Práctica 8. Seguridad básica: Firewalls y NAT

Objetivos de aprendizaje

- Ser capaz de construir reglas de filtrado en base a protocolos, direcciones y puertos.
- Ser capaz de construir reglas de filtrado para imponer una limitación temporal al tráfico.
- Conocer las acciones que se pueden llevar a cabo sobre el tráfico que casa con una regla.
- Saber cómo prevenir ataques desde la red externa utilizando filtrado de paquetes mediante reglas de control de acceso.
- Ser capaz de publicar algunos servicios comunes en Internet de forma segura.
- Entender el funcionamiento de NAT.

Introducción al firewall de RouterOS

Un firewall es un dispositivo hardware o un software que filtra paquetes que pasan a través de él.

En RouterOS el firewall se define a través de lo que se conoce como cadenas o `chain`. Una cadena es una secuencia de reglas consistentes en un patrón y una acción. El patrón es un criterio que permite decidir si a un paquete se le aplica la acción. En general las acciones consisten en dejar pasar el paquete (`accept`) o borrar el paquete sin más (`drop`). Además de estas dos acciones que están implementadas en todo firewall, RouterOS provee otras acciones como `reject`, `log` o `jump`, entre otras. La acción `reject` rechaza un paquete pero avisa al emisor del mismo de que el paquete ha sido eliminado. La acción `log` deja un rastro en el fichero de registro del sistema y pasa a la siguiente regla de la secuencia. La acción `jump` permite saltar de la cadena actual a otra cadena con el nombre especificado en el parámetro `jump-target`.

Veamos un ejemplo de una cadena filtrado en RouterOS:

```
/ip firewall filter
add chain=forward src-address=127.0.0.0/8 action=drop
add chain=forward protocol=tcp dst-port=111 action=drop
add chain=forward src-address=192.168.0.0/24 action=accept
add chain=forward action=accept
```

El nombre de la cadena es `forward`. Cada linea tipo `add` añade una regla a la cadena. El patrón se especifica mediante valores de determinados campos de las cabeceras TCP/IP y la acción correspondiente esa regla en el parámetro `action`. Cuando un paquete cruza el router, se analizan las reglas secuencialmente y si casa con el patrón se ejecuta la acción finalizando el análisis (excepto en la acción log). Si se llega al final de la cadena sin haber casado con ninguna regla, el paquete SE DEJA PASAR.

Para conocer los distintos criterios que se pueden utilizar para especificar los patrones se recomienda consultar el manual de RouterOS: `https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/Filter`

Los criterios más destacados son:

- `src-address` y `dst-address` que indican la dirección fuente y destino respectivamente, pudiéndose especificar un bloque de direcciones o un rango.

- `src-port` y `dst-port` que indican los puertos de origen y destino.
- `protocol` que indica el protocolo (tcp, udp, icmp, ...) Nótese que cuando se indican restricciones sobre números de puerto hay que indicar el protocolo tcp o udp.
- `in-interface` y `out-interface`: Interfaces por las que los paquetes entran o salen del router, respectivamente.
- `icmp-options`: Casa con los campos de tipo y código de un paquete ICMP. Revise `https://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml` para ver los tipos y códigos de los distintos paquetes ICMP. Es necesario que `protocol` sea ICMP cuando se define el patrón.

Cadenas por defecto

Aunque se pueden crear cadenas nuevas, en RouterOS existen tres cadenas por defecto que no se pueden borrar:

- `input`: Procesa paquetes entrantes destinados a alguna de las direcciones del router.
- `forward`: Procesa paquetes no destinados al router ni originados en el router.
- `output`: Procesa paquetes salientes originados en el router.

Las cadenas por defecto son el punto de inicio del proceso de firewalling, por lo que deberá haber al menos una regla en alguna de ellas para que el firewall funcione.

Creación de cadenas y saltos a otras cadenas

En RouterOS es posible crear cadenas adicionales con el fin de mejorar la eficiencia en el procesamiento y de organizar las reglas de manera lógica.

Por ejemplo, para crear una cadena con nombre `trafico_tcp` escribimos las reglas correspondientes pero en el atributo `chain` ponemos el nombre de la nueva cadena:

```
add chain=trafico_tcp protocol=tcp dst-port=69 action=drop
add chain=trafico_tcp protocol=tcp dst-port=111 action=accept
add chain=trafico_tcp protocol=tcp dst-port=135 action=accept
```

Ahora desde la cadena `forward` podemos añadir una regla que salte a la cadena `traficp_tcp`:

```
 add chain=forward protocol=tcp action=jump jump-target=trafico_tcp
```

Para ello utilizamos la acción `jump` y el atributo `jump-target` que indica el nombre de la cadena la que se va a saltar.

Alterar el orden de las reglas de una cadena

Como se ha indicado anteriormente, las reglas se ejecutan en el orden en el que se ejecutan dentro de una cadena. En algunos casos, puede ser necesario modificar el orden de las reglas de la cadena. Supongamos que tenemos implementadas las siguientes reglas para la cadena `forward`:

```
 [admin@MikroTik] /ip firewall filter> print chain=forward
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=forward action=drop src-address=192.168.0.0/16 in-interface=ether2

 1    chain=forward action=drop protocol=icmp icmp-options=8:0

 2    chain=forward action=accept protocol=icmp icmp-options=3:0

 3    chain=forward action=accept protocol=icmp icmp-options=4:0
```

```
 4    chain=forward action=accept protocol=icmp icmp-options=11:0

 5    chain=forward action=drop
```

Y ahora, deseamos añadir una regla que faltaba, pero en la primera posición. Podemos hacerlo de la siguiente manera:

```
[admin@MikroTik] /ip firewall filter>
   add chain=forward src-address=10.0.0.0/8 action=drop in-interface=ether2 place-before=0
```

El atributo `place-before` permite indicar el elemento de la lista de reglas ante el cual se va a insertar la nueva regla. En este caso, antes de la regla 0. Volviendo a imprimir la lista de reglas tenemos:

```
[admin@MikroTik] /ip firewall filter> print chain=forward
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=forward action=drop src-address=10.0.0.0/8 in-interface=ether2

 1    chain=forward action=drop src-address=192.168.0.0/16 in-interface=ether2

 2    chain=forward action=drop protocol=icmp icmp-options=8:0

 3    chain=forward action=accept protocol=icmp icmp-options=3:0

 4    chain=forward action=accept protocol=icmp icmp-options=4:0

 5    chain=forward action=accept protocol=icmp icmp-options=11:0

 6    chain=forward action=drop
```

Otra opción es mover la reglas mediante el comando `move`. Por ejemplo:

```
[admin@MikroTik] /ip firewall filter> move 0 2
```

Mueve la regla 0 y la coloca en el lugar de la regla con posición 2, desplazando esta regla y las restantes hacia abajo.

```
[admin@MikroTik] /ip firewall filter> print chain=forward
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=forward action=drop src-address=192.168.0.0/16 in-interface=ether2

 1    chain=forward action=drop src-address=10.0.0.0/8 in-interface=ether2

 2    chain=forward action=drop protocol=icmp icmp-options=8:0

 3    chain=forward action=accept protocol=icmp icmp-options=3:0

 4    chain=forward action=accept protocol=icmp icmp-options=4:0

 5    chain=forward action=accept protocol=icmp icmp-options=11:0

 6    chain=forward action=drop
[admin@MikroTik] /ip firewall filter>
```

Aspectos a tener en cuenta

- Todas las cadenas terminan con una acción implícita de `accept`, es decir, si ninguna de las reglas es aplicable el paquete se deja pasar.

- Hay que recordar que las reglas se aplican secuencialmente según el número de regla asignado en el momento de crearlas. Cuando una regla casa con el paquete, se aplica esta regla y termina el proceso. Por eso las reglas más restrictivas deben ir al principio.
- Cuando hay dos reglas cuyas condiciones no se solapan, es conveniente colocar primero la regla que se aplicará al mayor número de paquetes. Tenga en cuenta que cada vez que un paquete cruza la interfaz habrá que revisar las reglas. Si la regla aplicada al paquete está muy abajo en la lista, el tiempo de procesado será mayor.
- Recuerde que los protocolos de enrutamiento envían paquetes de actualización de la red. Si hay algún protocolo de enrutamiento operativo en la red recuerde dejar pasar el tráfico correspondiente.

Protección de la frontera con Internet: DMZ

En esta práctica se revisarán algunas medidas de seguridad que es conveniente aplicar a los routers frontera (aquellos que separan a nuestra organización de Internet), aunque algunas de ellas podrían aplicarse perfectamente en la red interna. Un router frontera está expuesto a Internet y, consecuentemente, a un importante grupo de potenciales atacantes.

Veamos algunos peligros potenciales:

- *Sniffing* o *snooping*: Gran parte de las comunicaciones de red sigue siendo en texto claro. Esto permite que un atacante que comprometa una red pueda escuchar el tráfico y obtener información privada. Este problema se resuelve mediante criptografía.
- Modificación de datos: Una vez que un atacante ha sido capaz de espiar los datos podría modificarlos. Se podrían alterar paquetes sin el conocimiento de emisor o el receptor. La solución a este problema son las firmas digitales.
- Spoofing: Se trata de suplantar la identidad de un usuario o una máquina. En esta práctica se tratará el problema del *spoofing* de IP. Un atacante podría esconderse utilizando direcciones IP aparentemente válidas para lograr sus propósitos.
- Ataques basados en contraseña: La mayoría de los sistemas operativos y dispositivos de red están protegidos mediante contraseñas. Si un atacante averigua la contraseña de un sistema o dispositivo de red podría modificar la configuración y comprometer el sistema.
- Denegación del servicio: Este tipo de ataque consiste en bloquear un servicio, como un servidor web, a base de peticiones masivas hasta que el servidor no es capaz de atender más peticiones. Una contramedida es limitar la cantidad de peticiones que un mismo usuario puede realizar.
- Hombre en el medio: Ocurre cuando alguien monitorea activamente la comunicación entre un emisor y un receptor, pudiendo incluso modificar la información.

Redes basadas en DMZ

En el momento de diseñar la topología es conveniente agrupar aquellos hosts con iguales requerimientos de seguridad en la misma red, de modo que resulte más sencillo construir las reglas de filtrado y el número de éstas sea menor. En el caso de la red interna, intervienen también otros factores como la distribución geográfica. Sin embargo, respecto de los servidores externos y del router externo es común aplicar un diseño basado en DMZ (*Demilitarized Zone*). Una DMZ es una subred física o lógica que contiene los servicios de red que la organización ofrece hacia una red poco confiable como Internet. La DMZ supone un nivel adicional de seguridad para la red interna, puesto que un atacante sólo tendrá acceso desde el exterior a los equipos situados en la DMZ. Los servicios que típicamente se sitúan en una DMZ son: Web, FTP o correo electrónico.

Generalmente, se utilizan dos diseños para crear una DMZ:

- Un firewall: Se utiliza un router con capacidad de firewalling con tres interfaces de red. La red externa es la red interconexión con el ISP que se conecta a la primera interfaz. La red interna se conecta a la segunda interfaz, mientras que la DMZ queda conectada a la tercera.

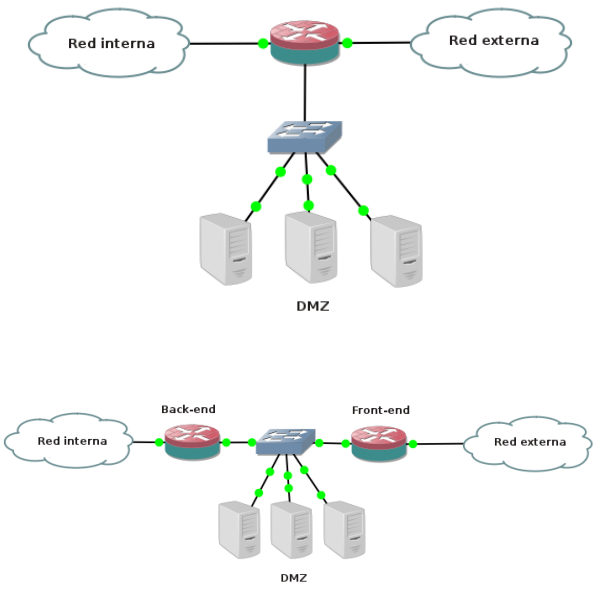

Figura 1: Diseños basados en DMZ: Arriba: Un firewall, Abajo: Firewall dual.

- Firewall dual: Es un diseño más seguro, aunque más caro. Consiste en situar un primer firewall entre la red externa y la DMZ. Y luego, situar un segundo firewall entre la DMZ y la red interna.

En esta práctica se revisarán los conjuntos de reglas que debemos crear para evitar distintas situaciones de peligro.

Contramedidas para el IP spoofing

El ataque de IP spoofing trata se suplantar la identidad de un host utilizando su dirección IP como dirección de origen. Para reducir el riesgo de que un atacante utilice direcciones no lícitas como direcciones de origen es conveniente incluir una serie de reglas en los router frontera.

- Para el tráfico entrante desde Internet habrá que tener en cuenta lo siguiente:
  - Las direcciones privadas especificadas en el [RFC 1918]. Este tipo de direcciones sólo podrían ser direcciones de origen de paquetes procedentes de redes internas.
  - Direcciones multicast (direcciones de multidifusión), que únicamente pueden ser direcciones de destino, nunca de origen.
  - Direcciones de loopback, que están pensadas para asignarlas a interfaces de loopback y se refieren al propio host, nunca a un host remoto.
  - Direcciones de la clase E (Están reservadas).
  - Si las direcciones de la red interna no coinciden con alguno de los bloques de direcciones para redes privadas, también es recomendable restringir el acceso a aquellos paquetes con direcciones de origen correspondientes a la red interna. Si un paquete con una dirección de origen de la red interna se envía desde de la red externa se trata de un IP spoofing.

  En el [RFC 5735] podrá encontrar una descripción detallada de estos y otros bloques de direcciones reservados para propósitos especiales.
- Para prevenir que se produzcan ataques de IP spoofing desde la red interna hacia la red externa es conveniente asegurarse de que el tráfico saliente de la red interna tiene una dirección IP de origen de dicha red, todas las demás direcciones IP deberían estar restringidas.

Reglas basadas en estado

Las reglas que hemos visto hasta ahora son reglas independientes del estado del sistema y de la red. La aplicación de la regla depende únicamente de los paquetes de tráfico que atraviesan el router. Sin embargo, en algunos casos, puede ser útil aplicar reglas basadas en estado, en las que el dispositivo basa la aplicación de la regla en sucesos ocurridos en instantes anteriores. Por ejemplo, si queremos evitar que entren paquetes en una red excepto si un host de esa red ha establecido una conexión previa con el host entrante, es necesario utilizar reglas basadas en estado. La regla se activa dependiendo de si hubo una conexión previa o no. Este tipo de reglas se suelen añadir en las interfaces externas de los firewalls para evitar que entren paquetes no autorizados en la red.

Para ello se suelen utilizar reglas similares a las siguientes:

```
/ip firewall filter
add chain=forward protocol=tcp connection-state=invalid action=drop
add chain=forward connection-state=established action=accept
add chain=forward action=drop
```

La primer regla borra los paquetes en los que el estado de la conexión es inválido. La segunda regla deja pasar los paquetes que corresponden a respuestas de conexiones previamente establecidas desde el interior.

Reglas para permitir servicios

En la DMZ habrá un conjunto de servidores que dan soporte a unos servicios. Por ejemplo, HTTP, FTP o SMTP. Las conexiones entrantes a la red deberán estar denegadas, salvo que correspondan a respuestas a paquetes de conexiones previas establecidas desde la red interna. Sin embargo, los que si deben permitirse son los paquetes dirigidos a puertos correspondientes a los servicios que deben ser visibles desde la red externa. Es importante prestar especial atención a los puertos menores que 1024 ya que estos son los que se suelen utilizar para servicios bien conocidos.

Recuerde que existen servicios que requieren más de un puerto. Por ejemplo, el protocolo FTP utiliza un puerto (21) para el tráfico de control, mientras que utiliza otro puerto secundario para transferir los ficheros. Por ello, puede no ser suficiente con contemplar únicamente el puerto 21. Asimismo, recuerde que en FTP existen dos modos: el activo y el pasivo. Este tipo de tráfico se puede permitir utilizando una regla como la siguiente, que deja pasar también el tráfico relacionado:

```
add chain=forward connection-state=related action=accept
```

Filtrado de tráfico ICMP

Los paquetes ICMP pueden utilizarse como parte de un ataque, ya que permiten reconocer la red. Sin embargo, como ICMP es un protocolo esencial para el funcionamiento de la red, no es buena idea denegar absolutamente todo el tráfico ICMP. Algunos paquetes deberán dejarse pasar:

- ICMP Echo e ICMP Echo Reply: A pesar de que estos paquetes son importantes para hacer comprobaciones en la red (son la base de los comandos `ping` y `traceroute`), también pueden utilizarse en ataques de denegación de servicios. Aunque conviene dejarlos pasar a través del router frontera, es conveniente controlar la cantidad de ellos que pasan. En muchas redes el administrador decide no dejar entrar este tipo de paquetes. Una opción menos restrictiva sería permitir el hacer ping a un grupo reducido de hosts.
- ICMP unreachable: Permiten identificar problemas de enrutamiento y se pueden utilizar para conocer la MTU de una red. Si un host activa el flag de no fragmentar en un paquete IP y además utiliza un tamaño de paquete demasiado grande, se genera un mensaje ICMP de destino inalcanzable. Si filtramos estos mensajes, el host de origen nunca sabría que está enviando paquetes demasiado grandes y no podría ajustar el tamaño. Por ello, es conveniente dejar pasar este tipo de paquetes.
- ICMP source quench: Es un paquete de *choke* que se utiliza para controlar flujos TCP y UDP [RFC 792]. Por lo que no es conveniente denegar este tipo de paquetes.

- ICMP Time Exceeded: Los paquetes IP se envían con un cierto valor en el campo de TTL que se decrementa en cada router. Cuando un router recibe un paquete con TTL cero, debe descartar el paquete y enviar una notificación de TTL excedido al host de origen. Es importante que estos mensajes puedan llegar a su destino por lo que es conveniente dejarlos pasar.

Habría que revisar y hacer consideraciones similares para el resto de tipos de paquetes ICMP, aunque estos suelen ser menos comunes por lo que muchos administradores deciden eliminarlos.

Limitar el acceso los dispositivos desde Internet

No debería ser posible acceder a la CLI de los dispositivos de red desde Internet. Un potencial atacante podría hacer un ataque por contraseña y ganar acceso al router pudiéndolo reconfigurar y comprometer la red. Igualmente, habría que limitar el acceso desde la red interna para que únicamente los administradores de red pudieran acceder.

NAT (Network Address Translation)

NAT es un servicio que modifica la dirección y/o el puerto de los paquetes IP que atraviesan una interfaz de red. El dispositivo que realiza NAT sobre los paquetes puede ser cualquier dispositivo situado entre el origen y el destino de éstos. Inicialmente, NAT fue diseñado para ahorrar direcciones IP versión 4, pero también constituye un elemento importante para la seguridad de las redes.

Los dispositivos de la red interna pueden utilizar direcciones IP reservadas para direccionamiento privado (RFC 1918). Dichas direcciones privadas no se utilizan en Internet, es decir, que un paquete con esta dirección no se puede encaminar en Internet. Los siguientes rangos de direcciones están reservados para direccionamiento privado:

- 10.0.0.0 - 10.255.255.255
- 172.16.0.0 - 172.31.255.255
- 192.168.0.0 - 192.168.255.255

Un router con NAT activado puede ocultar las direcciones IP de la red interna reemplazando la dirección interna privada con una dirección publica. Estas direcciones públicas son las únicas expuestas a Internet.

Existen distintos tipos de traducciones realizadas mediante NAT:

- Estática (NAT one-to-one): Cada dirección IP privada se traduce por una dirección IP pública que siempre es la misma. Esto permite a un servidor tener una dirección IP privada y ser visible en Internet mediante su dirección pública.
- Dinámica (NAT many-to-many): Una dirección privada se mapea a una dirección pública perteneciente a un pool de direcciones públicas. El router registra las direcciones del pool que están utilizadas y cada vez que un host de la red privada inicia una conexión con internet le asigna una dirección de este pool. Esto permite utilizar un conjunto de direcciones públicas menor que el número de hosts en la red privada.
- Sobrecarga (PAT - Port Address Tanslation, many-to-one): Se utiliza una única dirección IP y se utiliza el puerto de origen para multiplexar los datos hacia los distintos hosts de la red privada.

  En la figura 2, se muestra un ejemplo de funcionamiento de PAT. El host de la red interna con dirección 10.213.3.34 al abrir una conexión con otro host en Internet, envía un segmento TCP, con su dirección IP como dirección de origen, desde el puerto 20012. Dicho paquete al atravesar el router genera una entrada en la tabla NAT, en la que se registra un puerto (TCP) en la interfaz pública del router, en este caso el 3006. Al mismo tiempo, se sustituye la dirección de origen del paquete por la dirección pública del router (193.145.125.81) y el puerto de de origen del segmento TCP por el 3006, y se reenvía el paquete hacia el host situado en Internet. Una vez el host remoto recibe dicho paquete generará una respuesta con dirección de destino 193.145.125.81 y dirigido al puerto de destino 3006, es decir, al puerto 3006 del router. Cuando el paquete llega al router, éste consulta la tabla NAT y busca la entrada correspondiente al puerto 3006 y

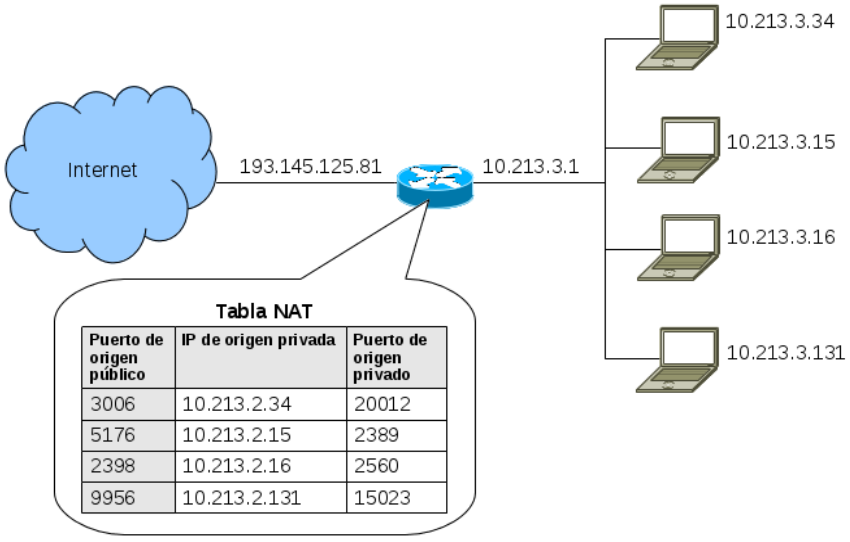

Figura 2: Ejemplo de funcionamiento de PAT.

sustituye la dirección y el puerto de destino públicos del paquete por la dirección y el puerto privados que figuran en la tabla. Finalmente, se reenvía el paquete de respuesta al host con dirección IP 10.213.3.34.

Nótese que a menos que un administrador añada manualmente una entrada a la tabla NAT, un host de Internet no podría iniciar una conexión con un host de la red interna. Haga la traza en sentido inverso y comprobará que esto no es posible, ya que el router no puede determinar a qué dirección de destino debe hacer la traducción.

Topología

En esta práctica vamos a utilizar una versión simplificada de una topología con firewalls para aprender a establecer las reglas de forma correcta. Consideraremos únicamente los flujos de tráfico entre la red externa y la DMZ y obviaremos la red interna, tal y como se muestra en la figura 3. Para ello utilizaremos dos de los PCs disponibles en el laboratorio y un router Mikrotik.

Paso 1. Asignar direcciones y comprobar conectividad

Realice el montaje físico que se muestra en el esquema. Asigne las direcciones a cada uno de los dispositivos y compruebe que existe conectividad entre el PC-1 y el PC-2.

Paso 2. Prevención de IP spoofing

En este paso vamos a crear un conjunto de reglas para controlar algunos ataques basados en IP spoofing. Vamos a controlar el acceso a la DMZ (PC-1) de paquetes con direcciones de origen ilícitas a través de la interfaz `ether2`. Para ello vamos a crear una cadena que denominaremos `anti_spoofing`. En ella vamos a introducir una serie de reglas con las que vamos a filtar paquetes IP cuya dirección de origen esté en alguno de los bloque reservados que se indican en el [RFC 5735]. No los incluiremos todos para simplificar este paso.

```
/ip firewall filter
add chain=anti_spoofing src-address=10.0.0.0/8 action=drop
add chain=anti_spoofing src-address=127.0.0.0/8 action=drop
add chain=anti_spoofing src-address=196.254.0.0/16 action=drop
```

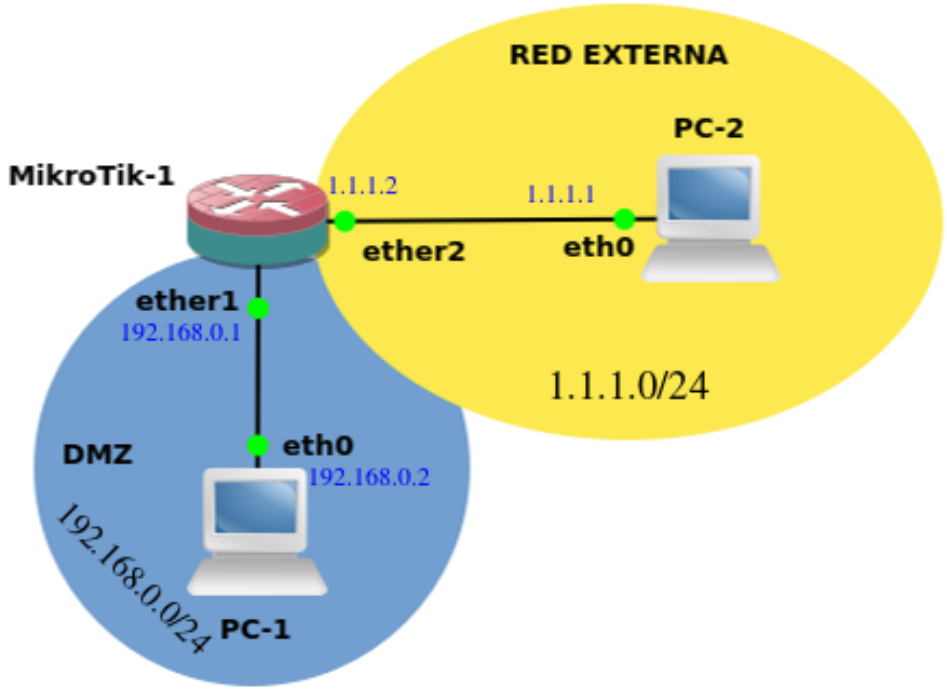

Figura 3: Topología de pruebas.

```
add chain=anti_spoofing src-address=172.16.0.0/12 action=drop
add chain=anti_spoofing src-address=192.168.0.0/16 action=drop
add chain=anti_spoofing src-address=240.0.0.0/4 action=drop
add chain=anti_spoofing src-address=224.0.0.0/4 action=drop
```

Vamos a realizar una captura mediante Wireshark en el PC-1, para comprobar qué paquetes llegan hasta la interfaz de este PC. Para ello ejecutamos el siguiente comando:

```
hping3 -S 192.168.0.2 -c 3 --source-ip 10.0.0.1
HPING 192.168.0.2 (virbr0 192.168.0.2): S set, 40 headers + 0 data bytes
len=40 ip=192.168.0.2 ttl=63 DF id=43042 sport=0 flags=RA seq=0 win=0 rtt=7.8 ms
len=40 ip=192.168.0.2 ttl=63 DF id=43180 sport=0 flags=RA seq=1 win=0 rtt=7.8 ms
len=40 ip=192.168.0.2 ttl=63 DF id=43200 sport=0 flags=RA seq=2 win=0 rtt=3.7 ms
```

Con esto se envían 3 intentos de conexión TCP al host 192.168.0.2 con una IP falseada 10.0.0.1 (paquetes marcados en gris, figura 4). Deberíamos observar algo similar a lo que se muestra en la figura 4. Se puede observar como en el PC-1 se capturan 3 paquetes con un intento de conexión TCP destinados a él.

Como aún no se han aplicado las reglas que hemos creado previamente, los paquetes pasan a través del router.

A continuación, vamos a saltar a la cadena `anti_spoofing` desde la cadena `forward` del firewall para que las reglas recién añadidas tengan efecto.

```
/ip firewall filter
add chain=forward in-interface=ether2 action=jump jump-target=anti_spoofing
```

Ahora vamos a repetir la prueba. Deberíamos observar que en la captura de Wireshark no aparecen los paquetes TCP SYN que se veían anteriormente. Esto deberían haber sido filtrados. Podemos comprobarlo mostrando las estadísticas de cada una de las reglas en nuestro firewall:

Finalmente, deberíamos comprobar que todas las reglas se están aplicando realizando las pruebas correspondientes.

No.	Time	Source	Destination	Protocol	Length	Info
9	13.530442	0c:30:56:10:ec:00	0c:30:56:e4:57:00	ARP	60	192.168.0.2 is at 0c:30:56:1
10	15.163655	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction
11	18.481638	10.0.0.1	192.168.0.2	TCP	54	1348 → 0 [SYN] Seq=0 Win=512
12	18.482073	192.168.0.2	10.0.0.1	TCP	60	0 → 1348 [RST, ACK] Seq=1 Ac
13	18.482550	192.168.0.1	192.168.0.2	ICMP	82	Destination unreachable (Net
14	19.481725	10.0.0.1	192.168.0.2	TCP	54	1349 → 0 [SYN] Seq=0 Win=512
15	19.482160	192.168.0.2	10.0.0.1	TCP	60	0 → 1349 [RST, ACK] Seq=1 Ac
16	19.482517	192.168.0.1	192.168.0.2	ICMP	82	Destination unreachable (Net
17	20.481776	10.0.0.1	192.168.0.2	TCP	54	1350 → 0 [SYN] Seq=0 Win=512
18	20.482203	192.168.0.2	10.0.0.1	TCP	60	0 → 1350 [RST, ACK] Seq=1 Ac
19	20.482659	192.168.0.1	192.168.0.2	ICMP	82	Destination unreachable (Net

Figura 4: Captura en el PC-1 sin activar la cadena `anti_spoofing`.

```
[admin@MikroTik] /ip firewall filter> print stats
Flags: X - disabled, I - invalid, D - dynamic
 #    CHAIN            ACTION                                BYTES      PACKETS
 0    anti_spoofing    drop                                    120            3
 1    anti_spoofing    drop                                      0            0
 2    anti_spoofing    drop                                      0            0
 3    anti_spoofing    drop                                      0            0
 4    anti_spoofing    drop                                      0            0
 5    anti_spoofing    drop                                      0            0
 6    anti_spoofing    drop                                      0            0
 7    forward          jump                                    120            3
[admin@MikroTik] /ip firewall filter>
```

Figura 5: Estadísticas de ejecución las reglas del firewall.

Paso 3. Publicación de servicios

Supongamos que el PC-1 es un servidor en el que se ejecutan servicios, los cuales deberíamos auditar y restringir aquellos puertos a los que no se debe tener acceso desde la red externa.

Para simular la presencia de un servicio web en el puerto 8080[1] del PC-1 vamos a utilizar `iperf`.

En el PC-1, ejecutamos el siguiente comando que abrirá el puerto 8080 en ese ordenador:

```
iperf -s -p 8080
```

Por otro, lado en el PC-2 vamos a ejecutar el comando para que actúe de cliente y conecte con la dirección IP del PC-1:

```
iperf -c 192.168.0.2 -p 8080
```

Ahora vamos a repetir lo mismo con el puerto 2525. Deberíamos observar que se produce una conexión con este puerto, porque aún no se han establecido reglas de filtrado.

Ahora vamos a publicar el servicio web del puerto 8080, mientras que restringimos el resto de servicios del PC-1. Para ello en el firewall del Mikrotik-1, vamos a crear una cadena para el filtrado de los servicios, que va a permitir el servicio web mientras se filtra el resto.

```
/ip firewall filter
add chain=servicios dst-port=8080 protocol=tcp dst-address=192.168.0.2 action=accept
add chain=servicios dst-address=192.168.0.2 action=drop
```

Ahora saltamos a la cadena de `servicios` desde la cadena `forward`:

[1] Los servicios web se ejecutan normalmente en el puerto 80. En esta práctica lo haremos con el puerto 8080 para que no sean necesarios privilegios de superusuario.

```
/ip firewall filter
add chain=forward in-interface=ether2 action=jump jump-target=servicios
```

Ahora repita la prueba que realizó al comienzo de este apartado. Debería observar como la conexión al puerto 8080 funciona, mientras que la conexión con el puerto 2525 no lo hace. Si hacemos un escaneo de puertos, deberíamos observar únicamente el puerto 8080, tal y como se muestra a continuación:

```
:~$ nmap -PN 192.168.0.2
Starting Nmap 7.80 ( https://nmap.org ) at 2021-04-13 10:53 WEST
Nmap scan report for 192.168.0.2
Host is up (0.00042s latency).
Not shown: 999 filtered ports
PORT     STATE SERVICE
8080/tcp open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 4.34 seconds
```

Para ello, debe volver a ejecutar previamente el comando iperf para este puerto en el PC-1.

Paso 4. Reglas basadas en estado

Ahora vamos a comprobar si desde el PC-1 es posible conectar con algún servicio en la red externa. Para ello, vamos a ejecutar el siguiente comando en el PC-2:

```
iperf -s -p 8080
```

Y ejecutamos el comando para que haga una conexión de cliente desde PC-1:

```
iperf -c 1.1.1.1 -p 8080
```

Es decir, que hacemos justo al contrario que en el paso anterior. Se debe observar que no es posible establecer una conexión. Esto se debe a que el conjunto de reglas que se habían establecido para los servicios filtran todo tráfico entrante desde la interfaz `ether2` y no permite que vuelvan las respuestas a la conexión. La dificultad con las conexiones desde el interior hacia el exterior es que no podemos predecir el puerto de origen de las peticiones realizadas y añadir las reglas al firewall para que deje pasar los paquetes destinados a estos puertos. Sin embargo esto último se consigue mediante reglas basadas en estado. Vamos añadir una regla de este tipo al firewall:

```
/ip firewall fitler
add chain=forward conection-state=established in-interface=ether2 action accept
```

Si ahora repetimos la prueba, seguirá sin funcionar. El motivo de esto no es otro que el orden de las reglas en la cadena `forward`.

```
[admin@MikroTik] /ip firewall filter> print chain=forward
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=forward action=jump jump-target=anti_spoofing

 1    chain=forward action=jump jump-target=servicios in-interface=ether2

 2    chain=forward action=accept connection-state=established
      in-interface=ether2
```

La regla que descarta los paquetes que está dentro de la cadena de servicios se aplica antes que la regla basada en estado que acabamos de añadir. Muévala a una posición anterior. Por ejemplo:

```
[admin@MikroTik] /ip firewall filter> print chain=forward
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=forward action=jump jump-target=anti_spoofing

 1    chain=forward action=accept connection-state=established
      in-interface=ether2

 2    chain=forward action=jump jump-target=servicios in-interface=ether2
```

Ahora compruebe si funciona correctamente.

Paso 5. Control de tráfico ICMP

En este punto vamos a crear una cadena para gestionar el tráfico ICMP.

Para comprobarlo enviamos paquetes de este tipo desde PC-1 hacia el PC-2. El comando `hping3` (`http://www.hping.org/manpage.html`) permite especificar el tipo de paquete que se desea enviar:

- TTL-exceeded:

  ```
  hping3 -1 -C 11 -K 0  1.1.1.1
  ```

- Destination unreachable:

  ```
  hping3 -1 -C 3 -K 0  1.1.1.1
  ```

- Source quench:

  ```
  hping3 -1 -C 4 -K 0  1.1.1.1
  ```

Para comprobar si los paquetes llegan al PC-2 debemos utilizar Wireshark.

Ahora construya una cadena con nombre `icmp` que permita el paso de los tres tipos de mensajes antes mencionados y restrinja el resto de mensajes ICMP desde el PC-1 hacia el PC-2. No olvide añadir el salto a esta cadena desde la cadena `forward`. Puede consultar el manual del firewall (`https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/NAT`). Deberá utilizar el atributo `icmp_options`. No olvide añadir también el atributo `protocol` indicando para aplicar las reglas únicamente a los paquetes ICMP.

Paso 6. Control de acceso al router

Además de controlar el acceso a los servicios y a la red interna es necesario asegurar que un atacante no pueda acceder al router y modificar su configuración o comprometer su servicio. Vamos a escanear los puertos del router desde el PC-2 mediante el comando nmap. Deberíamos observar algo similar a lo siguiente:

```
~$ nmap 1.1.1.2


Starting Nmap 7.60 ( https://nmap.org ) at 2021-04-11 19:52 WEST
Nmap scan report for _gateway (1.1.1.2)
Host is up (0.00046s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
80/tcp   open  http
```

```
2000/tcp open  cisco-sccp
8291/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 0.17 seconds
```

Como podemos observar, el router tiene algunos servicios activos. Para empezar deberíamos restringir el acceso desde la interfaz ether2, para evitar accesos desde la red externa. El segundo paso debería ser controlar los accesos desde la red interna para que únicamente puedan acceder los usuarios de la red de administración.

Para ello, vamos a añadir la siguiente regla a la cadena `input` del router.

```
/ip firewall filter
add chain=input in-interface=ether1 action=accept
add chain=input action=drop
```

Con la regla anterior se elimina cualquier tráfico destinado directamente al router a través de cualquier interfaz que no sea `ether1`.

Si vuelve a comprobar el escaneo de puertos, debería observar que los servicios del router no son visibles.

Paso 7. Traducción de direcciones mediante NAT.

Vamos a activar NAT de tipo masquerade para la interfaz `ether2`. Esto hace mediante el siguiente comando:

```
/ip firewall nat
add chain=srcnat src-address=192.168.0.0/24 out-interface=ether2 action=masquerade
```

Ahora para comprobar el correcto funcionamiento repita la prueba del paso 4 pero haciendo una captura de paquetes mediante Wireshark de forma simultánea en ambos PCs. Debería observar el efecto de la traducción de paquetes que se observan en el PC-2.

La pregunta que aún queda por resolver si con esta configuración podríamos conectar con un servicio en el puerto 8080 del PC-1, tal y como lo habíamos establecido en el paso 3. Para comprobarlo, repita la prueba realizada en este paso. ¿Sería esto correcto en un entorno con direccionamiento privado? Si lo considera necesario, revise la configuración del firewall y añada las reglas que considere oportunas. Igualmente, haga los cambios sobre la configuración de NAT que considere oportunos. Revise el manual del servicio NAT de MikroTik (`https://wiki.mikrotik.com/wiki/Manual:IP/Firewall/Filter`) para averiguar los comandos de configuración que debe llevar a cabo.

Referencias

[RFC 1918] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot y E. Lear. *Address Allocation for Private Internets*. RFC 1918 (Best Current Practice). Updated by RFC 6761. Internet Engineering Task Force, feb. de 1996. URL: `http://www.ietf.org/rfc/rfc1918.txt`.

[RFC 5735] M. Cotton y L. Vegoda. *Special Use IPv4 Addresses*. RFC 5735 (Best Current Practice). Obsoleted by RFC 6890, updated by RFC 6598. Internet Engineering Task Force, ene. de 2010. URL: `http://www.ietf.org/rfc/rfc5735.txt`.

[RFC 792] J. Postel. *Internet Control Message Protocol*. RFC 792 (INTERNET STANDARD). Updated by RFCs 950, 4884, 6633, 6918. Internet Engineering Task Force, sep. de 1981. URL: `http://www.ietf.org/rfc/rfc792.txt`.