FIREWALL (CORTAFUEGOS)

Introducción

En la unidad anterior hemos visto como implementar un servidor proxy con el que podamos controlar los accesos a Internet. Ahora veremos como con un firewall también conocido como muro de fuego o cortafuegos, controlaremos las redes conectadas permitiendo o denegando las comunicaciones entre dichas redes. También un firewall es considerado un filtro que controla el trafico de varios protocolos como TCP/UDP/ICMP que pasan por él para permitir o denegar algún servicio, el firewall examina la petición y dependiendo de este, la puede bloquear o permitirle el acceso. Un firewall puede ser un dispositivo de tipo Hardware como por ejemplo un router o software que se instala entre la conexión a Internet y las redes conectadas en el lugar.

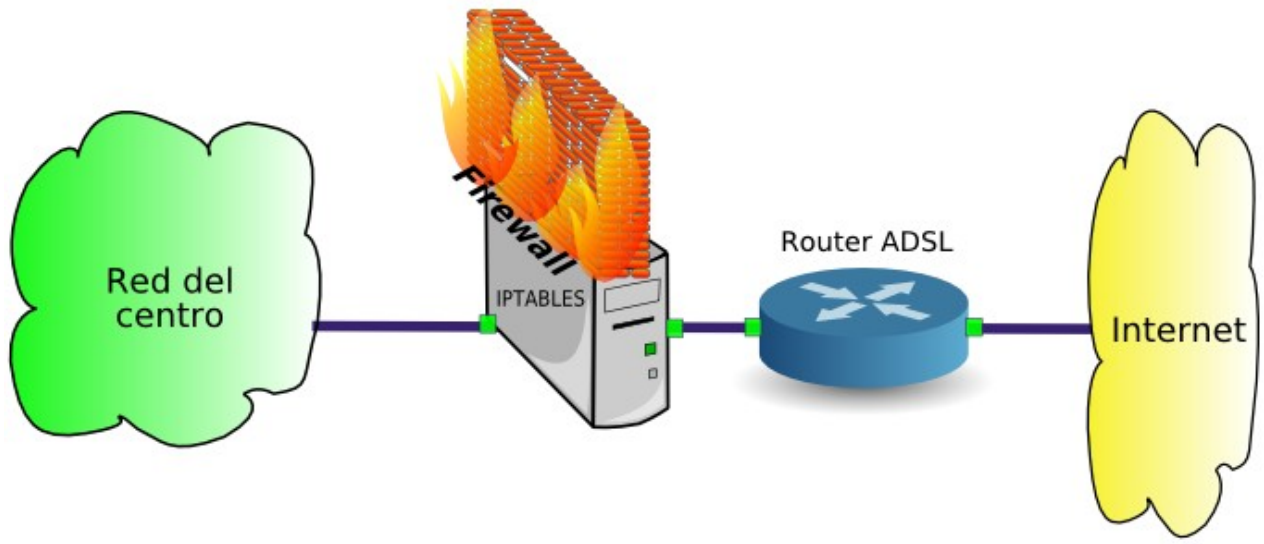
Firewall en Linux.

En Linux existe gran variedad de herramientas que nos permite controlar nuestro firewall desde un servidor que este conectado a Internet y a la red local. La más potente y difundida que suele venir por defecto en las distribuciones Linux es **iptables** (antes llamada **ipchains**). Aunque también existen otras herramientas como **Shorewall** que es una herramienta muy flexible, rápida y sencilla que permite crear reglas iptables usando archivos, o **ufw** que es un herramienta que nos permite crear reglas iptables de una forma muy simple dentro de distribuciones debian, ubuntu y derivados.

Más información de Shorewall y Ufw:

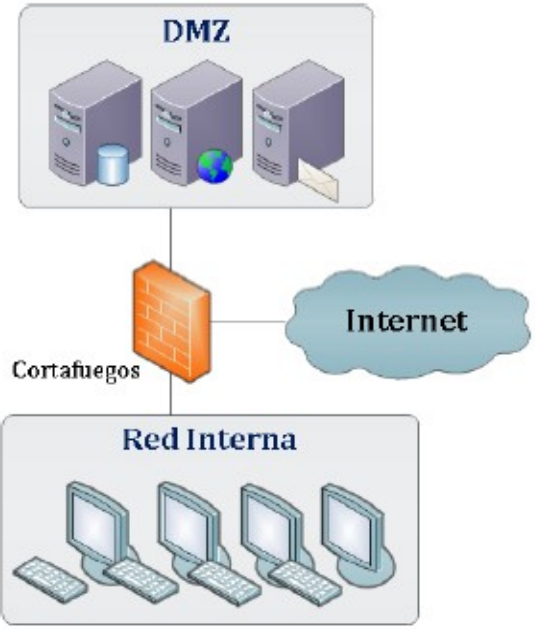
http://www.shorewall.net/

http://doc.ubuntu-es.org/UFW



DMZ.

Un firewall con configuración DMZ indica que va tener una zona Desmilitarizada o red perimetral, es una red local en la cual se encuentra dentro de una organización. Para poder ser una zona tipo DMZ deben haber servidores ofreciendo servicios de WWW, FTP, DNS, Samba, etc, esto permite ofrecer servicios de una red local hacia el exterior. Dentro de esta zona se podrá tener acceso desde la red local e Internet y firewall controlara los accesos a los servicios que se encuentren alojados dentro de la DMZ.



Mas información:

http://es.wikipedia.org/wiki/Zona_desmilitarizada_%28inform%C3%A1tica%29

Conceptos Iptables.

Antes de poder administrar nuestro firewall tendremos que saber para que nos sirve cada de una de las tablas que usa iptables para sus reglas.

Tablas

Cuando nosotros enviamos un paquete o una solicitud de servicio este pasa por tres tipos de tablas que debemos conocer.

NAT

Esta tabla que debe ser usada cuando se desea hacer los paquetes sean enrutados a una máquina cliente dentro de una red local o DMZ, pero también podremos enmascarar un red local y tener salida hacia Internet. Dentro de esta tabla tenemos las siguientes opciones:

- **POSTROUTING.** Permite establecer las comunicaciones desde la red interna al exterior. Por ejemplo, para hacer que la red interna tenga Internet.
- **PREROUTING.** Permite establecer las comunicaciones desde la red externa a la red interna. Por ejemplo, se utiliza para que desde el exterior se tenga acceso a un servidor interno.

•**DNAT:** Este parámetro se emplea cuando tenemos casos en donde se tiene un IP Publica y el servicio se encuentra dentro de la red local o DMZ y el firewall el encargado de redirigir esta petición a la máquina en donde se encuentre el servicio.
•**SNAT:** Esta opción se ocupa cuando queremos esconder nuestra IP de red local o DMZ, cambiándola dentro del firewall con la IP pública proporcionada por nuestro proveedor de Internet.
•**MASQUERADE:** Hace lo mismo que SNAT, pero MASQUERADE automáticamente convierte nuestra IP de la red local o DMZ a IP publica y se recomienda tener esta configuración cuando en nuestra red asignamos IP de forma DHCP.

MANGLE

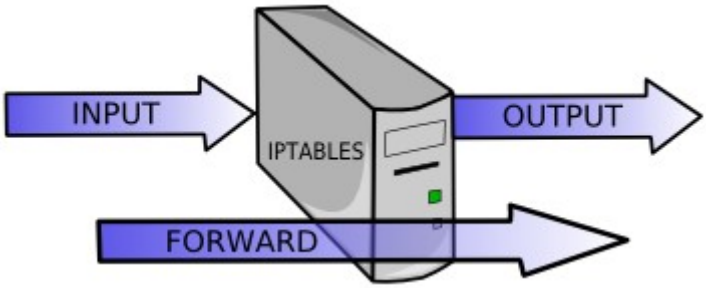
Esta tabla se usa principalmente para modificar paquetes. Dentro de esta tabla tenemos las siguientes opciones:

•**TOS:** Es usado para definir o cambiar el tipo de servicio de un paquete que puede ser usado para configurar políticas en la red considerando a ser enrutados los paquetes, no debemos usarlo para paquetes que vayan hacia Internet.
•**TTL:** Es usado para cambiar el campo tiempo de vida de un paquete y con ello conseguir un TTL especifico.
•**MARK:** Se usa para marca los paquetes con valores específicos, con estas marcas podremos limitar el ancho de banda y generar colas.

FILTER

Esta esta la tabla principal para el filtrado de paquetes donde vemos el tipo de paquete que podemos comparar y filtrar dentro del firewall. Dentro de esta tabla tenemos las siguientes tipos de paquetes:

•**INPUT:** Paquetes de entrada hacia nuestro firewall.
•**FORWARD:** Paquetes enrutados por medio del firewall a otra máquina.
•**OUTPUT:** Paquetes de salida de nuestro firewall.



Estados

Los estados en realidad son los seguimientos de conexiones dentro del firewall. Para esto tenemos las siguiente opciones:

- **ESTABLISHED:** El paquete seleccionado se asocia con otros paquetes en una conexión establecida.
- **INVALID:** El paquete seleccionado no puede ser asociado hacia ninguna conexión conocida.
- **NEW:** El paquete seleccionado esta creando una nueva conexión o bien forma parte de una conexión de dos caminos.
- **RELATED:** El paquete seleccionado esta iniciando una nueva conexión en algún punto de la conexión existente.

Podemos tomar decisiones a partir del estado del paquete por medio del modulo **state** con el parametro **“-m state”**, se refiere a la posibilidad de mantener información sobre el estado de la conexión en memoria. El seguimiento de conexiones se realiza en cadenas **PREROUTING y OUTPUT**, el numero máximo de conexiones esta guardada en **/proc/sys/net/ipv4/ip_conntrack_max.**

Protocolos

Todos los servicios manejan protocolos para su comunicaciones, por lo cual iptables podremos administrar servicios dentro de los protocolos:

- **TCP:** Protocolo de Control de Transmisión, este protocolo es mas utilizado por los servicios ofrecidos por algún servidor y en general en todo Internet y redes locales.
- http://es.wikipedia.org/wiki/Transmission_Control_Protocol
- **UDP:** Protocolo de Datagrama de Usuario, sirve para el envía de datagrama pero debe existir una conexión establecida.
- http://es.wikipedia.org/wiki/Udp
- **ICMP:** Protocolo de Mensajes de Control y Error de Internet, este protocolo solamente lo utilizamos cuando hacemos envío de paquetes de un máquina a otra como al hacer ping.
- http://es.wikipedia.org/wiki/Internet_Control_Message_Protocol

Para poder utilizar estos protocolos podremos usar el parámetro **“-p”.**

Objetivos/Acciones

Cuando nosotros creamos una regla iptables tenemos varias acciones básicas en las cuales podremos indicar al firewall que hacer con ellas. Estas acciones son:

- **ACCEPT:** Acepta los paquete que pase por el firewall.
- **DROP:** Deniega los paquete que pase por el firewall, cortando la comunicación.
- **REJECT:** Funciona básicamente igual que el objetivo DROP, aunque en este caso se devuelve un mensaje de error al host que envío el paquete bloqueado.

- **REDIRECT:** Sirve para redirigir paquetes y flujos hacia una máquina de la red local o DMZ. También sirve para redirigir peticiones entre puerto del mismo firewall para la activación de servicios.
- **MASQUERADE:** Hace lo mismo que SNAT, pero MASQUERADE automáticamente convierte nuestra IP de la red local o DMZ a IP publica y se recomienda tener esta configuración cuando en nuestra red asignamos IP de forma DHCP.
- **LOG:** Este objetivo funciona para registrar información detallada sobre los paquetes que pasan por el firewall.

Comando Iptables

- Hasta este momento solamente sabemos sobre los conceptos de iptables pero ahora aprenderemos la estructura de la creación de la reglas de iptables y con parámetros que podemos utilizar. El comando iptables contiene las siguientes opciones:

Opción	Descripción
-A	Agrega una cadena iptables al firewall.
-C	Verifica una cadena antes de añadirla al firewall.
-D	Borra una cadena de iptables en el firewall.
-E	Renombra una cadena de iptables.
-F	Libera o limpia de cadena en el firewall.
-I	Inserta una cadena en una cadena en un punto especificado por un valor entero definido por el usuario.
-L	Lista todas las cadena de iptables aplicadas en el firewall.
-N	Crea una nueva cadena con un nombre especificando por el usuario.
-P	Configura la política por defecto en una cadena en particular y puede ser ACCEPT o DROP.
-R	Reemplaza una regla en una cadena en particular, se debe especificar el numero de regla.
-X	Borra cadenas especificada por el usuario, no se permiten borrar cadenas no creada por el usuario.
-Z	Pone en ceros los contadores de bytes y de paquetes.

Parámetros Iptables

El comando iptables tiene varios parámetros que debemos conocer antes de ver algunos ejemplos ya que estos parámetros nos sirven para indicar alguna propiedad a nuestra regla creada dentro de firewall. Veámos los siguientes parámetros de iptables.

Parámetros	Descripción	Ejemplo
-d	Especifica IP destino, se usa para el redireccionamiento de servicio dentro de la red local o DMZ.	-d 192.168.2.100 -d 10.0.2.25
-i	Especificamos una interfaces de entrada. Se pueden especificar las conexiones que vienen de internet, red local o DMZ	-i eth0 -i wlan0 -i ppp0
-j	Especifica la acción a realizar.	-j ACCEPT -j DROP -j REJECT
-o	Indica una interfaz de salida. Se ocupa solamente para conexiones de la red local o DMZ	-o eth1 -o ppp0
-p	Especificamos el tipo de protocolo a utilizar en los paquetes.	-p tcp -p udp -p icmp
-s	Especificamos la dirección origen del envío de paquetes.	-s 192.168.2.0/24 -s 0.0.0.0/0
--dport	Puerto de entrada o destino de algún servicio.	--dport 22 --dport 80
--sport	Puerto de salida de algún recurso, utilizado dentro de la red local y DMZ	--sport 1863 --sport 2845
--to	IP destino del servicio.	--to 192.168.2.100:80

Con esto ya tenemos todas las opciones necesarias necesarias que podremos utilizar en iptables. Para cambiar las reglas en iptables tendremos que hacerlo **como root o con sudo** delante de la orden.

Estructura de las reglas en Iptables

Cuidado con el orden en el cual disponemos las reglas.
IPTABLES LEE DE MANERA SECUENCIAL LAS CADENAS DE REGLAS.

Es decir, comienza por la primera y verifica que se cumpla la condición y la ejecuta sin verificar las siguientes.

Por consiguiente, si la primera regla en la tabla filter de la cadena input es rechazar cualquier paquete,las siguientes reglas no serán verificadas, y se rechazará cualquier paquete. Como norma general estos son los pasos a seguir:

1. Borrar las reglas y las cadenas que hubiera, para asegurarnos de que sólo estén cargadas nuestras reglas.

2. Establecer las políticas por defecto para saber qué hacer si un paquete no coincide con ninguna regla.

3. Empezar el filtrado de paquetes con las reglas que queramos, cuidando el orden: pondremos las reglas de más específicas a más generales.

Ahora aprenderemos la nomenclatura.

iptables -A [Filtro] [parámetros de la regla] [objetivo]

Comenzaremos a ver algunas reglas de iptables.

Ejemplo 1: Se aceptarán todas la peticiones que vengan por la interfaz de red eth0.

iptables -A INPUT -i eth0 -j ACCEPT

Ejemplo 2: Se aceptan todas las peticiones tcp que vayan al puerto 80 por la interfaz eth0.

iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT

Ejemplo 3: Rechazamos todas las peticiones del protocolo icmp en todas las interfaces de red, no aceptamos ping.

iptables -A INPUT -p icmp -j REJECT

Firewall Básico

Para ver y entender mejor su funcionamiento ahora veremos la configuración básica de un iptables, creando nuestra reglas y describiéndo para que sirve cada una.

Primero limpiamos las reglas de iptables en todas las tablas.

iptables -F

iptables -X

iptables -Z

iptables -t nat -F

Establecemos política por defecto de cada de una de la tablas.

iptables -P INPUT ACCEPT

iptables -P OUTPUT ACCEPT

iptables -P FORWARD ACCEPT

Aceptamos conexiones locales en la interfaz lo (loopback) para evitar errores del sistema

iptables -A INPUT -i lo -j ACCEPT

Ahora aceptaremos todas las comunicaciones que nos interesan y luego denegamos el resto.

Aceptamos todas la conexiones entrantes al **puerto 22/ssh** por la interfaz de red eth0.

iptables -A INPUT -i eth0 -p tcp --dport 22 -j ACCEPT

Aceptamos todas la conexiones entrantes al **puerto 80/apache** por la interfaz de red eth0.

iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT

Rechaza todas la demás conexiones desde el **puerto 1 al 1024** por protocolo **tcp/udp** por la interfaz de red eth0.

iptables -A INPUT -i eth0 -p tcp --dport 1:1024 -j REJECT

iptables -A INPUT -i eth0 -p udp --dport 1:1024 -j REJECT

Solamente queda verificar que haya ejecutado las reglas correctamente, para verificarlo ejecutamos el siguiente comando.

iptables -nL

```
enrique@ServidorCep:~$ sudo iptables -nL
Chain INPUT (policy ACCEPT)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:22
ACCEPT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpt:80
REJECT     tcp  --  0.0.0.0/0            0.0.0.0/0           tcp dpts:1:1024 rej
ect-with icmp-port-unreachable
REJECT     udp  --  0.0.0.0/0            0.0.0.0/0           udp dpts:1:1024 rej
ect-with icmp-port-unreachable

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Si queremos probar la efectividad de nuestro firewall, podemos probar a bloquear el puerto 3128 usado por el proxy y veremos como los clientes que hubiéramos configurado para que navegaran a través de él dejarán de poder hacerlo

iptables -A INPUT -i eth0 -p tcp --dport 3128 -j REJECT

Usando esa regla cualquier petición tcp al puerto 3128 que llegara al servidor sería rechazada. Si queremos borrar esa última regla para volver a dejar el firewall como antes usamos el parámetro “-D”

iptables -D INPUT -i eth0 -p tcp --dport 1:1024 -j REJECT

Firewall LAN

Ahora veremos como configurar un firewall del tipo LAN:

- Los clientes de la red local podrán acceder a Internet pero sólo a los servicios de **HTTP/HTTPS y DNS**.
- Desde Internet se permitirá conectarse a servicios de **HTTP/FTP** que están dentro de la red local.

Como siempre que empezamos una configuración limpiamos las reglas de iptables en todas las tablas.

iptables -F

```
iptables -X
iptables -Z
iptables -t nat -F
```

Establecemos políticas por defecto

```
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
```

Todas la peticiones que vengan de Internet hacia el **puerto 80** redirigirlo a nuestro servidor apache con IP 192.168.2.100:80.

```
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 80 -j DNAT --to 192.168.2.100:80
```

Todas la peticiones que vengan de Internet hacia el **puerto 21** redirigirlo a la máquina que tenga el servidor FTP

```
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 21 -j DNAT --to 192.168.2.100:21
```

Aceptamos **conexiones locales en la interfaz lo**

```
iptables -A INPUT -i lo -j ACCEPT
```

Tenemos acceso al firewall desde el segmento de red 192.168.2.0 por la interfaz eth0

```
iptables -A INPUT -s 192.168.2.0/24 -i eth0 -j ACCEPT
```

Aceptamos que todo el trafico que viene desde la red local y vaya hacia los puertos

80/443 sean aceptadas, estas son **solicitudes http/https**

```
iptables -A FORWARD -s 192.168.2.0/24 -i eth0 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -s 192.168.2.0/24 -i eth0 -p tcp --dport 443 -j ACCEPT
```

Aceptamos las **consultas de DNS** de la red local

```
iptables -A FORWARD -s 192.168.2.0/24 -i eth0 -p tcp --dport 53 -j ACCEPT
iptables -A FORWARD -s 192.168.2.0/24 -i eth0 -p udp --dport 53 -j ACCEPT
```

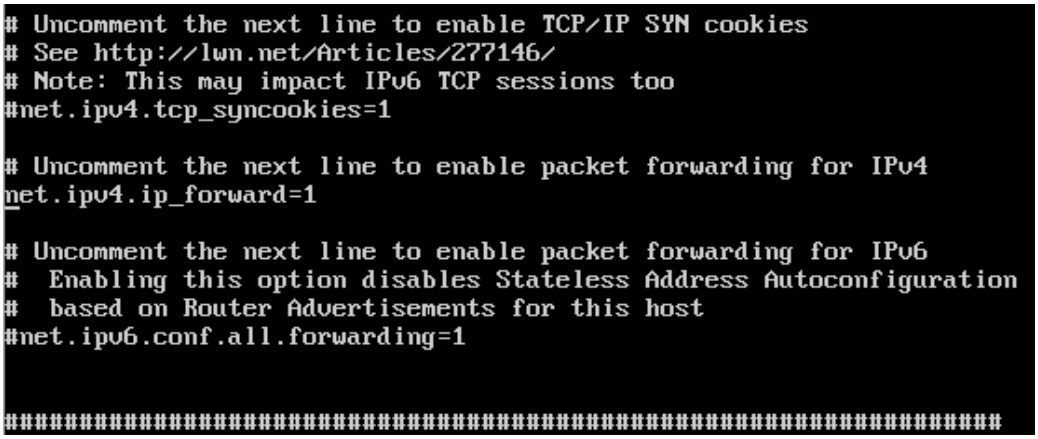
Denegamos el resto de los servicios

```
iptables -A FORWARD -s 192.168.2.0/24 -i eth0 -j REJECT
```

Ahora hacemos **enmascaramiento** de la red local

```
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth1 -j MASQUERADE
```

Establecemos el sistema como **router** para que permita el **FORWARD** para ello editamos el archivo **/etc/sysctl.conf** para establecer la variable ***net.ipv4.ip_forward=1*** descomentando la línea donde está (Quitamos la almohadilla de delante).



Si no queremos tener que cambiarle los permisos al archivo **/proc/sys/net/ipv4/ip_forward** vamos a tener que **loguearnos como root** con

su -

sino lo hemos hecho nunca en el sistema primero debemos darle un password a la cuenta root

sudo passwd root

una vez logueados como root ejecutamos

echo “1” > /proc/sys/net/ipv4/ip_forward

Ya podemos volver a nuestro usuario

su nombre_usuario

Por último rechazamos todas la demás conexiones de servicios desde el puerto 1 al 1024 por protocolo tcp/udp por la interfaz de red eth0.

iptables -A INPUT -s 0.0.0.0/0 -p tcp -dport 1:1024 -j DROP

iptables -A INPUT -s 0.0.0.0/0 -p udp -dport 1:1024 -j DROP

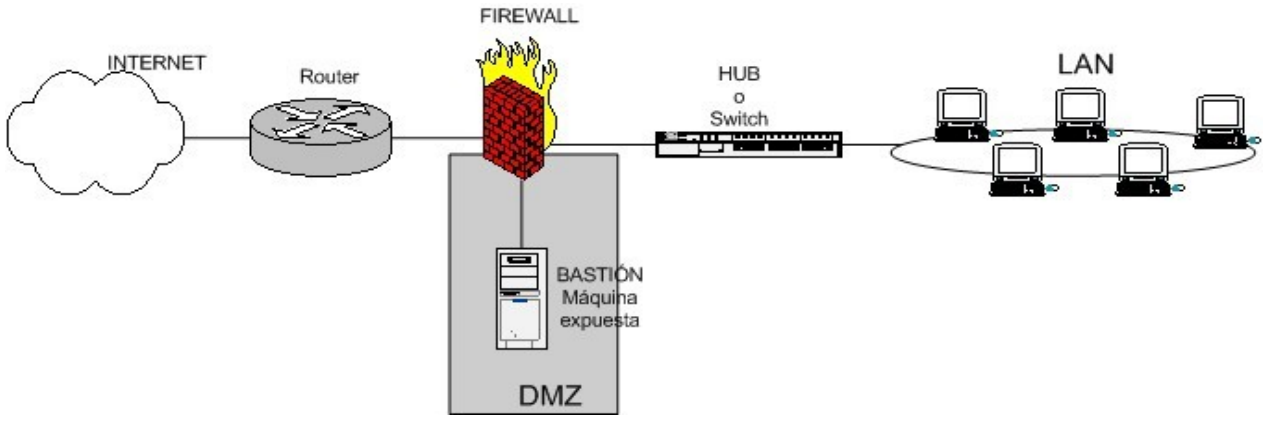
Firewall LAN/DMZ.

Ahora veremos como configurar nuestro firewall con la comunicación de la LAN/INTERNET a DMZ, es decir con un firewall con tres “patas” (interfaces de red):

eth0 conectada a Internet

eth1 conectada a una red local por eth1, red 192.168.2.0/24

eth2 a una zona de servidores DMZ, red 10.0.2.0/24



- Los clientes de la red local pueden conectarse a servicios del tipo APACHE en la DMZ,
- Desde Internet se permitirá conectarse a servicios de APACHE que se encuentran en DMZ, el servidor apache tiene la ip 10.0.2.30.

Limpiando reglas de iptables en todas las tablas.

```
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
```

Establecemos política por defecto

```
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -t nat -P PREROUTING ACCEPT
iptables -t nat -P POSTROUTING ACCEPT
```

Todas la peticiones que vengan de Internet hacia el puerto 8080 redirigirlo a la máquina de la DMZ con IP 10.0.2.30:80 que es el servidor Apache.

```
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 8080 -j DNAT --to 10.0.2.30:80
```

Aceptamos conexiones locales en la interfaz lo

```
iptables -A INPUT -i lo -j ACCEPT
```

Tenemos acceso al firewall desde la red local y DMZ

```
iptables -A INPUT -s 192.168.2.0/24 -i eth1 -j ACCEPT
iptables -A INPUT -s 10.0.2.0/24 -i eth2 -j ACCEPT
```

Ahora hacemos enmascaramiento de la Red Local y DMZ.

```
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
iptables -t nat -A POSTROUTING -s 10.0.2.0/24 -o eth0 -j MASQUERADE
```

echo “1” > /proc/sys/net/ipv4/ip_forward

Conexión del servidor Apache desde la red local a DMZ.

iptables -A FORWARD -s 192.168.2.0/24 -d 10.0.2.30 -p tcp –-dport 80 -j ACCEPT
iptables -A FORWARD -s 10.0.2.30 -d 192.168.1.0/24 -p tcp –-dport 80 -j ACCEPT

Denegamos los demás servicios.

iptables -A INPUT -s 0.0.0.0/0 -p tcp -dport 1:1024 -j DROP

iptables -A INPUT -s 0.0.0.0/0 -p udp -dport 1:1024 -j DROP

Reglas extras.

En esta sección sólo mostraremos algunas otras reglas que han faltado explicar.

Habilitando varios puerto en una regla.

Dentro de iptables tenemos la capacidad de hacer reglas para nuestro firewall con varios puerto de conexión al mismo tiempo.

Ejemplo 1:Permitimos las conexión desde cualquier equipo de la red local al servidor en los puerto 22 y 80.

iptables -A INPUT -s 192.168.2.0/24 -p tcp -m multiport --dport 22,80 -j ACCEPT

Ejemplo 2:Solamente permitiremos la conexión del cliente con la IP 192.168.2.50 a los puertos 20,21 y 23.

iptables -A INPUT -s 192.168.2.50 -p tcp -m multiport--dport 20,21,23 -j ACCEPT

Ejemplo 3: Rechazamos todas las peticiones entrantes desde el puerto 1 al 2500

iptables -A INPUT -s 0.0.0.0/0 -p udp -dport 1:2500 -j DROP

Proxy Transparente.

Como vimos en la unidad anterior con iptables podemos configurar el proxy para que sea transparente a los clientes sin tener que configurarlos uno a uno, redirigiendo las peticiones que le llegan a la interfaz de red que conecta con la LAN eth0 del puerto 80 al puerto del proxy 3128 (o 8080 según lo hayamos configurado).

Toda peticiones que venga por la interfaz de red eth0 y con salida al puerto 80 redireccionar al puerto 3128.

iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128

Bloquear pings.

Explicaremos varias reglas que podremos utilizar para bloquear los ping. Ejemplo 1:Podremos bloquear los pings que nos envíe un cliente, o un segmento de red.

iptables -A INPUT -p icmp -s 192.168.2.0/0 -j DROP

iptables -A INPUT -p icmp -s 192.168.2.20 -j DROP

Ejemplo 2:Pero si quisiéramos bloquear completamente los pings de cualquier fuente

iptables -A INPUT -p icmp -s 0.0.0.0/0 -j DROP

También es posible bloquear clientes, etc por la **MAC Address** un cliente.

iptables -A INPUT -m mac --mac-source 00:15:C5:B5:33:6C -j DROP

Guardando configuraciones

Después de haber introducido nuevas reglas en iptables si reiniciamos el servidor estas reglas se perderán. Hay varias formas de que estas reglas permanezcan si las queremos usar en más ocasiones, o cargarlas siempre que se arranque el sistema. Una de ellas es creando un archivo que usaremos **como un servicio** que podremos arrancar, detener o cargar al inicio como la mayoría de servicios del sistema.

Para ello nos iremos a la carpeta **/etc/init.d/** y allí creamos un archivo que llamaremos **iptables.cf**

sudo nano iptables.cf

editamos el archivo y metemos nuestras reglas iptables, para probarlo podemos introducir las reglas del firewall básico que hemos visto anteriormente. A continuación damos **permisos de ejecución** al archivo iptables.cf recién creado

sudo chmod a+x /etc/init.d/iptables.cf

y a partir de aquí ya podemos ejecutar el archivo como si de otro servicio se tratase cargando cada vez que lo iniciemos las reglas que hemos definido dentro de él

sudo service iptables.cf start

También podemos reiniciarlo o pararlo

sudo service iptables.cf restart

sudo service iptables.cf stop

comprobamos que efectivamente estas reglas están cargadas y correctas

sudo iptables -nL

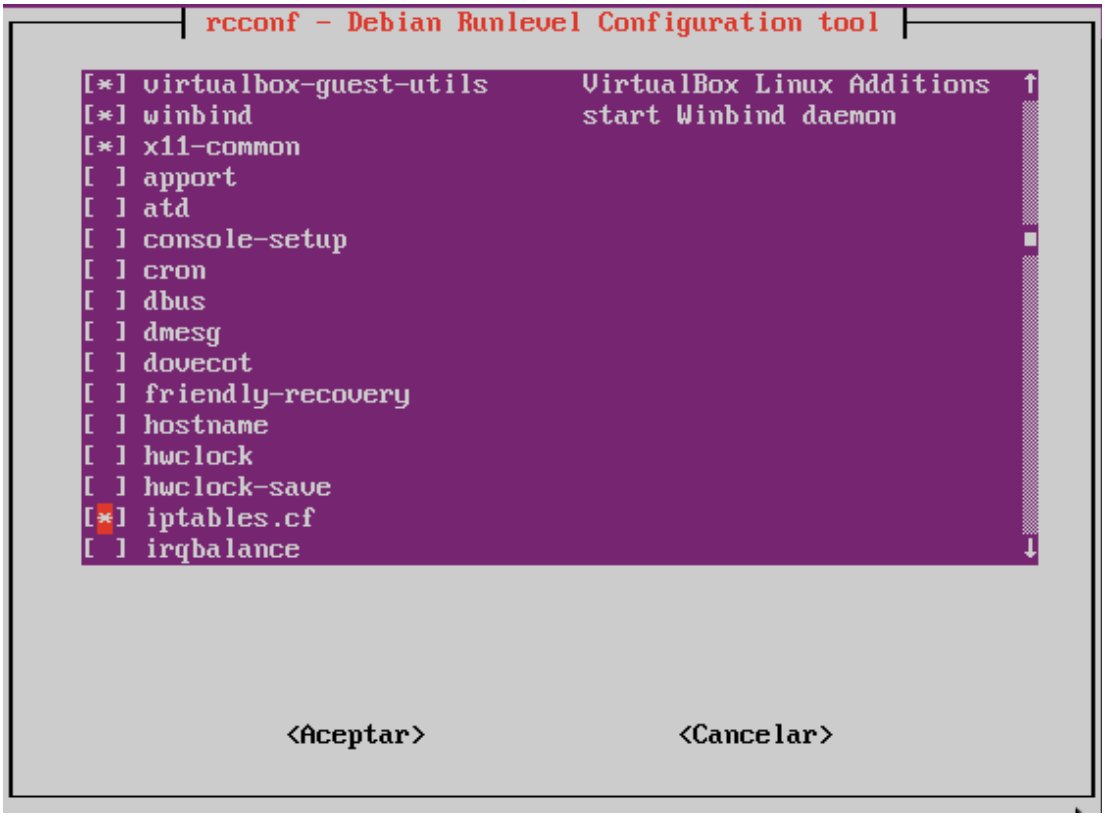
Si además quisiéramos que se ejecute al arranque del sistema **instalaremos un pequeño programa** que nos ayuda a configurar los archivos que se ejecutan al iniciar el equipo

sudo apt-get install rcconf

y lo ejecutamos

sudo rcconf

Seleccionando de la lista el archivo ejecutable que hemos creado **iptables.cf**



Otra forma si ya tenemos reglas que hemos introducido mediante líneas de comandos y queremos guardarlas, crearemos un archivo **/etc/iptables.reglas**

```
sudo nano iptables.reglas
```

Daremos permisos de escritura al archivo, o nos logueados como root como hemos visto previamente sino queremos modificar los permisos del archivo recién creado.

```
sudo chmod a+w /etc/iptables.reglas
```

y usaremos el comando **iptables-save** indicándole que queremos guardar las reglas en nuestro archivo iptables.reglas

```
sudo iptables-save > /etc/iptables.reglas
```

Con eso **pasaremos toda la configuración que tenga iptables en ese momento al archivo iptables.reglas**. Comprobamos que las reglas que tuviéramos definidas están dentro del archivo

```
sudo nano /etc/iptables.reglas
```

Si reiniciamos y queremos recargar la configuración de nuestro firewall podemos usar **iptables-restore**

```
sudo iptables-restore < /etc/iptables.reglas
```

Si además queremos que se carguen al iniciar el interfaz de red cuando se arranca el sistema, podemos añadir al archivo **/etc/network/interfaces** la línea

```
pre-up iptables-restore </etc/iptables.rules
```

Aparte de estas dos formas, también podemos almacenar las reglas que definamos en iptables como scripts para ejecutarlas cuando queramos como lo haríamos con cualquier otra cadena de comandos que formaran un script en Linux.

Otro supuesto práctico

Vamos a ver un último supuesto práctico para comprender mejor iptables, se va a configurar un servidor para que actúe como **router**. Para ello, es necesario realizar los siguientes pasos:

- Configurar las interfaces de red para que el servidor tenga acceso a las dos redes: Internet y la red interna.
- Configurar iptables para que permita el acceso de la red interna a Internet.

Configuración de las interfaces de red

Siguiendo el esquema de red propuesto, la interfaz de red *eth0* es la encargada de conectarse a Internet, mientras que la interfaz *eth1* pertenece a la red interna. Los parámetros de configuración de *eth0* los tiene que facilitar el proveedor de Internet o los puede obtener automáticamente utilizando DHCP.

Como ya sabemos la configuración de las interfaces de red se encuentra en el fichero **/etc/network/interfaces**.

```
auto eth0

iface eth0 inet dhcp


auto eth1

iface eth1 inet static

address 192.168.2.100

netmask 255.255.255.0

network 192.168.2.0

broadcast 192.168.2.255

# gateway 192.168.2.100
```

Establezca el sistema para que actúe como router (logueados como root):

```
echo “1” > /proc/sys/net/ipv4/ip_forward
```

Limpiamos la configuración del cortafuegos:

```
iptables -F
```

```
iptables -X
iptables -Z
iptables -t nat -F
```

Indicamos que la red interna tiene salida al exterior por NAT:

```
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -d 0/0 -j MASQUERADE
```

Permitimos todo el tráfico de la red interna y todo lo demás se deniega:

```
iptables -A FORWARD -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -j DROP
```

Guarde la configuración del cortafuegos ejecutando como root

```
iptables-save >/etc/iptables.rules
```

y modifique el fichero ***/etc/sysctl.conf*** para establecer la variable

net.ipv4.ip_forward=1.

Para comprender mejor iptables, se va a realizar una mejora del supuesto en la que la red interna sólo tiene acceso al exterior para ver páginas web (puerto 80/TCP) y para la resolución de nombres (53/UDP y 53/TCP). Además, se va a publicar un servidor web interno que se encuentra en la dirección 192.168.2.100.

Como siempre limpiamos la configuración del cortafuegos:

```
iptables -F
iptables -X
iptables -Z
iptables -t nat -F
```

Indicamos que la red interna tiene salida al exterior por NAT.

```
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -d 0/0 -j MASQUERADE
```

Se permite sólo el tráfico web (80/tcp) y DNS (53/udp y 53/tcp). Todo lo demás se deniega:

```
iptables -A FORWARD -s 192.168.2.0/24 -p TCP --dport 80 -j ACCEPT
iptables -A FORWARD -s 192.168.2.0/24 -p TCP --dport 53 -j ACCEPT
iptables -A FORWARD -s 192.168.2.0/24 -p UDP --dport 53 -j ACCEPT
iptables -A FORWARD -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -j DROP
```

Redirijimos el tráfico web que entra por la interfaz externa (eth0) al servidor de la red interna:

```
iptables -t nat -A PREROUTING -i eth0 –p tcp --dport 80 -j DNAT -- to 192.168.2.100:80
```

Guardamos la configuración del cortafuegos ejecutando:

```
iptables-save >/etc/iptables.rules
```

Como podemos ver las diferentes configuraciones y posibilidades de iptables son casi ilimitadas y se van comprendiendo mejor cuanto más ejemplos vamos viendo y probando.

Este artículo esta licenciado bajo Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License.

Servidores Linux Enrique Brotons