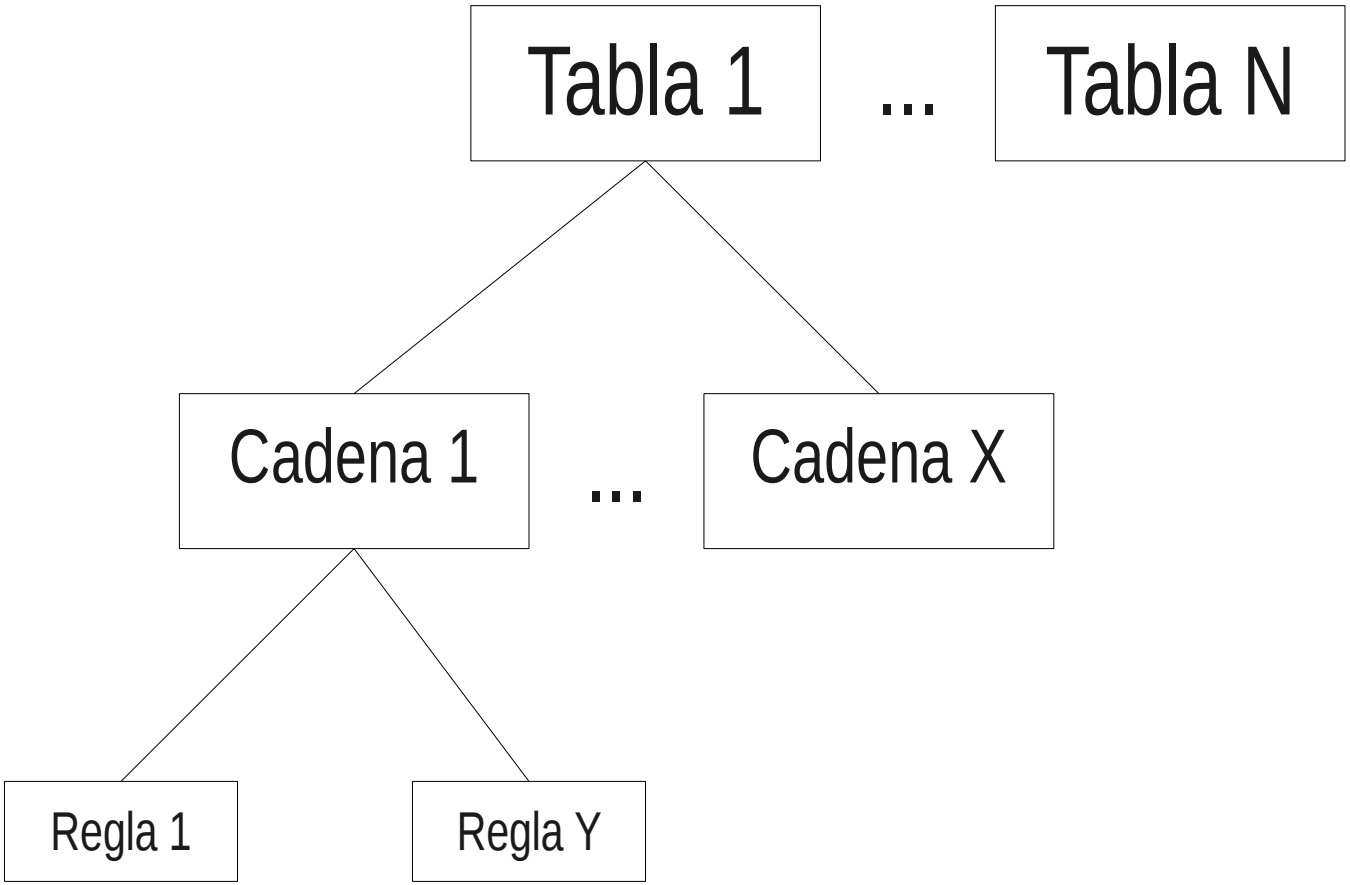
Introducción

- Un cortafuegos permite controlar el tráfico entre dos redes, generalmente una red interna y otra externa.
- Existen dos estrategias básicas:
  - Permiso predeterminado:
    - Las condiciones impiden el paso de datos a la red interna.
    - Cualquier ordenador, etc., no incluido en las condiciones tiene permitido el acceso a la red interna.
    - Más fácil de configurar pero más inseguro.
  - Denegación predeterminada:
    - Las condiciones permiten el paso de datos a la red interna.
    - Cualquier ordenador, etc., no incluido en las condiciones tiene denegado el acceso a la red interna.
    - Más difícil de configurar pero más seguro.

Iptables del kernel de Linux (I)

- Kernel de Linux:
  - <2.4: ipchains.
    - Solo filtrado y enmascaramiento.
    - No permiten realizar seguimiento de conexiones, etc.
  - >=2.4: iptables.
    - Filtrado y enmascaramiento.
    - Seguimiento de conexiones.
    - Modificación de campos de cabecera.
    - Etc.

Iptables del kernel de Linux (II)



Iptables del kernel de Linux (III)

- Iptables posee cinco tablas:
  - filter:
    - Tabla por defecto.
    - Filtra los paquetes de la red.
  - nat:
    - Altera las direcciones de origen y/o destino de los paquetes.
  - mangle:
    - Alteraciones locales del origen o destino de los paquetes para balancear tráfico, por ejemplo.
  - raw:
    - Configura excepciones en el seguimiento de los paquetes de las conexiones.
  - security:
    - Permite a módulos de seguridad de Linux (SELinux) implementar reglas de filtrado.

Iptables del kernel de Linux (IV)

- Cada tabla esta formada por cadenas:
  - Predefinidas.
  - Definidas por el usuario.
- Las cadenas predefinidas de cada tabla son:
  - filter:
    - INPUT: Paquetes destinados a un proceso local.
    - OUTPUT: Paquetes generados localmente por un proceso.
    - FORWARD: Paquetes recibidos por un dispositivo de red y que deben ser reenviados a la red sin ser procesados de forma local.

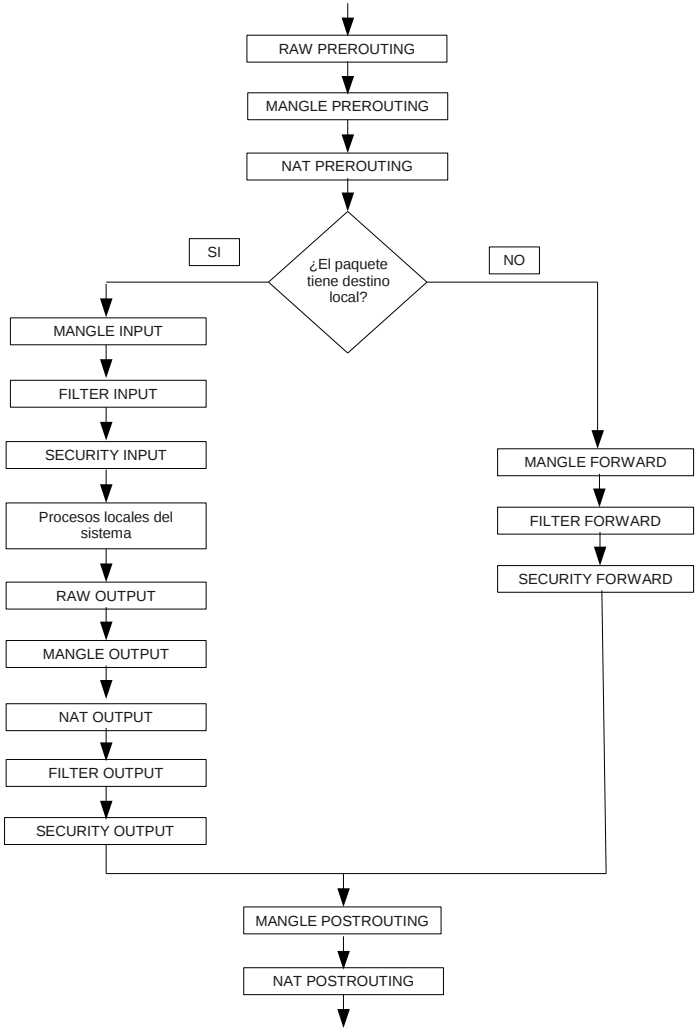
Iptables del kernel de Linux (V)

- nat:
  - PREROUTING: Paquetes recibidos y no procesados.
  - OUTPUT: Paquetes generados localmente y no enviados.
  - POSTROUTING: Paquetes que no han salido a la red.
- mangle:
  - PREROUTING: Paquetes recibidos y no enrutados.
  - INPUT: Paquetes destinados a un proceso local.
  - OUTPUT: Paquetes generados por un proceso local y no enrutados.
  - FORWARD: Paquetes reenviados entre dos dispositivos de red.
  - POSTROUTING: Paquetes que no han salido a la red.

Iptables del kernel de Linux (VI)

- raw:
  - PREROUTING: Paquetes recibidos por cualquier dispositivo de red.
  - OUTPUT: Paquetes generados por un proceso local.
- security:
  - INPUT: Paquetes destinados a un proceso local.
  - OUTPUT: Paquetes generados localmente por un proceso.
  - FORWARD: Paquetes recibidos por un dispositivo de red y que deben ser reenviados a la red sin ser procesados de forma local.

Iptables del kernel de Linux (VII)



Iptables del kernel de Linux (VIII)

- Un paquete modificado por una regla de una cadena de una tabla.
  - Aparece para el resto de reglas, cadenas y tablas con esa modificación.
  - No existe posibilidad de conocer el contenido inicial del paquete.
- Ejemplo:

```
iptables -t nat -A PREROUTING -p tcp -d 192.168.0.1
   --dport 80  -j DNAT --to-destination 192.168.0.2:80
```

- A partir de la modificación el paquete tiene como destino el puerto 80 de 192.168.0.2.

Acciones existentes en iptables

- Iptables tiene cuatro acciones por defecto:
  - ACCEPT: Aceptar un paquete sin ser analizado por el resto de reglas y tablas.
  - DROP: Rechazar un paquete sin enviar ningún mensaje al origen.
  - QUEUE: Enviar el paquete a un módulo de procesamiento en el espacio de usuario.
  - RETURN: Devolver el paquete a la regla siguiente a la regla que ocasionó la llamada a esta regla.
- Existen otras acciones que dependen de la tabla.

Comandos de iptables (I)

- Las tablas:
  - Poseen un comportamiento predefinido.
  - Puede alterarse mediante comandos de sintaxis:

```
iptables [-t <nombre de tabla>] <comando> <nombre de la cadena>
   <parámetro 1> <opción 1>...<parámetro N> <opción N>
```

    - <nombre de tabla>: Tabla sobre la que se ejecuta el comando (por defecto tabla filter).
    - <comando>: Acción a realizar sobre la cadena de la tabla.
    - <nombre de la cadena>: Cadena de la tabla sobre la que se ejecuta la acción.
    - <parámetro X> <opción X>: Especificación de la regla.
  - La complejidad de los comandos depende de su objetivo.

Comandos de iptables (II)

Comando	Descripción
-A	Añade la regla especificada al final de la cadena especificada.
-C	Chequea una regla y verifica su validez en la cadena especificada. Permite al usuario chequear una regla antes de que sea añadida a la cadena especificada.
-D	Borra una regla de la cadena especificada. Puede especificarse por un número que indique su posición, comenzando a contar siempre en 1, o bien escribir la regla completa a borrar.
-E	Renombra una cadena definida por el usuario. Esta acción no afecta a la estructura de la tabla donde se encuentra la cadena.
-F	Borrar todas la reglas de la cadena especificada. Si no se especifica la cadena, todas las reglas de todas las cadenas son borradas.
-h	Proporciona información de ayuda.
-I	Inserta una regla en la cadena en la posición indicada. Si no se indica ninguna posición la regla es insertada al principio de la cadena.

Comandos de iptables (II)

Comando	Descripción
-L	Lista todas las reglas. Los valores -v, -x y -n, permiten especificar que la salida sea más extensa (valor -v), que se de en valores exactos y no abreviados con K (miles), M (millones), etc., (valor -x), y que se de en valor numérico de direcciones IP y puertos (valor -n).
-N	Crea una nueva cadena con el nombre especificado por el usuario.
-P	Asigna la política por defecto a una cadena, de forma que si un paquete no corresponde a ninguna regla, esta será la acción por defecto a aplicar.
-R	Reemplaza la regla situada en la posición indicada de la cadena por la regla especificada. Como en la opción -D empieza a contar en 1.
-X	Borra una cadena especificada por el usuario. Borrar una cadena predefinida de una tabla no esta permitido.
-Z	Inicializa a cero el contador de bytes y paquetes en todas las cadenas de una tabla.

Comandos de iptables (IV)

- Ejemplos de comandos:

```
iptables -t nat -L -v
```

  - Muestra los datos de las reglas definidas en todas las cadenas de la tabla nat.

```
iptables -t mangle -N MI_CADENA
```

  - Crea una nueva cadena en la tabla mangle.

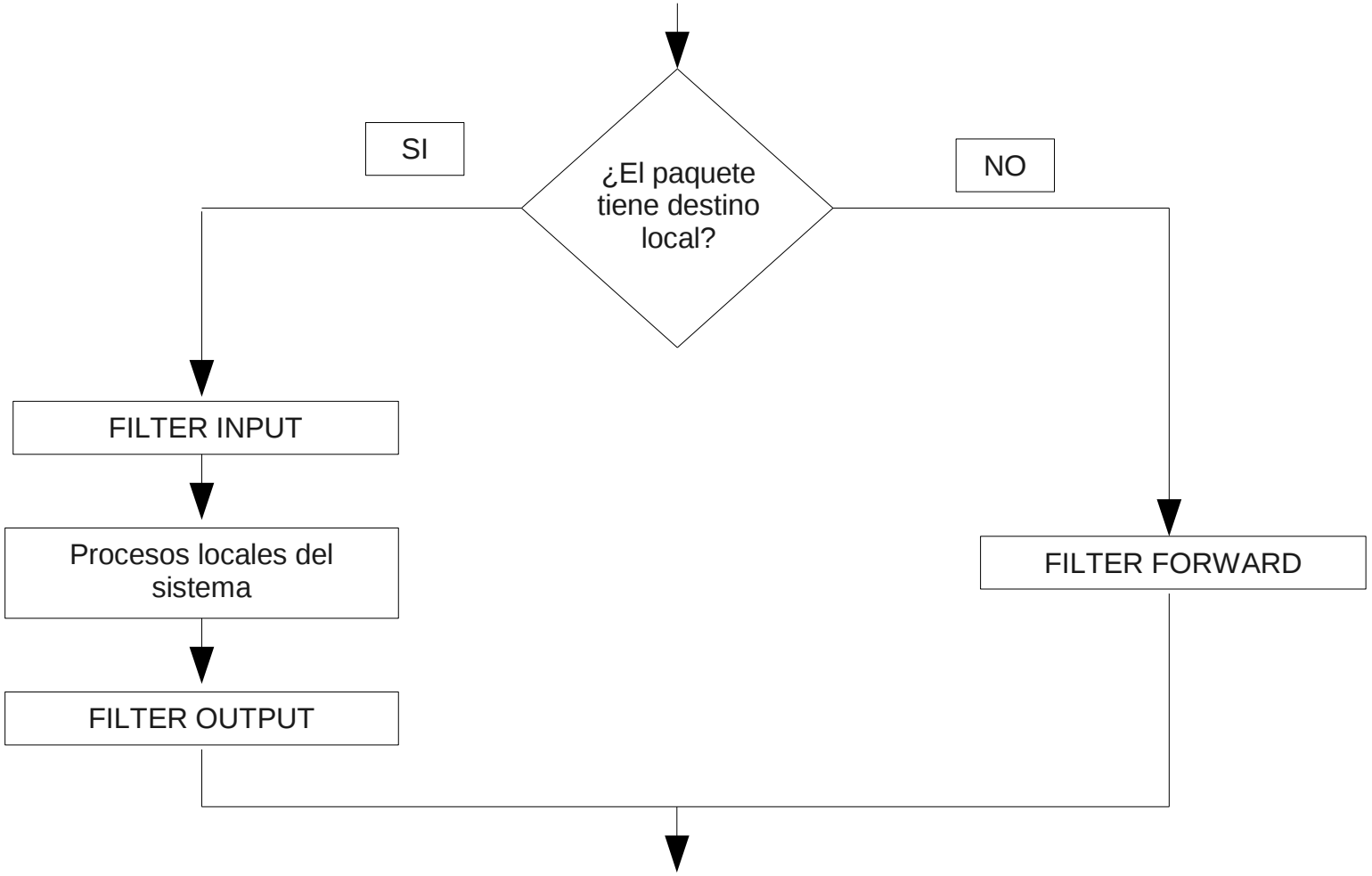
```
iptables -I INPUT 3 -p tcp -s 147.156.0.0/16 -j ACCEPT
```

  - Añade la regla especificada en la tercera posición de la cadena INPUT de la tabla filter.

La tabla filter (I)

- En iptables:
  - Cualquier tabla puede filtrar paquetes.
  - La más adecuada es la tabla filter.
- La tabla filter:
  - Filtra los paquetes:
    - Con destino un proceso local.
    - Con origen un proceso local.
    - Reenviados por el ordenador.
  - No realiza otras acciones como:
    - Alterar IPs de origen y/o destino.
    - Balancear tráfico.
    - Etc.

La tabla filter (II)



La tabla filter (III)

- Posee las siguientes extensiones a las acciones por defecto:
  - REJECT: Rechaza un paquete enviando un mensaje ICMP de error al origen.
    - El tipo de ICMP puede especificarse mediante --reject-with <tipo>, siendo el tipo por defecto icmp-port-unreachable.
  - LOG: Almacena información sobre el paquete en el log del sistema.
    - El paquete sigue analizándose por la regla siguiente.
    - --log-level <nivel> indica el nivel del log.
    - --log-prefix <cadena> es un texto de hasta 29 caracteres que se antepone al mensaje que genera iptables.

Parámetros de especificación de reglas (I)

- Los parámetros existentes son:
  - -c: Inicializa el contador de una regla durante:
    - Su inserción (-I).
    - Añadido (-A).
    - Reemplazo (-R).
    - Permite especificar el contador a inicializar:
      - PKTS para el contador de paquetes.
      - BYTES para el contador de bytes.
  - -d: Selecciona el ordenador/red destino del paquete.
    - El ordenador puede ser un nombre o IP.
    - La red puede especificarse como:
      - 192.168.0.0/255.255.0.0
      - 192.168.0.0/16
    - Si el parámetro se precede de ! la regla se aplica al resto de ordenadores/redes.

Parámetros de especificación de reglas (II)

- -f: Aplicar la regla a los paquetes fragmentados.
  - Si se precede de ! se aplica a los paquetes no fragmentados.
- -i: Dispositivo de red de entrada del paquete.
  - Solo puede aplicarse a las cadenas INPUT y FORWARD.
  - Si el parámetro se precede de ! se aplica la regla al resto de dispositivos de red.
  - El carácter + sustituye a un carácter y permite indicar un conjunto de dispositivos: eth+ -> eth0, eth1, etc.
- -j: Acción a realizar si el paquete coincide con la regla.

Parámetros de especificación de reglas (III)

- -m: Indica que se va a utilizar una extensión de los parámetros básicos.
- -o: Dispositivo de red de salida del paquete.
  - Solo puede aplicarse a las cadenas OUTPUT y FORWARD.
  - Opciones, etc., idénticas al parámetro -i.
- -p: Protocolo IP al que se le aplicará la regla:
  - TCP, UDP, ICMP, etc. o all, valor por defecto, para todos los protocolos.
  - Los protocolos validos son los existentes en /etc/protocols.
  - Precedido el protocolo de ! se aplica al resto de protocolos.
- -s: Selecciona el ordenador destino del paquete.
  - Mismas opciones, etc., que el parámetro -d.

Parámetros de especificación de reglas (IV)

- La opción -m permite especificar extensiones.
  - Si la extensión depende de un protocolo de red concreto.
  - El protocolo ha sido especificado con la opción -p.

  No es necesario poner la opción -m.
- Si el protocolo no ha sido especificado con la opción -p:
  - Debe ponerse opción -m <protocolo> para usar extensiones de ese protocolo.

Extensiones de TCP (I)

- --dport: Puerto de destino del paquete.
  - Puede especificarse mediante su nombre (www, smtp, etc.).
  - Número de puerto.
  - Rango de puertos: <puerto1>:<puerto2>
  - Si el puerto se precede de ! se aplica al resto de puertos.
- --sport: Puerto de origen del paquete.
  - Idéntica sintaxis, etc., que --dport.
- --syn: Paquete que inicia una conexión.
  - Si se precede de ! indica paquete que no inician una conexión.

Extensiones de TCP (II)

- --tcp-flags: Especificación de un paquete según el valor de los bits de bandera.
  - Los bits de bandera se indican como ACK, FIN, PSH, RST, SYN y URG.
  - Se utilizan dos listas separadas por un espacio.
    - La primera lista contiene, separadas por coma, las banderas a comprobar.
    - La segunda lista contiene, separadas por coma, las banderas que deben tener valor a 1.
  - Si las listas se preceden del símbolo ! las banderas deben tener valor 0.
- --tcp-option: Opción TCP que debe contener el paquete.

Extensiones de UDP

- --dport: Puerto de destino del paquete.
  - Puede especificarse mediante su nombre (www, smtp, etc.).
  - Número de puerto.
  - Rango de puertos: <puerto1>:<puerto2>
  - Si el puerto se precede de ! se aplica al resto de puertos.
- --sport: Puerto de origen del paquete.
  - Idéntica sintaxis, etc., que --dport.

Extensiones de ICMP

- --icmp-type:
  - Nombre o número del tipo ICMP que debe cumplir esta regla.
  - Precedida de ! indica que los paquetes no deben ser de este tipo.
  - Pueden obtenerse los tipos soportados ejecutando:

```
iptables –p icmp -h
```

Extensiones generales (I)

- No van unidas a ningún protocolo de red.
- Deben siempre especificarse usando el parámetro -m.
- Extensión mac:
  - Valida en las cadenas INPUT y FORWARD.
  - Solo posee una opción: --mac-source <MAC>
    - La dirección MAC se especifica como XX:XX:XX:XX:XX:XX.
    - Si se precede de la opción de ! indica que se aplique a las MACs distintas de la indicada.

Extensiones generales (II)

- Extensión recent:
  - Permite crear y buscar en una lista dinámica de direcciones IP de origen de los paquetes.
  - Sus opciones son:
    - --name <nombre>: Nombre de la lista. Si no se indica se utiliza la lista DEFAULT.
    - --set: Añade la IP de origen a la lista y devuelve verdad (mentira si se precede de !).
    - --rcheck: Comprueba si la IP de origen se encuentra en la lista.
    - --update: Mira si la IP de origen se encuentra en la lista y actualiza sus datos si existe.
    - --remove: Mira si la IP de origen se encuentra en la lista y la borra.
    - --seconds <segundos>:
      - Debe usarse junto con rcheck o update.
      - Devuelve verdad si la IP de origen esta en la lista y fue recibido hace más de “segundos”.
    - --hitcount <ocurrencias>:
      - Debe usarse junto con rcheck o update.
      - Devuelve verdad si la IP de origen esta en la lista y ha sido recibida un número mayor o igual que “ocurrencias”.
    - --rttl:
      - Debe usarse junto con rcheck o update.
      - Devuelve verdad si la IP de origen esta en la lista y el TTL se corresponde con el TTL del paquete que añadió la IP a la lista con la opción set.

Extensiones generales (III)

- Extensión state:
  - Permite el acceso de paquetes según su estado.
    - Realiza un seguimiento de las conexiones para conocer su estado.
  - Su única opción es --state <estado>:
    - INVALID: Paquete asociado a conexión desconocida.
    - ESTABLISHED: Paquete asociado a una conexión establecida y que envía paquetes en ambas direcciones.
    - NEW: Paquete que establece una conexión nueva.
    - RELATED: Paquete relacionado con una conexión existente (FTP en modo pasivo o error ICMP, por ejemplo).

Extensiones generales (IV)

- Extensión time:
  - Permite especificar valores temporales en las reglas.
  - Sus opciones son:
    - --timestart <valor>: Tiempo inicial.
    - --timestop <valor>: Tiempo final.
      - El tiempo se indica como hh:mm [00:00,23:59].
    - --days <lista de dias>: Dias de la semana separados por comas {Mon, Tue, Wed, Thu, Fri, Sat, Sun }.

Guardando la configuración

- Los comandos de iptables se ejecutan sobre la memoria RAM.
- Las reglas deben ser almacenadas en un fichero, en concreto /etc/sysconfig/iptables.

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -i eth0 -p udp -s 0/0 --sport 67:68 -d 0/0 --dport 67:68 -j ACCEPT
-A INPUT -p udp -j REJECT
-A INPUT -p tcp --syn -j REJECT
COMMIT
```

- Las reglas pueden guardarse mediante el comando:

```
service iptables save
```

El fichero iptables-config

- Dentro de /etc/sysconfig existe un fichero iptables-config:
  - Especifica el comportamiento al ser salvadas, cargadas, etc.
  - Permite indicar la inclusión de módulos opcionales para realizar el seguimiento de conexiones.
  - Los módulos se incluyen, como una lista entrecomillada, separada por espacios, en la línea IPTABLES_MODULES.
  - Realizar el seguimiento de FTP en modo pasivo:

```
IPTABLES_MODULES="ip_conntrack_ftp"
```

Ejemplos de cortafuegos (I)

- Cliente Linux:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -p udp -s 0/0 --sport 67:68 -d 0/0 --dport 67:68 -j ACCEPT
-A INPUT -p udp -j REJECT
-A INPUT -p tcp --syn -j REJECT
COMMIT
```

Ejemplos de cortafuegos (II)

- Servidor Linux:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT[0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p udp --sport 53 -j ACCEPT
-A INPUT -p udp -j REJECT
-A INPUT -p tcp --dport 22 --syn -j ACCEPT
-A INPUT -p tcp --dport 25 --syn -j ACCEPT
-A INPUT -p tcp --dport 80 --syn -j ACCEPT
-A INPUT -p tcp --dport 110 --syn -j ACCEPT
-A INPUT -p tcp --dport 995 --syn -j ACCEPT
-A INPUT -p tcp --syn -j REJECT
COMMIT
```

Ejemplos de cortafuegos (III)

- Servidor Linux como puerta de acceso a una subred:

```
*nat
:PREROUTING ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A POSTROUTING -o ppp0 -j MASQUERADE
COMMIT
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -i lo -j ACCEPT
-A INPUT -i eth0 -j ACCEPT
-A INPUT -p udp --sport 53 -j ACCEPT
-A INPUT -p udp -j REJECT
-A INPUT -p tcp --syn -j REJECT
-A FORWARD -i ppp0 -o eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
-A FORWARD -i eth0 -o ppp0 -j ACCEPT
-A FORWARD -j DROP
COMMIT
```