Práctica 8. Seguridad básica: Firewalls

Seguridad en un router frontera

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.

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 acton=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 a la posicion 2:

```
[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.
- 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.

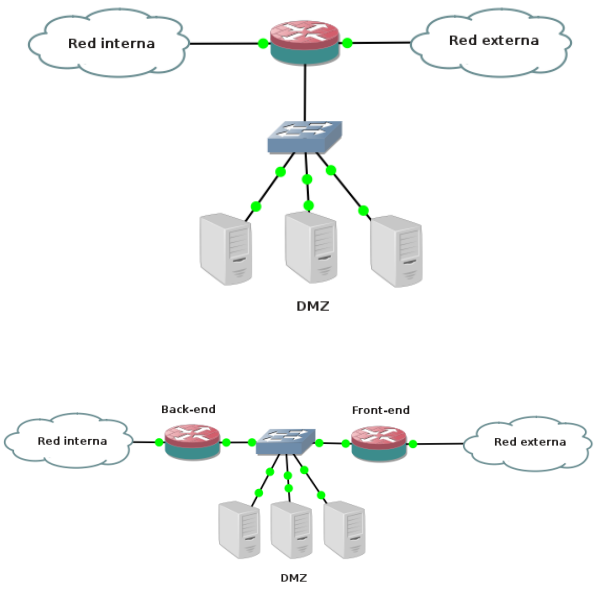

Figura 1: Diseños basados en DMZ: Arriba: Un firewall, Abajo: Firewall dual.

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.

Topología

En esta parte de la práctica, se utilizará una topología con una DMZ implementada con firewall dual como la que se muestra en la figura 2.

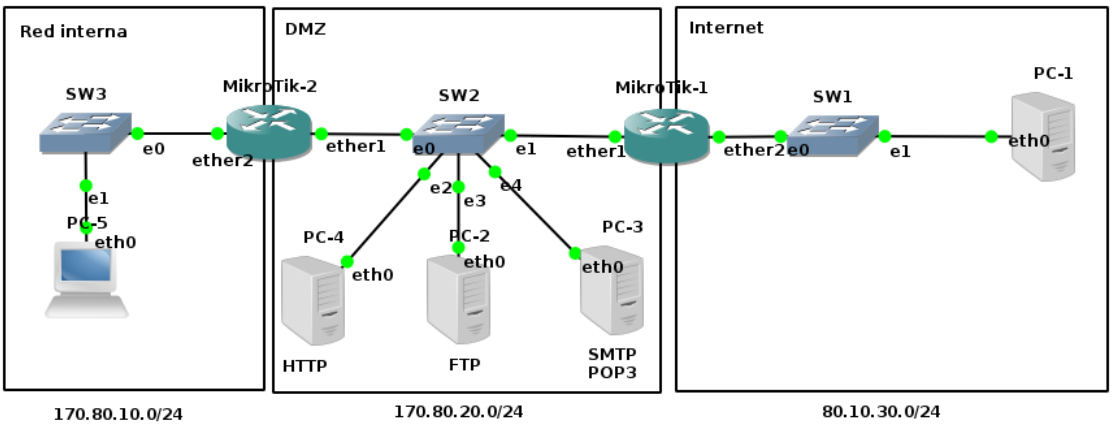

Figura 2: Esquema de la topología.

La red anterior tiene configuradas las direcciones IP, el enrutamiento dinámico mediante RIP, y existen algunos puertos abiertos que deberemos asegurar. **Compruebe la configuración de RIP que se ha implementado en cada uno de los routers y justifíquela.**

La asignación de direcciones IP de los distintos dispositivos es la siguiente:

Dispositivo	Interfaz	Dirección IP
MikroTik-1	ether1	170.80.20.1/24
MikroTik-1	ether2	80.10.30.1/24
MikroTik-2	ether1	170.80.20.2/24
MikroTik-2	ether2	170.80.10.1/24
PC-1	eth0	80.10.30.2/24
PC-2	eth0	170.80.20.6/24
PC-3	eth0	170.80.20.7/24
PC-4	eth0	170.80.20.5/24
PC-5	eth0	170.80.10.2/24

El objetivo es proteger tanto a los servidores como a los hosts de la red interna de potenciales ataques desde el exterior, utilizando los mecanismos que hemos aprendido en la práctica anterior, además de reglas basadas en estado, si son necesarias. También debemos proteger a los hosts de la red interna de posibles ataques procedentes desde alguno de los servidores de la DMZ que pueda haber sido comprometido y viceversa. Nótese que esta práctica es únicamente una introducción por lo que quedan aspectos por tratar.

Paso 1. Hacer un escaneo de hosts y puertos utilizando `nmap`.

Supondremos que el PC-1 es un host de Internet desde el que se van a perpetrar ataques contra nuestra red. Por ello, a dicho PC-1 se le ha instalado el comando nmap, que es un herramienta para la exploración de redes y auditoría de seguridad (`http://nmap.org/man/es/`). Permite enviar paquetes *raw* de forma que se pueden comprobar si nuestra red es vulnerable a distintos tipos de ataques realizados mediante el envío de paquetes IP cuya cabecera se modifica adecuadamente.

Como inicialmente la red está sin protección alguna, la única protección que tenemos es no instalar servicios que no sean estrictamente necesarios en los hosts y toda protección que tengamos a nivel de host. Por ello si

hacemos un escaneo de puertos y descubrimiento de hosts a las distintas redes debemos observar todos los hosts y los puertos (correspondientes a servicios) que están abiertos y pueden ser potenciales objetivos de un ataque. Ejecute:

```
root@box:~# nmap --open -n -T5 -PI 170.80.20.0/24
```

Este comando envía paquetes ICMP echo requests para detectar los hosts activos de la red 170.80.20.0/24. Una vez detectados se realiza un escaneo de puertos, mostrando qué puertos están activos en cada host. El flag `--open` hace que se muestren sólo los puertos abiertos. El flag `-n` sirve para desactivar la resolución inversa de nombres por DNS (al no existir en esta red un servidor DNS, se desactiva), mientras que el flag `-T5` aumenta la velocidad de escaneo de puertos.

```
Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-03 11:44 UTC
Warning: 170.80.20.2 giving up on port because retransmission cap hit (2).
Warning: 170.80.20.7 giving up on port because retransmission cap hit (2).
Warning: 170.80.20.1 giving up on port because retransmission cap hit (2).
Warning: 170.80.20.5 giving up on port because retransmission cap hit (2).
Warning: 170.80.20.6 giving up on port because retransmission cap hit (2).
Nmap scan report for 170.80.20.1
Host is up (0.00031s 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 scan report for 170.80.20.2
Host is up (0.00038s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
80/tcp   open  http
2000/tcp open  cisco-sccp
8291/tcp open  unknown

Nmap scan report for 170.80.20.5
Host is up (0.00052s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
80/tcp  open  http
443/tcp open  https

Nmap scan report for 170.80.20.6
Host is up (0.00049s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh

Nmap scan report for 170.80.20.7
Host is up (0.0011s latency).
```

```
Not shown: 997 closed ports
PORT    STATE SERVICE
22/tcp  open  ssh
110/tcp open  pop3
143/tcp open  imap

Nmap done: 256 IP addresses (5 hosts up) scanned in 17.55 seconds
```

Ahora haga lo mismo con la red interna con dirección 170.80.10.0/24.

Paso 2. Decidir el tráfico que vamos a permitir y cuál no

Las salidas anteriores revelan varios datos importantes:

- Todos los hosts (incluidos los routers) tienen un servicio de Secure Shell (`ssh`) que es accesible desde un host arbitrario de internet.
- Ambos routers permiten acceder desde Internet a distintos servicios.
- El PC-5 tiene varios puertos abiertos que no deberían ser accesibles desde Internet.
- Por último, es posible hacer ping desde Internet a todos los hosts de la red, pues aparentemente no hay restricción alguna para este tipo de mensajes.

¿Cuáles serían los riesgos potenciales?

- Habría que valorar si los accesos por Secure Shell (`ssh`) desde Internet, tanto a los servidores como a los routers, son realmente necesarios. Si no lo son, habría que bloquear este tipo de accesos desde el exterior.
- Los accesos por SSH, telnet, etc. deben permitirse sólo desde la red interna, y seguramente sólo ser accesibles para una red de administradores. Hay que recordar que el protocolo telnet no encripta la comunicación por lo que un potencial atacante podría espiar este tráfico para obtener información sobre la red.
- El mapeo de puertos anterior, permite detectar los hosts que hay en cada red y los servicios activos en los mismos. Para ello utiliza mensajes ICMP, como los empleados por el comando ping, para determinar los hosts activos y luego escanea los puertos para ver qué servicios están activos. Sería conveniente restringir este tipo de accesos por ICMP.
- Algunos hosts tienen servicios activos para su publicación en la red externa, mientras que otros no deberían ser públicos. Habrá que valorar que servicios son los que deseamos publicar y dejar pasar sólo aquellos que sean necesarios.
- En general, deberían estar restringidos todos los servicios que no sean públicos. En este caso, en la DMZ tenemos: HTTP, HTTPS, SMTP, POP3 e IMAP . Habría que valorar si es necesario publicar estos servicios al exterior. Además, como se dijo anteriormente, algunos servicios requieren dejar pasar también el tráfico relacionado.
- En general, cualquier atacante podría tratar de ocultar su identidad mediante un IP spoofing y habría que prevenir los ataques IP spoofing utilizando direcciones privadas o de la propia red desde internet, al igual que desde la DMZ hacia la red interna y desde la red interna hacia la DMZ.

Paso 3. Crear las cadenas de reglas necesarias para reducir los riesgos

Ahora deberá implementar las cadenas de reglas necesarias para filtrar el tráfico no deseado y dejar pasar el tráfico de interés para la organización.

Paso 4. Verificar las políticas establecidas

Una vez implementados distintas cadenas de filtrado, habrá que verificar que se cumplen los requisitos de seguridad. Para ello, utilizaremos nuevamente el comando `nmap`, que tiene gran cantidad de modificadores que permiten realizar distintas pruebas. Aquí se exponen sólo algunos para verificar que las reglas implementadas funcionan correctamente. Tenga en cuenta que unas reglas pueden interferir con otras dentro de la misma cadena o incluso con otras cadenas aplicadas en un punto distinto de la red, por lo que hay que verificar la implementación en su conjunto, no política por política.

Prevención de IP spoofing

Para verificar si el conjunto de reglas previene el IP spoofing, utilizamos nmap y le indicamos que queremos utilizar una dirección IP de origen distinta. Tenga en cuenta que el tráfico de vuelta no podrá volver al host de origen ya que dicho host realmente no tiene esa dirección IP. Para comprobar que el router elimina los paquetes con identidad falsa se puede comprobar el informe de estadísticas del router MikroTik. Supongamos que hemos puesto el siguiente conjunto de reglas en la cadena `forward`:

```
/ip firewall filter
add action=drop chain=forward in-interface=ether2 src-address=192.168.0.0/16
```

Esto lo que hace es eliminar todos los paquetes con dirección de origen en el rango privado de clase C que entren por la interfaz ether2. El resto de tráfico lo dejará pasar.

Ahora queremos comprobar si la regla que restringe el tráfico funciona correctamente. Para ello debemos generar un tráfico con dirección de origen en el rango 192.168.0.0/16 que pase por ether2. Esto se puede conseguir mediante el comando `hping3` desde el PC-1:

```
root@Debian:~# hping3 -c 10  -p 80 --spoof 192.168.0.1 172.80.20.5

HPING 172.80.20.5 (eth0 172.80.20.5): NO FLAGS are set, 40 headers + 0 data bytes

--- 172.80.20.5 hping statistic ---
10 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Esto enviará una serie de 10 paquetes IP falsos con dirección IP de origen 192.168.0.1 hacia 172.80.20.5.

Para comprobar si el firewall está actuando correctamente visualizamos las estadísticas:

```
 [admin@MikroTik] /ip firewall filter> print stats
Flags: X - disabled, I - invalid, D - dynamic
 #    CHAIN            ACTION                          BYTES         PACKETS
 0    forward          drop                              400              10
 1    forward          accept                        185 664           3 072
```

Esta verificación habría que realizarla con cada una de los rangos de direcciones susceptibles de utilizase como direcciones de origen en un IP spoofing desde Internet (véase el [RFC 5735]).

Por otra parte, un atacante podría haber comprometido la seguridad de uno de los servidores por lo que podría enmascarar su identidad utilizando IP spoofing atacando desde el servidor a la red interna. O bien, el atacante podría tratar de atacar uno de los servidores desde la propia red interna 170.80.10.0/24. Habría que comprobar que no es posible realizar IP spoofing desde la DMZ hacia la red interna, ni desde la red interna hacia los servidores. Para hacer esta verificación hemos instalado `nmap` en el PC-5 y en el PC-4. Haga los chequeos correspondientes.

Bloqueo del tráfico ICMP desde Internet

Si hemos decidido bloquear el `ping` desde Internet hacia la red interna, la comprobación más sencilla es hacer ping a todos los hosts desde el PC-1. Esta tarea la facilita el comando `nmap`:

```
root@Debian:~# nmap -n -T5 -sP -PI 170.80.20.0/24

Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-03 17:57 UTC
Nmap done: 256 IP addresses (0 hosts up) scanned in 13.05 seconds
root@Debian:~# nmap -n -T5 -sP -PI 170.80.10.0/24

Starting Nmap 6.47 ( http://nmap.org ) at 2018-04-03 17:58 UTC
Nmap done: 256 IP addresses (0 hosts up) scanned in 13.05 seconds
```

Fíjese que estamos haciendo un descubrimiento de hosts en las dos redes, la DMZ y la red interna y que en este caso no debería devolver el ping ninguno de los hosts.

Por otra parte, según las recomendaciones indicadas anteriormente si deberían pasar desde la red corporativa a Internet los paquetes ICMP del tipo: TTL-exceeded, destination unreachable, y source quench. Igualmente en el otro sentido si tenemos denegado el resto del tráfico ICMP.

Para comprobarlo enviamos paquetes de este tipo desde el interior hacia el PC-1. El comando `hping3` (`http://www.hping.org/manpage.html`) (instalado en PC-1, PC-4 y PC-5) permite especificar el tipo de paquete que se desea enviar:

- TTL-exceeded:

  ```
  hping3 -1 -C 11 -K 0  80.10.30.2
  ```

- Destination unreachable:

  ```
  hping3 -1 -C 3 -K 0  80.10.30.2
  ```

- Source quench:

  ```
  hping3 -1 -C 4 -K 0  80.10.30.2
  ```

Puede establecer un punto de captura en la interfaz del PC-1 para comprobar si llegan los paquetes desde el interior de la red. En la figura 3 se muestra la captura para los paquetes ICMP source quench que llegan al PC-1.

Tráfico correspondiente a los servicios

Como se ha señalado anteriormente, el tráfico correspondiente a los servicios activos en la DMZ debería estar permitido. Para comprobarlo, debemos hacer un nuevo escaneo de puertos desde el PC-1 hacia la red de la DMZ y desde la red interna hacia la DMZ.

Para ello se puede utilizar el comando:

```
nmap -PN <direccion IP>
```

para cada una de las direcciones IP de los servidores. Por ejemplo, para el servidor de correo deberíamos obtener:

```
root@box:~# nmap -n -PN 170.80.20.7

Starting Nmap 5.21 ( http://nmap.org ) at 2015-03-01 19:28 UTC
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 0.50% done
Nmap scan report for 170.80.20.7
Host is up (0.0010s latency).
Not shown: 995 filtered ports
PORT    STATE SERVICE
25/tcp  open  smtp
110/tcp open  pop3
143/tcp open  imap
```

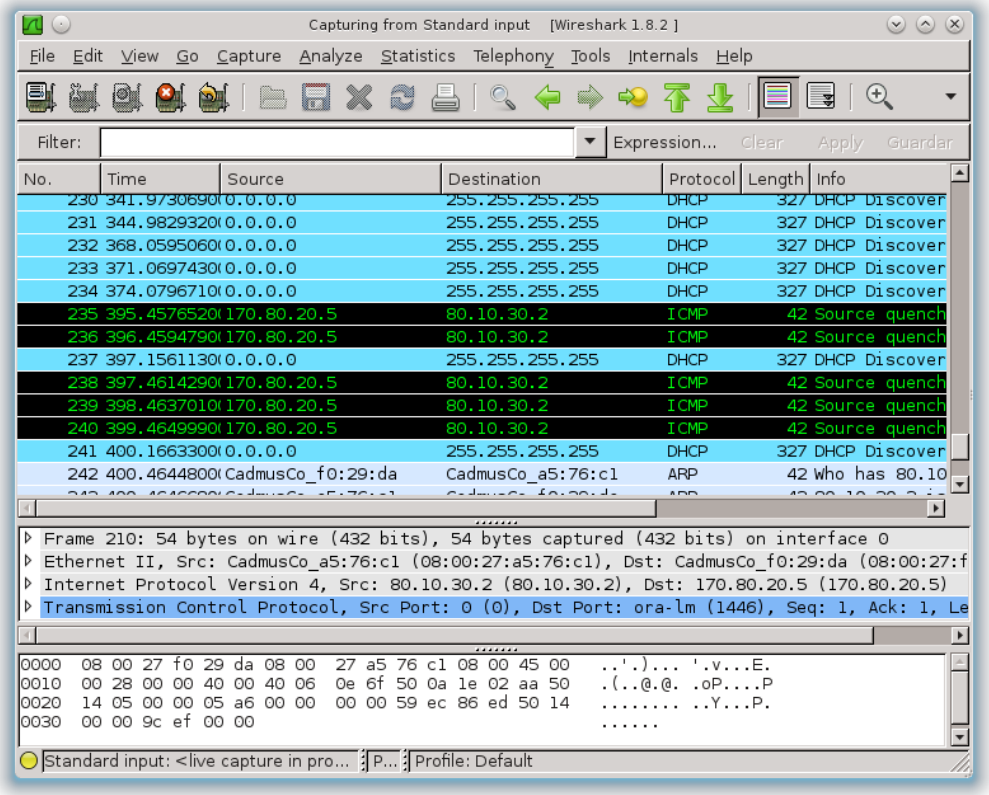

Figura 3: Captura de paquetes en la interfaz `eth0` del PC-1

donde ahora el puerto correspondiente a `ssh` estará cerrado.

También deberá valorar desde qué puntos de la red se debería poder acceder a los servicios abiertos en los routers.

Comprobar el tráfico de retorno

Como se ha indicado anteriormente en la frontera exterior no se debería permitir ningún tráfico entrante, excepto el que ya hemos mencionado anteriormente, y el tráfico correspondiente a las conexiones establecidas previamente desde el interior de la red. Esto se consigue mediante reglas basadas en estado. Para comprobar que esta característica funciona correctamente, en esta práctica estableceremos una sesión ssh desde el PC-5 al PC-1.

```
ssh testuser@80.10.30.2
```

La clave es `testuser` si se consigue establecer correctamente una sesión ssh, significa que los paquetes de respuesta no han sido bloqueados por el firewall y que todo funciona correctamente.

Pruebas desde la red interna hacia la DMZ y viceversa

A parte de comprobar el tráfico desde Internet hacia el interior de la red corporativa, también habrá que plantearse el caso en el que haya un ataque desde la red interna hacia la DMZ o viceversa. Por tanto, deberá realizar las pruebas oportunas para verificar que todos los posibles flujos de datos están cubiertos por los firewall.

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`.