Configuración básica de Firewall con IPtables

Asegurando la red

IPtables es el firewall que viene por default en la mayoría de las distribuciones de Linux. Por default no contiene ninguna regla configurada como es de esperarse, y podemos verificarlo tecleando en la terminal el comando `sudo iptables -L -v`, lo cual nos debe arrojar algo como lo siguiente:

```
```

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               desti

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               desti

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               desti
```

Esto cuenta con tres "cadenas" de reglas, las cuales son **INPUT**, **FORWARD** y **OUTPUT**.

INPUT se encarga de revisar el trafico de red que va a ingresar al servidor. **OUTPUT** hace exactamente lo opuesto. La cadena **FORWARD** controla el tráfico que se maneja entre diferentes redes en el servidor.

La política que tiene configurada por default es **ACCEPT** para cada una de las tres cadenas, lo que hace que ambas permitan todo el tráfico de red, además no contiene ninguna regla dentro de estas cadenas por lo que nuestro servidor está completamente abierto a todo el tráfico de red.

Principalmente vamos a trabajar con la cadena **INPUT** para proteger el servidor contra tráfico externo dirigido hacia él.

> *Es posible modificar las políticas default para cualquier cadena a **DROP** o **REJECT**. Para saber más al respecto aquí: http://serverfault.com/questions/157375/reject-vs-drop-when-using-iptables.*

Persistencia de reglas

Las reglas que indiquemos en nuestro firewall no son persistentes y se eliminarán tras cada reinicio que hagamos a nuestro servidor.

Sin embargo podemos hacer algo al respecto, existen un comando que nos va a permitir guardar las reglas que tengamos en un simple archivo de texto, también existe otro comando que nos permite restaurar estas reglas en caso de haberlas perdido.

```
# Mostrar las regulas hacia la salida estandar
sudo iptables-save

# Crear reglas a partir de un archivo de texto
sudo iptables-restore
```

Por ejemplo, si quisieramos guardar las reglas actuales podemos hacerlo con el siguiente comando:

```
sudo iptables-save > ~/rules.v4
```

Después, podemos restaurar estas reglas:

```
sudo iptables-restore < ~/rules.v4
```

Persistir reglas tras reinicios (Debian/Ubuntu)

También podemos hacer uso de un paquete que está disponible para distribuciones Debian y Ubuntu llamado *iptables-persistent* y podemos instalarlo con los siguientes comandos:

```
# Actualizar repositorios locales
sudo apt-get update
# Instalar iptables-persistent
sudo apt-get install -y iptables-persistent
# Iniciar el servicio de iptables-persistent
sudo service netfilter-persistent start
# Añadir el servicio al inicio del sistema
sudo invoke-rc.d netfilter-persistent save
# Detener el servicio (en caso de requerirlo)
sudo netfilter-persistent stop
```

Agregar reglas al Firewall

Ahora ya estamos listos para crear reglas en nuestro firewall.

Como lo mencionamos anteriormente, solamente estaremos ingresando reglas a la cadena de **INPUT**.

Primero vamos a manejar la información del loopback/localhost. El siguiente comando permitirá el flujo de datos entre los diferentes elementos de la red local *localhost* (interfaz loopback).

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

Con esto le decimos a iptables lo siguiente:

Con el parámetro **-A** vamos a adjuntar (append) la regla a la cadena INPUT.
Con el parámetro **-i** indicamos que vamos a agregar una interfaz, en este caso el loopback y lo indicamos con **lo**.
Por último indicamos el parámetro **-j** para indicar que este tráfico va a cer aceptado (ACCEPT) y se puede continuar con la siguiente regla.

Después vamos a hacer una configuración un poco más avanzada pero necesaria para asegurar que no nos vayamos a bloquear nosotros mismos del servidor (no bloquear nuestra conexión actual de SSH):

```
1  sudo iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCE
```

Esto va a permitir y mantener activas las conexiones que tengamos al momento de crear esta regla, por el momento solo estamos usando nuestra conexión SSH por lo que al crear esta regla no nos va a expulsar súbitamente.

Este comando hace lo siguiente:

Adjunta a la cadena *INPUT* con el parámetro **-A**
Utiliza el módulo *conttrack* con el parámetro **-m**
Revisa que el estado de las conexiones estén como *established* o *related* con el parámetro **--ctstate**
Indicamos que el tráfico para esta regla se va a aceptar (*ACCEPT*) con el parámetro **-j**

Si tecleamos el comando `netstat -a` podrémos ver que tenemos una conexión SSH establecida, que es como nos estamos conectando al servidor, de no agregar esta regla podríamos hacer que el firewall nos bloquee el acceso al servidor.

Reglas para el servidor WEB

Ahora vamos a añadir las reglas necesarias para permitir nuevas conexiones de SSH y el acceso al servidor WEB por el puerto 80 (default):

```
sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

Estos comandos hacen lo siguiente:

Adjuntan a la cadena *INPUT* con el parámetro **-A**
Aplica el protocolo *TCP* con el parámetro **-p**
Indica el puerto destino al puerto *22* y *80* con el parámetro **--dport**
Acepta el tráfico (*ACCEPT*) con el parámetro **-j**

Por último vamos a bloquear todo lo demás para que no se tenga acceso al servidor, salvo por lo que ya especificamos previamente. Esto lo hacemos con el siguiente comando:

```
sudo iptables -A INPUT -j DROP
```

Adjuntamos a la cadena *INPUT* con el parámetro **-A**
Ignoramos todo el tráfico (*DROP*) con el parámetro **-j**

Ahora vamos a revisar nuestras reglas

```
sudo iptables -L -v
```

Básicamente aquí podremos ver las reglas que acabamos de crear, en las que estamos especificando que vamos a permitir el tráfico del loopback/localhost, todas las conexiones que tengamos actualmente establecidas, el servidor web por el puerto 80 y el servidor SSH por el puerto 22. Por último podemos ver que tenemos una regla que va a bloquear todo lo que no esté dentro de las reglas que ya tenemos en nuestro firewall.

Insertar regla para HTTPS

Si necesitamos insertar una nueva regla no deberíamos hacer uso de **APPEND** cuando creemos la regla, ya que esto va a crear una nueva regla al final del listado que ya tenemos, para que se permita el paso de HTTPS tenemos que poner una regla antes de la que está bloqueando el resto de tráfico ya que de no hacerlo, incluso si creamos la regla bien, se estará bloqueando debido al orden de las reglas.

Supongamos que queremos poner nuestra nueva regla en la posición 5 del listado de reglas para la cadena **INPUT**, esto podemos hacerlo con el siguiente comando:

```
sudo iptables -I INPUT 5 -p tcp --dport 443 -j ACCEPT
```

Esto hará lo siguiente:

Creará una nueva regla en la posición *5* de la cadena *INPUT* con el comando **-I** (el listado de reglas comienza con el índice en el número 1 y no en el cero)
Indicamos que vamos a usar el protocolo *TCP* con el parámetro **-p**
Utilizamos el puerto default para HTTPS *(443)* con el parámetro **--dport**

Utilizamos el puerto default para HTTPS (443) con el parámetro **--dport**
Permitimos (*ACCEPT*) el tráfico para este puerto con el parámetro **-j**

Eliminar reglas

En ocasiones es necesario que eliminemos reglas del firewall, eso podemos hacerlo de una manera muy similar a como insertamos una regla por posición, sin embargo, también podemos especificar la regla completa tal y como se hizo cuando la creamos, la única diferencia es que en lugar de usar el parámetro **-A** o el parámetro **-I**, usaremos el parámetro **-D** (Delete). Veamos el siguiente comando como referencia:

```
sudo iptables -D INPUT 3
sudo iptables -D INPUT -p tcp --dport 22 -j ACCEPT
```

Estas dos reglas son equivalentes (suponiendo que la tercer regla del listado es la que permite utilizar SSH)

Encontrar reglas por comando

También es posible eliminar reglas si utilizamos el mismo comando que se uso para crearlas.

Para poder visualizar las reglas en forma de comando es tan sencillo como teclear `sudo iptables -S` .

Al hacer esto verás el listado de reglas que tenemos actualmente con el comando con el que las creamos, para poder eliminarlas simplemente debemos copiar todo el comando y cambiar el parámetro **-A** por **-D**.

Tomemos el siguiente ejemplo:

```
# Mostrar listado de reglas en formato de comando
sudo iptables -S
```

Esto nos devolverá lo siguiente:

```
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p tcp -m multiport --dports 22 -j f2b-sshd
-A INPUT -i lo -j ACCEPT
-A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -j DROP
```

Si queremos eliminar la regla que permite el paso de SSH debemos teclear en la terminal lo siguiente:

```
sudo iptables -D INPUT -p tcp -m tcp --dport 22 -j ACCEPT
```