Informe

Práctica 8:
Seguridad básica: Firewalls

Laboratorio de Redes

Diego Cruz Rodríguez
Universidad de La Laguna
Ingeniería Informática
3º Curso, 2º Semestre
06/05/2020

Índice

Objetivos **2**

Firewall **2**

RouterOS 2

Cadenas por defecto. 3

Creación de cadenas y saltos a otras cadenas 3

Alterar el orden de las reglas de una cadena 4

Aspectos a tener en cuenta 4

Protección de la frontera con Internet: DMZ 5

Redes basadas en DMZ 5

Contramedidas para el IP spoofing 6

Reglas basadas en estados 7

Reglas para permitir servicios 7

Filtrado de tráfico ICMP 7

Limitar el acceso a los dispositivos internos desde internet 8

Topología **8**

1º Hacer un escaneo de hosts y puertos utilizando nmap. 9

2º Decidir el tráfico que vamos a permitir y cuál no 10

3º Crear las cadenas de reglas necesarias para reducir los riesgos 11

4º Verificar las políticas establecidas 12

Referencias **13**

Objetivos

- 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.

Firewall

Un firewall es un dispositivo de seguridad de la red, monitoriza el tráfico y decide si debe permitir o bloquear un tráfico específico en función de un conjunto de restricciones de seguridad definidas de antemano.
Los firewalls establecen una barrera entre las redes internas seguras y las redes externas poco fiables como Internet.
Un firewall puede ser hardware, software o ambos.

RouterOS

El firewall se define a través de lo que se conoce como cadenas o chain. Una cadena consiste en un patrón y una acción. El patrón es un criterio que permite decidir si un paquete se le aplica una acción. Estas acciones pueden ser dejar pasar un paquete (accept) o desechar el paquete (drop), estas acciones se encuentran en todos los firewalls. Particularmente RouterOS también cuenta con *reject*, que rechaza el paquete y se lo notifica al emisor. *Log,* deja un rastro en los log del sistema y pasa a la siguiente regla de la secuencia y la acción jump que permite saltar de la cena actual a otra cadena con el nombre especificado en el campo jump-target.

Criterios más destacados para esesificarpatrones RouterOS:

- **src-address y dst-address** , dirección fuente y destino respectivamente pudiendo especificar un bloque de direcciones o rango.
- **src-port y dst-port** indican los puertos de origen y destino.
- **protocol** que indica el protocolo (tcp,udp,icmp...) Nótese que cuando se indican restricciones sobre el números de puertos hay que indicar el protocolo tcp o udp.
- **in-interface y out-interface** interfaz por la que entran o salen del router respectivamente.
- **icmp-options**: permite diferenciar los diferentes tipos icmp. Es necesario que el *protocol* sea ICMP.

Cadenas por defecto.

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.

Deberá haber al menos una regla en alguna de ellas para que el firewall funcione.

Creación de cadenas y saltos a otras cadenas

Para mejorar la eficiencia en el procesamiento y organizar las reglas de manera lógica. Podemos emplear los saltos, creamos cadenas con el nombre concretos y posteriormente la enlazamos con un salto en forward, input o output, para ello empleamos el campo *jump-target*.

Alterar el orden de las reglas de una cadena

Cuando se añade una nueva regla a una cadena se añade por defecto en última posición. Si queremos que no sea así existen varias forma de cambiarlo si es una cadena nueva que vamos añadir, podemos usar atributo place-before seguido del número que queremos que tenga la nueva regla. En caso de ser una regla ya existente lo que podemos hacer es moverla, para ello podemos usar el atributo move seguido del número de la regla que queremos mover y el número de la posición que quiere que ocupe (Este procedimiento se hace añadiendo una copia de la regla en la posición indicada y posteriormente se elimina la anterior).

Aspectos a tener en cuenta

- Si ningunas de las reglas de una cadena es aplicable el paquete se deja pasar.
- Las reglas se aplican secuencialmente. Cuando una regla casa con un paquete, se aplica esa regla y termina el proceso. Por dicha razón, las reglas más restrictivas deben ir al principio.
- Cuando existan 2 reglas que no se solapen hay que poner primera la regla que se aplique a mayor número de paquetes. Al ser un procesamiento secuencial, si la regla se encuentra muy abajo el tiempo de procesamiento será mayor.
- Hay que recordar que los protocolos de enrutamiento envían paquetes de actualización de la red. Si hay un protocolo de enrutamiento activo en la red hay que recordad dejar pasar el tráfico correspondiente.

Protección de la frontera con Internet: DMZ

Peligros potenciales sobre routers frontera.

- **Sniffing o snooping**: Gran parte de las comunicaciones de red siguen siendo de texto plano. Esto permite que un atacante que comprometa una red pueda escuchar 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. Un atacante podríía 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

Una DMZ es una subred fı́sica 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.

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

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:

- Las direcciones privadas 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.

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 deberı´an estar restringidas.

Reglas basadas en estados

El dispositivo basa la aplicación de la regla en sucesos ocurridos en instantes anteriores. Para poder comprobar este tipo de conexiones existen campos como *connection-state* con valor established para comprobar que existe ya una conexión en curso o invalid para decir que es el primer paquete buscando la conexión.

Reglas para permitir servicios

En la DMZ habrá un conjunto de servidores que dan soporte a unos servicios. 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.
También hay que tener en cuenta conexiones de variadas que pueden pertenecer un un mismo protocolo esto lo podemos controlar con el valor *related* en el campo connection-state.

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.
- 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.

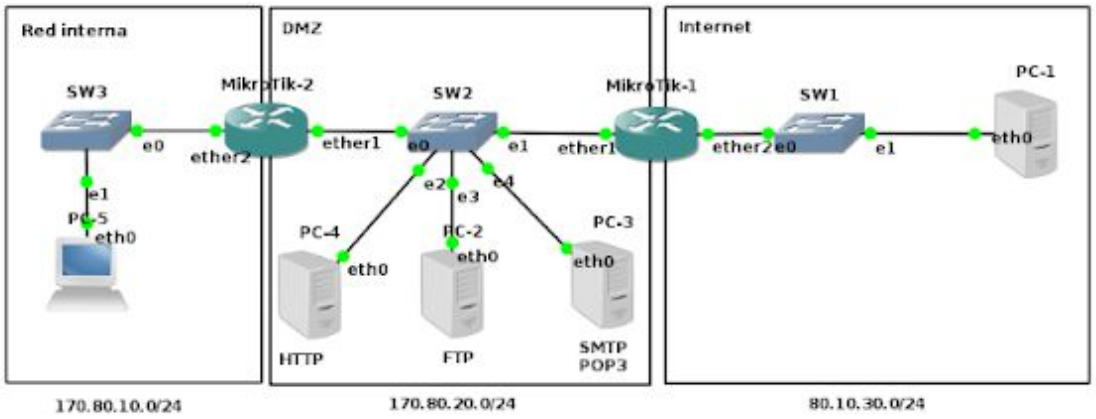
- ICMP source quench: Es un paquete de choke que se utiliza para controlar flujos TCP y UDP. Por lo que no es conveniente denegar este tipo de paquetes.
- ICMP Time Exceeded: Los paquetes IP se envían con un cierto valor en el campo de TTL que se decrementa en cada router. Cuando un router recibe un 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 a los dispositivos internos 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 pudiéndose reconfigurar y comprometer la red. Igualmente, habríía que limitar el acceso desde la red interna para que únicamente los administradores de red pudieran acceder.

Topología



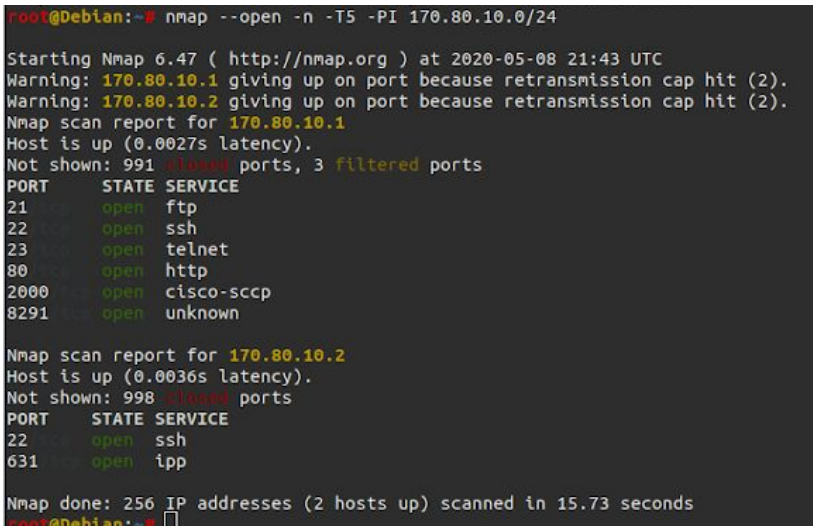
Esquema de la topología de red.

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

Direcciones de red en cada interfaz.

1º Hacer un escaneo de hosts y puertos utilizando *nmap*.

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

Starting Nmap 6.47 ( http://nmap.org ) at 2020-05-08 21:38 UTC
Warning: 170.80.20.1 giving up on port because retransmission cap hit (2).
Warning: 170.80.20.2 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).
Warning: 170.80.20.7 giving up on port because retransmission cap hit (2).
Nmap scan report for 170.80.20.1
Host is up (0.0013s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21       open  ftp
22       open  ssh
23       open  telnet
80       open  http
2000     open  cisco-sccp
8291     open  unknown

Nmap scan report for 170.80.20.2
Host is up (0.0024s latency).
Not shown: 994 closed ports
PORT     STATE SERVICE
21       open  ftp
22       open  ssh
23       open  telnet
80       open  http
2000     open  cisco-sccp
8291     open  unknown

Nmap scan report for 170.80.20.5
Host is up (0.0025s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22      open  ssh
80      open  http
443     open  https

Nmap scan report for 170.80.20.6
Host is up (0.0025s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE
21     open  ftp
22     open  ssh

Nmap scan report for 170.80.20.7
Host is up (0.0027s latency).
Not shown: 997 closed ports
PORT    STATE SERVICE
22      open  ssh
110     open  pop3
143     open  imap

Nmap done: 256 IP addresses (5 hosts up) scanned in 18.50 seconds
```



2º Decidir el tráfico que vamos a permitir y cuál no

Mikrotik 1

Acepta tráfico puerto tcp 80/443 con destino 170.80.20.5
Acepta tráfico puerto tcp 21 y relacionado con destino 170.80.20.6
Acepta tráfico puerto tcp 110/143 y relacionado con destino 170.80.20.7
Rechaza todo tráfico no previamente establecido con destino la red 170.8.10.0/24
Rechaza todo tráfico no previamente establecido con destino la red 170.8.20.0/24
Rechazara paquetes procedentes de la red externa con ip origen de la red 170.80.20.0/24
Rechazara paquetes procedentes de la red externa con ip origen de la red 170.80.10./24

Mikrotik 2
Rechaza todo tráfico no previamente establecido con destino la red 170.8.10.0/24
Rechazara paquetes procedentes de la red externa con ip origen de la red 170.80.10./24

3º Crear las cadenas de reglas necesarias para reducir los riesgos

Mikrotik 1

```
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=forward action=jump jump-target=snif

 1    chain=forward action=jump jump-target=serverftp dst-address=170.80.20.6

 2    chain=forward action=jump jump-target=serverweb protocol=tcp
      dst-address=170.80.20.5

 3    chain=forward action=jump jump-target=serversmtppop
      dst-address=170.80.20.7

 4    chain=forward action=jump jump-target=redinterna
      dst-address=170.80.10.0/24

 5    chain=forward action=jump jump-target=reddmz dst-address=170.80.20.0/24

 6    chain=serverweb action=accept protocol=tcp dst-address=170.80.20.5
      dst-port=80

 7    chain=serverweb action=accept protocol=tcp dst-address=170.80.20.5
      dst-port=443

 8    chain=serverftp action=accept protocol=tcp dst-address=170.80.20.6
      dst-port=21

 9    chain=serverftp action=accept connection-state=related
      dst-address=170.80.20.6

10    chain=serversmtppop action=accept connection-state=related
      dst-address=170.80.20.7

11    chain=serversmtppop action=accept protocol=tcp dst-address=170.80.20.7
      dst-port=110

12    chain=serversmtppop action=accept protocol=tcp dst-address=170.80.20.7
      dst-port=143

13    chain=redinterna action=drop connection-state=invalid
      dst-address=170.80.10.0/24

14    chain=redinterna action=accept connection-state=established
      dst-address=170.80.10.0/24

15    chain=redinterna action=drop dst-address=170.80.10.0/24

16    chain=reddmz action=accept connection-state=established
      dst-address=170.80.20.0/24

17    chain=reddmz action=drop connection-state=invalid
      dst-address=170.80.20.0/24

18    chain=reddmz action=drop dst-address=170.80.10.0/24

19    chain=snif action=drop src-address=170.80.10.0/24 in-interface=ether2

20    chain=snif action=drop src-address=170.80.20.0/24 in-interface=ether2
```

Mikrotik 2

```
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=forward action=jump jump-target=snif

 1    chain=forward action=jump jump-target=redinterna
      dst-address=170.80.10.0/24

 2    chain=redinterna action=drop connection-state=invalid
      dst-address=170.80.10.0/24

 3    chain=redinterna action=accept connection-state=established
      dst-address=170.80.10.0/24

 4    chain=redinterna action=drop dst-address=170.80.10.0/24

 5    chain=snif action=drop src-address=170.80.10.0/24 in-interface=ether1
```

4º Verificar las políticas establecidas

Prevención de IP spoofing

```
root@Debian:~# hping3 -c 10 -p 80 --spoof 170.80.10.6 170.80.20.5
HPING 170.80.20.5 (eth0 170.80.20.5): NO FLAGS are set, 40 headers + 0 data byte
s

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

Bloqueo Tráfico ICM

```
root@Debian:~# nmap -n -T5 -sP -PI 170.80.10.0/24

Starting Nmap 6.47 ( http://nmap.org ) at 2020-05-09 03:02 UTC
Nmap done: 256 IP addresses (0 hosts up) scanned in 13.65 seconds
root@Debian:~# nmap -n -T5 -sP -PI 170.80.10.0/24

Starting Nmap 6.47 ( http://nmap.org ) at 2020-05-09 03:04 UTC
Nmap done: 256 IP addresses (0 hosts up) scanned in 13.32 seconds
```

Tráfico correspondiente a los servicios

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

Starting Nmap 6.47 ( http://nmap.org ) at 2020-05-09 03:29 UTC
Nmap scan report for 170.80.20.7
Host is up (0.0033s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
110/tcp open  pop3
143/tcp open  imap

Nmap done: 1 IP address (1 host up) scanned in 8.78 seconds
root@Debian:~# nmap -n -PN 170.80.20.6

Starting Nmap 6.47 ( http://nmap.org ) at 2020-05-09 03:29 UTC
Nmap scan report for 170.80.20.6
Host is up (0.0034s latency).
Not shown: 999 filtered ports
PORT   STATE SERVICE
21/tcp open  ftp

Nmap done: 1 IP address (1 host up) scanned in 7.15 seconds
root@Debian:~# nmap -n -PN 170.80.20.5

Starting Nmap 6.47 ( http://nmap.org ) at 2020-05-09 03:29 UTC
Nmap scan report for 170.80.20.5
Host is up (0.0033s latency).
Not shown: 998 filtered ports
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https

Nmap done: 1 IP address (1 host up) scanned in 8.77 seconds
```

Comprobar el tráfico de retorno
Conexión ssh desde PC-5 aPC-1

```
rtt min/avg/max/mdev = 1.172/5.139/203.219/5.680 ms
root@Debian:~# ssh testuser@80.10.30.2
testuser@80.10.30.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Apr  3 13:12:14 2018 from 170.80.10.2
testuser@Debian:~$
```

Referencias

- https://www.cisco.com/c/es_es/products/security/firewalls/what-is-a-firewall.html

- Enunciado de la práctica