

Práctica 07. Una configuración sencilla de Firewall con IPtables

Seguridad de Sistemas Informáticos

Cheuk Kelly Ng Pante

November 10, 2023

Contents

1	Configurar las interfaces de red del firewall	2
2	Asignar máquina virtual “cliente” como cliente	3
3	Configurar el Firewall	5
4	Configuración de un proxy	7

1. Configurar las interfaces de red del firewall

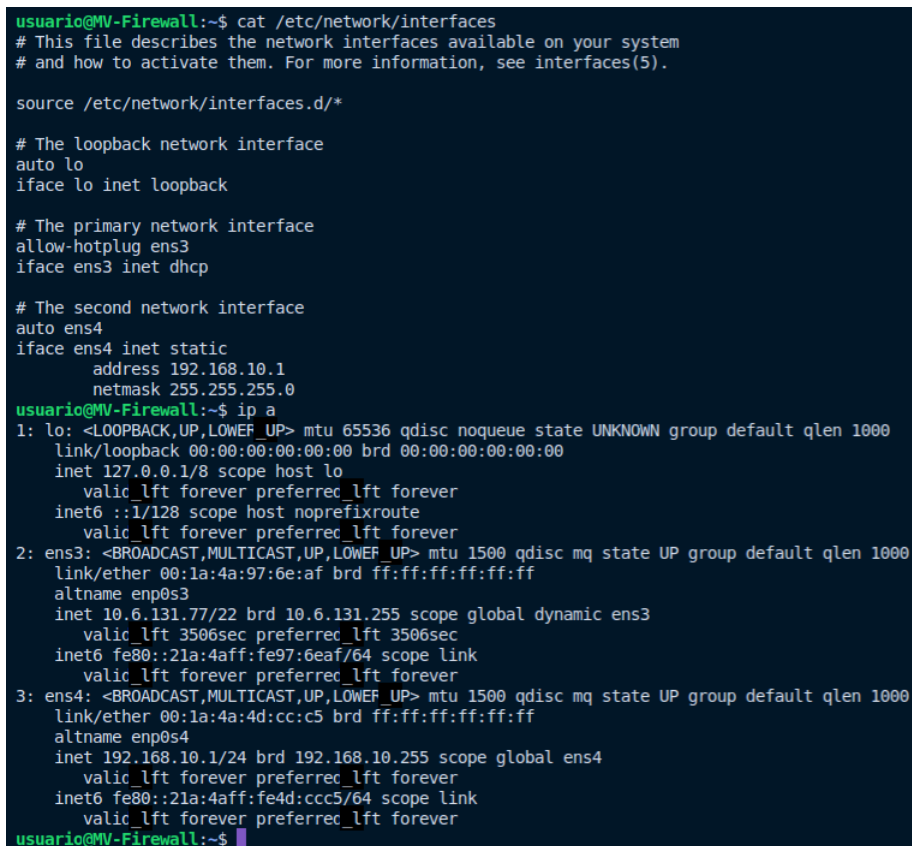
Para este punto debemos activar dos interfaces de red, esto lo hicimos en la práctica anterior, ya que una de las interfaces (DOCINT1) no se configuró ya que no teníamos la IP. En mi caso he seleccionado la IP 192.168.10.0/24. Para configurar la interfaz DOCINT1, ens4 en la MV del firewall, vamos a configurar el fichero `/etc/network/interfaces` y metemos lo siguiente:

```
auto ens4
iface ens4 inet static
address 192.168.10.1
netmask 255.255.255.0
```

Una vez añadido la configuración reiniciamos los servicios de red con el siguiente comando:

```
usuario@debian# sudo systemctl restart networking
```

Una vez reiniciado ya tenemos configurado la interfaz que se utilizará para trabajar sobre una red privada. Aquí una captura de pantalla de la configuración hecha:



```
usuario@MV-Firewall:~$ cat /etc/network/interfaces
# 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 ens3
iface ens3 inet dhcp

# The second network interface
auto ens4
iface ens4 inet static
    address 192.168.10.1
    netmask 255.255.255.0
usuario@MV-Firewall:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferrec_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferrec_lft forever
2: ens3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:1a:4a:97:6e:af brd ff:ff:ff:ff:ff:ff
    altname enp0s3
    inet 10.6.131.77/22 brd 10.6.131.255 scope global dynamic ens3
        valid_lft 3506sec preferrec_lft 3506sec
    inet6 fe80::21a:4aff:fe97:6eaf/64 scope link
        valid_lft forever preferrec_lft forever
3: ens4: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:1a:4a:4d:cc:c5 brd ff:ff:ff:ff:ff:ff
    altname enp0s4
    inet 192.168.10.1/24 brd 192.168.10.255 scope global ens4
        valid_lft forever preferrec_lft forever
    inet6 fe80::21a:4aff:fe4d:ccc5/64 scope link
        valid_lft forever preferrec_lft forever
usuario@MV-Firewall:~$
```

Figure 1.1: Configuración de las interfaces de red

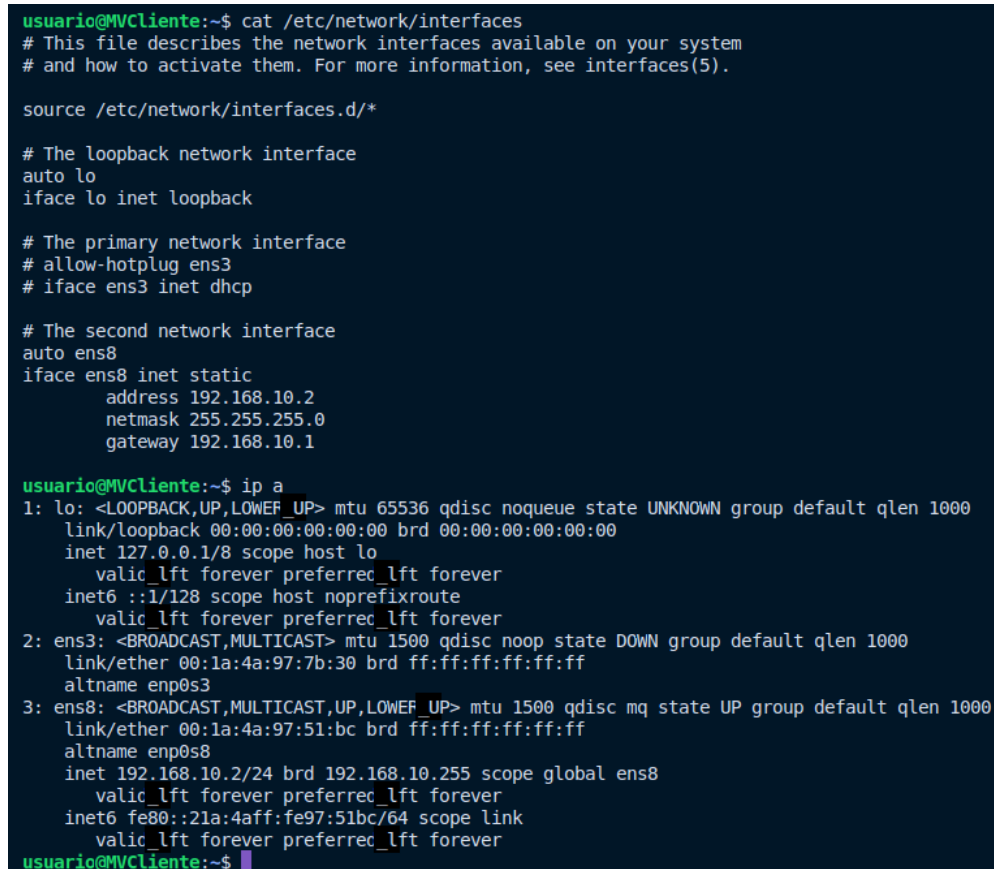
2. Asignar máquina virtual “cliente” como cliente

Para este punto vamos a configurar el fichero `/etc/network/interfaces` de la máquina virtual “cliente” y configuramos otra segunda interfaz y hacemos lo mismo que en el apartado anterior con la única diferencia que en esta interfaz que vamos a configurar añadimos un gateway que va a ser la de la “Red interna”. El `/etc/network/interfaces` vamos a añadir los siguiente:

```
auto ens8
iface ens8 inet statis
    address 192.168.10.2
    netmask 255.255.255.0
    gateway 192.168.10.1
```

Una vez añadido la configuración reiniciamos lo servicios de red con el siguiente comando:

```
usuario@debian# sudo systemctl restart networking
```

A terminal window showing the configuration of network interfaces. The user runs 'cat /etc/network/interfaces' and then 'ip a'. The output shows the configuration for 'lo', 'ens3', and 'ens8'. 'ens8' is configured with a static IP of 192.168.10.2 and a gateway of 192.168.10.1. The 'ip a' command shows the status of these interfaces, with 'ens8' being up and running.

```
usuario@MVCliente:~$ cat /etc/network/interfaces
# 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 ens3
# iface ens3 inet dhcp

# The second network interface
auto ens8
iface ens8 inet static
    address 192.168.10.2
    netmask 255.255.255.0
    gateway 192.168.10.1

usuario@MVCliente:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:1a:4a:97:7b:30 brd ff:ff:ff:ff:ff:ff
    altnam enp0s3
3: ens8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:1a:4a:97:51:bc brd ff:ff:ff:ff:ff:ff
    altnam enp0s8
    inet 192.168.10.2/24 brd 192.168.10.255 scope global ens8
        valid_lft forever preferred_lft forever
    inet6 fe80::21a:4aff:fe97:51bc/64 scope link
        valid_lft forever preferred_lft forever
usuario@MVCliente:~$
```

Figure 2.1: Configuración de las interfaces de red

A continuación, instalamos un navegador en modo texto, en este caso instalamos *links* y lo hacemos de la siguiente manera:

```
usuario@debian# sudo apt install links
```

```
usuario@MVCliente:~$ cat /etc/network/interfaces
# 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 ens3
# iface ens3 inet dhcp

# The second network interface
auto ens8
iface ens8 inet static
    address 192.168.10.2
    netmask 255.255.255.0
    gateway 192.168.10.1

usuario@MVCliente:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: ens3: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN group default qlen 1000
    link/ether 00:1a:4a:97:7b:30 brd ff:ff:ff:ff:ff:ff
    altnam enp0s3
3: ens8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group default qlen 1000
    link/ether 00:1a:4a:97:51:bc brd ff:ff:ff:ff:ff:ff
    altnam enp0s8
    inet 192.168.10.2/24 brd 192.168.10.255 scope global ens8
        valid_lft forever preferred_lft forever
    inet6 fe80::21a:4aff:fe97:51bc/64 scope link
        valid_lft forever preferred_lft forever
usuario@MVCliente:~$
```

Figure 2.2: Captura de pantalla del navegador *links*

3. Configurar el Firewall

- Tenga política por defecto ACCEPT para la cadena OUTPUT y DROP para INPUT y FORWARD:

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT
```

- Permita conectividad total desde la interfaz de loopback (lo) para hacer pruebas desde la consola del firewall:

```
iptables -A INPUT -i lo -j ACCEPT
```

- Acepte conexiones desde la red interna a los puertos 80 (web) y 22 (servicio SSH en el FW)

```
iptables -A INPUT -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -p tcp --sport 80 -j ACCEPT
```

- Crear una cadena "custom" denominada "SERVICES" para la gestión de los servicios anteriores:

```
# Crear la cadena "SERVICES"
iptables -N SERVICES

# Añadir reglas a la cadena "SERVICES"
iptables -A FORWARD -i ens4 -s 192.168.10.0/24 -j SERVICES

# Crear una cadena "SERVICES" y agregar reglas a ella
# Permitir trafico a los puertos 80 y 22
iptables -A SERVICES -p tcp --dport 22 -j ACCEPT
iptables -A SERVICES -p tcp --dport 80 -j ACCEPT
iptables -A SERVICES -p tcp --dport 443 -j ACCEPT
```

- Permita el tráfico a una impresora a todo el rango IP de la clase C especificada en la red interna:

```
iptables -A FORWARD -i ens4 -o ens4 -s 192.168.10.2 -d
192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -i ens4 -o ens4 -s 192.168.10.0/24 -d
192.168.10.2 -j ACCEPT
```

```

#!/bin/bash

iptables -F
iptables -t nat -F

# Establecer politicas por defecto
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT ACCEPT

# Permitir la conectividad desde la interfaz de loopback
iptables -A INPUT -i lo -j ACCEPT

# Permitir SSH, HTTP, DNS
iptables -A INPUT -p tcp --dport ssh -j ACCEPT
iptables -A INPUT -p tcp --sport 53 -j ACCEPT
iptables -A INPUT -p udp --sport 53 -j ACCEPT
iptables -A INPUT -p tcp --sport 80 -j ACCEPT
iptables -A INPUT -p tcp --sport 443 -j ACCEPT

# Permitir DNS en el cliente
iptables -A FORWARD -p tcp --sport 53 -j ACCEPT
iptables -A FORWARD -p udp --sport 53 -j ACCEPT
iptables -A FORWARD -p tcp --dport 53 -j ACCEPT
iptables -A FORWARD -p udp --dport 53 -j ACCEPT

# Crear una cadena "SERVICES" y agregar reglas a ella
# Permitir trafico a los puertos 80 y 22
iptables -N SERVICES
iptables -A FORWARD -i ens4 -s 192.168.10.0/24 -j SERVICES
iptables -A SERVICES -p tcp --dport 22 -j ACCEPT
iptables -A SERVICES -p tcp --dport 80 -j ACCEPT
iptables -A SERVICES -p tcp --dport 443 -j ACCEPT

# Para poder hacer ping
iptables -A INPUT -p icmp -j ACCEPT

# Permitir trafico a una impresora desde la red interna
iptables -A FORWARD -i ens4 -o ens4 -s 192.168.10.2 -d 192.168.10.0/24 -j ACCEPT
iptables -A FORWARD -i ens4 -o ens4 -s 192.168.10.0/24 -d 192.168.10.2 -j ACCEPT

# Permitir NAT y redirigir el tráfico del puerto 80 al 8080
iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.10.1:8080
iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
iptables -A INPUT -p tcp --dport 8080 -j ACCEPT

```

Figure 3.1: Captura de pantalla del script de *firewall*

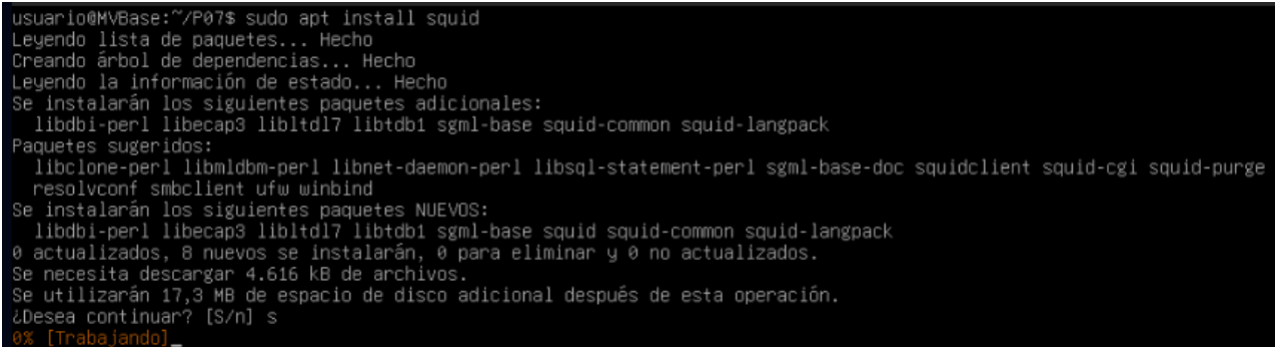
4. Configuración de un proxy

Para implementar un proxy transparente para que los clientes de la red interna puedan navegar por internet vamos a implementar otro script en bash que va a tener el siguiente contenido:

```
#!/bin/bash

iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to-destination 192.168.10.1:8080
iptables -A INPUT -p tcp --dport 8080 -j ACCEPT
iptables -t nat -A POSTROUTING -o ens3 -j MASQUERADE
```

Para poder utilizar un proxy transparente para dar servicio de navegación a los clientes de la red interna, vamos a instalar *Squid Cache*:



```
usuario@MVBBase:~/P07$ sudo apt install squid
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Se instalarán los siguientes paquetes adicionales:
  libdbi-perl libecap3 libltdl7 libtdb1 sgml-base squid-common squid-langpack
Paquetes sugeridos:
  libclone-perl libmldbm-perl libnet-daemon-perl libsql-statement-perl sgml-base-doc squidclient squid-cgi squid-purge
  resolvconf smbclient ufw winbind
Se instalarán los siguientes paquetes NUEVOS:
  libdbi-perl libecap3 libltdl7 libtdb1 sgml-base squid squid-common squid-langpack
0 actualizados, 8 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
Se necesita descargar 4.616 kB de archivos.
Se utilizarán 17,3 MB de espacio de disco adicional después de esta operación.
¿Desea continuar? [S/n] s
0% [Trabajando]
```

Figure 4.1: Instalación del *squid*

Una vez instalado el *squid* vamos a configurar el fichero `/etc/squid/squid.conf` y añadimos lo siguiente:

```
acl localnet src 192.168.10.0/24
http_port 192.168.10.1:8080 intercept
```

y buscamos la línea `http_access deny CONNECT !SSL_Ports` y la comentamos, quedandose el fichero de configuración como se muestra en la figura 4.2


```

acl localnet src 0.0.0.1-0.255.255.255 # RFC 1122 "this" network (LAN)
acl localnet src 10.0.0.0/8           # RFC 1918 local private network (LAN)
acl localnet src 100.64.0.0/10        # RFC 6598 shared address space (CGN)
acl localnet src 169.254.0.0/16       # RFC 3927 link-local (directly plugged) machines
acl localnet src 172.16.0.0/12        # RFC 1918 local private network (LAN)
acl localnet src 192.168.0.0/16       # RFC 1918 local private network (LAN)
acl localnet src fc00::/7             # RFC 4193 local private network range
acl localnet src fe80::/10           # RFC 4291 link-local (directly plugged) machines

acl localnet src 192.168.10.0/24

acl SSL_ports port 443
acl Safe_ports port 80               # http
acl Safe_ports port 21               # ftp
acl Safe_ports port 443              # https
acl Safe_ports port 70               # gopher
acl Safe_ports port 210              # wais
acl Safe_ports port 1025-65535       # unregistered ports
acl Safe_ports port 280              # http-mgmt
acl Safe_ports port 488              # gss-http
acl Safe_ports port 591              # filemaker
acl Safe_ports port 777              # multiling http

http_access deny !Safe_ports
#http_access deny CONNECT !SSL_ports
http_access allow localhost manager
http_access deny manager
include /etc/squid/conf.d/*.conf
http_access allow localnet
http_access allow localhost
http_access deny all
http_port 3128

http_port 192.168.10.1:8080 intercept

coredump_dir /var/spool/squid
refresh_pattern ^ftp:               1440  20%  10080
refresh_pattern ^gopher:            1440  0%   1440
refresh_pattern -i (/cgi-bin/|\?)  0     0%    0
refresh_pattern .                    0     20%  4320

```

Figure 4.2: Configuración del *squid*

Ahora, reiniciamos el servicio de *squid*:

```
usuario@debian# sudo systemctl restart squid.service
```

Una vez que se haya reiniciado probamos en la máquina cliente si hay conexión a internet:

```

usuario@MVCiente:~$ wget www.google.es
--2023-11-09 17:14:26-- http://www.google.es/
Resolviendo www.google.es (www.google.es)... 142.250.184.163, 2a00:1450:4003:80c::2003
Conectando con www.google.es (www.google.es)[142.250.184.163]:80... conectado.
Petición HTTP enviada, esperando respuesta... 200 OK
Longitud: no especificado [text/html]
Grabando a: «index.html.1»

index.html.1 [ <=> ] 19,11K --.-KB/s en 0,03s
2023-11-09 17:14:26 (616 KB/s) - «index.html.1» guardado [19567]

```

Figure 4.3: wget a *Google*

[illegible]

9