# Administración de un cortafuegos con iptables

## 1. Políticas

### a. Fundamentos de las políticas de un cortafuegos

Un cortafuegos puede funcionar según dos modelos distintos: «todo lo que no está autorizado está prohibido» o «todo lo que no está prohibido está autorizado». Para establecer el comportamiento por defecto, iptables permite definir para cada cadena una acción por defecto.

*Definición de la política por defecto de iptables*

```
iptables -P cadena acción
```

Donde *cadena* representa el tipo de tráfico (INPUT, OUTPUT y FORWARD), y *acción* el comportamiento deseado (DROP o ACCEPT).

*Ejemplo de definición de política*

*En este ejemplo, se prohíbe todo el tráfico saliente del host aplicando una política de descarte de paquetes salientes.*

```
root@test:~$ ping -c 1 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
64 bytes from 192.168.0.10: icmp_seq=1 ttl=64 time=0.880 ms
--- 192.168.0.10 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.880/0.880/0.880/0.000 ms
root@test:~$ iptables -P OUTPUT DROP
root@test:~$ ping -c 1 192.168.0.10
PING 192.168.0.10 (192.168.0.10) 56(84) bytes of data.
ping: sendmsg: Operation not permitted
--- 192.168.0.10 ping statistics ---
1 packets transmitted, 0 received, 100% packet loss, time 0ms
root@test:~$
```

### b. Configuración de una política básica

Si el host que se desea configurar se sabe que se convertirá en un cortafuegos, es probable que todo el tráfico esté prohibido por defecto. Esta configuración común consiste en establecer para las tres cadenas, INPUT, OUTPUT y FORWARD, una política de descarte de paquetes.

*Configuración de una política restrictiva*

```
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

## 2. Filtrado de paquetes

### a. Política y reglas

Después de haber configurado una política que describe el comportamiento básico del cortafuegos, hay que crear las reglas específicas para el tráfico que se desea dejar pasar o prohibir. La filosofía del cortafuegos es: se define el comportamiento general con las políticas y se gestiona caso por caso el comportamiento específico con reglas.

## b. Creación de reglas

Para cada elemento de tráfico que debe estar permitido o prohibido, habrá que crear una regla específica.

*Sintaxis de creación de una regla de gestión de tráfico*

```
iptables -A cadena -s ip_origen -d ip_destino -p protocolo --dport puerto
-j acción
```

| iptables: creación de reglas | |
|---|---|
| -A *cadena* | Se añade una regla en la cadena *cadena* (INPUT, OUTPUT o FORWARD). |
| -s *ip_origen* | Opcional: la dirección IP origen de donde provienen los paquetes sometidos a la regla. Si la dirección es una dirección de red, hay que especificar la máscara. |
| -d *ip_destino* | Opcional: la dirección IP destino a la que van los paquetes sometidos a la regla. Si la dirección es una dirección de red, hay que especificar la máscara. |
| -p *protocolo* | Indica el protocolo utilizado en el paquete sometido a la regla. Valores comunes: udp, tcp, icmp. |
| --dport *puerto* | Opcional: indica el puerto de destino del paquete sometido a la regla. |
| -j *acción* | Indica cómo tratar el paquete sometido a la regla (ACCEPT o DROP). |

*Autorización de ping salientes y entrantes*

*Cada tipo de flujo tiene que ser el objetivo de una regla de iptables.*

```
alfa:~# iptables -A OUTPUT -p icmp -j ACCEPT
alfa:~# iptables -A INPUT -p icmp -j ACCEPT
alfa:~#
```

*Autorización del tráfico http que pase por la máquina con origen una red determinada*

```
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p tcp -dport 80 -j ACCEPT
alfa:~#
```

Una configuración errónea en un cortafuegos puede tener consecuencias drásticas. Para comprobar que la configuración está realizada correctamente, se recomienda utilizar un escáner de puertos desde una máquina remota. El comando nmap -F seguido de la dirección IP de la máquina protegida permite comprobar muy rápido (Fastmode) que los puertos están adecuadamente bloqueados o abiertos.

## c. Gestión de reglas

Las reglas se aplican en su orden de creación y el sistema les asigna automáticamente un número de orden.

*Visualización de los números de reglas efectivas*

```
iptables -L cadena --line-numbers -n
```

Donde *cadena* representa la cadena de tratamiento (INPUT, OUTPUT o FORWARD). El parámetro -n no es obligatorio, pero acelera notablemente la visualización del resultado, ya que no fuerza a que el comando tenga que resolver las direcciones en nombres.

*Eliminación de una regla*

```
iptables -D cadena número
```

Donde *número* representa el número de la línea obtenido con el comando anterior y *cadena* representa la cadena de tratamiento (INPUT, OUTPUT o FORWARD).

*Inserción de una regla*

```
iptables -I cadena número condiciones -j acción
```

Donde *condiciones* representa los criterios de selección del paquete sometido a la regla (direcciones IP, puertos y protocolos).

*Ejemplo de gestión de reglas*

*La gestión dinámica de reglas es tan pesada que su uso se ha establecido en un archivo de script que incluye todas las reglas y se recarga por completo después de cada cambio.*

```
alfa:~# iptables -L FORWARD --line-numbers -n
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:23
2    ACCEPT     udp  --  192.168.1.0/24       0.0.0.0/0           udp dpt:53
3    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80
alfa:~# iptables -D FORWARD 1
alfa:~# iptables -L FORWARD --line-numbers -n
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     udp  --  192.168.1.0/24       0.0.0.0/0           udp dpt:53
2    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80
alfa:~# iptables -I FORWARD 1 -s 192.168.1.0/24 -p tcp --dport 22 -j ACCEPT
alfa:~# iptables -L FORWARD --line-numbers -n
Chain FORWARD (policy DROP)
num  target     prot opt source               destination
1    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:22
2    ACCEPT     udp  --  192.168.1.0/24       0.0.0.0/0           udp dpt:53
3    ACCEPT     tcp  --  192.168.1.0/24       0.0.0.0/0           tcp dpt:80
alfa:~#
```

### d. Gestión de flujos de retorno

En la mayoría de las aplicaciones de red, un host envía un paquete con destino otro host que le responde. Por lo tanto, se establece una comunicación bidireccional. Ahora bien, en la configuración de un cortafuegos, se visualiza perfectamente la comunicación de ida: por ejemplo, desde un navegador a un servidor web a través del puerto 80; en cambio, no se ve tan bien las respuestas que se realizan por un puerto aleatorio, por iniciativa del cliente, mayor que 1024.

En los inicios de los cortafuegos, la solución consistía en autorizar cualquier tráfico entrante cuyo puerto era superior al 1024. Los cortafuegos tenían entonces más tendencia a impedir a los usuarios salir antes que evitar las intrusiones en la red.

Pasados unos años, los cortafuegos llamados «stateful» (con estado) son capaces de autorizar dinámicamente los flujos de retorno siempre que sean respuesta a un flujo de salida explícitamente autorizado.

*Autorización implícita de flujos de retorno*

```
iptables -A cadena -m state --state ESTABLISHED,RELATED -j ACCEPT
```

La opción -m state permite realizar un filtro en función del estado del paquete tratado. Los estados aceptados son ESTABLISHED y RELATED y representan respectivamente paquetes en respuesta a un flujo autorizado y paquetes enviados para una nueva conexión, pero con la iniciativa de una conexión establecida y autorizada (por ejemplo, el tráfico de datos ftp relativo al tráfico de comandos ftp).

*Ejemplo de configuración completa de un cortafuegos*

*Se configura en este caso el cortafuegos para que no deje pasar nada, a excepción de las respuestas a cada una de las comunicaciones establecidas, así como los protocolos necesarios para navegar por Internet (http, https y dns).*

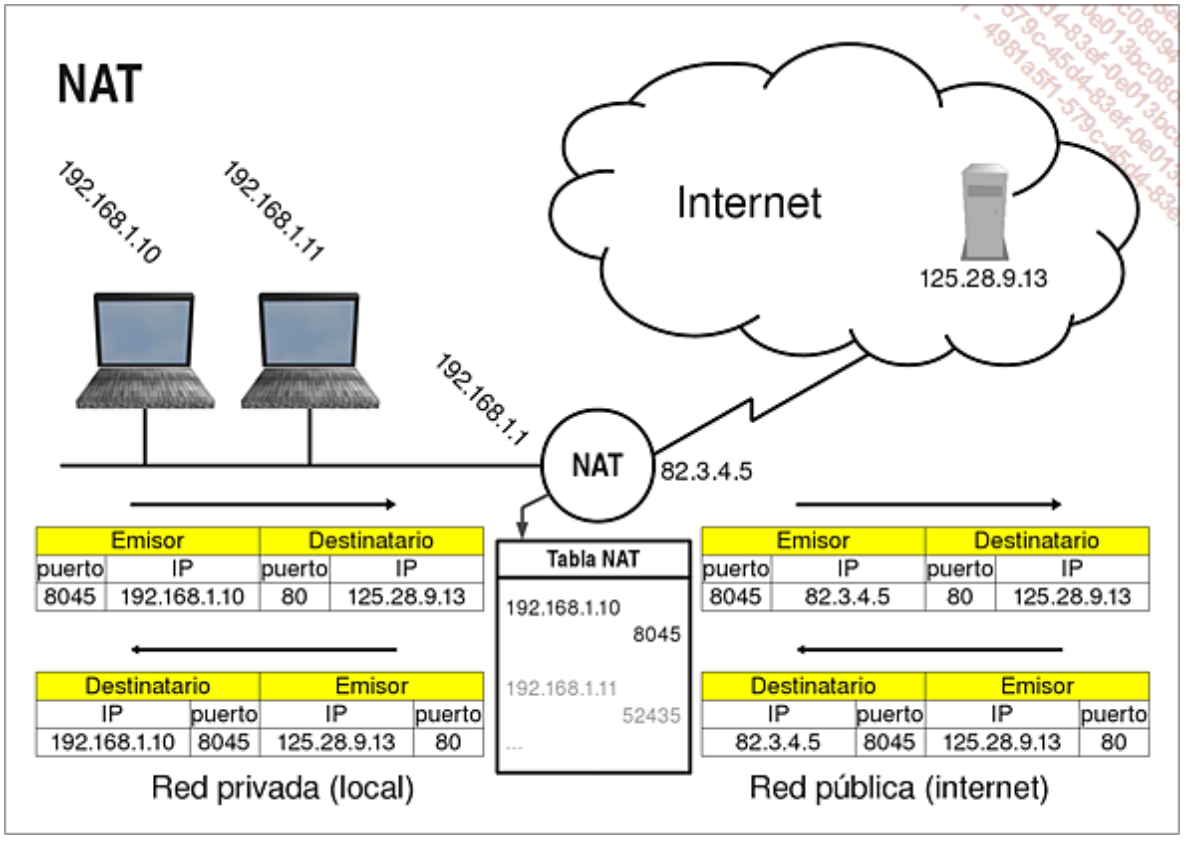
```
alfa:~# iptables -P INPUT DROP
alfa:~# iptables -P OUTPUT DROP
alfa:~# iptables -P FORWARD DROP
alfa:~# iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p tcp --dport 80 -j ACCEPT
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p tcp --dport 443 -j ACCEPT
alfa:~# iptables -A FORWARD -s 192.168.1.0/24 -p udp --dport 53 -j ACCEPT
```

En este ejemplo, se configura el cortafuegos para que no deje pasar nada excepto las respuestas a los tráficos establecidos, así como los protocolos necesarios para navegar por Internet (http, https y dns).

La aplicación **fail2ban**, en el caso de que se generen muchos intentos fallidos de conexión a aplicaciones o al propio sistema, permite crear dinámicamente una regla que bloqueará cualquier comunicación del atacante. El conocimiento detallado de su configuración no se exige para la certificación LPI.

## 3. Gestión de NAT

### a. Recordatorio del principio de NAT




NAT consiste en reescribir la cabecera IP de un paquete que viaja de una red pública a una red privada y viceversa.

Las direcciones IP privadas no se pueden enrutar por Internet. Un paquete proveniente de una dirección privada no podría encontrar una ruta de retorno, porque ningún router aceptaría devolverlo a su origen. De todos modos, las redes privadas se usan en todas partes (hay millones de redes 192.168.1.0), sería imposible mantener en las tablas de enrutamiento de los routers de Internet una ruta coherente con la red de origen.

La solución para salir de una red privada consiste en reemplazar la dirección IP privada del emisor por la dirección IP pública (único tipo de dirección en Internet) del router realizando NAT. La trazabilidad de las traducciones (reemplazo de direcciones IP privadas) se realiza

gracias al puerto emisor utilizado: para cada traducción realizada, el router se guarda en memoria el puerto emisor empleado. Como el paquete de respuesta llega al router con la dirección pública de este y al mismo puerto que usó en la emisión, la dirección original del cliente se averigua fácilmente por parte del router NAT.

### b. Diagnóstico de la configuración NAT de un router

NAT se gestiona en una tabla específica llamada **NAT**. Cualquier configuración vinculada con NAT se realiza con el comando **iptables** especificando que se trabaja en la tabla NAT. Las cadenas que se tratan en la tabla NAT son **PREROUTING**, **POSTROUTING** y **OUTPUT**, que representan el tráfico que hay que modificar antes del enrutamiento, después del enrutamiento o directamente en la salida de la máquina.

*Visualización de la configuración NAT*

```
iptables -t nat -L
iptables -t nat -S
```

### c. Conexión de una red privada a una red pública

En esta configuración, que también es la más corriente, la dirección IP del emisor de los hosts de la red privada se reemplaza por la dirección pública del router NAT.

*Configuración de NAT*

```
iptables -t nat -A POSTROUTING -o tarjeta_exterior -j acción_nat
```

| NAT con iptables: opciones y parámetros | |
|---|---|
| -t nat | La regla afecta a la tabla NAT. |
| -A POSTROUTING | Se añade una regla a la cadena POSTROUTING, para el procesado después del enrutamiento. |
| -o *tarjeta_exterior* | Identifica la tarjeta de red por la cual salen los paquetes del cortafuegos. |
| -j *acción_nat* | Identifica el modo de acción de NAT, soporta dos opciones: SNAT si la dirección pública es fija y MASQUERADE si la dirección pública es dinámica. |

*Ejemplo de configuración de NAT*

```
alfa:~# iptables -t nat -A POSTROUTING -o eth1 -j MASQUERADE
alfa:~#
```

En este ejemplo, eth1 es la interfaz conectada a la red pública.

## 4. Scripts de configuración de reglas de filtrado

### a. Red Hat e iptables

Los sistemas Red Hat y sus derivados ofrecen un servicio iptables que permite aplicar una configuración de filtrado o NAT automáticamente. El arranque del servicio aplica la configuración y su parada anula todos los filtros. Este funcionamiento es extremadamente práctico y permite gestionar un cortafuegos RedHat cómodamente.

### b. Creación de servicios personalizados de cortafuegos con iptables

Se comprueba bastante rápido que la creación de reglas de filtrado y de NAT con iptables tiene aspectos pesados. Por consiguiente, después de haber determinado qué reglas se necesitan, es interesante escribirlas en un script.

*Ejemplo de script de configuración de cortafuegos*

*Este tipo de script no fuerza a que se tengan que gestionar las reglas una por una en el caso de que se modifique la configuración. Es mucho más fácil insertar una línea en el script que desplazar la numeración de las reglas en memoria. Sin embargo, hay que anular cualquier regla antes de cada ejecución del script.*

```
#!/bin/bash
# nombre del archivo: /etc/cortafuegos_on
# Política básica
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
# NAT con eth0 como interna y eth1 como externa - dirección IP pública fija
iptables -t nat -A POSTROUTING  -o eth1 -j SNAT --to-source 81.2.3.4
# gestión de paquetes devueltos
iptables -A FORWARD -i eth1 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
# tráfico saliente autorizado
iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 80 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -p tcp --dport 443 -j ACCEPT
iptables -A FORWARD -i eth0 -o eth1 -p udp --dport 53 -j ACCEPT
```

Por supuesto, no hay que olvidar convertirlo en ejecutable.

También será útil crear un script de anulación de todas las reglas de filtrado. En efecto, puede ser útil autorizar más o menos provisionalmente todo el tráfico para una actualización del cortafuegos o para usar una aplicación puntual.

*Ejemplo de script de anulación de filtrado*

```
#!/bin/bash
# nombre del archivo: cortafuegos_off
# Borrado de reglas
iptables -F
# Política permisiva
iptables -P INPUT ACCEPT
iptables -P OUTPUT ACCEPT
iptables -P FORWARD ACCEPT
```

Finalmente, se puede crear un script de gestión de servicio estándar.

*Ejemplo de script de servicio del cortafuegos*

*Naturalmente, hay que poner este script en el directorio /etc/init.d.*

```
#!/bin/bash
# nombre del archivo: cortafuegos
case $1 in
start)
  /etc/cortafuegos_on
  ;;
stop)
  /etc/cortafuegos_off
  ;;
status)
```

```
  iptables -L
  ;;
*)
  echo "Syntaxis: /etc/init.d/cortafuegos start|stop|status
  ;;
esac
```