JUNTA DE ANDALUCIA
CONSEJERIA DE EDUCACION
CEP Lora del Río

Cortafuegos en GNU/Linux con iptables

Alberto Molina Coballes, José Domingo Muñoz Rodríguez y José Luis Rodríguez Rodríguez.

15 de abril de 2010

En este documento se describe la configuración de un cortafuegos y un dispositivo de NAT con iptables en GNU/Linux. En concreto se utilizarán las tablas filter y nat de esta aplicación del proyecto netfilter. Este documento forma parte del curso *Servicios en GNU/Linux. Portal Educativo*, organizado por el CEP de Lora del Río (Sevilla) en 2010.

Índice

1. Introducción

Un cortafuegos es un componente de un sistema o una red que tiene dos funciones principales:

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

Los cortafuegos pueden implementarse sobre dispositivos de hardware específicos, que se denominan cortafuegos hardware; o bien sobre dispositivos genéricos mediante software, obviamente denominados cortafuegos software. En este documento se verá la forma de implementar un cortafuegos software en GNU/Linux con la aplicación *iptables* del proyecto http://netfilter.org.

2. Configuraciones de red

Los sistemas operativos de escritorio suelen incorporar software de cortafuegos que es necesario configurar en cada uno de los equipos. En una red local gestionada de forma adecuada, en lugar de configurar el software de cortafuegos puesto por puesto, se configura de forma precisa un solo cortafuegos que proteja toda la red local del exterior.

Dado que el cortafuegos debe analizar todo el tráfico entre la red local y el exterior en los dos sentidos, lo más habitual es que sea el dispositivo que conecta la red local con el exterior como en el esquema de la figura 1.

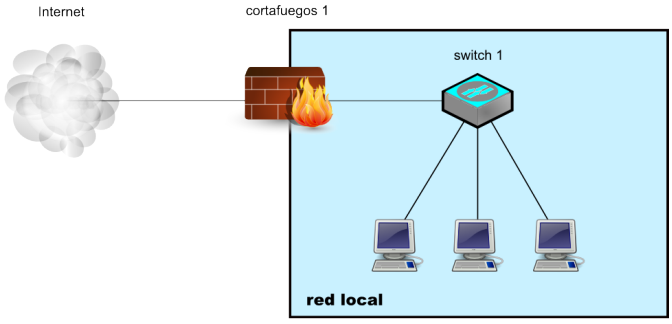

Figura 1: Cortafuegos simple que se interpone entre la red local y el exterior.

El esquema anterior es perfectamente válido cuando no es necesario acceder a nuestros servidores desde el exterior, pero cuando tenemos servidores que ofrecen servicios a Internet es muy recomendable colocar estos es un segmento de red independiente de la red local, ya que existe riesgo de que estos equipos sean atacados desde el exterior y debemos impedir que ese potencial ataque se pueda propagar por el resto de la red local.

La forma tradicional de implementar un segmento de red específico para los servidores es colocar los servidores delante del cortafuegos de la red local y a su vez instalar un nuevo cortafuegos que proteja el segmento de red de los servidores como se puede ver en la figura 2. De esa forma la zona de servidores es accesible tanto desde la red local como desde Internet, pero la red local está protegida de los accesos desde Internet. Alguien consideró que este esquema era muy similar a lo que se conoce en términos *militares* como una zona desmilitarizada o DMZ por sus siglas en inglés, por lo que se suele denominar así al segmento de red en el que se encuentran los servidores que ofrecen servicios a Internet.

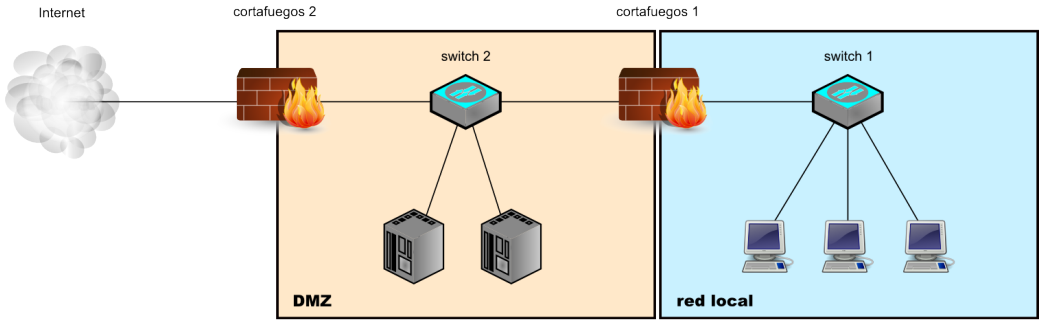

Figura 2: Cortafuegos clásico con DMZ.

Una forma habitual de implementar una DMZ sin necesidad de utilizar dos cortafuegos es el esquema que utiliza un cortafuegos de "3 patas", para lo que es necesario que el dispositivo que actúa como cortafuegos disponga de tres interfaces de red conectada cada una de ellas a uno de los segmentos de red, tal como se puede ver en la figura 3.

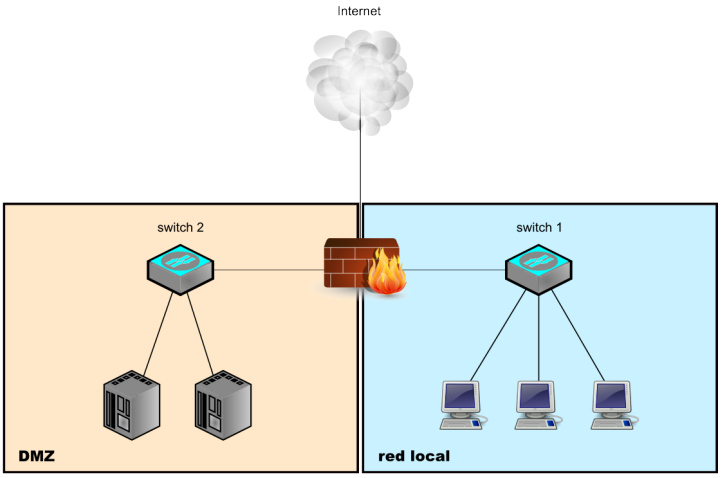

Figura 3: Cortafuegos de "3 patas" que separa la red local de la DMZ e Internet.

Como en este curso el esquema de red utilizado desde el principio se corresponde con el de la figura 1, será el que utilicemos en los ejemplos de este documento (incluso pondremos reglas de acceso a servidores dentro de la red local), pero eso se hace exclusivamente por no complicar el esquema de red del curso. En una implementación más realista recomendamos siempre ubicar los servidores accesibles desde Internet en una DMZ.

3. Tipos de cortafuegos

Además del esquema de red, también hay que detallar el tipo de cortafuegos que vamos a utilizar, ya que existen diferentes tipos de cortafuegos en función del análisis del tráfico que hagan, siendo los más significativos:

Cortafuegos de filtrado de paquetes Examinan los paquetes en diferentes niveles de la pila TCP/IP. Como mínimo a nivel de red en el que examinan parámetros como la dirección IP origen o destino, aunque también es posible que estos cortafuegos trabajen a nivel de enlace o nivel de transporte.

Cortafuegos de capa de aplicación Examinan los paquetes en el nivel de aplicación, por lo que son cortafuegos específicos para cada protocolo.

Cortafuegos con seguimiento de la conexión Son capaces de analizar si los paquetes pertenecen o no a una conexión establecida, están relacionados con una conexión previa o son paquetes que establecen una conexión nueva.

En el caso de iptables es un cortafuegos de filtrado de paquetes que puede analizar el tráfico desde en nivel de enlace al nivel de transporte, además iptables puede realizar seguimiento de la conexión[1] y que puede asociarse con otras aplicaciones o extensiones para funcionar como cortafuegos de capa de aplicación. En este documento nos centraremos en las características de filtrado de paquetes y el seguimiento de la conexión de forma simple.

3.1. Políticas de cortafuegos

Un cortafuegos tras examinar una serie de parámetros de un paquete y analizar las reglas que para él son aplicables, realiza una determinada acción con él: aceptarlo, rechazarlo, marcarlo, etc. En cualquier caso tiene que haber una decisión por defecto cuando no haya ninguna regla aplicable a un paquete, es lo que se denomina la política por defecto del cortafuegos o simplemente política. En el caso de iptables existen dos políticas por defecto:

ACCEPT Se aceptan por defecto todos los paquetes salvo aquellos para los que haya reglas específicas que no lo permitan.

DROP Se deniegan por defecto todos los paquetes salvo aquellos para los que haya reglas específicas que lo permitan.

La primera política es más fácil de implementar si hay pocas reglas, aunque mucho menos segura, por lo que en este documento nos centraremos en la implementación de un cortafuegos con política DROP.

4. Características de iptables

Iptables es un componente del núcleo linux que suele incluirse habitualmente en forma de módulos, esto implica que no hay un proceso (un demonio) como en el caso de otras aplicaciones que hemos visto en el curso que pueda pararse, lanzarse o recargarse. Iptables está siempre presente en nuestro equipo[2], la forma de trabajar con él es simplemente ejecutar una serie de reglas con ayuda de la instrucción iptables y una determinada sintaxis que veremos a continuación.

Además de todo esto, iptables no es sólo una aplicación de cortafuegos, sino que puede funcionar como dispositivo de nat o puede alterar los paquetes que lo atraviesan. En este documento se explicará la forma de trabajar con iptables como aplicación de cortafuegos mediante filtrado de paquetes, seguimiento de la conexión y dispositivo de NAT.

4.1. Tablas y cadenas

Iptables incluye de forma estándar tres **tablas**, dentro de las que se definen una serie de **cadenas** y que a su vez están constituidas por una agrupación de **reglas**. Cada una de estas

[1] Para un seguimiento de la conexión más avanzado puede utilizarse conntrack del mismo proyecto netfilter.
[2] Salvo en el caso de un núcleo compilado sin incluir iptables, algo muy poco frecuente

reglas tiene una serie de parámetros que especifican el paquete al que se van a aplicar y finalmente una **acción**, que es la encargada de decir qué destino tiene el paquete.

Las tablas predefinidas de iptables son:

filter Permite generar las reglas de filtrado o sea, que paquetes aceptar, cuales rechazar o cuales omitir. Es la tabla por defecto. Las cadenas que componen la tabla filter son:

INPUT Filtrado de paquetes que llegan al cortafuegos.

OUTPUT Filtrado de los paquetes de salen del cortafuegos.

FORWARD Filtrado de los paquetes que atraviesan el cortafuegos.

nat Con esta tabla es posible realizar enmascaramiento de IP, redireccionar puertos o cambiar las direcciones IP de origen y destino de los paquetes. Se utiliza para modificar la dirección IP origen o destino de los paquetes. Las cadenas que componen la tabla nat son:

PREROUTING Permite realizar alguna acción sobre el paquete antes de que se tome la decisión de encaminamiento. Utilizada fundamentalmente para realizar Destination NAT o DNAT.

POSTROUTING Permite realizar alguna acción sobre el paquete antes de que salga del cortafuegos. Utilizada principalmente para realizar enmascaramiento IP o SNAT.

OUTPUT Permite modificar los paquetes generados en el propio cortafuegos antes de enrutarlos. No vamos a utilizar ningún ejemplo en este documento con esta cadena, por lo que no la tendremos en cuenta a partir de ahora.

mangle Tabla para la modificación de paquetes con la que no vamos a trabajar en este documento.

Dependiendo del tipo de proceso, se aplican las reglas de una cadena u otra y además se hace en un determinado orden. Las cadenas utilizadas en cada tipo de proceso y el orden en el que se hace se muestran de forma esquemática en la figura 4 y se explican a continuación:

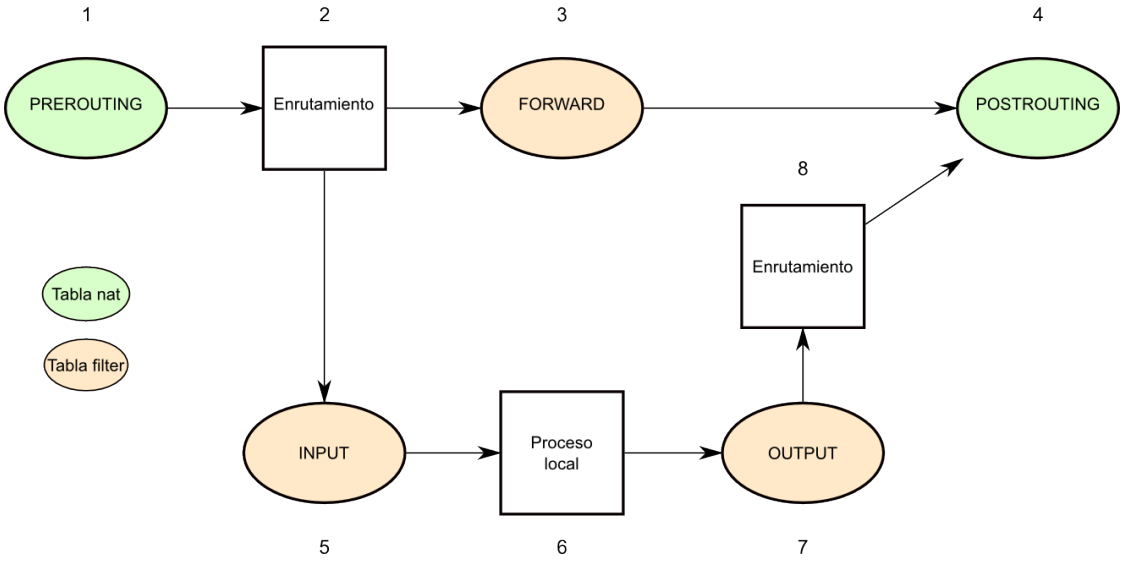

Figura 4: Cadenas de iptables implicadas en los diferentes procesos que se pueden producir.

- El paquete llega al cortafuegos y se le aplican las reglas de la cadena PREROUTING (1), se toma la decisión de encaminamiento (2), que obligan al paquete a atravesar

el cortafuegos, a continuación se aplican las reglas de la cadena FORWARD (3) y por último, antes de salir se aplican las reglas de POSTROUTING.

- Al igual que en el caso anterior el paquete llega al cortafuegos y se le aplican las reglas de la cadena PREROUTING (1), se toma la decisión de encaminamiento (2), aunque en este caso obligan al paquete a dirigirse al propio cortafuegos, a continuación se aplican las reglas de la cadena INPUT (5) y termina el proceso (6).
- El paquete se origina en un proceso local (6) dentro del cortafuegos, antes de salir se le aplican las reglas de la cadena OUTPUT (7), se toma la decisión de encaminamiento (8) y por último se aplican las reglas de la cadena POSTROUTING (4).

4.2. Formato de las reglas

La estructura o el esqueleto de una regla es:

```
iptables -t [tabla] operación cadena parámetros acción
```

Donde el valor de tabla y cadena se corresponde con los especificados anteriormente. Veamos los valores más significativos de los demás componentes de la regla:

4.2.1. Operaciones habituales

Especifican qué se va a hacer con la regla y puede tomar los valores:

-A, --append Añade una o más reglas al final de la cadena especificada.

-D, --delete Borra una o más reglas de la cadena. Se puede especificar la regla mediante un número o mediante parámetros.

-I, --insert Inserta una regla en una determinada posición de la cadena. Si no se especifica una posición, inserta la regla al principio de la cadena.

-L, --list Lista las reglas de una cadena concreta o todas las de la tabla si no se especifica.

-F, --flush Borra todas las reglas de una cadena o todas las de la tabla si no se especifica.

-Z, --zero Pone a cero los contadores de una cadena o los de todas las cadenas de una tabla si no se especifica.

4.2.2. Parámetros más comunes

Se utilizan para definir el paquete al que se le va a aplicar la regla.

-p, --protocol Especifica el protocolo (tcp, udp, icmp, . . .). Si se quiere especificar el puerto destino u origen, se acompaña del parámetro **--dport** o **--sport**.

-s, --source Especifica la dirección o direcciones IP origen.

-d, --destination Especifica la dirección o direcciones IP destino.

-i, --in-interface Especifica la interfaz de red de entrada.

-o, --out-interface Especifica la interfaz de red de salida.

4.2.3. Acciones

Una vez que se define completamente la regla hay que determinar la acción a realizar con los paquetes que la cumplan. Las acciones se especifican con el parámetro **-j** seguido de:

ACCEPT Se acepta el paquete.

DROP Se elimina el paquete (no se responde al equipo que realiza la petición, que pasado un tiempo mostrará un mensaje de tiempo excedido en la petición).

REJECT Equivalente a DROP, pero envía un mensaje ICMP al equipo que realiza la petición para que sepa que no está permitida (por defecto port-unreachable).

DNAT Utilizada en la cadena PREROUTING para modificar la dirección IP destino. Debe llevar asociado el parámetro **--to, --to-destination**.

SNAT Utilizada en la cadena POSTROUTING para modificar la dirección IP origen si tenemos dirección IP estática en la interfaz de red de salida. Debe llevar asociado el parámetro **--to, --to-source**.

MASQUERADE Equivalente a SNAT pero utilizado cuando tenemos direción IP dinámica en la interfaz de red de salida.

REDIRECT Utilizada en la cadena PREROUTING para modificar la dirección IP a la que tenga la interfaz de red de entrada. Puede llevar el parámetro **--to, --to-ports** para especificar un cambio en el puerto destino.

Por ejemplo, si queremos permitir una consulta DNS simple (`53/udp`) al equipo cliente (`192.168.2.2`), se podría poner una regla de iptables como la siguiente[3]:

```
avatar:~# iptables -A FORWARD -p udp --dport 53 -s 192.168.2.0/24 \
-i eth1 -o eth0 -j ACCEPT
```

Esto permitiría realizar la consulta DNS, pero como tenemos política por defecto DROP, no se aceptaría la respuesta del servidor DNS. Es por esto por lo que si utilizamos política DROP, las reglas son siempre dobles (dos reglas de FORWARD para paquetes que atraviesan el cortafuegos y una regla de INPUT y otra de OUTPUT para paquetes generados en el propio equipo que actúa de cortafuegos). En el ejemplo anterior tendríamos que añadir una segunda regla que permitiese la respuesta del servidor DNS:

```
avatar:~# iptables -A FORWARD -p udp --sport 53 -d 192.168.2.0/24 \
-o eth1 -i eth0 -j ACCEPT
```

En el caso de que quisiéramos permitir consultas DNS a avatar, el par de reglas de iptables serían:

```
avatar:~# iptables -A OUTPUT -p udp --dport 53 -s 192.168.1.1 -o eth0 \
-j ACCEPT
avatar:~# iptables -A INPUT -p udp --sport 53 -d 192.168.1.1 -i eth0 \
-j ACCEPT
```

4.3. Seguimiento de la conexión

Una de las formas habituales de mejorar el funcionamiento de un cortafuegos de filtrado de paquetes es añadir seguimiento de la conexión *connection tracking o conntrack*. Además hay

[3] \ significa que la línea continúa

algunos protocolos que establecen más de una conexión entre el cliente y el servidor (FTP, TFTP, IRC y PPTP por ejemplo) que no pueden funcionar en un cortafuegos con política DROP sin conntrack.

Para el seguimiento de la conexión se definen cuatro estados posibles:

NEW Para una conexión nueva.

ESTABLISHED Existe previamente tráfico en esa conexión.

RELATED Conexión relacionada con una establecida (como en el caso del FTP pasivo)

INVALID Paquete que no siguen un comportamiento esperado y que pueden establecerse reglas para denegarlos o analizarlos.

El seguimiento de la conexión en iptables se especifica mediante el parámetro **-m state --state ESTADO**, donde ESTADO es alguno de los estados definidos anteriormente.

El ejemplo anterior en el que se permitían consultas DNS a cliente y avatar se modificaría de la siguiente manera al introducir el seguimiento de la conexión[4]:

```
iptables -A FORWARD -p udp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT
```

5. Configuración paso a paso de un cortafuegos

En esta sección vamos a construir reglas de iptables de forma paulatina, de manera que cada ejemplo sea completo, pero vayamos incrementando las prestaciones en cada caso. Empezaremos todos los ejemplos con líneas de iptables en las que se borran todas las reglas que estuvieran creadas previamente para evitar cualquier conflicto.

Después de cada paso es recomendable ejecutar la instrucción:

```
iptables -t tabla -L -n -v
```

Donde tabla puede ser filter o nat y que sirve para comprobar las reglas que se han aplicado y el número de paquetes que las utilizan.

Empezaremos desarrollando reglas elementales para que los equipos de la red local accedan a Internet y se permita todo el tráfico de forma explítica:

```
# Borramos las reglas que haya definidas y ponemos los contadores
# a cero
iptables -F
iptables -t nat -F
iptables -Z

# Establecemos la política por defecto de las cadenas de la tabla
# filter
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Permitimos el tráfico TCP entre la red cliente y el exterior
```

[4] A partir de ahora no incluiremos el prompt de la línea de comandos para mayor claridad

```
iptables -A FORWARD -p tcp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p tcp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Permitimos el tráfico UDP entre la red cliente y el exterior
iptables -A FORWARD -p udp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p udp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Permitimos el tráfico ICMP entre la red cliente y el exterior
iptables -A FORWARD -p icmp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p icmp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Hacemos masquerade para los paquetes que salen hacia Internet
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
```

A continuación incluímos seguimiento de la conexión, de manera que sólo se permiten paquetes provenientes de Internet que se correspondan con respuestas a solicitudes hechas desde la red local.

```
# Borramos las reglas que haya definidas y ponemos los contadores
# a cero
iptables -F
iptables -t nat -F
iptables -Z

# Establecemos la política por defecto de las cadenas de la tabla
# filter
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Permitimos el tráfico TCP entre la red cliente y el exterior,
# pero sólo para conexiones iniciadas desde la red local
iptables -A FORWARD -p tcp --sport 1024:65535 -i eth1 -o eth0 \
-s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 -o eth1 -i eth0 \
-d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos el tráfico UDP entre la red cliente y el exterior,
# pero sólo para conexiones iniciadas desde la red local
iptables -A FORWARD -p udp --sport 1024:65535 -i eth1 -o eth0 \
-s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --dport 1024:65535 -o eth1 -i eth0 \
-d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos el tráfico ICMP entre la red cliente y el exterior
iptables -A FORWARD -p icmp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p icmp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Hacemos REJECT con el resto de paquetes que provengan de la
# red local
iptables -A FORWARD -i eth1 -o eth0 -s 192.168.2.0/24 -j REJECT

# Hacemos masquerade para los paquetes que salen hacia Internet
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
```


Hasta ahora se ha permitido que los equipos de la red local accedan a todos los servicios que quieran de Internet, pero si queremos restringir el tráfico de salida a determinados protocolos,

debemos poner un par de reglas de iptables para cada protocolo permitido. A continuación se van a permitir sólo consultas DNS (UDP y TCP), y HTTP:

```
#Limpiamos las reglas anteriores y ponemos los contadores a cero:
iptables -F
iptables -t nat -F
iptables -Z

# Establecemos la política por defecto de las cadenas de la tabla
# filter
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Permitimos conexiones HTTP
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 80 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 80 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos consultas DNS por tcp
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos consultas DNS por udp
iptables -A FORWARD -p udp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos el tráfico ICMP entre la red cliente y el exterior
iptables -A FORWARD -p icmp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p icmp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Hacemos REJECT con el resto de paquetes que provengan de la
# red local
iptables -A FORWARD -i eth1 -o eth0 -s 192.168.2.0/24 -j REJECT

# Hacemos masquerade para los paquetes que salen hacia Internet
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
```

Con la configuración anterior, avatar está completamente incomunicado (incluso consigo mismo a través de localhost), por lo que vamos a incluir algunas reglas que permitan la conexión a localhost y además de algunos protocolos para Internet y para la red local:

```
#Limpiamos las reglas anteriores y ponemos los contadores a cero:
iptables -F
iptables -t nat -F
iptables -Z

# Establecemos la política por defecto de las cadenas de la tabla
# filter
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP
```

```
# Permitimos todo el tráfico a localhost
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -o lo -j ACCEPT

# Permitimos consultas DNS desde avatar (UDP y TCP)
iptables -A OUTPUT -p udp --sport 1024:65535 --dport 53 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp --dport 1024:65535 --sport 53 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 53 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 53 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos acceso a servicio SMTP externo
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 25 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 25 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos acceso a un servidor LDAP que estuviera en cliente
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 389 -o eth1 -d \
192.168.2.2 -s 192.168.2.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 389 -i eth0 -s \
192.168.2.2 -d 192.168.2.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos conexiones HTTP
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 80 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 80 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos consultas DNS por tcp
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos consultas DNS por udp
iptables -A FORWARD -p udp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos el tráfico ICMP entre la red cliente y el exterior
iptables -A FORWARD -p icmp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p icmp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Hacemos REJECT con el resto de paquetes que provengan de la
# red local
iptables -A FORWARD -i eth1 -o eth0 -s 192.168.2.0/24 -j REJECT

# Hacemos masquerade para los paquetes que salen hacia Internet
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
```


Veamos ahora en el caso de que tuviéramos un servidor en avatar (SMTP para la red local e Internet):

```
#Limpiamos las reglas anteriores y ponemos los contadores a cero:
iptables -F
iptables -t nat -F
iptables -Z

# Establecemos la política por defecto de las cadenas de la tabla
# filter
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Permitimos todo el tráfico a localhost
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -o lo -j ACCEPT

# Permitimos consultas DNS desde avatar (UDP y TCP)
iptables -A OUTPUT -p udp --sport 1024:65535 --dport 53 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp --dport 1024:65535 --sport 53 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 53 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 53 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos acceso a servicio SMTP externo
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 25 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 25 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos acceso al servidor SMTP desde fuera
iptables -A INPUT -p tcp --sport 1024:65535 --dport 25 -i eth0 \
-d 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --dport 1024:65535 --sport 25 -o eth0 \
-s 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos acceso al servidor SMTP desde la red local
iptables -A INPUT -p tcp --sport 1024:65535 --dport 25 -i eth1 \
-d 192.168.1.1 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED \
-j ACCEPT
iptables -A OUTPUT -p tcp --dport 1024:65535 --sport 25 -o eth1 \
-s 192.168.1.1 -d 192.168.2.0/24 -m state --state ESTABLISHED \
-j ACCEPT

# Permitimos acceso a un servidor LDAP que estuviera en cliente
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 389 -o eth1 -d \
192.168.2.2 -s 192.168.2.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 389 -i eth0 -s \
192.168.2.2 -d 192.168.2.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos conexiones HTTP
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 80 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 80 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT
```

```
# Permitimos consultas DNS por tcp
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos consultas DNS por udp
iptables -A FORWARD -p udp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos el tráfico ICMP entre la red cliente y el exterior
iptables -A FORWARD -p icmp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p icmp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Hacemos REJECT con el resto de paquetes que provengan de la
# red local
iptables -A FORWARD -i eth1 -o eth0 -s 192.168.2.0/24 -j REJECT

# Hacemos masquerade para los paquetes que salen hacia Internet
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
```

En el caso de que el servidor SMTP no estuviera en avatar sino en un equipo de la red local (`192.168.2.2`), habría que cambiar las reglas anteriores por reglas de FORWARD y además redirigir las peticiones exteriores al puerto `25/tcp` mediante DNAT[5]:

```
#Limpiamos las reglas anteriores y ponemos los contadores a cero:
iptables -F
iptables -t nat -F
iptables -Z

# Establecemos la política por defecto de las cadenas de la tabla
# filter
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Redirigimos las peticiones al servidor SMTP de la red local
iptables -t nat -A PREROUTING -p tcp --dport 25 -d 192.168.1.1 \
-j DNAT --to 192.168.2.2

# Permitimos todo el tráfico a localhost
iptables -A OUTPUT -o lo -j ACCEPT
iptables -A INPUT -o lo -j ACCEPT

# Permitimos consultas DNS desde avatar (UDP y TCP)
iptables -A OUTPUT -p udp --sport 1024:65535 --dport 53 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p udp --dport 1024:65535 --sport 53 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 53 -o eth0 \
-s 192.168.1.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 53 -i eth0 \
-d 192.168.1.1 -m state --state ESTABLISHED -j ACCEPT
```

[5] Lo sensato sería tener este equipo en una DMZ, pero como ya explicamos anteriormente se hace aquí con el esquema de red del curso

```
# Permitimos acceso a servicio SMTP externo desde 192.168.2.2
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 25 -o eth0 \
-i eth1 -s 192.168.2.2 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 25 -i eth0 \
-o eth1 -d 192.168.2.2 -m state --state ESTABLISHED -j ACCEPT

# Permitimos acceso al servidor SMTP desde fuera
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 25 -i eth0 \
-o eth1 -d 192.168.2.2 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 25 -o eth0 \
-i eth1 -s 192.168.2.2 -m state --state ESTABLISHED -j ACCEPT

# Permitimos acceso a un servidor LDAP que estuviera en cliente
iptables -A OUTPUT -p tcp --sport 1024:65535 --dport 389 -o eth1 -d \
192.168.2.2 -s 192.168.2.1 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A INPUT -p tcp --dport 1024:65535 --sport 389 -i eth0 -s \
192.168.2.2 -d 192.168.2.1 -m state --state ESTABLISHED -j ACCEPT

# Permitimos conexiones HTTP
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 80 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 80 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos consultas DNS por tcp
iptables -A FORWARD -p tcp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p tcp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos consultas DNS por udp
iptables -A FORWARD -p udp --sport 1024:65535 --dport 53 -i eth1 -o \
eth0 -s 192.168.2.0/24 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A FORWARD -p udp --dport 1024:65535 --sport 53 -o eth1 -i \
eth0 -d 192.168.2.0/24 -m state --state ESTABLISHED -j ACCEPT

# Permitimos el tráfico ICMP entre la red cliente y el exterior
iptables -A FORWARD -p icmp -i eth1 -o eth0 -s 192.168.2.0/24 -j ACCEPT
iptables -A FORWARD -p icmp -o eth1 -i eth0 -d 192.168.2.0/24 -j ACCEPT

# Hacemos REJECT con el resto de paquetes que provengan de la
# red local
iptables -A FORWARD -i eth1 -o eth0 -s 192.168.2.0/24 -j REJECT

# Hacemos masquerade para los paquetes que salen hacia Internet
iptables -t nat -A POSTROUTING -s 192.168.2.0/24 -o eth0 -j MASQUERADE
```


6. Ejecución automática al iniciar

Las reglas de iptables se pueden ejecutar directamente desde la línea de comandos y por tanto se pueden asociar a un fichero de script, que tenga en la primera línea *#!/bin/sh*). Para ejecutar este script de forma automática, basta con añadir permiso de ejecución al script y ubicarlo en alguno de los directorios que se ejecutan al iniciar la máquina: */etc/rc.local*,

/etc/rc2.d/, etc.

También se puede optar por utilizar las instrucciones *iptables-save* e *iptables-restore*, que básicamente hacen lo mismo aunque incluyen pequeñas modificaciones en el formato de las líneas del fichero.