Fundamentos de Iptables

Todos los datos se envían en forma de paquetes a través de Internet. El kernel de Linux proporciona una interfaz que filtra los paquetes de tráfico entrante y saliente usando tablas de filtros de paquetes. Iptables es una aplicación de línea de comandos y un firewall de Linux que puedes configurar, mantener e inspeccionar estas tablas. Puedes definirse varias tablas. Cada tabla puede contener múltiples cadenas. Una cadena no es más que un conjunto de reglas. Cada regla define qué hacer con el paquete si coincide con ese paquete. Cuando el paquete es emparejado, se le da un **TARGET**. Un objetivo puede ser otra cadena que coincida con o uno de los siguientes valores especiales:

- **ACCEPT**: Significa que el paquete podrá pasar.
- **DROP**: Significa que no se permitirá que el paquete pase.
- **RETURN**: Significa omitir la cadena actual y volver a la siguiente regla de la cadena en la que fue llamado.

En esta actividad de iptables, vamos a trabajar con una de las tablas por defecto llamada **filtro (filter)**. La tabla de filtros tiene tres cadenas (conjuntos de reglas) :

- **INPUT:** Esta cadena se utiliza para controlar los paquetes entrantes al servidor. Puede bloquear / permitir conexiones basadas en puerto, protocolo o dirección IP de origen.
- **FORWARD:** Esta cadena se utiliza para filtrar los paquetes que entran al servidor pero que deben ser reenviados en otro lugar.
- **OUTPUT:** Esta cadena se utiliza para filtrar los paquetes que salen del servidor.

Paso 1 – Instalación de Iptables Firewall de Linux

1. Instalación de Iptables

Iptables viene preinstalado en casi todas las distribuciones de Linux. Pero si no lo tienes instalado en el sistema Ubuntu

```
alex@alex-VirtualBox:~$ sudo apt-get update
```

```
alex@alex-VirtualBox:~$ sudo apt-get install iptables
```

2. Comprobación del estado actual de los iptables

Con este comando, puedes comprobar el estado de su configuración actual de Iptables. Aquí se utiliza la opción -L para listar todas las reglas y la opción -v es para una lista más tediosa. Ten en cuenta que estas opciones distinguen entre mayúsculas y minúsculas.

```
alex@alex-VirtualBox:~$ sudo iptables -L -v
Chain INPUT (policy ACCEPT 234 packets, 353K bytes)
 pkts bytes target     prot opt in     out     source               destination

  289  360K ufw-before-logging-input  all  --  any    any     anywhere
   anywhere
  289  360K ufw-before-input  all  --  any    any     anywhere             anywh
ere
  234  353K ufw-after-input  all  --  any    any     anywhere             anywhe
re
  234  353K ufw-after-logging-input  all  --  any    any     anywhere
  anywhere
  234  353K ufw-reject-input  all  --  any    any     anywhere             anywh
```

En estas líneas por ejemplo podemos observar:

```
Chain OUTPUT (policy ACCEPT 242 packets, 15682 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Esta cadena se establece en la política **ACCEPT** predeterminada. Actualmente no hay reglas para ninguna de las cadenas.

Para hacer más práctica esta actividad de Iptables, modificaremos la cadena **INPUT** para filtrar el tráfico entrante.

Paso 2 – Definición de reglas de cadena

Definir una regla significa añadirla a la lista (cadena). Aquí está el comando Iptables formateado con opciones regulares. No tenemos que especificar todos ellos.

sudo iptables -A -i -p -s --dport -j

Aquí -A significa añadir. La cadena se refiere a la cadena en la que queremos añadir nuestras reglas. Interface es la interfaz de red en la que se desea filtrar el tráfico. Protocol se refiere al protocolo de la red de los paquetes que desea filtrar. También puede especificar el puerto, no el del puerto en el que desea filtrar el tráfico.

1. Habilitar el tráfico en localhost

Queremos que todas las comunicaciones entre aplicaciones y bases de datos en el servidor continúen como de costumbre.

```
alex@alex-VirtualBox:~$ sudo iptables -A INPUT -i lo -j ACCEPT
```

-A se utiliza una opción para añadir la regla a la cadena INPUT, aceptar todas las conexiones en la interfaz lo. **lo significa la interfaz de loopback**. Se utiliza para todas las comunicaciones en el localhost, como las comunicaciones entre una base de datos y una aplicación web en la misma máquina.

2. Habilitación de conexiones en el puerto HTTP, SSH y SSL

Si queremos que nuestras conexiones regulares **HTTP (puerto 80), https (puerto 443), ssh (puerto 22)** continúen como de costumbre. Introduce los siguientes comandos para habilitarlos. En los comandos siguientes, hemos especificado el **protocolo con la opción -p** y el puerto correspondiente para cada protocolo con la **opción -dport** (puerto de destino).

```
alex@alex-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 22 -j ACCEPT
alex@alex-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 80 -j ACCEPT
alex@alex-VirtualBox:~$ sudo iptables -A INPUT -p tcp --dport 443 -j ACCEPT
```

Ahora se aceptarán todas las conexiones de protocolo TCP con puertos especificados.

3. Filtrado de paquetes basados en la fuente

Si deseas aceptar o rechazar paquetes basados en la dirección IP de origen o en el intervalo de direcciones IP, puedes especificarlo con la opción -s. Por ejemplo, para aceptar paquetes desde la dirección 192.168.1.130

sudo iptables -A INPUT -s 192.168.1.130 -j ACCEPT

Puedes eliminar paquetes de una dirección IP con un comando similar con la opción DROP.

sudo iptables -A INPUT -s 192.168.1.130 -j DROP

Si deseas eliminar paquetes de un rango de direcciones IP, debes utilizar el módulo Iprange con la opción -m y especificar el intervalo de direcciones IP con -src-range.

sudo iptables -A INPUT -m iprange --src-range 192.168.1.100-192.168.1.200 -j DROP

4. Eliminar el resto del tráfico

Nota: Es importante eliminar el resto del tráfico después de definir las reglas, ya que impide el acceso no autorizado a un servidor desde otros puertos abiertos.

sudo iptables -A INPUT -j DROP

Este comando descarta todo el tráfico entrante distinto de los puertos mencionados en los comandos anteriores. Puede comprobar su conjunto de reglas ahora con:

sudo iptables -L -v

5. Eliminación de reglas

Si deseas eliminar todas las reglas y comenzar con una pizarra limpia, puede utilizar el comando flush.

sudo iptables -F

Este comando borra todas las reglas actuales. Si deseas eliminar una regla específica, puede hacerlo con la opción -D. En primer lugar, lista todas las reglas con números introduciendo el comando siguiente:

sudo iptables -L --line-numbers

A continuación, obtendrá una lista de reglas con números.

```
11   ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
12   ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
13   ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
```

Para eliminar una regla, especifica el número en la lista y la cadena de la regla. En nuestro caso, la cadena INPUT y el número 12.

```
alex@alex-VirtualBox:~$ sudo iptables -D INPUT 12
alex@alex-VirtualBox:~$ sudo iptables -L --line-numbers
```

```
.virginm.net
10   ACCEPT     tcp  --  anywhere             0.0.1.187
11   ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
12   ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:https
```

Paso 3 – Cambios persistentes

Las reglas de Iptables que hemos creado se guardan en la memoria. Eso significa que tenemos que redefinirlos en el reinicio. Para que estos cambios sean persistentes después del reinicio, utiliza el siguiente comando en los sistemas Ubuntu:

sudo /sbin/iptables-save

Este comando guarda las reglas actuales en el archivo de configuración del sistema que se utiliza para reconfigurar las tablas en el momento del reinicio. Debes ejecutar este comando cada vez que realice cambios en las reglas. **Para desactivar este cortafuegos, simplemente limpia todas las reglas y haz que los cambios sean persistentes.**

sudo iptables -F
sudo /sbin/iptables-save

Conclusión de la práctica.

hemos utilizado el firewall de Iptables Linux para permitir solamente tráfico en puertos específicos. También nos hemos asegurado de que nuestras reglas se guardarán después del reinicio. Este firewall de Linux eliminará los paquetes no deseados, pero hay una advertencia aquí que Iptables puede gobernar solo el tráfico de ipv4. Si tu casilla de servidor virtual VPS ha habilitado la red ipv6, debe establecer diferentes reglas para ese tráfico con ip6tables.