Servidor Firewall en Linux (IPTABLES)

Servidor Firewall
Etiquetas: firewall iptables

ÍNDICE SEGURIDAD

Tabla de Contenidos [-+]

Que es un Firewall?

Un firewall también es conocido como muro de fuego, este funciona entre 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 el para permitir o denegar algún servicio, el firewall examina la petición y dependiendo de este lo puede bloquear o permitirle el acceso Un firewall puede ser un dispositivo de tipo Hardware o software que de instala entre la conexión a Internet y las redes conectadas en el lugar, como podemos ver en la Figura 1.1.

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 ver 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, como podemos ver en la Figura 1.2.

Firewall GNU/Linux.

En GNU/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. Esta herramientas son: Ipchains: Esta herramienta ya quedo en el olvido ya que se usaba para kernel 2.4. Iptables: Esta herramienta es la que se esta ocupando actualmente y apareció a partir del kernel 2.4 y 2.6 en adelante. Con Iptables crea las reglas mas rápidas y sencillas que ipchains. Shorewall: Es una herramienta muy flexible, rápida y sencilla que permite crear reglas iptables, en shorewall se configuran varios archivos para poder controlar el firewall de nuestra red. ufw: Esta es un herramienta que nos permite crear reglas iptables de una forma demasiado sencilla dentro de distribuciones debian, ubuntu y derivados.

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 3tipos 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:

* **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 Publica del servidor.
* 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 lo uses 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 especifico, con estas marcas podremos limitar el ancho de banda y generar colas.

FILTER
Esta esta la tabla principal para el filtrado de paquetes que podemos comparar y filtar paquetes dentro del firewall. Dentro de esta tabla tenemos las siguientes opciones:

* **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 conexion 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.
* **UDP:** Protocolo de Datagrama de Usuario, sirve para el envía de datagrama pero debe existir una conexión establecida.
* **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, en resumen es un ping.

Para poder ocupar estos protocolos podremos ocupar el parámetro -p.

Objetivos

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 su nomenclatura ya que estos parámetros nos sirve para indicar alguna propiedad a nuestra regla creada dentro de firewall. Entonces revisemos los siguientes paramentas de iptables.

Con esto ya tenemos todas las opciones necesarias necesarias que podremos utilizar en iptables.

Nomenclatura Iptables

Ahora aprenderemos la nomenclatura.

iptables -A [parametros de la regla]

Comenzaremos a ver algunas reglas de iptables.

Ejemplo 1: Se aceptaran 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 de 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

Ahora veremos la configuración básica de un iptables, creando nuestra reglas y describiéndola para que sirve cada una.

```
## Limpiando reglas de iptables en todas las tablas.
iptables -F
iptables -X
iptables -Z

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

#Aceptamos todas la conexiones 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 al puerto 80/apache por la interfaz de red eth0.
iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT

#Rechaza todas la demas 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.

```
lucifer:~# iptables -nL
```

Firewall LAN

Ahora veremos como configurar un firewall del tipo LAN:

* Los clientes de la red local podrán acceder a internet pero solo a servicio de HTTP/HTTPS y

DNS.
* Desde internet se permitirá conectarse a servicios de HTTP/FTP que están dentro de la red local.

```
## 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 80 redirigirlo a la máquina de la red
local con IP 192.168.1.12:80.
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 80 -j DNAT --to 192.168.1.12:80

### Todas la peticiones que vengan de internet hacia el puerto 21 redirigirlo a la máquina de la red
local con IP 192.168.1.52:21.
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport 21 -j DNAT --to 192.168.1.52:21

### Aceptamos conexiones locales en la interfaz lo
iptables -A INPUT -i lo -j ACCEPT

### Tenemos acceso al firewall desde el segmento de red 192.168.1.0 por la interfaz eth1
iptables -A INPUT -s 192.168.1.0/24 -i eth1 -j ACCEPT

### Aceptamos que todo el trafico que viene desde la red local vaya hacia los puertos 80/443 sean aceptadas estas son solicitudes http/https
iptables -A FORWARD -s 192.168.1.0/24 -i eth1 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -s 192.168.1.0/24 -i eth1 -p tcp --dport 443 -j ACCEPT

### Aceptamos que consultas de DNS de la red local
iptables -A FORWARD -s 192.168.1.0/24 -i eth1 -p tcp --dport 53 -j ACCEPT
```

```
iptables -A FORWARD -s 192.168.1.0/24 -i eth1 -p udp --dport 53 -j ACCEPT

### Denegamos el resto de los servicios
iptables -A FORWARD -s 192.168.1.0/24 -i eth1 -j REJECT

### Ahora hacemos enmascaramiento de la red local
iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j MASQUERADE
echo 1 > /proc/sys/net/ipv4/ip_forward

#Rechaza todas la demas conexiones 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:

* Los clientes de la red local pueden conectarse a servicios del tipo APACHE y SAMBA en la DMZ,
* Desde internet se permitirá conectarse a servicios de APACHE que se encuentran en DMZ.

```
## 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.
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 192.168.1.0/24 -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 192.168.1.0/24 -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

### APACHE1 “Conexión del servicio desde la red local a DMZ”
iptables -A FORWARD -s 192.168.1.0/24 -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

### Samba “Conexión del servicio desde la red local a DMZ”
iptables -A FORWARD -s 192.168.1.0/24 -d 10.0.2.50 -p tcp –-dport 139 -j ACCEPT
iptables -A FORWARD -s 10.0.2.50 -d 192.168.1.0/24 -p tcp –-dport 139 -j ACCEPT

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 este capitulo solo mostraremos algunas otras reglas que han faltado explicar.
Habilitando varios puerto en una regla.
Dentro de iptables tenemos la capacidad de armar reglas para nuestro firewall con varios puerto de conexión al mismo. Ejemplo 1: Permitimos las conexión desde cualquier equipo de la red local al servidor en los puerto 22 y 80.

```
iptables -A INPUT -s 192.168.1.0/24 -p tcp --dport 22:80 -j ACCEPT
```

Ejemplo 2: Solamente permitiremos la conexion del cliente con la IP 192.168.1.50 a los puertos 20 y 21.

```
iptables -A INPUT -s 192.168.1.50 -p tcp --dport 20:21 -j ACCEPT
```

Proxy Transparente.

Esta regla es de mucha ayuda para los administradores no tener que ir cliente por cliente en la red a configurar sus navegadores web que con esta hacemos un redireccionamiento de puertos en el mismo servidor.

Toda peticiones que venga por la interfaz de red eth1 y con salida al puerto 80 redirecciona al puerto 3128.

```
iptables -t nat -A PREROUTING -i eth1 -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 192.168.1.0/0 -j DROP
iptables -A INPUT -p icmp -s 192.168.1.100 -j DROP
```

Ejemplo 2: Pero si quisiéramos bloquear completamente hacia cualquier interfaz entonces esta seria la regla.

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

Bloquear mac También es posible bloquear clientes, etc por la MAC Address de su máquina cliente.

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

VPN

Para un servicio como OpenVPN también es necesaria tener sus propias reglas de iptables para hacer la conexiones a los túneles. Aceptamos el trafico que entrada y salida por el protocolo UDP por el servicio OpenVPN.

```
iptables -A INPUT -i ppp0 -p udp --dport 1194 -j ACCEPT
iptables -A OUTPUT -o ppp0 -p udp --sport 1194 -j ACCEPT
```

En este caso la interfaz de escucha del servicio es ppp0 pero también puede ser eth0. Permitimos la conexión desde cualquier equipo por la interfaz tun.

```
[root@test ~]# iptables -A INPUT -i tun+ -j ACCEPT
[root@test ~]# iptables -A OUTPUT -o tun+ -j ACCEPT
```

Permitimos que los equipos de las otras redes accedaan a nuestra red..

```
[root@test ~]# iptables -A FORWARD -i tun+ -j ACCEPT
[root@test ~]# iptables -A FORWARD -o tun+ -j ACCEPT
```