Ejercicios Sencillos `iptables`

Ismael Macareno Chouikh

2025-01-31

Índice

1. Entorno

Es aconsejable tener varias máquinas virtuales, una principal donde configuraremos el *firewall* y una máquina adicional que representará un equipo de otra red que atraviesa el *firewall* para salir a internet.

1.1. Configuración para equipo que hace de *firewall*

1. Dos tarjetas de red
   - Una en modo puente (red 192.168.1.X/24)
   - La otra en red interna dentro de la red 10.0.1.10/24
2. Servicios:
   - FTP - puerto 21 (`vsftpd`)
   - Web - puerto 80 (`apache2`)
   - MySQL - puerto 3306

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet dhcp

allow-hotplug enp0s8
iface enp0s8 inet static
        address 10.0.1.10
        netmask 255.255.255.0
```

Listing 1: Macareno, Ismael (2025). Configuración de fichero `/etc/network/interfaces` [BASH]. Propio

1.2. Configuración para equipo que hace de cliente

1. Tarjeta de red
   - Configurada en modo red interna dentro de la red 10.0.1.11/24

ADVERTENCIA

Los ficheros .yaml son de sintáxis estrícta, nada de tabulaciones, solo espacios

1.3. Comprobaciones

Una vez establecida la configuración realiza un `ping` para comprobar la conectividad entre equipos conectados a la misma red interna.

```
# This file is generated from information provided by the datasource.  Changes
# to it will not persist across an instance reboot.  To disable cloud-init's
# network configuration capabilities, write a file
# /etc/cloud/cloud.cfg.d/99-disable-network-config.cfg with the following:
# network: {config: disabled}
network:
    ethernets:
        enp0s3:
            dhcp4: no
            addresses:
              - 10.0.1.11/24
            routes:
              - to: default
                via: 10.0.1.10
            nameservers:
              addresses: [8.8.8.8, 8.8.4.4]
    version: 2
```

Listing 2: Macareno, Ismael (2025). Configuración de fichero `/etc/netplan/X.yaml` [BASH]. Propio

2. Ejercicios

2.1. Ejercicio 1

Activar el enrutamiento en la máquina *firewall* indicando los pasos

Para activar el enrutamiento en la máquina *firewall* lo que habrá que hacer será modificar el fichero /etc/sysctl.conf de la siguiente manera

```
# Uncomment the next line to enable packet forwarding for IPv4
net.ipv4.ip_forward=1
```

2.2. Ejercicio 2

Indicar el comando para ver el contenido de las tablas de `iptables`

Para ver el contenido habrá que ejecutar el comando `iptables -L` y si quisiéramos especificar las tablas NAT `iptables -t nat -L`

2.3. Ejercicio 3

Indicar el comando que nos permite establecer una política restrictiva para *forward*, es decir, ningún paquete atraviesa el *server*

- `sudo iptables -P FORWARD DROP`

2.4. Ejercicio 4

¿Qué ocurre ahora una vez establecida esta política? ¿Atraviesan los paquetes el servidor? ¿Responde el servidor a un `ping`? ¿Y el equipo cliente a un `ping` desde el servidor?

1. No atraviesan los paquetes el servidor (la regla decía claramente que ningún paquete atraviesa el servidor)

2. Si responde a un `ping`, esto es debido a que la regla es para el reenvío no para tráfico
3. Si, el servidor podrá hacer `ping` a un cliente

2.5. Ejercicio 5

Revisa ahora el contenido de la tabla de filtros, ¿Ha cambiado algo con respecto a su estado inicial?

```
root@debian11:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
```

Como se puede apreciar no ha cambiado nada al estado inicial

2.6. Ejercicio 6

Deja la tabla *filter* como estaba

- `sudo iptables -F`

2.7. Ejercicio 7

Deniega todo el tráfico `icmp` y comprueba que efectivamente el *firewall* deja de responder a las peticiones `ping`

- `sudo iptables -A INPUT -p icmp --icmp-type echo-request -j DROP`

```
root@debian11:~# tcpdump -i enp0s8 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on enp0s8, link-type EN10MB (Ethernet), snapshot length 262144 bytes
22:10:25.025220 IP 10.0.1.11 > 10.0.1.10: ICMP echo request, id 5073, seq 1, length 64
22:10:26.065225 IP 10.0.1.11 > 10.0.1.10: ICMP echo request, id 5073, seq 2, length 64
22:10:27.088262 IP 10.0.1.11 > 10.0.1.10: ICMP echo request, id 5073, seq 3, length 64
22:10:28.112424 IP 10.0.1.11 > 10.0.1.10: ICMP echo request, id 5073, seq 4, length 64
22:10:29.136161 IP 10.0.1.11 > 10.0.1.10: ICMP echo request, id 5073, seq 5, length 64
22:10:30.161058 IP 10.0.1.11 > 10.0.1.10: ICMP echo request, id 5073, seq 6, length 64
^C
6 packets captured
6 packets received by filter
0 packets dropped by kernel
```

2.8. Ejercicio 8

Comprueba la existencia de esta última regla mostrando el número de línea y bórrala

- `sudo iptables -L INPUT --line-numbers`

```
root@debian11:~# iptables -L INPUT --line-numbers
Chain INPUT (policy ACCEPT)
num  target     prot opt source               destination
1    DROP       icmp --  anywhere             anywhere             icmp echo-request
```

- `sudo iptables -D INPUT 1`

2.9. Ejercicio 9

Deniega todo el tráfico de `ping` que venga desde la red 172.26.50.0

- `sudo iptables -A INPUT -s 172.26.50.0/24 -p icmp --icmp-type echo-request -j DROP`

2.10. Ejercicio 10

Bloquea todo el tráfico que venga de la IP 192.168.60.2

- `sudo iptables -A INPUT -s 192.168.60.2 -j DROP`

2.11. Ejercicio 11

Bloquea todo el tráfico que llega al puerto 80 (para la comprobación será necesario que instales un servidor web en dicho puerto)

- `sudo iptables -A INPUT -p tcp --dport 80 -j DROP`

```
maka:~/ $ wget http://192.168.1.128                                    [2
--2025-01-30 23:32:49--  http://192.168.1.128/
Connecting to 192.168.1.128:80...
```

como se puede apreciar en el bloque de código de arriba si intentamos desde otra máquina acceder al servidor web de la máquina se queda pensando.

2.12. Ejercicio 12

Bloquea todos el tráfico que llega al puerto 80 por la interfaz `enp0s3`

- `sudo iptables -A INPUT -i enp0s3 -p tcp --dport 80 -j DROP`

2.13. Ejercicio 13

Bloquea todo el tráfico que llegue al puerto 21 desde la IP 192.168.60.2

- `sudo iptables -A INPUT -s 192.168.60.2 -p tcp --dport 21 -j DROP`

2.14. Ejercicio 14

Bloquear todo el tráfico saliente a la IP 192.168.1.X/24

- `sudo iptables -A OUTPUT -d 192.168.1.128/24 -j DROP`

2.15. Ejercicio 15

Cerrar todos los puertos bien conocidos (1-1024)

- `sudo iptables -A INPUT -p tcp --dport 1:1024 -j DROP`