. Todos los paquetes que vienen de nuestra red local dejaran el router (equipo maestro) en su dispositivo de salida, y es conveniente monitorear únicamente los paquetes de este dispositivo de salida y enmascarar los paquetes que saldrán con la dirección IP válida. La opción `-i` hace esta tarea y toma nuestro dispositivo de salida como su argumento. De esta forma:

```
#/sbin/ipchains -A forward -s 192.168.0.0/24 -d 0/0 -i ppp0 -j user_msq
```

Esta regla concuerda con todos los paquetes que se envíen a través de este dispositivo (`ppp0`) con la dirección fuente de la clase C de la red `192.168.0.0` destinada a cualquier dirección y lo pone en la cadena `user_msq`.

La instalación esta casi terminada. La cadena IP esta instalada y todos los paquetes que son originados por nuestra LAN se encuentran enmascarados. Ahora, que es lo que hace falta más?

Algunos protocolos tienen alguna extraña definición que hace un poco más difícil reconocer los paquetes de regreso. El `ftp` por ejemplo es tal protocolo. Las conexiones no se establecen únicamente del cliente al servidor, pero el servidor también se conecta con el cliente. Esto significa, que el núcleo del linux tiene que reconocer esas conexiones como parte de una conversación entre la máquina enmascarada del cliente y el servidor FTP en Internet, y transmitir estos paquetes a la máquina en la LAN. Esto no es parte del enmascaramiento estándar en el kernel.

Sin embargo el kernel tiene módulos para utilizar un número de protocolos que necesitan la dirección especial. El tabla 2.3 muestra una lista de estos módulos. El módulo del `ip_masq_user` se

puede utilizar para poner la dirección en ejecución especial en el espacio del usuario para los protocolos que (todavía) no son utilizados por otros módulos.

Tabla 2.3: Módulos especiales de enmascaramiento

Tabla 2.3: Módulos especiales de enmascaramiento

	ip_masq_cuseeme	CuSeeMe protocolo para video conferencia	
	ip_masq_irc	Chat	
	ip_masq_raudio	Real Audio	
	ip_masq_vdolive	VDO Live	
	ip_masq_ftp	File Transfer Protocol	
	ip_masq_quake	Quake	
	ip_masq_user	Control especial de enmascaramiento	

Si se van a utilizar estos módulos en nuestro router, es necesario cargar las referencias de estos protocolos de esta manera:

```
# /sbin/insmod ip_masq_cuseeme
# /sbin/insmod ip_masq_irc
# /sbin/insmod ip_masq_raudio
# /sbin/insmod ip_masq_vdolive
# /sbin/insmod ip_masq_ftp
# /sbin/insmod ip_masq_quake
```

Ahora hemos terminado y el enmascaramiento debe trabajar absolutamente bien. Ahora puedes controlar si todas las reglas se encuentran trabajando correctamente, listandolos con el comando ipchains de la siguiente forma:

```
# ipchains -L forward -n
Chain forward (policy ACCEPT):
target prot opt source destination ports
user_msq all ----- 192.168.0.0/24 0.0.0.0/0 n/a

# ipchains -L user_msq -n
Chain user_msq (1 references):
target prot opt source destination ports
MASQ all ----- 0.0.0.0/0 0.0.0.0/0 n/a
```

Como se ve, se puede listar la cadena usando el parámetro *-L* y el nombre de la cadena. No dar ningún nombre enumerará todas las cadenas que existen. El parámetro *-n* asigna a ipchains para que este imprima la salida en valores numéricos que se resuelve de la resolución de las direcciones IP e impresión de los nombres.

RPM en el disco LinuxPPP 6.2

-Tema aún no terminado.-

Script para hacer funcionar el enmascaramiento

Si existe la necesidad de utilizar el enmascaramiento, entonces es probable activar el script cada vez que la maquina se inicializa. Entonces es necesario utilizar un script que se almacene dentro de */etc/rc.d/init.d* y realice esta tarea por nosotros. Como se mostró en el apartado anterior, el RPM que esta incluido dentro del disco de LinuxPPP instalara varios programas, scripts y manuales para que todo funcione correctamente. El script *masquerade* que se almacena dentro de */etc/rc.d/init.d* funcionara para que cada vez que se levante el equipo este arranque el programa para funcionar el enmascaramiento. Este programa funciona exactamente igual como se mostró en la sección anterior. Este levanta la cadena *user_msq* y envía el trafico a ser enmascarado a nuestra cadena. Las cuatro variables que existen para levantar las reglas son las siguientes:

MSQ_START

El enmascaramiento se levantara cuando esta variable se encuentre con la opción "yes". De otra forma tu debes de inicializar el script manualmente si esta opción se encuentra en "no".

MSQ_DEV

Dispositivo donde el enmascaramiento toma efecto. Este dispositivo es la interfaces de salida de tu router (equipo maestro) que puede ser *ppp0* o *eth0*.

MSQ_NETWORKS

aquí se agregan las direcciones de las redes locales a ser enmascaradas separadas por un espacio. Tu puedes especificar direcciones IP o redes en esta opción.

MSQ_MODULES

aquí se agregan las direcciones de las redes locales a ser enmascaradas separadas por un espacio. Tu puedes especificar direcciones IP o redes en esta opción.

MSQ_MODULES

Los módulos son necesarios para el enmascaramiento (Ver tabla 2.3)

Para tener la misma funcionalidad como la que se mostró en la sección anterior el archivo `/etc/sysconfig/firewall/config` debe de quedar de la siguiente forma:

```
# Masquerading settings
#
MSQ_START="yes"
MSQ_NETWORKS="192.168.0.0/24"
MSQ_DEV="ppp0"
MSQ_MODULES="ip_masq_cuseeme ip_masq_ftp ip_masq_irc \
ip_masq_quake ip_masq_raudio ip_masq_vdolive"
```

El script `-masquerade-` soporta las opciones "Start, stop, reload". También puedes ver la lista de conexiones enmascaradas con la opción "status". Un ejemplo de ella sería así:

```
# /sbin/init.d/masquerade status
BB Masquerading v2.1
IP masquerading entries
prot expire source destination ports
UDP 04:51.46 Netwinder.suse.com norad-48.mcdn.net 1177 (61031) -> domain
UDP 04:36.02 Netwinder.suse.com norad-48.mcdn.net 1175 (61028) -> domain
TCP 01:11.29 Netwinder.suse.com www.apple.com 2153 (61027) -> telnet
TCP 01:56.27 Netwinder.suse.com sfbay1.yahoo.com 2155 (61032) -> www
```

Como se muestra en el ejemplo de arriba, se ven cuatro conexiones enmascaradas originadas por el equipo Netwinder.suse.com. Existen dos peticiones de DNS a norad-48.mcdn.net, una sesión telnet a www.apple.com y una conexión a sfbay1.yahoo.com

El Firewall

Antes de proseguir con la instalación del firewall y las reglas de filtrado de la información, tenemos que pensar de lo que quisiéramos que el firewall hiciera. La primera cosa por supuesto es proteger nuestra red de área local contra intrusos de el Internet. Pero no deseamos probablemente bloquear todo el tráfico del exterior. Podemos tener servicios que deban ser accesibles, como el WEB, ftp o mail server. También puede haber los equipos en los que confiemos y puedan tener acceso a nuestra red local. Por ejemplo los contratistas, o las cuentas del mantenimiento son visitantes posibles que no deseamos negarles el acceso. Podemos desear controlar quién puede tener acceso desde Internet a los equipos de trabajo locales.

Aspectos generales

La primera parte es como negar el acceso desde el exterior, mientras que el tráfico salida pueda tener acceso al exterior. ¿Este es en realidad un problema? El filtrado de paquetes puede ser muy grande, y no se saben si algún paquete que llegó fue iniciado desde dentro de alguna máquina de la red local, o un o de alguna máquina del exterior, por lo que en estos momentos no se sabe sobre conexiones de todos modos. Las reglas del filtro observan cada paquete como una entidad separada.

Para distinguir entre el tráfico de salida y de entrada, necesitamos saber un poco sobre la estructura del servicio del TCP/IP. Vamos a tomar como muestra SMTP (protocolo simple del transporte del correo) como ejemplo. El smtp es utilizado por MTAs (agentes de transporte del correo) para transportar el correo a partir de un ordenador principal a otro. Para hacer esto la máquina principal (o los MUA en este ordenador principal) que desea entregar el correo abre una conexión al puerto 25 en la máquina receptora, sabiendo (o esperando) que un servidor del smtp esté escuchando las conexiones entrantes en este puerto. Si es así ambos agentes del correo negociarán algunos parámetros y el correo será enviado a la máquina receptora. Esto quiere decir, que si bloqueamos el puerto 25 en nuestra máquina, nadie podrá entrar en contacto con nuestro mail server. Casi todos los servicios del TCP trabajan esta manera. Hay los accesos especiales donde un demonio espera conexiones entrantes, bloqueando este acceso dará lugar a una negación de este servicio. Esto nos da una política para la protección selectiva de los servicios para el tráfico entrante. ¿Pero qué hay sobre las conexiones salientes? Bien, el TCP/IP tiene números de acceso a partir de la 0 a 65535. Los primeros 1024 accesos (0 a 1023) son reservados para los servicios de sistema. Esto quiere decir, que las conexiones salientes tienen números de acceso más arriba de 1023, y todos los paquetes entrantes que intenten alcanzar los puertos más arriba de 1023, son contestaciones a las conexiones iniciadas por peticiones internas.

Entonces, lo que hacemos es bloquear todos los puertos debajo de 1024 y permitir que el tráfico a los puertos más altos pase. Esto es bueno como regla general, pero la vida es dura, y hay anomalías. Algunos servicios sensibles están situados en números de acceso más altos. El ya popular servicio de HTTP utiliza el número de puerto 3128 por valor por defecto. Los servidores con bases de datos también tienden a utilizar altos números de puertos. Por ejemplo, mysql utiliza el puerto 3306. Entonces, bloquear los puertos del 0 al 1023 son lo

bueno como regla general, pero la vida es dura, y hay anomalías. Algunos servicios sensibles están situados en números de acceso más altos. El ya popular servicio de HTTP utiliza el número de puerto 3128 por valor por defecto. Los servidores con bases de datos también tienden a utilizar altos números de puertos. Por ejemplo, mysql utiliza el puerto 3306. Entonces, bloquear los puertos del 0 al 1023 son lo suficiente para controlar nuestros accesos. Deberas que observar que es lo que esta en funcionamiento en tu sistema y cerciorarse de que tienes una lista completa de todos los accesos que puedan ser una blanco para los ataques del exterior y verificar que estos estan bloqueados.

Script para funcionar el firewall

Esta vez no intentaremos instalar todo a la vez. El firewall es mucho más complejo que el enmascaramiento. La forma de instalación del enmascaramiento se vio en la sección 2,2,1 y se proporciono una comprensión general de cómo el comando ipchains es utilizado para instalar reglas del firewall dando un ejemplo de mundo verdadero. Pero como el mundo verdadero no es siempre tan fácil como en este caso. Si estas interesado, en leer el script, este se localiza en /etc/rc.d/init.f/firewall. Con la descripción dada en esta sección, no es realmente duro entender qué va allí.

Para entender cómo trabaja, tendremos una red de ejemplo otra vez. A continuación se muestra la figura 2.2, y como se ve, es un poco más compleja que en ejemplo de enmascaramiento.

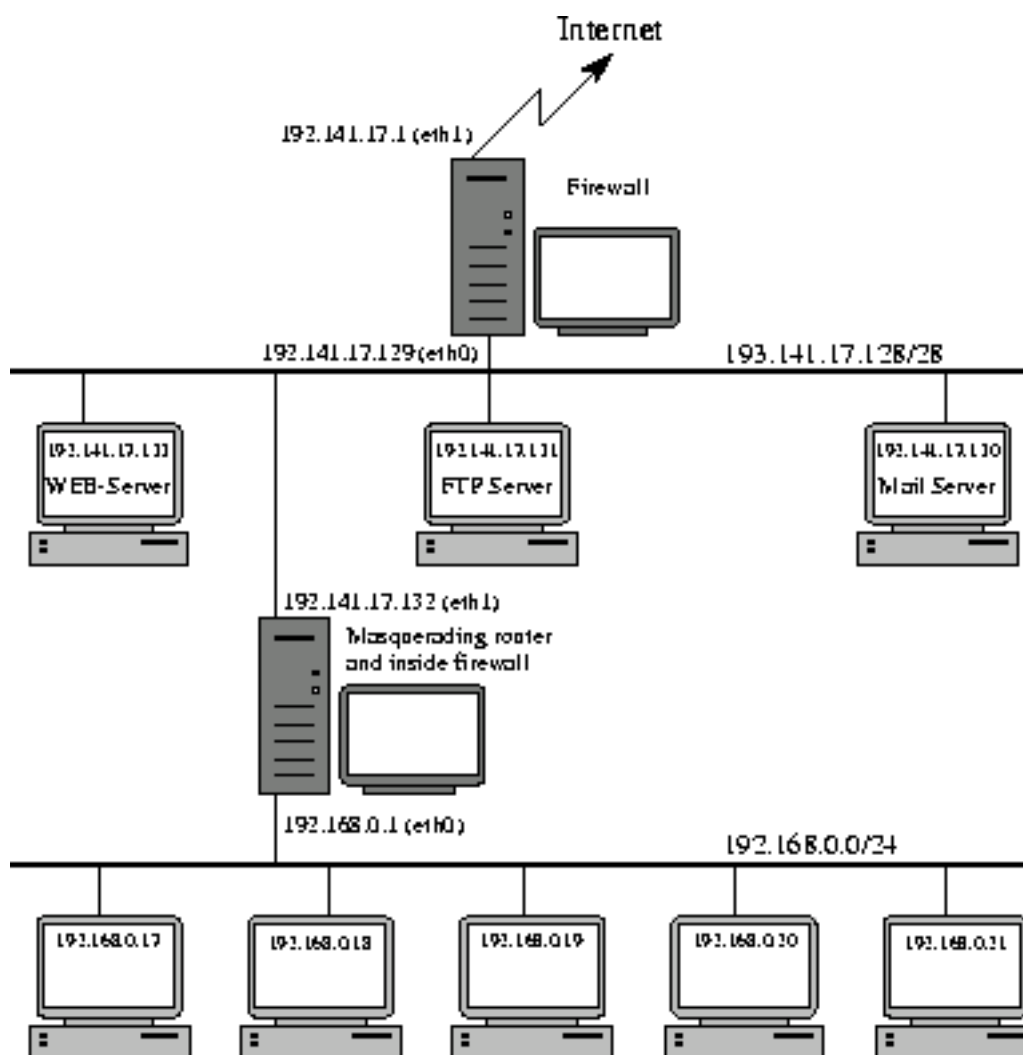


Figura 2.2: Ejemplo de una red con un firewall instalado.

Como vez, ahora tenemos dos segmentos esta vez. Las direcciones privadas IP de las maquinas internas dentro del segmento (192,168,0,0/24) están conectados con el otro segmento a través de un firewall enmascarado. Las maquinas en el segmento exterior tienen direcciones IP validas y estan conectados atravez de este equipo, que proporcionan los servicios que pueden y deben ser accesibles del Internet. La conexión de este segmento a Internet es realizada por una máquina que esta como gateway y firewall. Esta topologie tiene algunas ventajas sobre todas las máquinas que pudieran estas únicamente en un solo segmento. La estación de trabajo local esta protegida por dos firewall. Si alguien desea entrar a esta maquina por el primer firewall, esta persona únicamente tendra permiso a las máquinas que proporcionan los servicios, como HTTP o FTP que se les proporciona a las personas del exterior, pero no podra entrar a nuestro segundo segmento que esta protegido por el otro firewall. Por supuesto él podra dañar o destruir nuestro Web site, pero los datos sensibles verdaderos todavía estan protegidos por el segundo firewall. Bajo ninguna circunstancia podra tener acceso directo al segmento interior si este no se le permite.

Aspectos generales de bash2

Aspectos generales de bash2

-Tema aún no terminado.-

Como funciona este script

Con el conocimiento básico mencionado en el capítulo anterior sobre servicios del TCP/IP, es fácil diseñar un filtro. Para agregar las reglas necesarias a la cadena de entrada de información al que remiten todos los paquetes de entrada que provienen del dispositivo que apunta al mundo exterior e ira a la dirección IP que deseamos proteger con la variable `user_fw` de la cadena definida. Todos los paquetes que vienen de la red local y que van al exterior serán enviados a la variable `user_out` de la cadena definida.

Entonces agregamos las reglas a la variable `user_fw` de la cadena. La primer cosa a hacer es evitar el spoofing. El Spoofing significa que alguien está fingiendo estar en el interior de nuestra red local, pero realmente está viniendo del exterior. Esto se hace generalmente falsificando o substituyéndolo por una dirección de nuestra red local. La detección de esta clase de paquetes es algo fácil, como ningún paquete de nuestra LAN están viniendo del dispositivo que señala al exterior, podemos eliminar todos los paquetes que tengan una dirección local y que provengan del dispositivo de entrada de nuestro firewall agregando las reglas de negación para estos paquetes. Lo siguiente es validar los paquetes de las maquinas de las cuales confiamos el acceso dentro de la regla. Con la variable `ACCEPT`, esta aceptara todos los paquetes de estas direcciones que le mencionemos. Hay que tener cuidado con esto, pues estos paquetes pueden ser "spoofed". Entonces hay que crear una lista con reglas validas para los diferentes tipos de servicio como ftp, HTTP, HTTP seguro (SSL), SSH, smtp, NNTP y DNS. Esto se verá en el párrafo siguiente donde se explica la configuración. Entonces si tomaremos un rango de puertos cerrarlos para el tráfico de UDP y de TCP. Hay que recordad, que que las reglas se llevan un orden secuencial. Puedes aquí dar el acceso para los servicios que quieras permitir, pues los paquetes legales han sido detectados por las reglas en la cadena antes de que los paquetes vengan del dispositivo. Otro punto interesante, es validar todo que no se ha negado hasta ahora.

Cuando la cadena para el tráfico entrante se instala, las reglas para el tráfico saliente se agregan a la variable `user_out` de la cadena. Esto se hace solamente si se desea vigilar el tráfico saliente de todos los equipos de nuestra LAN. Si éste es el caso, una regla para validar se agrega para cada maquina que requiera tener el acceso. Entonces una regla de negar bloquea tráfico de el resto de las maquinas.

Las opciones se fijan para validar y para negar las reglas para registrar si están especificados en la instalación. Antes de que se haga cualquier otra cosa, es cierto que debemos de controlar el flujo del servicio, y las condiciones de la maquina para tener un buen firewall (tener una dirección IP valida y soporte del Kernel de Linux para el firewall).

Soporte para el servicio proxy

Una cosa no mencionada hasta ahora es el soporte para el servicio proxying transparente. Pues no se relaciona directamente con el firewall y se a pospuesto hasta esta sección. El Kernel de Linux utiliza (si está configurado correctamente) el cambio de dirección de paquetes destinados a las máquinas remotas a los accesos locales. Estes puede ser utilizado para la entrada y salida tráfico. Por ejemplo, se puede volver a redirigir todas las conexiones salientes al puerto 80 (HTTP) a un acceso local, para filtrarlas a un proxy server local. Tambien para esto, las reglas se agregan a la cola de las variables `user_fw` y `user_out` si este servicio es especificado dentro de la configuración.

Configuración del firewall

Al igual que el script para el funcionamiento de enmascaramiento, el firewall también tiene un script para su configuración y puesta en marcha dentro del directorio `/etc/rc.d/init.d/firewall` y el archivo de configuración se localiza en: `/etc/sysconfig/firewall/config`

Estas variables tienen el prefijo `FW_` y siguen el mismo formato. Contienen una lista de IP's o de las direcciones de red con las máscaras opcionales, separada por espacios en blanco. La descripción de estas reglas se describen abajo. Hay que utilizar una dirección IP, los nombres de las maquinas que no tendran resolución de IP. Durante el proceso de instalación del firewall todo el tráfico de la red sera bloqueado, así que ninguna petición del servidor de nombres se pueden utilizar para resolver nombres de direcciones IP.

En vez de especificar una dirección IP se puede utilizar también la cadena especial `IP@device`. Este substituirá la dirección IP que tiene el dispositivo de entrada de la red que tiene en ese momento que se ejecuta el script del firewall. Esto es útil si se tiene una conexión dialup donde su dirección IP se cambia cada vez que se conecta el ISP.

A continuación se muestra una lista con la mayoría de las variables para configurar el firewall. La mayoría de las variables usadas tienen significados obvios:

FW_START

El firewall comenzara solamente si el script se encuentra dentro de los archivos de arranque y la variable se fija a " yes ". No obstante se puede inicializar el script de forma manual incluso si este se fija a " no ".

FW_WORLD_DEV

Dispositivo que debe ser protegido. Puedes tener una lista de dispositivos aquí, si es que

FW_WORLD_DEV

Dispositivo que debe ser protegido. Puedes tener una lista de dispositivos aquí, si es que tienes más de un dispositivo de salida *-es decir dispositivos virtuales para los servidores del WEB-*. Todo el tráfico en estos dispositivos será vigilado por las reglas del firewall. Si tienes conexión dialup PPP éste puede ser el dispositivo ppp0. Puede ser un dispositivo ISDN si marcas hacia fuera con una tarjeta del ISDN.

FW_LOCALNETS

Lista de redes locales. Solamente las direcciones IP listadas aquí estarán protegidas. Si deseas proteger el firewall por sí mismo, la dirección IP de él debe de estar listado aquí.

FW_TCP_LOCKED_PORTS

Los números de acceso TCP que se deseen bloquear aquí deben de listarse en un rango que consista en pares de números separados por dos puntos. Por ejemplo: "1:6 8:1023". Los puertos 1 a 6 y 8 a 1023 se encuentran bloqueados. El valor por omisión es bloquear todos los puertos hasta 1023. Verifica que tengas los puertos más importantes aquí. Revisa la sección 2,3,2 para entender el significado de este parámetro.

FW_UDP_LOCKED_PORTS

Los números de puertos UDP que deben ser bloqueados siguen la misma sintaxis que con los puertos TCP. Se recomienda para fijar esto a 1:1023 así que todos los accesos reservados estarán bloqueados.

FW_INT_DEV

Dispositivo para la red interna. Se monitorea el tráfico de salida usando este dispositivo. Como con el dispositivo que va hacia la red exterior, puedes enumerar más de un dispositivo aquí.

FW_LOG_DENY

Si esta variable esta marcada con " yes " todas las violaciones de las reglas del firewall se registran dentro de */var/log/messages*. Esto significa que cada tentativa de romper el firewall sera registrada.

FW_LOG_ACCPET

Si esta variable esta marcada con " yes " todos los paquetes que esten permitidos dentro de las reglas del firewall seran registrados dentro de */var/log/messages*. Esto significa que cada uno los paquetes que pasan por el firewall (permitidos) seran registrados dentro de ese archivo. Hay que tener cuidado con esta opción, pues crea muchos registros de entradas del registro.

FW_FTPSERVER

Direcciones de los sitios de ftp que son libremente accesibles del exterior. Esto no significa que todos los servicios de esta máquina están disponibles. Solamente el tráfico de ftp será permitido, el resto del tráfico todavía será bloqueado (igual que para el resto de los servicios cubierto por las configuraciones separadas del firewall).

FW_WWWSERVER

Direcciones de los sitios de WWW que son accesibles del exterior. Iguales que para ftp, solamente conexiones del HTTP se pueden hacer a este equipo.

FW_SSLSERVER

Direcciones de los sitios de Secure-Socket-Layer (SSL) WWW que son accesibles del exterior. Es necesario, que el acceso del SSL se especifique en FW_SSLPORT.

FW_SSLPORT

Puerto donde el SSL espera recibir peticiones. Aquí usted puede incorporar solamente un número.

FW_MAILSERVER

Direcciones de los sitios SMTP que son accesibles del exterior.

FW_DNSSERVER

Direcciones de los sitios DNS que son accesibles del exterior.

FW_NNTPSERVER

Las direcciones de los sitios del NNTP que son accesibles para noticias *-véase abajo para las alimentaciones de las noticias-*.

FW_NEWSFEED

Direcciones de los servicios de noticias que se permiten conectar con los servidores del NNTP. Las dos variables *FW_NNTPSERVER* y *FW_NEWSFEED* necesitan ambas ser instaladas.

FW_ROUTER

Direcciones del router de Internet. Esta unicamente puede ser utilizada si la direccion del router no proporciona los rangos que estan en la variable *FW_LOCALNETS*, pero está

Direcciones del router de Internet. Esta únicamente puede ser utilizada si la dirección del router no proporciona los rangos que están en la variable `FW_LOCALNETS`, pero está situada en un lado desprotegido del firewall. La figura 2,3 ilustra este esquema. Se observa que la conexión a Internet está hecha con el router en uno de sus dispositivos, y otro de ellos conectado con el firewall hacia nuestro segmento por su ethernet (esto es importante, ya que ninguna otra máquina tiene que estar en este segmento).

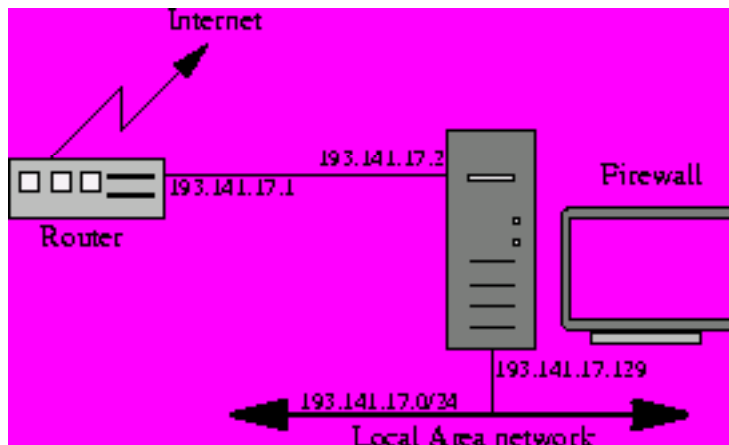


Figura 2.3: Instalación con un router dedicado

Otra cosa importante (y desafortunada) es, que la dirección IP de este router está dentro de nuestra red que deseamos proteger, en este caso la red 193,141,17,0/24. Sin tomar en consideración, que todo el tráfico que viene del router será eliminado, pues parecería un ataque tipo spoofing (hay que recordar que las direcciones IP de la red protegida no deben de provenir de nuestro dispositivo de salida, como se mencionó anteriormente). Fijando la dirección del router dentro de la variable `FW_ROUTER`, los paquetes con esta dirección origen pueden desviar el filtro spoofing. Pero aquí se ve por qué debemos de evitar una entrada como esta. Si alguien realmente hace un spoofing hacia la dirección del router, él pueden pasar el firewall.

Intente evitar esto, es decir, usando subredes para dar a su router una dirección que no pertenezca a la red interna. Este parámetro fue agregado al firewall para poder manejar estas opciones en redes existentes, donde no era posible cambiar a una topología mejor. Si diseña una red este tipo, no utilice esta característica!

FW_INOUT

Si esta variable se cambia a "yes" lea el archivo `/etc/sysconfig/firewall/fw-inout`. Las direcciones IP listadas dentro de este archivo podrán tener acceso al otro extremo del firewall.

FW_TRANSPORXY_IN

Aquí puedes incorporar una lista de puertos y de direcciones IP para redirigir el tráfico entrante a los puertos locales. Cada entrada consiste en una dirección IP origen de los paquetes de entrada, la dirección IP destino y el puerto, y el puerto local que debe ser redireccionado, todo esto separado por comas:

Source IP, Target IP, Target Port, Local Port

Esto quiere decir que si un paquete que provenga de un equipo y este trae la dirección IP fuente (Source IP) y este desea ir a un equipo que contenga el destino (Target IP) y el número de puerto al que se desea conectar, este es redireccionado al puerto local. Es decir si se desea volver a redirigir todo el tráfico a cualquier WEB SERVER en su red al web server local, puedes utilizar la variable `FW_TRABS_PROXY_IN` de esta forma:

0/0,192.141.17.0/0,80,80

FW_TRANS_PROXY_OUT

De igual forma que la variable de arriba, pero para conexiones salientes.

`FW_TRANS_PROXY_OUT` tiene el mismo significado para el tráfico saliente que

`FW_TRANS_PROXY_IN` para los paquetes entrantes. La diferencia es que la variable `_OUT` filtra el tráfico del dispositivo que se determinó en la variable `FW_INT_DEV`, mientras que `_IN` hace lo mismo con `FW_WORLD_DEV`.

filtra el tráfico del dispositivo que se determino en la variable FW_INT_DEV, mientras que _IN hace lo mismo con FW_WORLD_DEV.

FW_FRIENDS

Si esta variable se cambia a " yes " entonces el archivo /etc/sysconfig/firewall/fw-friends se lee. Si no ninguna máquina en la red local tiene acceso a salir a la internet -véase *abajo para más*-.

FW_SSH

Esta variable te da acceso del tipo SSH -acceso 22- para esos equipos que se encuentren dentro del archivo: /etc/sysconfig/firewall/fw-ssh -véase *abajo para más detalles*-.

Además de leer el archivo /etc/sysconfig/firewall de configuración, el firewall además lee algunos otros archivos que están dentro del directorio /etc/sysconfig/firewall, para conseguir mayor información sobre algunos parámetros de configuración.

/etc/sysconfig/firewall/fw-friends

Las máquinas que tienen acceso ilimitado a nuestra red local se agregan dentro de este archivo. Debes de agregar la dirección IP de cada máquina en líneas separadas por equipo. Los comentarios se pueden insertar en las líneas que comienzan con el signo de "#". Este archivo será leído solamente si FW_FRIENDS se fija a "yes". Si no ninguna máquina del exterior tiene acceso completo a la red local.

/etc/sysconfig/firewall/fw-inout

Solamente los equipos listados aquí tienen acceso directo a internet o salir de la red local. Recuerda que para que funcione la variable FW_INOUT se fija a "yes". Cada máquina no incluida en esta lista será bloqueada. Los comentarios están marcados con el signo "#" al inicio de cada línea. Si FW_INOUT se fija a "no" entonces cualquier máquina de la red local puede tener acceso al Internet.

/etc/sysconfig/firewall/fw-ssh

Si FW_SSH se encuentra en "yes" entonces todas las máquinas mencionadas tienen acceso al puerto 22. Esto significa que pueden tener acceso demonio sshd -*demonio seguro del shell*- en la red local.

Utilizando el script para funcionar el Firewall

La red del ejemplo mostrada en la figura 2.2 hace que dos máquinas estén como firewall. La tabla 2.4 muestra las configuraciones para estas dos máquinas.

Tabla 2.4: Parámetros del firewall para la red ejemplo.

	Variable	inside Firewall	outside firewall	
	FW_START	yes	yes	
	FW_WORLD_DEV	eth1	Eth1	
	FW_LOCALNETS	192.168.0.0/24	193.141.17.128/28	
	FW_FTPSERVER		193.141.17.131	
	FW_WWWSERVER		193.141.17.133	
	FW_SSLSERVER			
	FW_SSLPORT			
	FW_MAILSERVER		193.141.17.130	
	FW_DNSSERVER			
	FW_NNTPSERVER			
	FW_NEWSFEED			
	FW_INT_DEV	eth0	eth0	

	FW_INT_DEV	eth0	eth0	
	FW_LOG_ACCEPT	no	no	
	FW_LOG_DENY	yes	yes	
	FW_ROUTER			
	FW_FRIENDS	no	no	
	FW_INOUT	no	no	
	FW_SSH	no	No	
	FW_TRANSPROXY_OUT			
	FW_TRANSPROXY_IN			
	FW_TCP_LOCKED_PORTS	1 al 1023	1al 1023	
	FW_UDP_LOCKED_PORTS	1 al 1023	1 al 1023	

Lo más importante es que las variables FW_WORLD_DEV y FW_LOCALNETS estén con los valores correctos . Estas variables especifican donde el filtro de paquetes estará vigilando el tráfico, y que direcciones IP deben de estar protegidas. También, como se menciona anteriormente, las variables _LOCKED_PORTS son importantes, pues que servicios estarán bloqueados por el filtro.

El script para levantar el firewall, contiene los comandos necesarios para levantarlo. La estructura es muy similar al script de enmascaramiento. Este script tiene las opciones:

```

start
    Inicia el proceso de levantar las reglas del firewall, el filtrado de paquetes, definiciones
    realizadas por el usuario, etc.

stop
    Detiene las reglas, cadenas y definiciones realizadas por el usuario.

Reload, restart
    Lo mismo que hacen stop y start

status
    Imprime la lista que está manejando en ese momento el firewall.

```

Ahora que sabemos un poco más sobre cómo instalar un firewall. Te recomiendo echarle un vistazo al comando ipchains para conocer más y poder definir más reglas o mejorar las que se muestran en este ejemplo. Los scripts para levantar el firewall y el enmascaramiento puedes leerlos y modificarlos para configuraciones especiales que tú desees instalar en tu centro de cómputo. Recuerda que estos scripts son independientes uno del otro y debes de tener cuidado si haces modificaciones a los mismos.