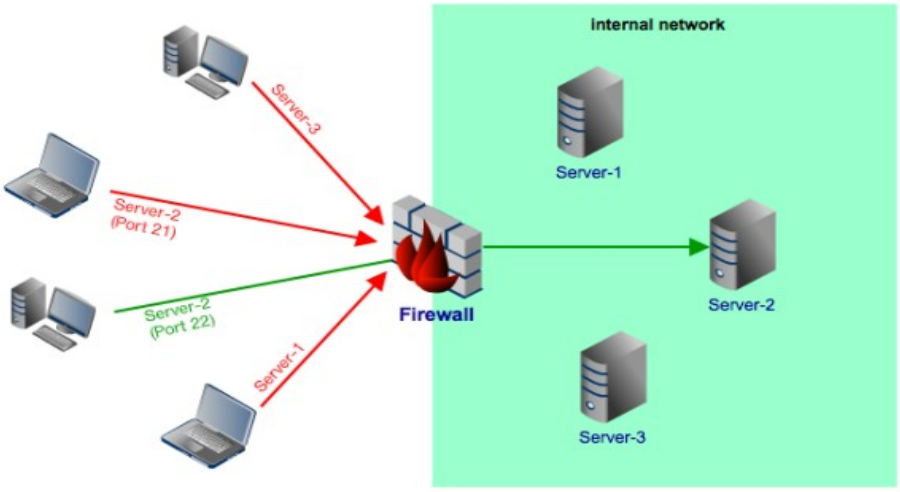
Control de acceso a servicios: Firewall

Prof. Manuel Castillo
Administración de Redes
Escuela Profesional de Ciencia de la Computación
Facultad de Ciencias
Universidad Nacional de Ingeniería

Definición

- Firewall permite establecer seguridad a nivel de paquetes y conexiones de comunicación.
- Un cortafuegos permite controlar el tráfico entre dos redes, generalmente una red interna y otra externa.
  - Es un mecanismo de control de acceso sobre la capa de red.
- Evitan el acceso a usuarios no autorizados a un host determinado.
- Instalación teniendo en cuenta estructura de red y servicios que queden libres.
- Importante su utilización para evitar ataques, por ejemplo, SYN.
- Dispositivos que funcionan como firewall:
  - Software's
  - Routers
  - Computadoras dedicadas como proxy
- Resumen: decide si un paquete:
  - pasa, se modifica, se convierte o se descarta.



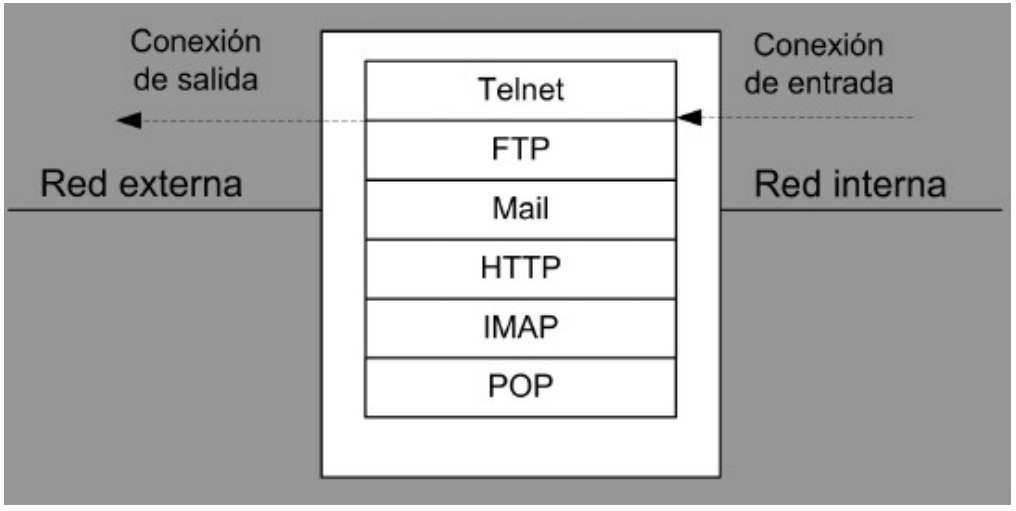
Tips antes de configurar

- No usar en lugar de otras herramientas, sino conjuntamente.
- Centraliza las medidas de seguridad de la red en un único sistema, pero no tiene por que ser el único.
  - Proxy, Snort...
- No implica seguridad al 100%, hay que vigilar equipos internos.
- Políticas diferentes cuando hay subredes diferentes:
  - Jerarquías de firewall → Perímetros de seguridad.
- Inversión proporcional al nivel de seguridad deseado.
- El propio cortafuegos debe estar protegido contra posibles intrusiones.
- Tres tecnologías:
  - Encaminadores con filtrado de paquetes.
  - Pasarelas a nivel de aplicación.
  - Pasarelas a nivel de circuito.

Pasarelas a nivel de aplicación

Definición

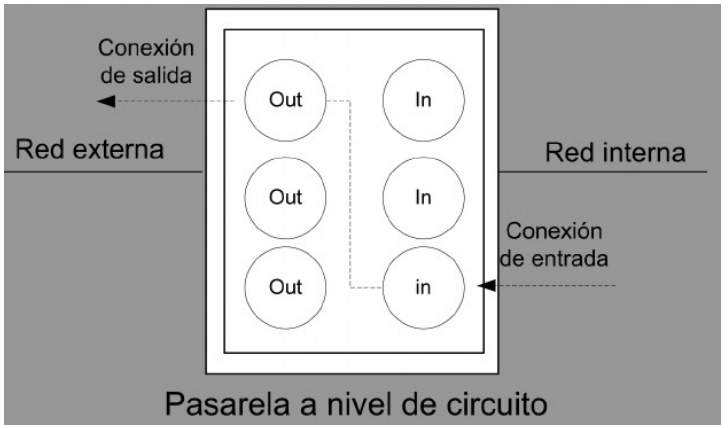
- Servidor intermediario (Servidor proxy).
- No encamina paquetes a nivel de red sino que actúa como retransmisor a nivel de aplicación.
- El servicio proxy se encargará de realizar las conexiones solicitadas con el exterior y, cuando reciba una respuesta, se encargará de retransmitirla al equipo que había iniciado la conexión.
- Por tanto, el servicio proxy ejecutado en la pasarela aplicará las normas para decidir si se acepta o se rechaza una petición de conexión.



Pasarelas a nivel de circuito

Definición

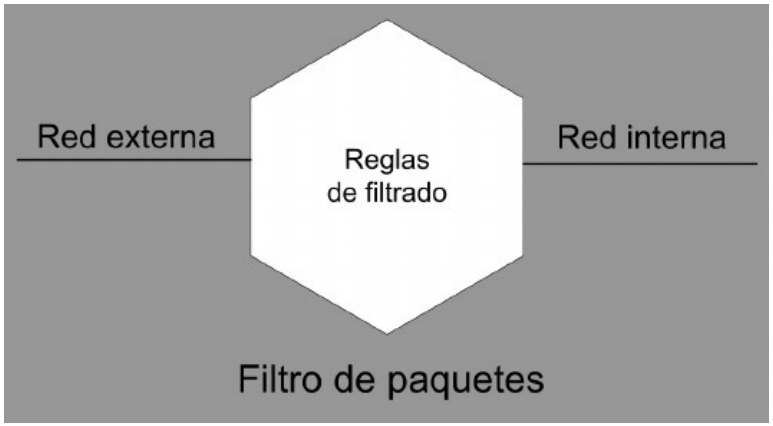
- Híbrido entre los esquemas de filtrado de paquetes y el uso de servidores intermediarios.
- Una vez establecida la conexión, el dispositivo se encargará de retransmitir todo el tráfico entre ambas partes sin inspeccionar el contenido de los paquetes a nivel de aplicación. Por tanto, evalúa la conexión solamente.
- La función de seguridad que ofrece este tipo de dispositivo consiste en determinar qué conexiones están permitidas, antes de bloquear conexiones hacia el exterior.

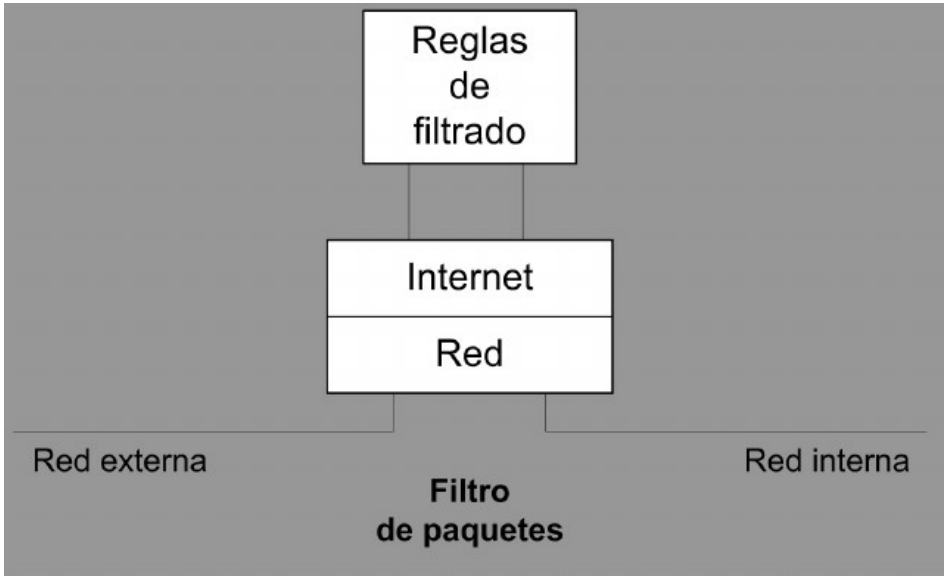


Encaminadores con filtrado de paquetes

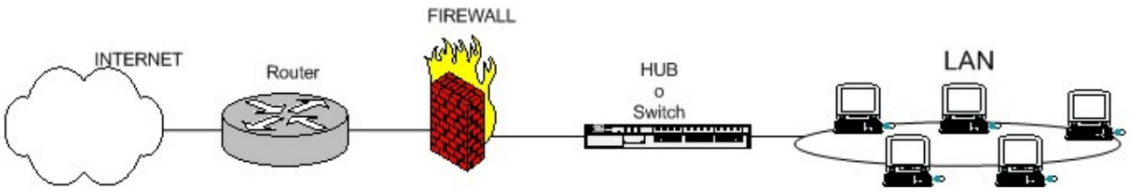
Definición

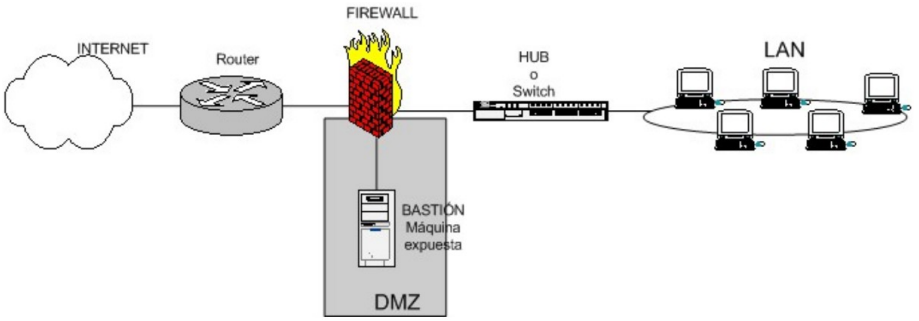
- Dispositivo que encamina el tráfico TCP/IP (router de TCP/IP) sobre la base de una serie de reglas de filtrado que deciden qué paquetes se encaminan a través suyo y cuales se descartan.
- Opciones que se pueden denegar:
  - Direcciones de origen y de destino.
  - Tipos de protocolo e indicadores (flags) especiales.
  - Puertos de origen y de destino o tipos de mensaje (según el protocolo).
  - Contenido de los paquetes.
  - Tamaño del paquete.

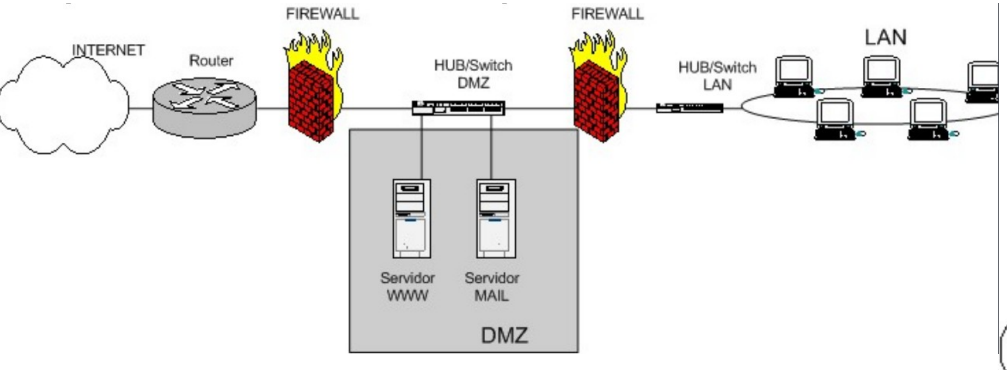


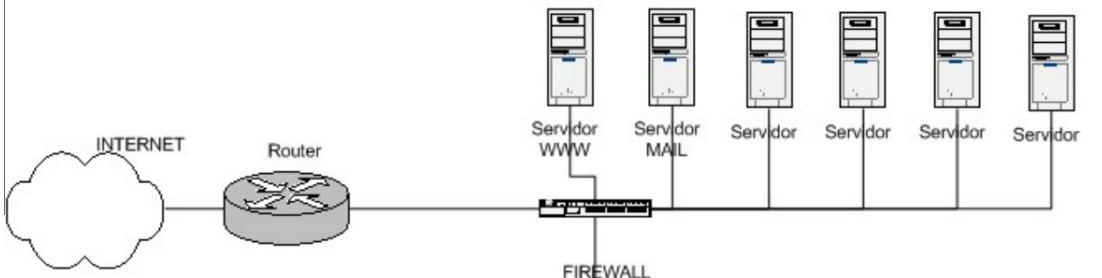


Esquemas de red

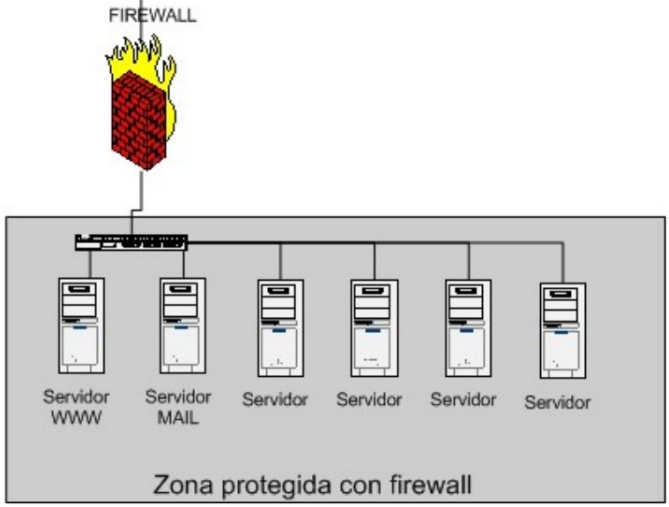








Este podría ser el esquema de un ISP en el que se coloca el firewall para proteger determinados servidores como un servicio adicional. El firewall se pone en medio entre el resto de la red y la zona de servidores protegidos. Para llegar a estos servidores se tiene que pasar por el firewall por narices.



Políticas

- Política **permisiva**:
  - Se permite todo, excepto todo lo que que se ha denegado.
  - 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.
- Política **restrictiva**:
  - Se deniega todo lo que explícitamente no se se permite.
  - 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.

Algunas preguntas

- ¿Que problema hay con no controlar que todo está abierto (política permisiva)?

Algunas preguntas

- ¿Que problema hay con no controlar que todo está abierto (política permisiva)?
  - Instalar un software nuevo que abra un puerto determinado, o que no sepamos que determinados paquetes ICMP son peligrosos.
  - Con política denegar, a no ser que lo permitamos explícitamente, el firewall se convierte en un auténtico MURO infranqueable.

Algunas preguntas

- ¿Abrimos/Cerramos todos los puertos?

Algunas preguntas

- ¿Abrimos/Cerramos todos los puertos?
  - ¿Para qué abrir un puerto que nunca se va a utilizar o que no esté en uso actualmente?

Algunas preguntas

- ¿Como podemos ver los puertos abiertos?

Algunas preguntas

- ¿Como podemos ver los puertos abiertos?
  - *netstat –ln* o *netstat –an* o *netstat –puta* | *grep LISTEN*

Algunas preguntas

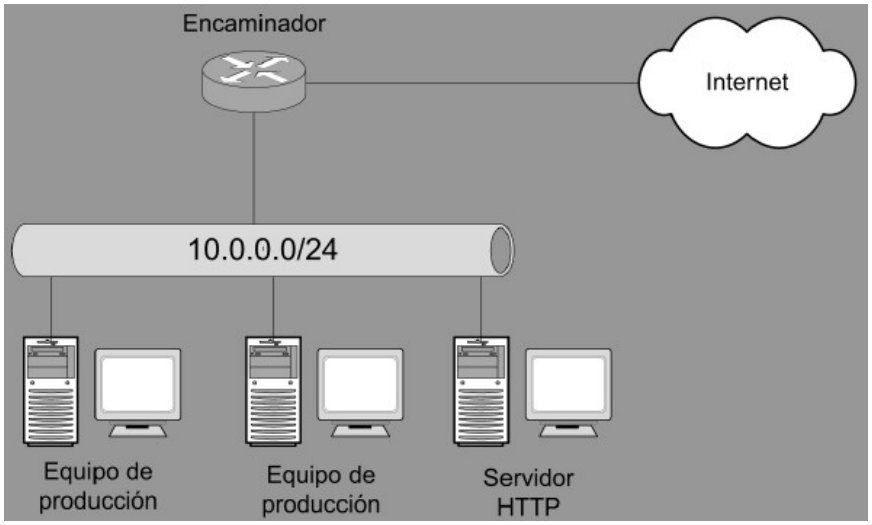
- ¿El orden de las reglas es determinante?

Algunas preguntas

- ¿El orden de las reglas es determinante?
  - Si ponemos reglas muy permisivas entre las primeras del firewall, puede que las siguientes no se apliquen y no sirvan de nada.
  - Hay que tener muy en cuenta el orden en las reglas.

Ejemplo 1

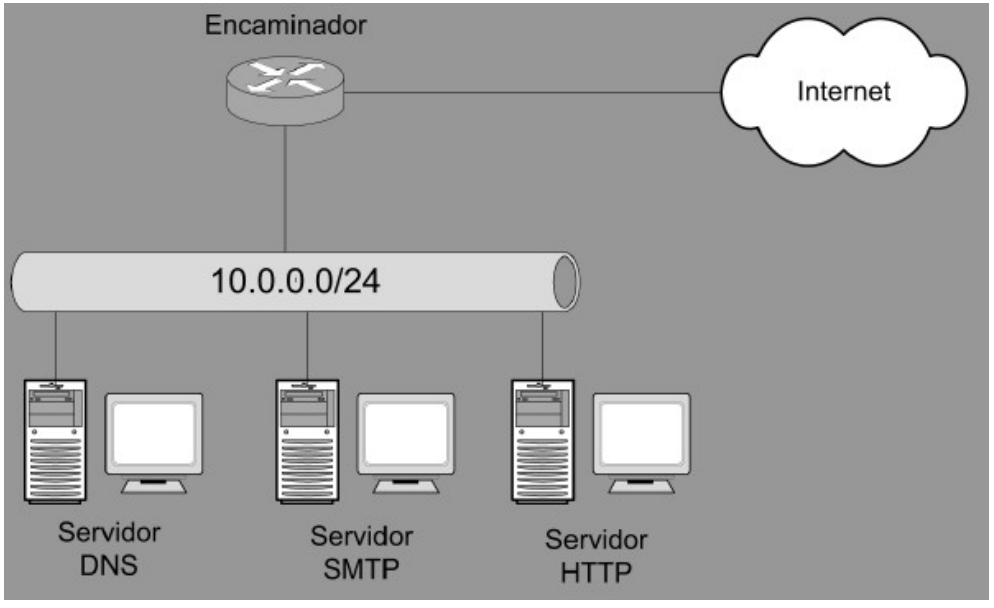
- Reglas:
  - Todos los sistemas de la red interna 10.0.0.0 pueden acceder a cualquier servicio TCP de Internet.
  - El tráfico ICMP sólo está permitido de salida, no de entrada (para evitar la extracción de información mediante este protocolo).
  - Los sistemas externos no se pueden conectar a ningún sistema interno, excepto al servidor de HTTP (10.0.0.1).



Regla	Acción	Origen	Puerto de origen	Destino	Puerto de destino	Indicador	Descripción
1	Permite	10.0.0.0	*	*	*	ICMP	Permite tráfico ICMP de salida
2	Permite	10.0.0.0	*	*	*	TCP	Permite conexiones TCP de salida
3	Permite	*	*	10.0.0.1	80	TCP	Permite conexiones HTTP de entrada
4	Rechaza	*	*	10.0.0.0	*	*	Rechaza cualquier otra conexión a la red interna

Ejemplo 2

- Reglas:
  - Todos los sistemas de la red interna 10.0.0.0 pueden acceder a cualquier servicio TCP de la red Internet, exceptuando HTTP.
  - Se deben de autorizar accesos al servidor de DNS (10.0.0.3).
  - Los sistemas externos no se pueden conectar a ningún sistema interno, excepto al servidor de HTTP (10.0.0.1) y de SMTP (10.0.0.2).
- Realice la tabla de reglas.



Ejemplo 2 – Solución

Regla	Acción	Origen	Puerto de origen	Destino	Puerto de destino	Indicador	Descripción
1	Rechaza	10.0.0.0	*	*	80	TCP	Rechaza cualquier conexión a servidores HTTP
2	Permite	10.0.0.0	*	*	*	TCP	Permite conexiones TCP de salida
3	Permite	*	*	10.0.0.1	80	TCP	Permite conexiones HTTP entrantes
4	Permite	*	*	10.0.0.2	25	TCP	Permite conexiones SMTP entrantes
5	Permite	*	*	10.0.0.3	53	UDP	Permite conexiones DNS entrantes
6	Rechaza	*	*	10.0.0.0	*	*	Rechaza cualquier otra conexión a la red interna

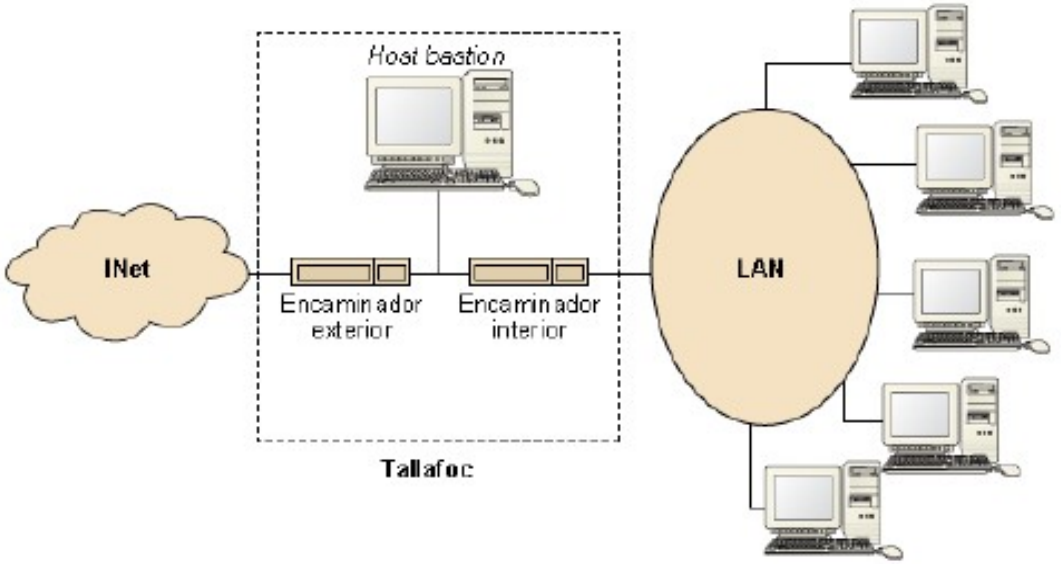
Ventajas y desventajas

- Ventajas:
  - Económica.
  - Rendimiento alto a redes con carga de tráfico elevada.
  - Parte de las políticas de seguridad de cualquier institución plasmando los conceptos de seguridad en el mundo real.
- Desventajas:
  - Muchos de los routers pueden ser vulnerables a ataques.
  - Su capacidad de actuación puede llegar a deteriorarse a causa de la utilización de un filtro excesivamente estricto.
  - Las reglas de filtrado pueden ser muy complejas, y en ocasiones sucede que posibles distracciones en su configuración sean aprovechadas por un atacante.

Zonas desmilitarizadas (DMZ)

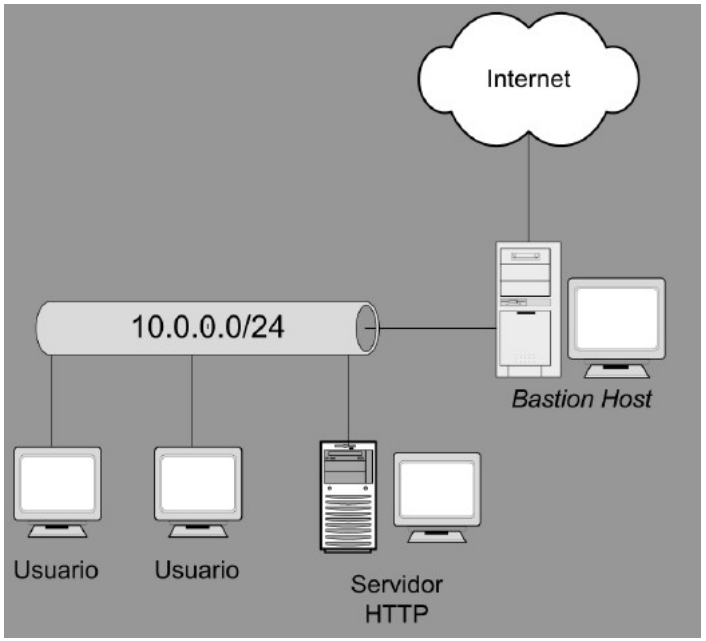
Concepto

- Intenta aislar una red perimetral.
- La seguridad se centra en el host bastión.
- Actualmente se usan dos routers (interior y exterior) conectados a la red perimetral.
- Posibilidad de instalar un servidor de correo que seria visible desde el exterior.
- Un atacante tendría que romper la seguridad de ambos routers para acceder a la red.

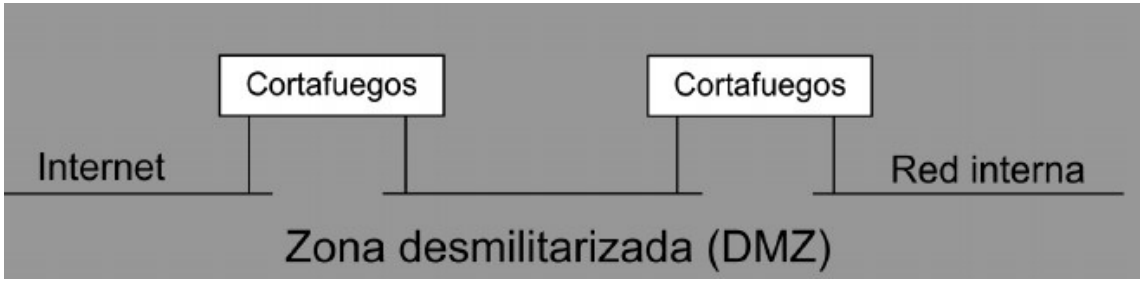


Bastion host Vs. DMZ

- Un *bastion host* es un equipo que ha sido fuertemente protegido para soportar los supuestos ataques desde un lugar hostil (en este caso, Internet) y que actúa como punto de contacto entre el interior y el exterior de una red.

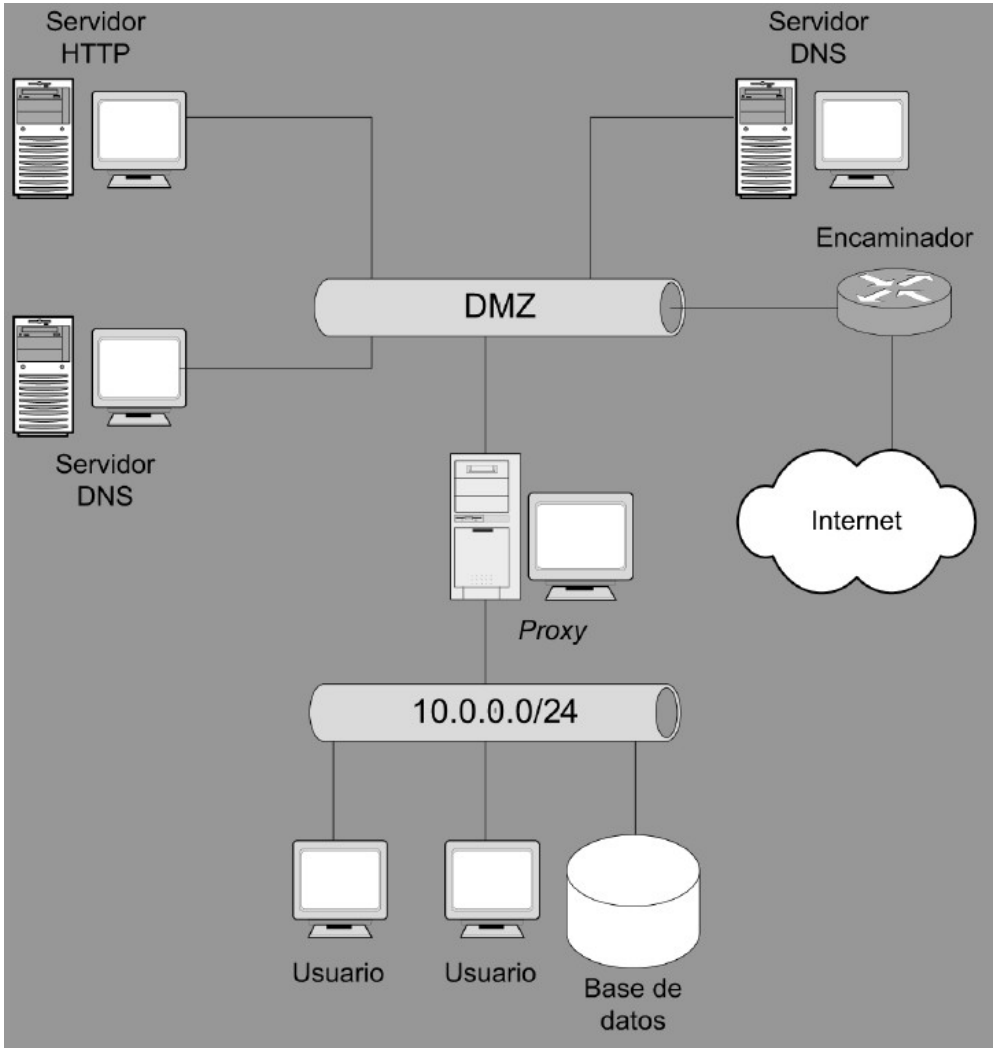


- Un cortafuegos separa el exterior de la red del segmento desmilitarizado (la DMZ) y los servidores que tienen que ser públicos desde el exterior de la red.
- El segundo cortafuegos, que hace de punto de contacto entre la red interna y la zona desmilitarizada, se configurará para que rechace todos los intentos de conexión que vayan llegando desde el exterior.
- Si un atacante consigue introducirse en uno de los servidores de la zona desmilitarizada, será incapaz de atacar **inmediatamente** una estación de trabajo.

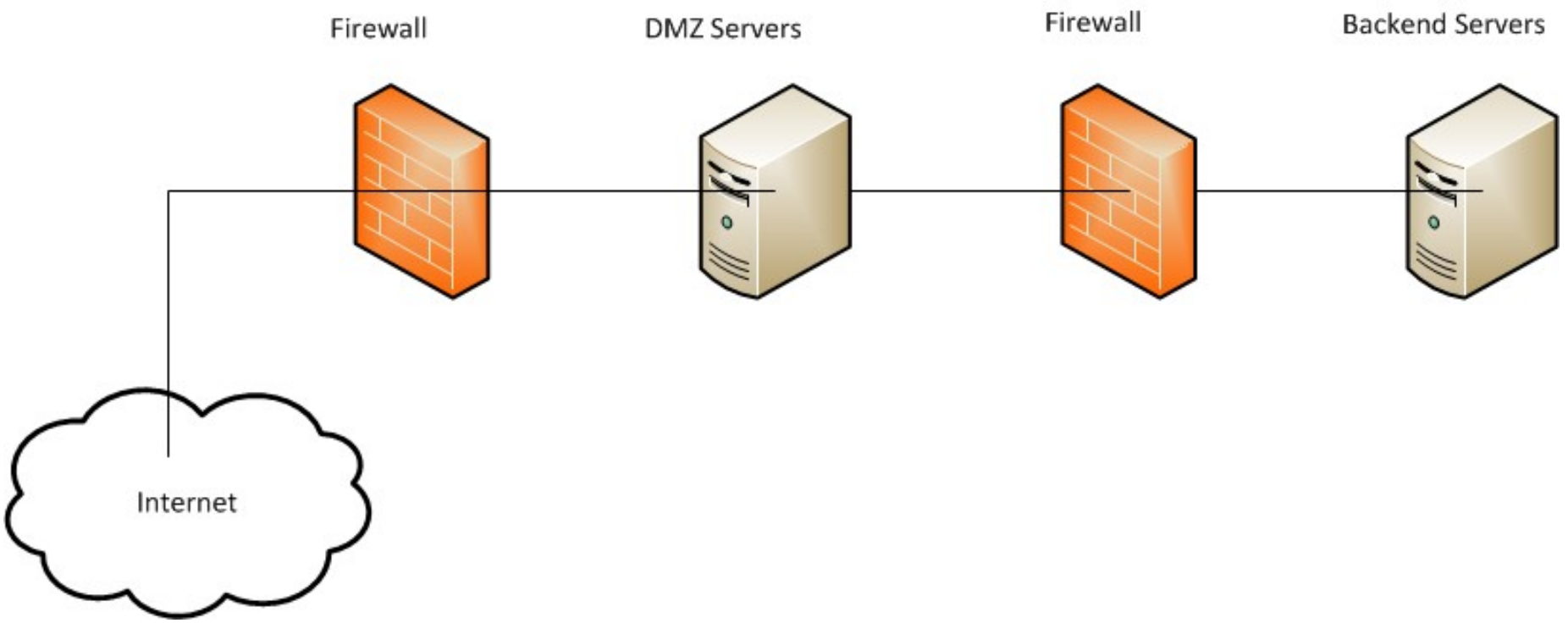


Tecnologías para una DMZ (I)

- En la figura siguiente podemos ver el uso de un encaminador con filtrado de paquetes.
- Adicionalmente, con la utilización de un servidor intermediario para el establecimiento de una zona desmilitarizada.
- ¿Que ocurre exactamente en esta Figura?



Tecnologías para una DMZ (II) – Típica

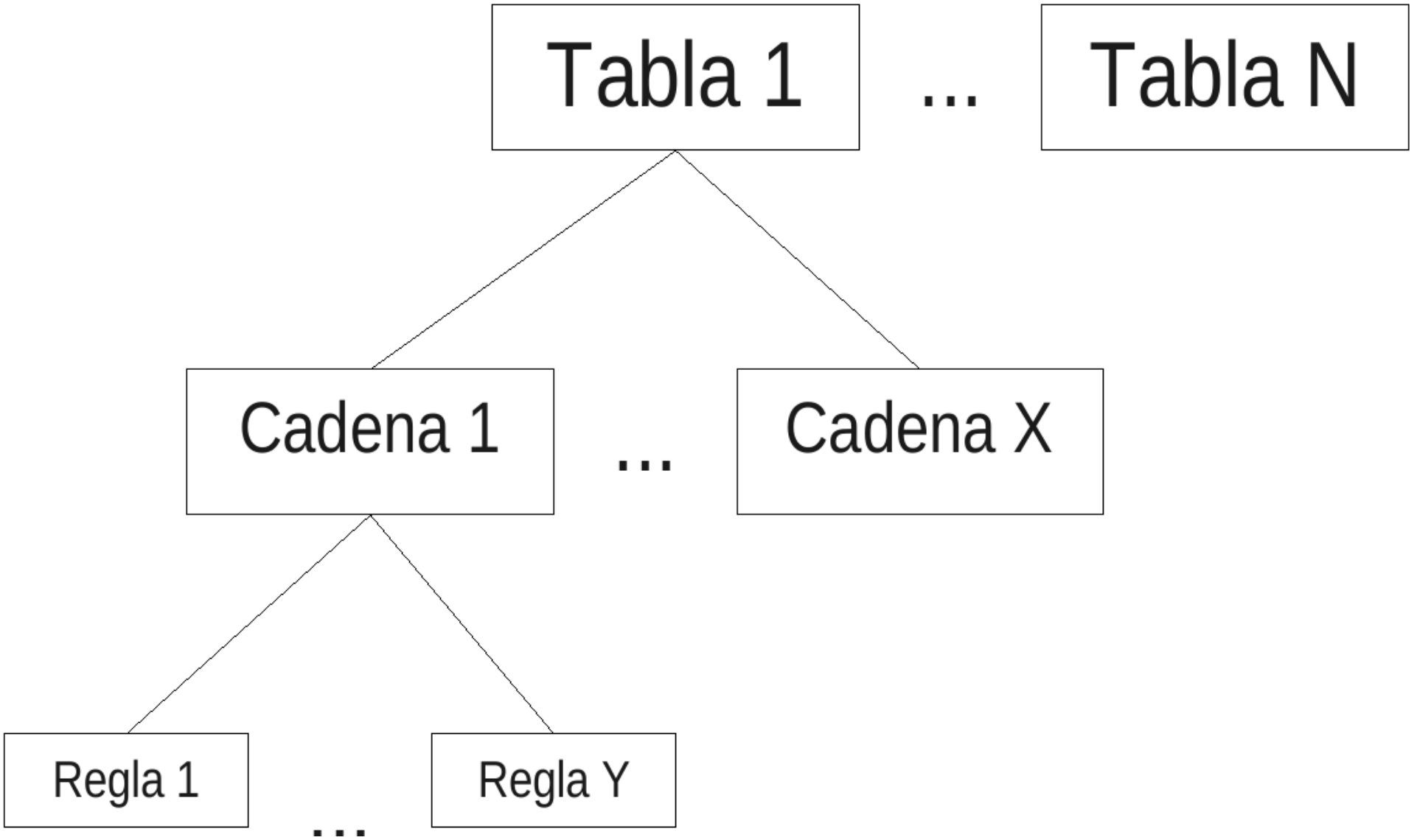


Netfilter: *iptables*

iptables del kernel de Linux

- Netfilter para Linux:
  - http://www.netfilter.org
- Kernel de Linux (versión):
  - <2.4: ***ipchains***.
    - Solo filtrado y enmascaramiento.
    - No permiten realizar seguimiento de conexiones, etc.
  - >=2.4: ***iptables***.
    - Filtrado y enmascaramiento (NAT).
    - Seguimiento de conexiones.
    - Modificación de campos de cabecera.
    - Etc.

Estructura de *iptables*



Tablas de *iptables* (I)

- ***Filter***:
  - Tabla por defecto.
  - Normas propias de filtrado → Filtra los paquetes de la red.
- ***NAT***:
  - Altera las direcciones de origen y/o destino de los paquetes → Traslación de direcciones.
- *Mangle*:
  - Especifica algunas opciones de control de paquetes y cómo gestionarlos.
  - 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 Mandatory Access Control que permitan filtrar paquetes.

Cadenas de *iptables* (I)

- *Filter*:
  - ***INPUT***: Paquetes destinados a un proceso local → De entrada.
  - ***OUTPUT***: Se aplica a los paquetes generados de forma local por un proceso y que van a ser enviados por la red → De salida.
  - ***FORWARD***: Paquetes recibidos por un dispositivo de red y que deben ser reenviados a la red sin ser procesados de forma local.
- *NAT*:
  - ***POSTROUTING:*** para Source NAT, según los paquetes salen, cambia la dirección de origen de las conexiones a alguna diferente.
  - ***PREROUTING***: para Destination NAT, según los paquetes entran. Las reglas han de decidir si hay que hacer algún enrutamiento del paquete y cuál será la dirección de destino.
  - *OUTPUT*: para Destination NAT con los paquetes generados en la propia máquina.

Cadenas de *iptables* (II)

- *Mangle*:
  - *PREROUTING*: Se aplica a los paquetes recibidos por un dispositivo de red antes de ser enrutados.
  - *INPUT*: Se aplica a los paquetes destinados a un proceso local.
  - *OUTPUT*: Se aplica a los paquetes generados de forma local por un proceso antes de ser enrutados.
  - *FORWARD*: Se aplica a los paquetes que son reenviados a través de dos dispositivos de red del ordenador sin la intervención de ningún proceso local.
  - *POSTROUTING*: Se aplica a los paquetes antes de salir a la red.
- *Raw*:
  - *PREROUTING*: Se aplica a los paquetes recibidos por cualquier dispositivo de red.
  - *OUTPUT*: Se aplica a los paquetes generados por un proceso local.
- *Security*:
  - *INPUT*: Paquetes destinados a un proceso local.
  - *OUTPUT*: Paquetes generados localmente por un proceso y que van a ser enviados por la red.
  - *FORWARD*: Paquetes recibidos por un dispositivo de red y que deben ser reenviados a la red sin ser procesados de forma local.

Comandos de *iptables*

- 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*>: permite indicar sobre qué tabla se ejecuta el comando (por defecto tabla *filter*).
  - <*comando*>: indica la acción a realizar, como puede ser añadir una regla, borrar una regla, etc..
  - <*nombre de la cadena*>: cadena de la tabla sobre la que se ejecuta la acción.
  - <*parámetro X*> <*opción X*>: parámetros y opciones siguientes definen la regla, que acción debe realizar, etc..
- La complejidad de los comandos depende de su objetivo.

Reglas *iptables* (I)

- Sintaxis:
  - *iptables -A chain -j target*
    - *chain: input | output | forward*
    - *target:* Destino que se le dará al paquete.
    - -A: opción que se añadirá.
- Opciones:
  - *-j*: Qué hacer con los paquetes (*accept, reject, drop*).
    - *Reject:* informa al emisario
    - *Drop:* no envía respuesta al emisario.
  - *log:* envía el paquete al sistema de registros.
  - *-I*: Indica posición.
  - *-s*: IP fuente.
  - *-D*: Borra una regla. Puede ser el número de regla o la regla exacta.
  - *-P*: Permiten definir una regla por defecto.

Comando 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.

Comando iptables (III)

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.

Reglas *iptables* (II) – *filter*

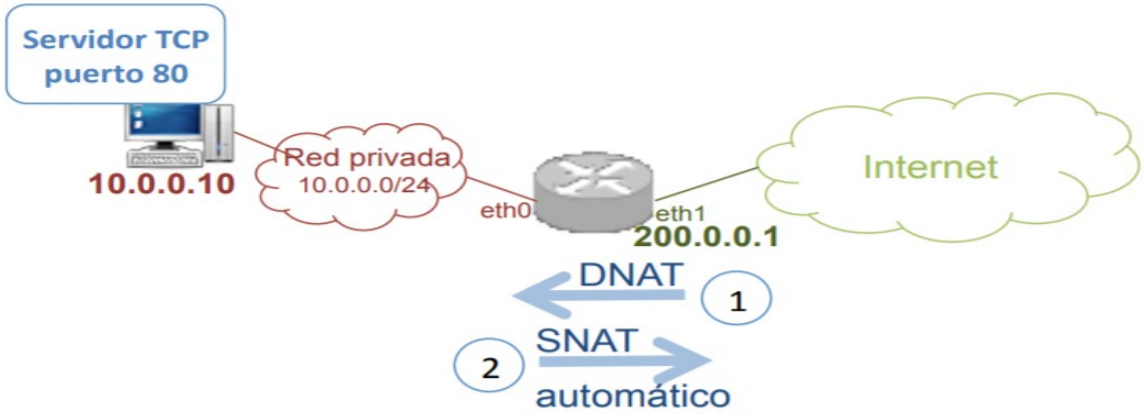
- Ejemplos:
  - *iptables -A INPUT -s 10.0.0.0/8 -j ACCEPT*
  - *iptables -D INPUT 1*
  - *iptables -D INPUT -s 10.0.0.0/8 -j ACCEPT*
  - *iptables -P INPUT DENY*
  - *iptables -P OUTPUT REJECT*
  - *iptables -P FORWARD REJECT*

Reglas *iptables* (III) – *NAT*

- Borrar reglas y reiniciar contadores
  - *iptables -t nat -F #Borra las reglas*
  - *iptables -t nat -Z #Reinicia los contadores*
- Si suponemos que la IP de nuestro ordenador es *192.168.0.1*, podemos reenviar el tráfico destinado al puerto TCP *80* de nuestro ordenador a otro ordenador, de IP *192.168.0.2*.
  - *iptables -t nat -A PREROUTING -p tcp -d 192.168.0.1 --dport 80 -j DNAT --to-destination 192.168.0.2:80*

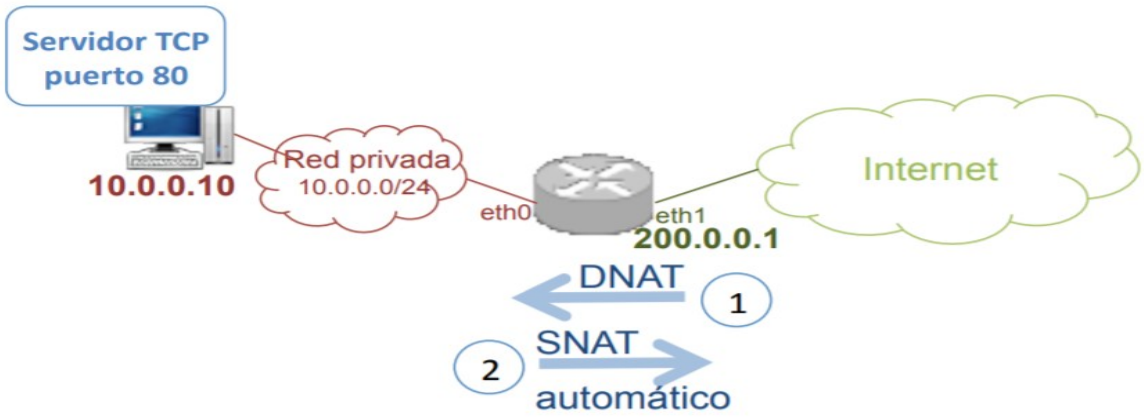
Reglas *iptables* (IV) – *NAT*

- Modificar la dirección IP origen de los paquetes al salir de una red privada (*10.0.0.0/24*) a través de la interfaz de salida (*eth1*) de un router NAT. Todos los paquetes llevarán la dirección IP pública del router NAT (*200.0.0.1*)
  - *Indicar Regla*
- Modificar la dirección IP y puerto destino de los paquetes al entrar dentro de una red privada (*10.0.0.0/24*). Los segmentos van dirigidos inicialmente a la dirección IP del router NAT (*200.0.0.1*) y puerto *8080*. Se modificará su dirección IP destino a *10.0.0.10* y puerto destino *80*.
  - *Indicar Regla*



Reglas *iptables* (IV) – *NAT*

- Modificar la dirección IP origen de los paquetes al salir de una red privada (*10.0.0.0/24*) a través de la interfaz de salida (*eth1*) de un router NAT. Todos los paquetes llevarán la dirección IP pública del router NAT (*200.0.0.1*)
  - *iptables -t nat -A POSTROUTING -s 10.0.0.0/24 -o eth1 -j SNAT --to-source 200.0.0.1*
- Modificar la dirección IP y puerto destino de los paquetes al entrar dentro de una red privada (*10.0.0.0/24*). Los segmentos van dirigidos inicialmente a la dirección IP del router NAT (*200.0.0.1*) y puerto *8080*. Se modificará su dirección IP destino a *10.0.0.10* y puerto destino *80*.
  - *iptables -t nat -A PREROUTING -p tcp -i eth0 -d 200.0.0.1 --dport 8080 -j DNAT --to-destination 10.0.0.10:80*



Reglas *iptables* (V)

- ¿Que hacen la siguientes reglas?
  - *iptables -A INPUT -s 10.0.0.0/8 -d 192.168.1.2 -j DROP*
  - *iptables -A INPUT -p tcp --dport 113 -j REJECT --reject-with tcp-reset*
  - *iptables -A INPUT -p tcp --dport 113 -s 10.0.0.0/8 -j ACCEPT*

Reglas *iptables* (V)

1) *iptables -A INPUT -s 10.0.0.0/8 -d 192.168.1.2 -j DROP*

2) *iptables -A INPUT -p tcp --dport 113 -j REJECT --reject-with tcp-reset*

3) *iptables -A INPUT -p tcp --dport 113 -s 10.0.0.0/8 -j ACCEPT*

1) Perdemos los paquetes que vengan de 10.x.x.x con destino a 192.168.1.2.

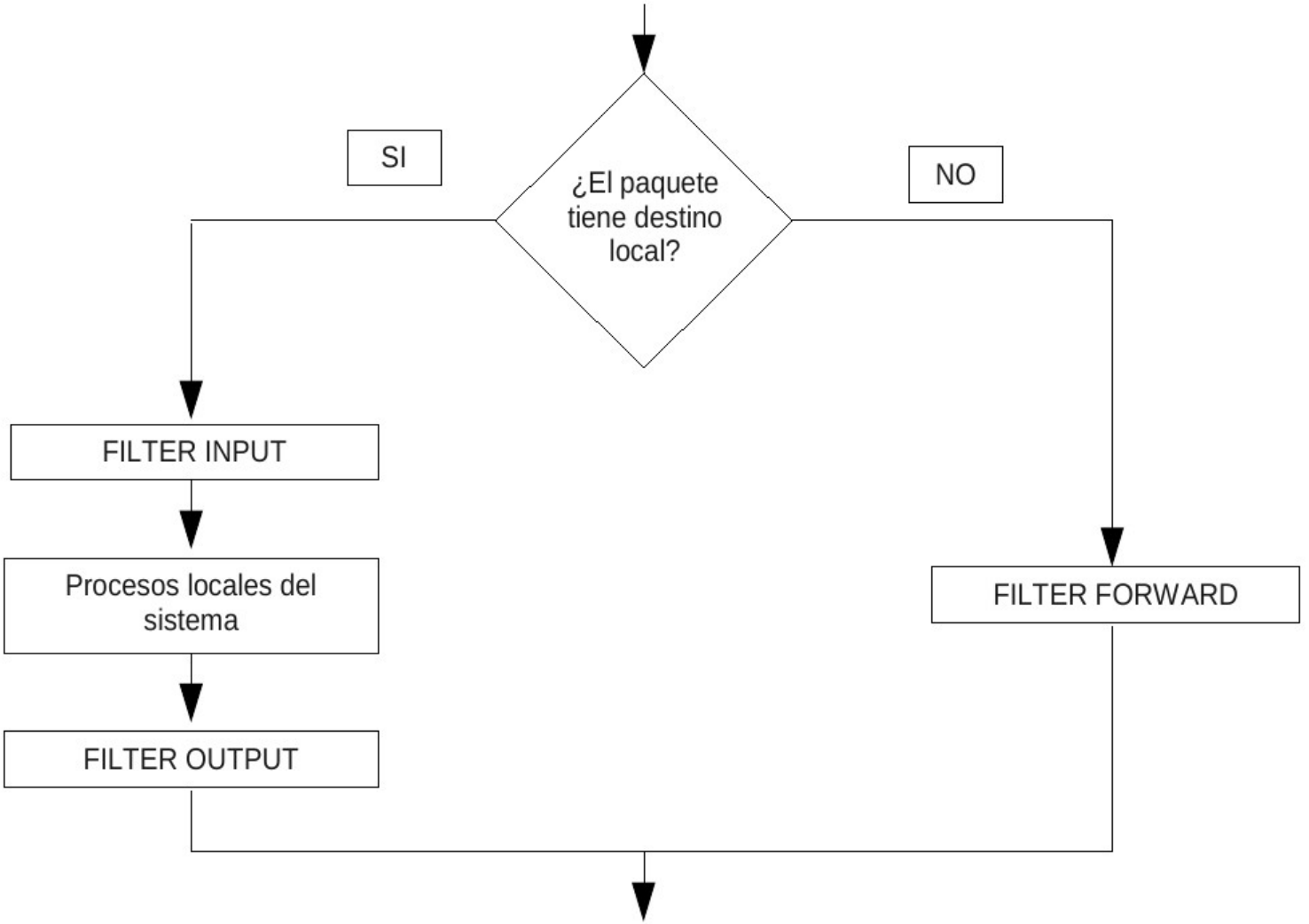
2) Rechazamos los paquetes TCP con destino al puerto 113, emitiendo una respuesta de tipo *tcp-reset*.

3) Los mismos paquetes que en 2) pero que provengan de 10.x.x.x serán aceptados.

Ejemplos

- *iptables -t nat -L -v*
  - Muestra los datos de las reglas definidas en todas las cadenas de la tabla *NAT*.
- *iptables -I INPUT 3 -p tcp -s 147.156.0.0/16 -j ACCEPT*
  - Que añade la regla especificada (*-p tcp -s 147.156.0.0/16 -j ACCEPT*) en la tercera posición de la cadena *INPUT* de la tabla *filter*, pues al no especificar ninguna tabla se utiliza la tabla por defecto.

La tabla *filter*



Recordamos

- 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.

Expresiones 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* u otros como uno de los valores *icmp-net-unreachable, icmp-host-unreachable*, etc
- *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)

- *-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.
- *-j*: Acción a realizar si el paquete coincide con la regla.

  iptables -A INPUT -j ACCEPT

  - Aañade, al final de la cadena INPUT, una regla que indica que se acepten todos los paquetes que lleguen a esta regla.

  iptables -A INPUT -j MI_CADENA

  - Se añada, al final de la cadena INPUT, una regla que indica que los paquetes que lleguen a esta regla sean enviados para su análisis a las reglas de la cadena MI_CADENA.
  - Si una cadena no tiene un parámetro -j, el paquete pasa a ser comprobado por la siguiente regla, siendo el contador de está regla incrementado en una unidad, pues el paquete cumplió la regla.

Parámetros de especificación de reglas (II)

- *-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 ó 192.168.0.0/16

  iptables -A INPUT -d 192.168.0.0/24 -j ACCEPT

    - Añade una regla, al final de la cadena INPUT, indicando que se admitan todos los paquetes de entrada que tengan como destino la subred 192.168.0.0/24.
- *-f*: Aplicar la regla a los paquetes fragmentados.

  iptables -A OUTPUT -f -j DROP

    - Indica que se añada, al final de las reglas de la cadena OUTPUT, la regla que indica que se deniegue la salida de todo paquete que este fragmentado.

Parámetros de especificación de reglas (III)

- *-i*: identifica el dispositivo de red de entrada, como *eth0*, al que se debe aplicar la regla.
  - Solo puede aplicarse a las cadenas *INPUT* y *FORWARD*.

    iptables -I INPUT 1 -i lo -j ACCEPT

    - Indica que se añada, en primer lugar de las reglas de la cadena INPUT, una regla que especifica que todo el tráfico entrante del interfaz de loopback (lo) sea aceptado.
  - El carácter "+" sustituye a un carácter y permite indicar un conjunto de dispositivos: *eth+* -> *eth0, eth1, etc.*
- *-o*: identifica el dispositivo de red de salida para una regla
  - Solo puede aplicarse a las cadenas *OUTPUT* y *FORWARD*.
  - Opciones, etc., idénticas al parámetro *-i*.

Parámetros de especificación de reglas (IV)

- *-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*.

    iptables -A OUTPUT -p icmp -j ACCEPT
    - Indica que se añada, al final de la cadena OUTPUT, una regla que indica que si el protocolo de transporte es ICMP se acepte la salida del paquete.
  - Precedido el protocolo de ! se aplica al resto de protocolos.
- *-s*: Selecciona el ordenador destino del paquete.
  - Mismas opciones que el parámetro *-d*.

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*>

    iptables -I INPUT 3 -p tcp --dport 80 -j ACCEPT
    - Indica que si el protocolo de transporte es TCP y el puerto destino es el 80 (www) el paquete sea aceptado.
  - 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 TCP (II)

- *--syn*: Paquete que inicia una conexión.

  iptables -A INPUT -p tcp --syn -j MI_CADENA

  - Indica que si un paquete posee protocolo de transporte TCP y establece una conexión, se pase su análisis a las reglas de la cadena indicada por MI_CADENA.

  – Si se precede de ! indica paquete que no inician una conexión.

  iptables -A INPUT -p tcp ! --syn -j ACCEPT

  - indica que si un paquete posee protocolo de transporte TCP y no establece una conexión, se acepte su salida.

Extensiones de TCP (III)

- *--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, sino se encuentra 0.

    iptables -A INPUT -p tcp SYN,RST,ACK SYN -j MI_CADENA

    - Se añada al final de la cadena INPUT una regla que indica que si un paquete posee protocolo de transporte TCP y de los bits de bandera SYN está activo y RST y ACK inactivo, el paquete se pase para su análisis a las reglas de la cadena indica por MI_CADENA.
  - 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.
  - Igual que TCP

    iptables -I INPUT 3 -p udp --dport 1024: -j ACCEPT

    - Indica que si el protocolo de transporte es UDP y el puerto destino es mayor de 1024 se acepte el paquete.
- *--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 *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 *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 *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 }.

Ejemplos

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 / /tabla filer*

  :INPUT ACCEPT [0:0] / / indica el valor inicial que tienen los contadores de paquetes y bytes de esa regla al inicializar las iptables

  :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*

Ejemplo en 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

Ejemplo en cliente Linux

- La primera regla permite (-j ACCEPT) que sea aceptado todo el tráfico de entrada proveniente de la red de loopback (-i lo).
- La segunda regla indica que el se acepte (-j ACCEPT) el tráfico de entrada udp (-p udp) con origen en cualquier ordenador (-s 0/0) y puertos de origen 67 o 68 (--sport 67:68) y con destino cualquier ordenador (-d 0/0) y puertos de destino 67 o 68 (--dport 67:68).
- La tercera regla indica que todo el tráfico de entrada udp (-p udp) sea rechazado (-j REJECT).
- Por último, la cuarta regla indica que todo el tráfico de entrada tcp (-p tcp) que quiera iniciar una conexión tcp (--syn), sea rechazado (-j REJECT).
- El resto de tráfico de red es permitido, pues la política por defecto es aceptar (ACCEPT) el tráfico de red entrante.

Ejemplo en 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
```

Ejemplo en servidor Linux

- La primera regla permite (-j ACCEPT), igual que en el ejemplo anterior, que sea aceptado todo el tráfico de entrada proveniente de la red de loopback (-i lo).
- Las dos siguientes reglas permiten que el ordenador funcione como servidor de nombres (DNS), pues habilitan (-j ACCEPT) la recepción de paquetes udp (-p udp) que tengan como destino el puerto 53 (--dport 53) o como origen el puerto 53 (--sport 53).
- La siguiente regla indica que se rechace todo el tráfico UDP que no haya cumplido alguna de las reglas anteriores.
- Las siguientes cinco reglas permiten que cualquier dispositivo de red 23, pues este no se encuentra especificado, admita (-j ACCEPT) los paquetes tcp (-p tcp) que quieran iniciar una conexión (--syn) con los puertos especificados como destino (--dport 22, --dport 25, --dport 80, --dport 110 y --dport 995).
- Por último, la última regla indica que se rechace todo el tráfico TCP que quiera establecer una conexión y que no haya cumplido alguna de las reglas anteriores.
- El resto de tráfico de red, igual que en el ejemplo anterior, es permitido, pues la política por defecto es aceptar (ACCEPT) el tráfico de red entrante.

Ejercicio

- Aceptar el servicio SSH para un host únicamente.

Ejercicio

:FORWARD DROP [0:0]

:INPUT DROP [0:0]

:OUTPUT ACCEPT [0:0]

-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

-A INPUT -i lo -j ACCEPT

-A INPUT -s 1.8.9.7 -p tcp --dport 22 -j ACCEPT

COMMIT

Ejemplo servidor Linux con puerta de acceso a 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
```

Ejemplo servidor Linux con puerta de acceso a subred

- En primer lugar, podemos ver como este ejemplo implica la modificación de dos tablas, las tablas nat y filter.
- En la tabla nat, además de establecer como política por defecto de sus tres cadenas de reglas el que todo sea aceptado, añadimos una regla:
  - La regla añadida indica que a todo el tráfico que vaya a ser enviado (POSTROUTING) por el modem (-o ppp0), se le modifique la dirección IP y se le asigne la dirección IP que le ha sido proporcionada al modem (-j MASQUERADE).
- Por su parte, en la tabla filter se establecen las políticas por defecto en sus cadenas de reglas y se añaden las siguientes reglas:
  - En las dos primeras reglas permiten (-j ACCEPT), que sea aceptado todo el tráfico de entrada proveniente de la red de loopback (-i lo) y de la Intranet (-i eth0), pues consideramos confiable toda nuestra Intranet, por lo que no limitamos su tráfico.
  - En la siguiente regla aceptamos (-j ACCEPT) todo el tráfico udp (-p udp) proveniente del puerto 53 (--sport 53). De esta forma permitimos la resolución de nombres mediante el uso del servicio de DNS.
  - Las dos reglas siguientes indican, al igual que en los anteriores ejemplos, que se rechace todo el tráfico UDP y que se rechace el tráfico TCP que intente establecer una conexión.

Ejemplo servidor Linux con puerta de acceso a subred

- Aunque aparentemente el cortafuegos ya esta configurado correctamente, esto no es así, pues debemos todavía configurar las reglas de la tabla filter que afectan al reenvió de paquetes, esto es, las reglas que afectan a la cadena FORWARD. Las tres reglas establecidas son las siguientes:
  - La primera regla indica que se acepten (-j ACCEPT) todos los paquetes que tengan como origen el modem (-i ppp0) y como destino la Intranet (-o eth0), siempre que su estado (-m state --state ESTABLISHED,RELATED) corresponda a una conexión ya establecida (ESTABLISHED) o a una nueva conexión cuya apertura esta relacionada con otra conexión ya existente (RELATED).
  - La segunda regla indica que se acepten (-j ACCEPT) los paquetes que tengan como origen la Intranet (-i eth0) y destino Internet, esto es, el modem (-o ppp0).
  - Por último, la tercera regla indica que todo lo que no cumpla las reglas anteriores de reenvió sea rechazado (-j DROP).