Firewalling

Última actualización en 2020

Un cortafuegos es un sistema (hardware o software) usado para separar una subred protegida de otra red de riesgo (de la que no tenemos tenemos control) y establece políticas de control entre ambos entornos (controlamos que queremos que entre y salga de esa red).

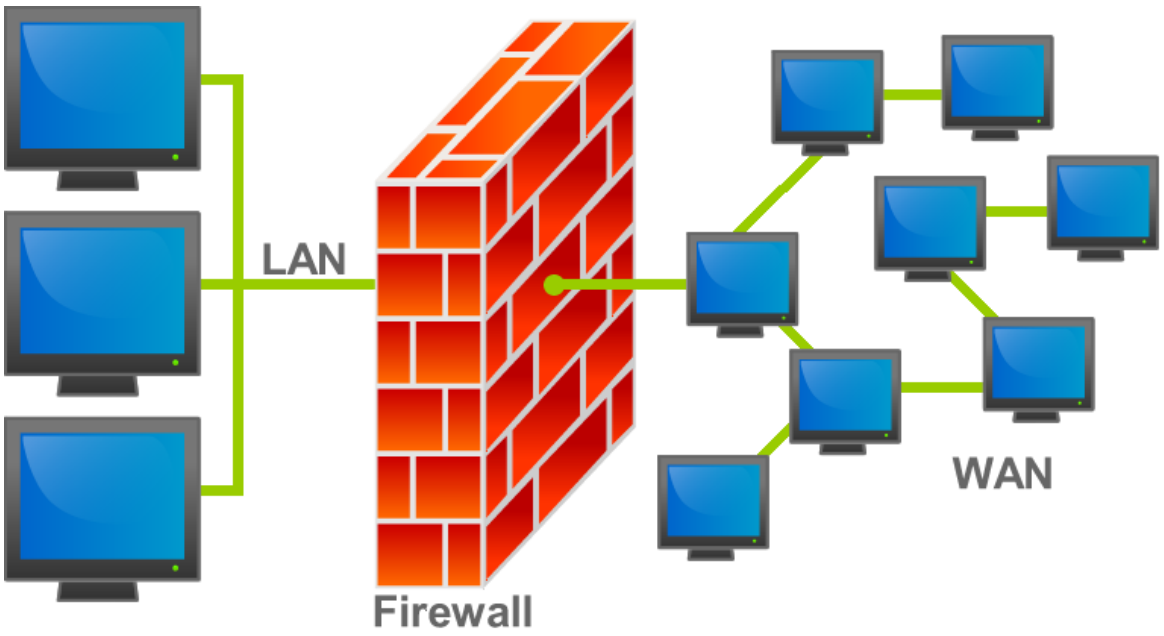


Imagen obtenida de: Wikipedia

El **filtrado de paquetes** es un proceso que deniega o permite el flujo de información y datos entre la red que se desea proteger (la red interna) del resto.

- Los firewalls trabajan sobre las **cabeceras de los paquetes IP**.
- Estableciendo una serie de reglas se podrán ejecutar acciones de aceptación o rechazo sobre los paquetes.
- Existen varios tipos de filtrado de paquetes:
    - **Estático (*stateless*)**: analiza las cabeceras de cada paquete sin establecer relación con otros.
    - **Dinámico (*stateful*)**: permite control de un flujo de datos dentro de la misma conexión TCP o varias conexiones haciendo uso de la memoria.

1. NAT (Network Address Translation)

Mecanismo que altera las cabeceras de los paquetes IP pudiendo cambiar las direcciones y puertos origen o destino.

- Source NAT (SNAT): se produce cuando alternamos la **dirección de origen** del paquete. Estamos cambiando el sitio del cual viene la conexión (la dirección del usuario que envía los datos) justo **después del encaminamiento**, antes de que el paquete salga por el cable o por el aire (en conexiones inalámbricas).
- Destination NAT (DNAT): se produce cuando se alterna la dirección de destino del paquete. El DNAT siempre se hace **después del encaminamiento**, cuando el paquete entra por el cable. El *port forwarding*, balanceo de carga o proxy transparente son formas de DNAT.

2. Netfilter

Netfilter es un *framework* de Linux que permite interceptar y manipular paquetes de red. Iptables es su componente más popular (es una interfaz de alto nivel) y actúa como cortafuegos en capas 3 y 4 (red y transporte). Además, iptables es la herramienta más utilizada para implementar *firewalls*.

2.1. Reglas

Con **iptables** se pueden establecer una serie de reglas para establecer qué es lo que hay que hacer con un paquete. Para cada regla que se cree **hay que tener claras dos cosas**:

- La **condición** para aplicar esa regla (p. ej: que un paquete vaya al puerto 80).
- La **acción** a realizar en caso de que un paquete coincida con la condición (p. ej: para la condición de paquetes que van al puerto 80 la acción es aceptar). Las acciones más importantes son las siguientes:
    - **ACCEPT** (aceptar).
    - **DROP** (no hace nada con el paquete, lo rechaza pero para el emisor parece más bien que no hay nada ahí).
    - **REJECT** (rechaza el paquete).
    - **REDIRECT**: redirección de puertos (solo en la misma máquina).
    - **DNAT**: cambia la dirección de destino del paquete.
    - **SNAT**: cambia la dirección de origen del paquete.

2.2. Cadenas (chains)

Una **cadena** es una **lista ordenada de reglas**. Las cadenas existentes en iptables son las siguientes:

- **INPUT (entrada)**: reglas que se realizarán cuando el paquete entre a la propia máquina.
- **OUTPUT (salida)**: reglas que se ejecutarán cuando se trate de paquetes que salgan de la máquina hacia fuera.
- **FORWARD (reenvío)**: cuando el paquete se envía de una interfaz a otra (podéis imaginaros un *router* con dos interfaces y de una interfaz de entrada tienes que mandarlo por la de salida.
- **PREROUTING (pre-enrutamiento)**: primera acción a realizar cuando entra el paquete al sistema.

- **POSTROUTING (post-enrutamiento)**: acción a ejecutar antes de sacar el paquete por otra interfaz.
- **Cadenas personalizadas.**

En caso de llegada de un paquete que no coincide con ninguna de las reglas definidas para una cadena **se aplicará la política por defecto para esa cadena**. Esta puede ser: aceptar todo o rechazar todo.

2.2.1. Ejemplo

Si decido rechazar (REJECT) los paquetes de entrada (cadena INPUT) al puerto 53, 80 y 8080 (en este orden) y llega un paquete al *firewall* con el puerto de destino 22 mirará, por orden, cada una de esas tres reglas y aplicará la primera que coincida:

1. ¿El paquete va al puerto 53? NO
2. ¿El paquete va al puerto 80? NO
3. ¿El paquete va al puerto 8080? NO
4. Aplico la política por defecto (rechazar todo o aceptar todo lo que entre).

2.3. Filtrado de paquetes

Con Netfilter podemos hacer lo siguiente:

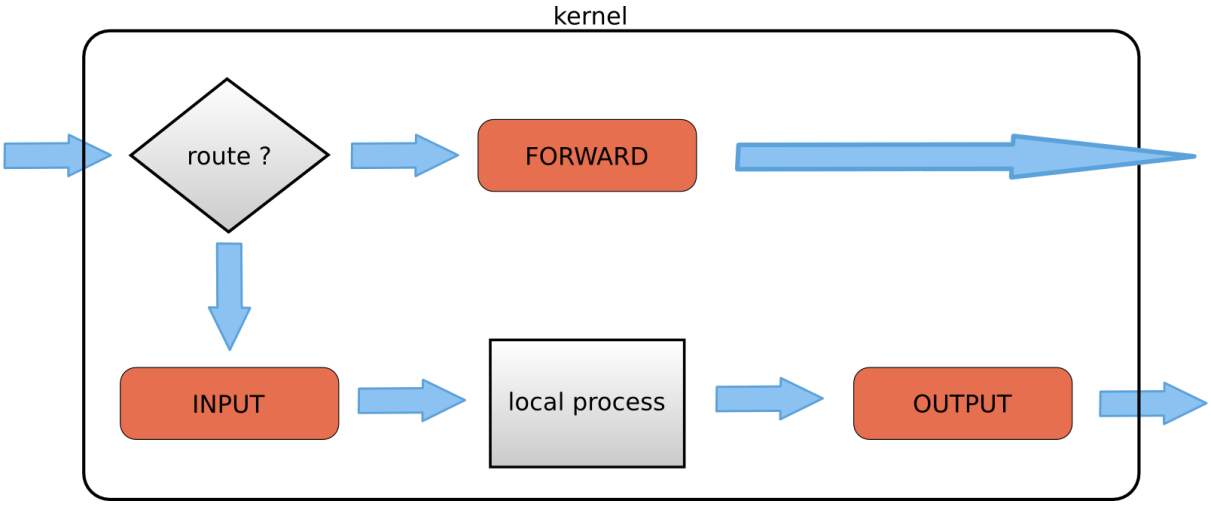
- Filtrar paquetes.
- Traducir direcciones y puertos origen y destino (NAT).
- Manipular sobre paquetes IP.
- Seguimiento de conexiones.

Funciona tanto con **IPv4** como con **IPv6.**

2.4. Tablas

Netfilter tiene las siguientes tablas:

- **FILTER**: filtrado de paquetes (decide qué paquetes pasan y cuáles no). Solo la usamos con las *chains* INPUT, OUTPUT y FORWARD.
- **NAT**: para traducción de direcciones. Permite cambiar direcciones origen y destino de los paquetes por otras. Se usa en las *chains* de *prerouting, postrouting* y *output*.
- **MANGLE**: analiza el paquete y lo etiqueta para que reciba un tratamiento concreto. Usa las *chains prerouting, postrouting, input, forward y output*. Puedes pensar en MANGLE como una serie de casos que pueden ocurrir (imagínate 20 casos similares), puedes etiquetar todos los casos comunes con una etiqueta y hacer cosas en relación a esa etiqueta. Es la tabla menos utilizada.



3. Uso de iptables

3.0.1. Sintaxis

iptables [-t tabla] COMANDO CADENA condición acción [opciones adicionales]

Los corchetes indican que se puede o no poner (no se pone siempre).

1. Comando "iptables".
2. Selecciono tabla: -t INPUT, -t FORWARD, -t OUTPUT, etc.

3.0.2. Ejemplo

Rechazar todos los paquetes que lleguen al *firewall* (que nos entren, INPUT) del origen (source, -s) de dirección 134.12.32.10.

Tabla filter, entrada (input), condición (que vengan de 134.12.32.10) acción (-j) rechazar (reject).

iptables -t **filter** -A **INPUT** -s 134.12.32.10 -j **REJECT**

- -t -> tabla filter (tabla de filtrado de tráfico)
- -A -> Añadir regla a la lista (en este caso a la lista de entrada, INPUT)
- -s -> significa source (Fuente, origen del tráfico).
- -j -> acción a realizar (aceptar o rechazar la conexión. accept para aceptar y reject o drop para rechazar).

3.1. Uso de iptables

A continuación se muestra como leer, añadir, actualizar, eliminar reglas con ***iptables***.

3.1.1. Listado de reglas

A continuación se irán listando y explicando los principales parámetros de iptables. Se aconseja ir probandolos en una máquina virtual.

Para listar las reglas que hay insertadas actualmente se usa el **parámetro -L**:

```
iptables [-t tabla] -L [CHAIN -v] [--line-numbers]
```

Por ejemplo, para mirar la lista de reglas de la tabla **FILTER** y la cadena **INPUT** podemos hacer lo siguiente:

```
iptables -t filter -L INPUT
```

Por defecto (si no se pone información sobre la tabla) **ya se utiliza filter**, por tanto cuando queremos hacer algo sobre esa tabla no es necesario indicarlo. El comando anterior podría quedar de la siguiente manera:

```
iptables -L INPUT
```

3.1.1.1. Añadido de reglas

Para añadir una nueva regla al final de la lista se puede utilizar el **parámetro -A**:

```
iptables -A CADENA.............................
```

Para añadir la regla en una posición concreta se utilizaría **el parámetro -I** (la vocal i en mayúscula):

```
iptables -I <numero> CADENA.....................
```

3.1.2. Inserción de reglas

iptables -F

iptables -F -t nat

3.1.3. Eliminación de reglas

Para eliminar TODAS las reglas de una cadena concreta usamos **el parámetro -F**:

```
iptables -F CADENA
```

Hay varias maneras de eliminar reglas. Si sabes como es la regla concreta solo tienes que repetirla y cambiar el **parámetro** de añadido (-A o -I) por **-D:**

```
iptables [-t tabla] -D INPUT ....................................
```

Si se quiere eliminar una regla en una posición concreta:

```
iptables -D CADENA <numero>.....................
```

El borrado de reglas por posición o línea debería hacerse desde abajo hacia arriba. Esto es, eliminar primero las últimas. Si eliminamos una regla la posición de todas las reglas posteriores se reducirá en uno y esto **puede dar lugar a eliminar reglas no deseadas**.

3.1.4. Reemplazo de reglas

Para reemplazar una regla en una posición por otra nueva se usa el **parámero -R**:

```
iptables -R CHAIN <posicion> <nueva_regla>
```

3.1.5. Políticas por defecto

Se puede establecer una acción por defecto a realizar cada vez que llegue un paquete. Es decir, si no hay una regla definida sobre qué hacer con ese paquete que nos llega se utiliza la política por defecto. Esta política puede ser:

- ACCEPT: acepta el paquete.
- REJECT: rechaza el paquete.
- DROP: hace como si el pc no existiera (no deja pasar el paquete, similar a rechazarlo).

Por ejemplo, para poner una política por defecto en que se cierre absolutamente todo el tráfico de entrada utilizaríamos **REJECT** o **DROP**.

```
iptables -P INPUT DROP
```

Cuando establecemos el *reject* o *drop* sobre todo el tráfico decimos que utilizamos una política de "**lista blanca**". Cerramos todo y añadimos reglas a iptables para permitir acceso solo a lo que queramos.

Cuando establecemos el *accept* por defecto sobre todo el tráfico decimos que utilizamos una política de "**lista negra**". Aceptamos, por defecto, todo el tráfico y añadimos reglas solo para el que queremos bloquear.

3.1.5.1. Ejercicio

Escribe esta regla en una máquina virtual e intenta acceder a www.google.com . Razona lo que ocurre (por qué te funciona el acceso o por qué no no lo hace).

3.1.6. Seguimiento de conexiones (Connection Tracking)

Cuando establecemos reglas que bloquean conexiones entrantes (cadena INPUT) podemos comprobar que nosotros mismos nos estamos bloqueando. La razón de esto no tiene nada que ver con el tráfico de salida (cadena OUTPUT), solamente con el de entrada.

Al bloquear el tráfico de entrada, cuando salimos a internet y la respuesta vuelve a nosotros ese tráfico es bloqueado (la petición llega pero no vuelve). Para esto iptables provee un módulo de *connection tracking* llamado **conntrack**: si una conexión ya ha sido establecida no la bloquea. Los estados en los que puede estar este módulo son los siguientes:

- **NEW:** primer paquete de la conexión.
- **ESTABLISHED:** se usa para paquetes que son parte de una conexión ya existente. Para que una conexión esté en este estado tiene que haber recibido una respuesta de otro host.
- **RELATED**: se usa para conexiones que están relacionadas con una conexión ya establecida (established). Por ejemplo: tráfico FTP.
- **INVALID:** el paquete no tiene un estado válido.
- **UNTRACKED**: cuando no hay seguimiento sobre el paquete.

Ejercicio: Para cargar el módulo de **conntrack** se utiliza la opción **-m**. Búscala en el manual.

Como ejemplo, solucionaremos el problema que teníamos al bloquear todos los paquetes de INPUT en la política por defecto (cuando nos conectamos a una web nuestra petición llega pero la respuesta no):

```
iptables -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT
```

3.2. Reglas de reenvío (FORWARD) y traducción (NAT)

3.2.1. Reenvío de paquetes

Para que funcionen las reglas de redirección es necesario activar el redireccionamiento IP en la máquina. Para ver si está activado puedes escribir el siguiente comando:

```
cat /proc/sys/net/ipv4/ip_forward
```

Si el fichero contiene un 0 el reenvío IP está desactivado. Para activarlo hay que poner un 1 en el fichero:

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

3.2.2. Ejemplo de reenvío y traducción de peticiones web

Si deseamos reenviar tráfico desde un ordenador origen a otro de destino podemos usar la cadena **FORWARD**. Por ejemplo, para redirigir todo el tráfico por la interfaz *eth1* que llega al *firewall* hacia otro ordenador:

```
iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth1 -o eth0 -p tcp --dport 80 -j ACCEPT
```

En la regla anterior estamos redirigiendo el tráfico a otra red (por otra interfaz). Para hacer esto, además, necesitamos hacer NAT:

Cuando llega la petición web: Tenemos que cambiar la dirección de destino a la del ordenador de destino.

```
iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 80 -j DNAT
--to-destination 192.168.1.125:80
```

Cuando nos llega la respuesta: viene de nuestra red. Hay que cambiar la dirección de origen por la de la interfaz de salida.

```
iptables -t nat -A POSTROUTING -o eth1 -p tcp --dport 80 -d
192.168.1.125 -j SNAT --to-source 192.168.1.1
```

Ejercicio: las reglas establecidas de forward redirigen todo el tráfico entrante y saliente al puerto 80, pero las reglas deberían ser un poco más restrictivas. Intenta mejorar los comandos de FORWARD para hacer ***connection tracking*** y permitir que lleguen desde el exterior solamente las conexiones nuevas (NEW) y de vuelta lo ya establecido (RELATED,ESTABLISHED).

3.2.3. Parámetros

Algunos ejemplos de parámetros útiles para la inserción de reglas son los siguientes:

- **--destination-port o --dport:** puerto de destino.

- **--source-port o --sport:** puerto de origen.
- **-p protocolo**: protocolo a usar, p. ej: -p tcp, -p udp (mirar manual). Siempre que se utilicen puertos hay que indicar también el protocolo.
- **-m o --match**: proporciona una extensión con más funcionalidades de iptables. Algunos módulos disponibles son:
    - **conntrack**: permite establecer reglas para conexiones ya establecidas o nuevas.
    - **iprange:** permite establecer reglas con rangos de IPs. Por ejemplo: … *-m iprange --src-range 192.168.1.1-192.168.1.254 -j …*
    - **multiport**: permite establecer en una sola regla varios puertos (o rangos de puertos:
        - … --match multiport --dports 80,443 -j …
        - … --match multiport --dports 20:60 -j …
- **-i**: para una interfaz (*interface*) determinada, p. ej: -i eth0.
- **-s:** *source*, fuente origen del paquete (una ip origen).
- **-j**: acción a realizar en la regla (aceptar, rechazar, enmascarar, etc.).

4. Guardado de reglas

Las reglas que se ejecutan en consola con **iptables** no se almacenan de forma permanente. Para almacenar las reglas que has ejecutado de manera permanente puedes consultar **este enlace**.

6. Referencias

IPTables. Openwebinars: hacking ético: https://openwebinars.net/

IPTables. https://www.linuxito.com/seguridad/793-tutorial-basico-de-iptables-en-linux

Cortafuegos. https://es.wikipedia.org/wiki/Cortafuegos_(inform%C3%A1tica)

NAT. https://netfilter.org/documentation/HOWTO/es/NAT-HOWTO-3.html

IPTables: https://www.booleanworld.com/depth-guide-iptables-linux-firewall/#Targets

IPTables: https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands

Tabla nat: https://www.karlrupp.net/en/computer/nat_tutorial

Nat/Forward: https://www.digitalocean.com/community/tutorials/how-to-forward-ports-through-a-linux-gateway-with-iptables

Documentación creada por: Marcos Núñez Celeiro

Realizada para: segundo curso de S.M.R