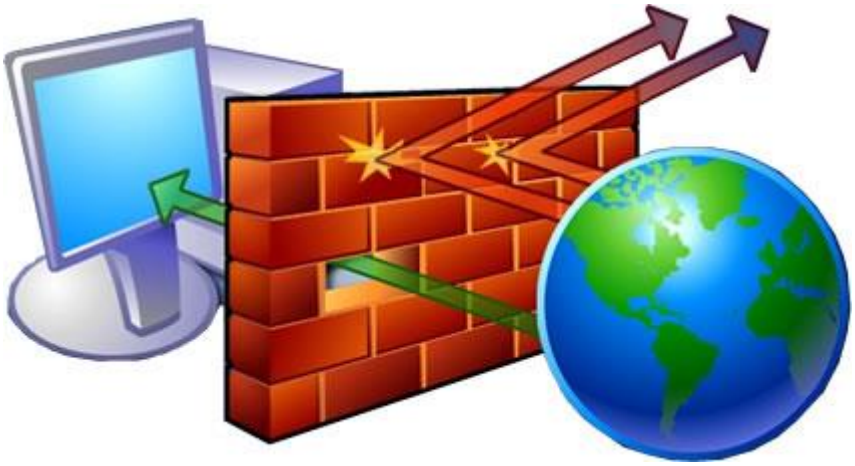
Iptables, herramienta para controlar el tráfico de un servidor



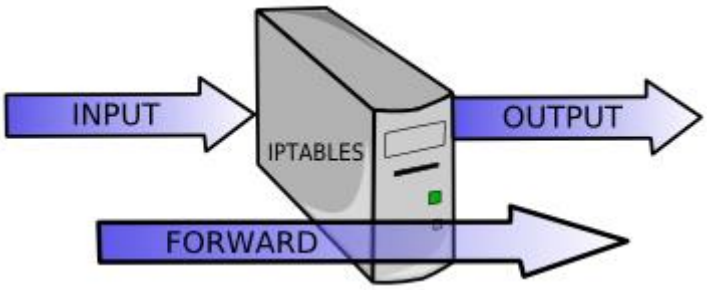
La seguridad es punto muy importante a tener en cuenta en cualquier organización de ahí que sea fundamental hacer uso de aquellos mecanismos que tengamos a nuestro alcance para poner el mayor número de barreras posibles a los atacantes.

En una estructura de servidores, el firewall ocupa un papel fundamental para salvaguardar la información que se almacena en su interior. Esta herramienta, totalmente configurable, suele tener dos funciones básicas:

- Bloquear el acceso no permitido desde una red externa.
- Restringir las conexiones de la red local con el exterior.

A lo largo de este White Paper nos centraremos en la aplicación iptables del proyecto Netfilter, pensado para sistemas de tipo Unix.

¿Qué es iptables?



Iptables es un módulo del núcleo de Linux que se encarga de filtrar los paquetes de red, es decir, es la parte que se encarga de determinar qué paquetes de datos queremos que lleguen hasta el servidor y cuáles no.

Al igual que ocurre con otros sistemas de cortafuegos, iptables funciona a través de reglas. Es decir, el usuario mediante sencillas instrucciones indica al firewall el tipo de paquetes que debe permitir entrar, los puertos por donde se pueden recibir esos paquetes, el protocolo utilizado para el envío de datos y cualquier otra información relacionada con el intercambio de datos entre redes.

Cuando en el sistema se recibe o se envía un paquete, se recorren en orden las distintas reglas hasta dar con una que cumpla las condiciones. Una vez localizada, esa regla se activa realizando sobre el paquete la acción indicada.

Gracias a su robustez, iptables se ha convertido hoy por hoy en una de las herramientas más utilizadas para el filtrado de tráfico en sistemas Linux.

Tablas y cadenas

En iptables existen tres grandes grupos de reglas o tablas, dentro de las que se definen una serie de cadenas predefinidas y cualquier paquete pasará por una de ellas. Veamos a continuación cuáles son estos tres grandes grupos.

1.- Reglas de filtrado o filter

Dentro de esta tabla se indican las reglas de filtrado, es decir, indicamos qué paquetes serán aceptados, cuáles rechazados o cuáles son omitidos. Las cadenas que forman parte de esta tabla son:

- **INPUT**: Hace referencia a los paquetes que llegan al sistema
- **OUTPUT**: Filtrado de los paquetes que salen de nuestra red
- **FORWARD**: Referencia al tráfico que el router reenvía a otros equipos

2.- Tabla NAT

Mediante esta tabla, se indican las reglas para realizar el enmascaramiento de la dirección IP del paquete, redirigir puertos o bien cambiar la dirección IP de origen y destino de los paquetes. Dentro de esta tabla nos podemos encontrar las siguientes cadenas predefinidas:

- **PREROUTING**: Mediante esta cadena, indicamos que se realice una determinada acción sobre el paquete antes de que sea enrutado. Por ejemplo, para que desde el exterior se tenga acceso a un servidor.
- **POSTROUTING**: Permite realizar una determinada acción antes de que el paquete salga del cortafuegos.
- **OUTPUT**: Permite modificar los paquetes generados en el propio cortafuegos antes de ser enrutados.

3. Tabla Mangle

Es la tabla que recoge las distintas reglas que actúan sobre los flags (opciones) de los paquetes. Todos los paquetes pasan por esta tabla.

Formatos de las reglas

```
# By default this script does nothing.
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
iptables -A INPUT -p udp -i eth0 --dport 1194 -j ACCEPT
iptables -A FORWARD -i eth0 -o tun0 -m state --state ESTABLISHED,RELATED -j ACC
iptables -A FORWARD -s 10.8.0.0/24 -o eth0 -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.8.0.0/24 -o eth0 -j MASQUERADE
```

Una vez vistas las distintas tablas que trae predefinidas la herramienta iptables, es hora de ver la estructura de las reglas que se pueden crear dentro de cada una de esas tablas.

La estructura general de una regla es la siguiente.
Iptables –t [tabla] operación cadena parámetros acción

En ese esquema, los valores de tabla y cadena corresponden con las opciones vistas anteriormente. El resto de datos los veremos a continuación.

1.- Parámetros

Los parámetros son utilizados para definir el paquete al que se le aplicará la regla. A continuación os detallamos los principales parámetros que podemos utilizar tanto en su forma abreviada como completa.

- **-p, --protocol**: Sirve para especificar el protocolo que se utiliza (tcp, udp, icmp, etc. Además si queremos especificar el puerto, se acompaña del parámetro –dport.
- **-s, --source**: Con este parámetro indicamos la dirección IP de origen.
- **-d, --destination**: Especificamos la dirección IP de destino.
- **-i, --in-interface**: Especifica la interfaz de red de entrada (eth0, eth1,…)
- **-o, --out-interface**: Especifica la interfaz de red de salida (eth0, eth1,…)

2.- Acciones

Una vez definida la regla, hay que indicar la acción que realizaremos sobre aquellos paquetes que la cumplan. Para indicar esta acción, haremos uso del parámetro **–j** seguido de alguno de los siguientes valores.

- **ACCEPT**: Mediante esta acción estamos indicando que el paquete sea aceptado.
- **DROP**: Se elimina el paquete y no se le envía al equipo que hizo la petición ningún mensaje de respuesta.
- **REJECT**: Similar al caso anterior, pero en esta ocasión se manda un paquete ICMP al equipo que hizo la petición para indicarle que no está permitida.
- **DNAT**: Esta acción es utilizada en la cadena PREROUTING de la tabla NAT para modificar la IP de destino. Tiene que llevar asociado el parámetro **–to**.
- **SNAT**: Acción asociada en la cadena POSTROUTING para modificar la IP origen. Al igual que el caso anterior le tiene que acompañar el parámetro **–to**.
- **MASQUERADE**: Acción equivalente a SNAT pero utilizada cuando tenemos una dirección IP dinámica en la interfaz de salida.
- **REDIRECT**: Se utiliza en la cadena PREROUTING para modificar la dirección IP que tenga la interfaz de red de entrada.

3.- Operaciones

Mediante las operaciones, especificamos qué se hará con la regla. A continuación os indicamos los posibles valores que pueden tomar, tanto en su formato abreviado como completo.

a) –A, --append

Añade una regla al cortafuegos para que realice una acción sobre los paquetes. Por ejemplo, si queremos añadir una regla INPUT de la tabla filter que descarte todos los paquetes enviados por el protocolo TCP, haríamos lo siguiente:

iptables –t filter –A INPUT –p tcp –j DROP

La tabla por defecto sobre la que se trabaja en iptables es filter, por lo que si queremos añadir una regla sobre ella, no haría falta indicar la tabla. De esta forma, la anterior regla podría quedar de la siguiente forma.

iptables –A INPUT –p tcp –j DROP

b) –D, --delete

Operación que elimina una regla en el firewall. Si quisiéramos borrar la regla anterior, tendríamos que hacer lo siguiente.

iptables –D INPUT –p tcp –j DROP

Si hubiera varias reglas que cumplieran el patrón, únicamente se borraría la primera aparición.

c) –I, --insert

Permite añadir una regla en una determinada posición. Si no se indica la posición, la regla es insertada al principio. Por ejemplo si queremos añadir la regla que rechaza todos los paquetes tcp en la cuarta posición, tendríamos que indicarlo de la siguiente forma.

iptables –I INPUT 4 –p tcp –j DROP

d) –L, --list

Se utiliza para listar todas las reglas de una cadena. Por ejemplo, si queremos listarnos las reglas que tiene asignadas la cadena OUTPUT lo haríamos de la siguiente manera.

iptables –L OUTPUT

Esto nos sacaría por pantalla un listado de todas las reglas que hayamos definido para la cadena indicada.

e) –F, --flush

Se utiliza para borrar todas las reglas que existan para una determinada cadena.

iptables –F INPUT

En este ejemplo, estaríamos indicando que se borrasen todas las reglas que existan para la cadena INPUT.

Ejemplos de reglas

Veamos a continuación algunos ejemplos de reglas de filtrado.

1.- Reenvío de paquetes desde la interfaz eth1 hacia la interfaz eth0

iptables –A FORWARD –i eth1 –o eth0 –j ACCEPT

2.- Permitir todo el tráfico entrante desde cualquier dirección (0/0) de la red eth1 hacia cualquier destino (0/0)

iptables –A INPUT –i eth1 –s 0/0 –d 0/0 –j ACCEPT

3.- Denegar todo el tráfico entrante desde la interfaz eth2 que intente utilizar alguna dirección IP de la red local (192.168.0.0/24)

iptables –A INPUT –i eth2 –s 192.168.0.0/24 –j DROP

4.- Aceptar todos los paquetes enviados por el protocolo tcp por el puerto 25 en el servidor con ip 217.81.148.217 procedente desde cualquier sitio.

iptables –A INPUT –p tcp –dport 25 –s 0/0 –d 217.81.148.217 –j ACCEPT

5.- Hacer NAT si ip origen es 217.51.222.183 y sale por eth1

iptables -t nat -A POSTROUTING -s 217.51.222.183 -o eth1 -j MASQUERADE

Arrancar iptables

Las reglas de iptables que vayamos creando se van almacenando en memoria, por lo que cada vez que reiniciáramos el servidor, habría que volver a meter de nuevo las reglas.

Una opción para salvar las reglas que hemos creado y que sean restauradas en caso de reinicio de la máquina es por medio de la instrucción “iptables save”. Al ejecutar esta acción, estamos diciéndole al sistema que almacene las reglas en el archivo “/etc/sysconfig/iptables”. La próxima vez que se inicie el sistema, el script de inicio de iptables volverá a cargar las reglas almacenadas en ese archivo.

Otra opción que tenemos para lograr esto, es la de crearnos nuestro propio script donde iremos escribiendo las distintas reglas. Para lograr esto, debemos hacer lo siguiente:

1. Nos creamos el script que contendrá todas las reglas que vayamos creando y lo guardaremos en el directorio /etc. Supongamos que lo llamamos "reglas-iptables"
2. Le asignamos permisos de ejecución al fichero que contiene las reglas mediante la instrucción: chmod +x /etc/reglas-iptables
3. Por último debemos indicar al sistema donde se encuentra nuestro archivo para que sea ejecutado cada vez que se inicie el sistema

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

/etc/reglas-iptables

exit 0
```

Para lograr eso, añadimos en nuestro archivo "**/etc/rc.local**", un archivo que contendrá algo similar a lo que podéis ver en la imagen anterior.

Con esta modificación, al iniciar la máquina se cargarán las reglas de nuestro cortafuegos de forma automática.