PRÁCTICA 4: FILTRADO DE PAQUETES

Inés del Río y Paloma Pérez de Madrid

ÍNDICE

ÍNDICE

1. CORTAFUEGOS PERSONAL

Índice

1. CORTAFUEGOS PERSONAL

1.1 Configuración básica del cortafuegos: reglas de filtrado sin estado.

Establecer distintas reglas de filtrado sin estado sobre las cadenas INPUT y OUTPUT. Filtraremos ambos sentidos de la conexión, con reglas complementarias. Fijar en primer lugar la política DROP para el tráfico entrante y saliente del equipo, y habilitar a continuación el intercambio ICMP con el resto de los equipos de nuestra subred. Permitir el tráfico saliente hacia los puertos 80/tcp, 443/tcp y 53/udp (nuestro equipo actúa como cliente). Habilitar el tráfico entrante al puerto 22/tcp y 80/tcp (nuestro equipo actúa como servidor).

Kali
192.168.64.128

Ubuntu
192.168.64.129

1. CORTAFUEGOS PERSONAL

1.1 Configuración básica del cortafuegos: reglas de filtrado sin estado.

- Fijar en primer lugar la política DROP para el tráfico entrante y saliente del equipo, y habilitar a continuación el intercambio ICMP con el resto de los equipos de nuestra subred.

```
# iptables -P INPUT DROP (Denegar todo tráfico entrante)
# iptables -P OUTPUT DROP (Denegar todo tráfico saliente)
```

```
root@server:~# iptables -P INPUT DROP
root@server:~# iptables -P OUTPUT DROP
palomaperezdemadrid@MBP-de-Paloma ~ % ping 192.168.64.129
PING 192.168.64.129 (192.168.64.129): 56 data bytes
Request timeout for icmp_seq 0
Request timeout for icmp_seq 1
Request timeout for icmp_seq 2
Request timeout for icmp_seq 3
Request timeout for icmp_seq 4
Request timeout for icmp_seq 5
^C
--- 192.168.64.129 ping statistics ---
7 packets transmitted, 0 packets received, 100.0% packet loss
```

```
palomaperezdemadrid@MBP-de-Paloma ~ % ping 192.168.64.129
PING 192.168.64.129 (192.168.64.129): 56 data bytes
64 bytes from 192.168.64.129: icmp_seq=0 ttl=64 time=0.747 ms
64 bytes from 192.168.64.129: icmp_seq=1 ttl=64 time=0.747 ms
^C
--- 192.168.64.129 ping statistics ---
2 packets transmitted, 2 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.747/0.747/0.747/nan ms
```

- Permitir el tráfico saliente hacia los puertos 80/tcp, 443/tcp y 53/udp (nuestro equipo actúa como cliente).

  - Permitimos tráfico ICMP (ping) entre Ubuntu (192.168.64.129) y la subred (192.168.64.0/24):

```
# iptables -A INPUT -p icmp -s 192.168.64.0/24 -j ACCEPT
# iptables -A OUTPUT -p icmp -d 192.168.64.0/24 -j ACCEPT
```

```
root@server:~# iptables -A INPUT -p icmp -s 192.168.64.0/24 -j ACCEPT
root@server:~# iptables -A OUTPUT -p icmp -d 192.168.64.0/24 -j ACCEPT
```

1. CORTAFUEGOS PERSONAL

1.1 Configuración básica del cortafuegos: reglas de filtrado sin estado.

- Permitir tráfico saliente para puertos específicos (80/tcp, 443/tcp, 53/udp)

```
# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
# iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
# iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
```

```
root@server:~# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
root@server:~# iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
root@server:~# iptables -A OUTPUT -p udp  --dport 53 -j ACCEPT
```

- Permitir tráfico entrante en puertos específicos (22/tcp, 80/tcp)

```
# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
root@server:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@server:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

1. CORTAFUEGOS PERSONAL

1.1 Configuración básica del cortafuegos: reglas de filtrado sin estado.

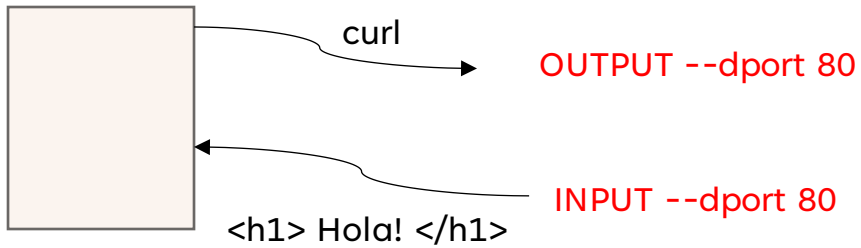
- Permitir tráfico saliente para puertos específicos (80/tcp, 443/tcp, 53/udp) [Actúa como **cliente**]

```
# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
# iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
# iptables -A OUTPUT -p udp --dport 53 -j ACCEPT
```

```
root@server:~# iptables -A OUTPUT -p tcp --dport 80 -j ACCEPT
root@server:~# iptables -A OUTPUT -p tcp --dport 443 -j ACCEPT
root@server:~# iptables -A OUTPUT -p udp  --dport 53 -j ACCEPT
```

Configuramos también la respuesta

```
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
# iptables -A INPUT -p tcp --dport 443 -j ACCEPT
# iptables -A INPUT -p udp --dport 53 -j ACCEPT
```



- Permitir tráfico entrante en puertos específicos (22/tcp, 80/tcp) [Actúa como **servidor**]

```
# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
```

```
root@server:~# iptables -A INPUT -p tcp --dport 80 -j ACCEPT
root@server:~# iptables -A INPUT -p tcp --dport 22 -j ACCEPT
```

```
# iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
# iptables -A OUTPUT -p tcp --sport 80 -j ACCEPT
```

Nota: Una máquina se puede conectar a mi (INPUT), pero yo le tendré que enviar los datos (OUTPUT)

1. CORTAFUEGOS PERSONAL

1.2 Efecto del orden en las reglas de filtrado.

Comprobar el efecto que tiene el orden de las reglas de filtrado. Partiendo de una configuración sin filtros, bloquear todo el tráfico TCP saliente salvo el dirigido a una dirección IP concreta de nuestra subred, viendo el efecto del orden en la entrada de las dos reglas necesarias.

Permitir tráfico TCP saliente hacia la IP concreta (192.168.64.128, Kali).

```
# iptables -A OUTPUT -p tcp -d 192.168.64.128 -j ACCEPT
```

Bloquear todo el tráfico TCP saliente como regla general.

```
# iptables -A OUTPUT -p tcp -j DROP
```

Nota: borramos reglas # iptables –F

También hay que cambiar las políticas por defecto (iptables –P) :

```
# iptables -P INPUT ACCEPT
# iptables -P OUTPUT ACCEPT
```

1. CORTAFUEGOS PERSONAL

1.2 Efecto del orden en las reglas de filtrado.

- Comprobar configuración actual (iptables –L –v)

```
root@server:~# iptables -L -v
Chain INPUT (policy DROP 45 packets, 3660 bytes)
 pkts bytes target     prot opt in     out     source               destination

    7   588 ACCEPT     icmp --  any    any     192.168.64.0/24      anywhere
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:http
    9   576 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:ssh

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy DROP 30 packets, 3610 bytes)
 pkts bytes target     prot opt in     out     source               destination
    7   588 ACCEPT     icmp --  any    any     anywhere             192.168.64.0/24
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:http
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:https
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:http
    0     0 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp dpt:https
   16  1232 ACCEPT     udp  --  any    any     anywhere             anywhere             udp dpt:domain
root@server:~#
```

1. CORTAFUEGOS PERSONAL

1.2 Efecto del orden en las reglas de filtrado.

Comprobar el efecto que tiene el orden de las reglas de filtrado. Partiendo de una configuración sin filtros, bloquear todo el tráfico TCP saliente salvo el dirigido a una dirección IP concreta de nuestra subred, viendo el efecto del orden en la entrada de las dos reglas necesarias.

```
root@server:~# curl 192.168.64.128
Esta es la Maquina Kali
root@server:~# curl 192.168.64.1
<!DOCTYPE HTML>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Directory listing for /</title>
</head>
<body>
<h1>Directory listing for /</h1>
<hr>
<ul>
<li><a href=".CFUserTextEncoding">.CFUserTextEncoding</a></li>
<li><a href=".DS_Store">.DS_Store</a></li>
<li><a href=".gitconfig">.gitconfig</a></li>
<li><a href=".idlerc/">.idlerc/</a></li>
<li><a href=".node_repl_history">.node_repl_history</a></li>
<li><a href=".npm/">.npm/</a></li>
```

ANTES de las reglas iptables.

1. CORTAFUEGOS PERSONAL

1.2 Efecto del orden en las reglas de filtrado.

Comprobar el efecto que tiene el orden de las reglas de filtrado. Partiendo de una configuración sin filtros, bloquear todo el tráfico TCP saliente salvo el dirigido a una dirección IP concreta de nuestra subred, viendo el efecto del orden en la entrada de las dos reglas necesarias.

```
root@server:~# iptables -A OUTPUT -p tcp -d 192.168.64.128 -j ACCEPT
root@server:~# iptables -D OUTPUT -p tcp -j DROP
iptables: Bad rule (does a matching rule exist in that chain?).
root@server:~# iptables -A OUTPUT -p tcp -d 192.168.64.128 -j ACCEPT
root@server:~# iptables -A OUTPUT -p tcp -j DROP
root@server:~# curl 192.168.64.128
Esta es la Maquina Kali
root@server:~# curl 192.168.64.1
^C
```

DESPUÉS de las reglas iptables.
No se puede llegar al servidor de 192.168.64.1 (TCP) pero sí al de Kali (.128)

Si cambiamos el orden de las reglas, no se puede acceder a ningún servicio web de ninguna máquina

```
root@server:~# iptables -F
root@server:~# iptables -A OUTPUT -p tcp -j DROP
root@server:~# iptables -A OUTPUT -p tcp -d 192.168.64.128 -j ACCEPT
root@server:~# curl 192.168.64.1
^C
root@server:~# curl 192.168.64.128
curl: (7) Couldn't connect to server
root@server:~#
```

1. CORTAFUEGOS PERSONAL

1.3 Redirección de tráfico SSH saliente.

Definir una regla que redirija todo el tráfico SSH saliente que va hacia una dirección IP concreta (por ejemplo, 1.2.3.4) para que se redirija a la IP de un servidor real que tenga el servicio SSH activo. Para probar su funcionamiento, abrir una sesión SSH contra 1.2.3.4.

iptables -t nat -A OUTPUT -p tcp --dport 22 -d 1.2.3.4 -j DNAT --to-destination 192.168.64.128:22

- -t nat: trabajando con la tabla de NAT.
- -A OUTPUT
- -p tcp --dport 22
- -d 1.2.3.4: IP de destino (1.2.3.4) hacia la cual está siendo dirigido el tráfico.
- -j DNAT --to-destination 192.168.64.128:22: Redirige el tráfico a la dirección IP interna (IP de Kali) y al puerto 22 (el puerto SSH del servidor real).

1. CORTAFUEGOS PERSONAL

1.3 Redirección de tráfico SSH saliente.

```
root@server:~# ssh kali@1.2.3.4
^C
root@server:~# iptables -t nat -A OUTPUT -p tcp --dport 22 -d 1.2.3.4 -j DNAT --to-destination 192.168.64.128:22
root@server:~# ssh kali@1.2.3.4
The authenticity of host '1.2.3.4 (1.2.3.4)' can't be established.
ED25519 key fingerprint is SHA256:cahk3X8QK77XC4Be2FMd11VCHE1Ec3z7kPyKSMHfAZo.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '1.2.3.4' (ED25519) to the list of known hosts.
kali@1.2.3.4's password:
Linux kali 6.8.11-arm64 #1 SMP Kali 6.8.11-1kali2 (2024-05-30) aarch64

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

Kali GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Dec 13 17:26:07 2024 from 192.168.64.129
┌──(kali㉿kali)-[~]
└─$ 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: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:73:ac:aa brd ff:ff:ff:ff:ff:ff
    inet 192.168.64.128/24 brd 192.168.64.255 scope global dynamic noprefixroute eth0
       valid_lft 1739sec preferred_lft 1739sec
    inet6 fe80::20c:29ff:fe73:acaa/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

1. CORTAFUEGOS PERSONAL

1.4 Protección contra escaneo de sistema operativo con nmap

Desde una máquina remota, utilizar **nmap** para identificar el sistema operativo de nuestra máquina virtual Linux. A continuación, aplicar una regla con estado para impedir conexiones TCP inválidas (las que no se inician con el segmento SYN). Volver a utilizar la herramienta nmap para identificar el SO y comentar los nuevos resultados obtenidos.

```
┌──(root㉿kali)-[/var/www/html]
└─# nmap 192.168.64.129 -O
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-12-13 17:32 CET
Nmap scan report for 192.168.64.129
Host is up (0.00057s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:C5:48:05 (VMware)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.49 seconds
```

Antes de añadir un filtro

Añadiremos el siguiente filtro:

iptables -A INPUT -p tcp ! --syn -m state --state NEW -j DROP

- -p tcp: tráfico TCP.
- ! --syn: Filtra las conexiones que no inician con el segmento SYN.
- -m state --state NEW: Aplica esta regla solo a las nuevas conexiones.
- -j DROP: Las conexiones que no cumplen con estos requisitos se bloquean.

1. CORTAFUEGOS PERSONAL

1.4 Protección contra escaneo de sistema operativo con nmap

Desde una máquina remota, utilizar **nmap** para identificar el sistema operativo de nuestra máquina virtual Linux. A continuación, aplicar una regla con estado para impedir conexiones TCP inválidas (las que no se inician con el segmento SYN). Volver a utilizar la herramienta nmap para identificar el SO y comentar los nuevos resultados obtenidos.

Después de añadir el filtro

```
Nmap scan report for 192.168.64.129
Host is up (0.0011s latency).
Not shown: 999 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
MAC Address: 00:0C:29:C5:48:05 (VMware)
No exact OS matches for host (If you know what OS is running on it, see https
://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.94SVN%E=4%D=12/13%OT=22%CT=1%CU=42343%PV=Y%DS=1%DC=D%G=Y%M=000C
OS:29%TM=675C6313%P=aarch64-unknown-linux-gnu)SEQ(SP=105%GCD=1%ISR=10A%TI=Z
OS:%CI=Z%II=I%TS=A)SEQ(SP=106%GCD=1%ISR=10A%TI=Z%CI=Z%II=I%TS=A)OPS(O1=M5B4
OS:ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%O5=M5B4ST11NW7%O6=
OS:M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W6=FE88)ECN(R=Y%DF=
OS:Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+%F=AS%RD=0%Q
OS:=)T2(R=N)T3(R=N)T4(R=N)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(
OS:R=N)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=1
OS:64%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)

Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.o
rg/submit/ .
```

1. CORTAFUEGOS PERSONAL

1.5 Reglas con estado para conexiones y tráfico ICMP.

Partir nuevamente de una configuración sin filtros y bloquear todo el tráfico entrante y saliente. A continuación, definir reglas con estado que permitan iniciar conexiones TCP o UDP hacia el exterior, así como el tráfico ICMP siempre que sea iniciado por nosotros o relacionado con nuestras conexiones. Comprobarlo iniciando conexiones exteriores, intentando conectar desde el exterior (por ejemplo, al servidor SSH) y lanzando o recibiendo mensajes de alcanzabilidad (ping).

```
# iptables -P INPUT DROP (Denegar todo tráfico entrante)
# iptables -P OUTPUT DROP (Denegar todo tráfico saliente)
```

Permitir conexiones TCP y UDP

```
# iptables -A OUTPUT -p tcp -m state --state NEW,ESTABLISHED -j ACCEPT
# iptables -A OUTPUT -p udp -m state --state NEW,ESTABLISHED -j ACCEPT

# iptables -A INPUT -p tcp -m state --state ESTABLISHED -j ACCEPT
# iptables -A INPUT -p udp -m state --state ESTABLISHED -j ACCEPT
```

Permitir tráfico ICMP saliente y entrante que esté relacionado con nuestras conexiones

```
# iptables -A OUTPUT -p icmp -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT
# iptables -A INPUT -p icmp -m state --state RELATED,ESTABLISHED -j ACCEPT
```

-m state --state NEW,RELATED,ESTABLISHED: Permite tráfico ICMP que sea parte de una nueva conexión, relacionada o parte de una conexión establecida.

1. CORTAFUEGOS PERSONAL

1.5 Reglas con estado para conexiones y tráfico ICMP.

Partir nuevamente de una configuración sin filtros y bloquear todo el tráfico entrante y saliente. A continuación, definir reglas con estado que permitan iniciar conexiones TCP o UDP hacia el exterior, así como el tráfico ICMP siempre que sea iniciado por nosotros o relacionado con nuestras conexiones. Comprobarlo iniciando conexiones exteriores, intentando conectar desde el exterior (por ejemplo, al servidor SSH) y lanzando o recibiendo mensajes de alcanzabilidad (ping).

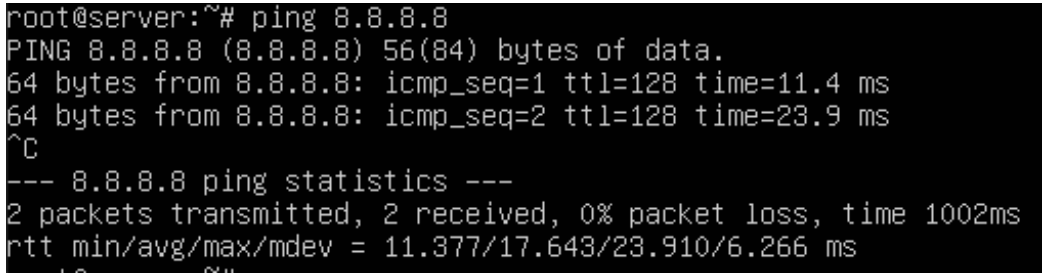
```
root@server:~# iptables -L -v
Chain INPUT (policy DROP 104 packets, 7596 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     icmp --  any    any     anywhere             anywhere             state RELATED,ESTABLISHED
    0     0 ACCEPT     all  --  any    any     anywhere             anywhere             state RELATED,ESTABLISHED

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy DROP 19 packets, 2188 bytes)
 pkts bytes target     prot opt in     out     source               destination
    6   360 ACCEPT     tcp  --  any    any     anywhere             anywhere             tcp flags:FIN,SYN,RST,ACK/SYN state NEW
    5   622 ACCEPT     udp  --  any    any     anywhere             anywhere             state NEW
    0     0 ACCEPT     icmp --  any    any     anywhere             anywhere             state NEW,RELATED,ESTABLISHED
```

1. CORTAFUEGOS PERSONAL

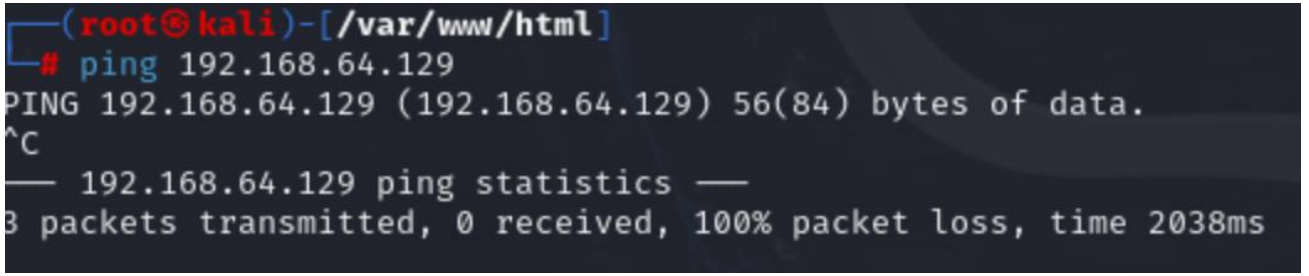
1.5 Reglas con estado para conexiones y tráfico ICMP.

Partir nuevamente de una configuración sin filtros y bloquear todo el tráfico entrante y saliente. A continuación, definir reglas con estado que permitan iniciar conexiones TCP o UDP hacia el exterior, así como el tráfico ICMP siempre que sea iniciado por nosotros o relacionado con nuestras conexiones. Comprobarlo iniciando conexiones exteriores, intentando conectar desde el exterior (por ejemplo, al servidor SSH) y lanzando o recibiendo mensajes de alcanzabilidad (ping).

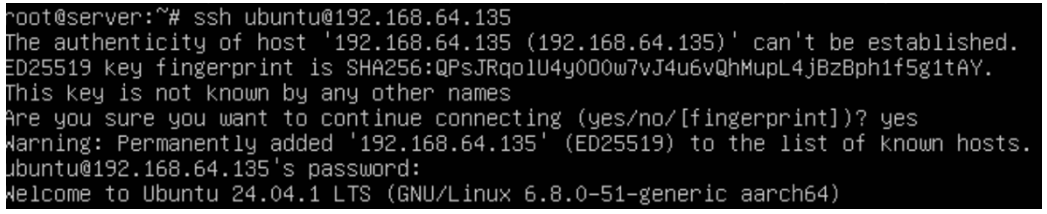
ICMP desde Ubuntu hacia el exterior (funciona)



ICMP desde exterior (Kali) a Ubuntu (no debería funcionar)



SSH (TCP) desde Ubuntu hacia el exterior (funciona)



SSH desde exterior (Kali) a Ubuntu (no debería funcionar)

1. CORTAFUEGOS PERSONAL

1.6 Limitación de accesos al servidor SSH para mitigar ataques de fuerza bruta.

Considere la siguiente situación: Se están produciendo accesos continuos a nuestro servidor OpenSSH para intentar descubrir mediante fuerza bruta usuarios y contraseñas válidas del sistema. Nos gustaría limitar el número de accesos desde una única dirección IP origen (por ejemplo, un máximo de 2 conexiones cada 180 segundos desde cada IP), para minimizar el riesgo este tipo de intentos sin afectar a los usuarios legítimos. Busque una solución para este problema mediante iptables, e indique la secuencia de reglas que habría que aplicar para implantar esta política y la función que cumple cada de ellas. Para acotar más aún estos intentos de acceso es posible limitar también el número máximo de reintentos de autentificación en cada sesión (por ejemplo, a dos). ¿Cómo podríamos configurar esta nueva restricción?

Limitar el número de intentos de conexión por IP

```
# iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
# iptables -A INPUT -p tcp --dport 22 -m recent --name ssh --set
```

- **-m recent --name ssh**: Utiliza el módulo recent para hacer un seguimiento de las direcciones IP que han intentado conectarse. Usamos el nombre ssh para identificar las entradas relacionadas con intentos de SSH.
- **--set:** se registra la IP de la máquina que intenta conectarse al servidor SSH en el conjunto ssh

1. CORTAFUEGOS PERSONAL

1.6 Limitación de accesos al servidor SSH para mitigar ataques de fuerza bruta.

Limitar el número de intentos de conexión por IP

```
# iptables -A INPUT -p tcp --dport 22 -m recent --name --update --seconds 180 --hitcount 3 -j DROP
```

- **--update:** indica a iptables que debe **actualizar** el tiempo de la IP dentro del conjunto ssh en lugar de simplemente registrarla
- **--seconds 180:** Establece un periodo de tiempo de 180 segundos para realizar el seguimiento.
- **--hitcount 3:** Si una IP realiza más de 2 intentos en ese intervalo de tiempo (lo que sería el tercer intento), se aplica la acción especificada (-j DROP).

```
palomaperezdemadrid@MBP-de-Paloma ~ % ssh root@192.168.64.129
root@192.168.64.129's password:
Permission denied, please try again.
root@192.168.64.129's password:
Permission denied, please try again.
root@192.168.64.129's password:
root@192.168.64.129: Permission denied (publickey,password).
palomaperezdemadrid@MBP-de-Paloma ~ % ssh root@192.168.64.129
root@192.168.64.129's password:
Permission denied, please try again.
root@192.168.64.129's password:

palomaperezdemadrid@MBP-de-Paloma ~ % ssh root@192.168.64.129
```

1

2

3

3º intento, bloqueado

1. CORTAFUEGOS PERSONAL

1.6 Limitación de accesos al servidor SSH para mitigar ataques de fuerza bruta.

```
root@server:~# iptables -P FORWARD DROP
root@server:~# iptables -P INPUT DROP
root@server:~# iptables -P OUTPUT DROP
root@server:~# iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
root@server:~# iptables -A INPUT -p tcp --dport 22 -m recent --name ssh --set
root@server:~# iptables -A INPUT -p tcp --dport 22 -m recent --name ssh --update --seconds 180 --hitcount 3 -j DROP
root@server:~#
```

Limitar el número de reintentos de autenticación dentro de una sesión

Editar archivo ssh:

vim /etc/ssh/sshd_config

Añadir Max Tries (MaxAuthTries 2)

systemctl restart sshd


```
┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
Permission denied, please try again.
root@192.168.64.129's password:
Received disconnect from 192.168.64.129 port 22:2: Too many authentication failures
Disconnected from 192.168.64.129 port 22
```

Sólo deja 2 intentos, antes dejaba 6

ÍNDICE

2. CORTAFUEGOS PERSONAL CON UFW

Índice

a) Verificación del estado del cortafuegos y reglas iptables existentes
b) Arranque de un servicio para pruebas (Apache2, OpenSSH Server, etc.)
c) Arranque de UFW y configuración de su arranque automático
d) Comprobación del estado de UFW y visualización de reglas iptables
e) Configuración de la política por defecto de UFW (permitir/denegar tráfico entrante/saliente)
f) Establecimiento de una política por defecto para conexiones salientes y entrantes
g) Permitir conexiones entrantes a los servicios específicos
h) Limitación de conexiones desde una IP origen para mitigar ataques de diccionario (Acceso limitado al servidor SSH)

2. CORTAFUEGOS PERSONAL CON UFW

a) Verificación del estado del cortafuegos y reglas iptables existentes

```
root@server:~# iptables -L -v
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@server:~# _
```

No hay reglas iptables

b) Arranque de un servicio para pruebas (Apache2, OpenSSH Server, etc.)

```
# apt update
# apt install apache2
# systemctl start apache2
# systemctl enable apache2
# vim /var/www/html/index.html (cambiar el index)
```

```
root@server:~# curl localhost
Maquina Ubuntu, IP 192.168.64.129
```

2. CORTAFUEGOS PERSONAL CON UFW

c) Arranque de UFW y configuración de su arranque automático

apt install ufw

ufw enable (iniciar cortafuegos)

ufw status (verificar)

```
root@server:~# ufw enable
Firewall is active and enabled on system startup
root@server:~# ufw status
Status: active
```

d) Comprobación del estado de UFW y visualización de reglas iptables

ufw status

iptables –L -n

```
Chain INPUT (policy DROP)
target     prot opt source               destination
ufw-before-logging-input  all  --  anywhere             anywhere
ufw-before-input  all  --  anywhere             anywhere
ufw-after-input  all  --  anywhere             anywhere
ufw-after-logging-input  all  --  anywhere             anywhere
ufw-reject-input  all  --  anywhere             anywhere
ufw-track-input  all  --  anywhere             anywhere
ACCEPT     tcp  --  anywhere             192.168.64.0/24

Chain FORWARD (policy DROP)
target     prot opt source               destination
ufw-before-logging-forward  all  --  anywhere             anywhere
ufw-before-forward  all  --  anywhere             anywhere
ufw-after-forward  all  --  anywhere             anywhere
```

Son las reglas ufw por defecto, también las puedes ver con:

ufw show raw

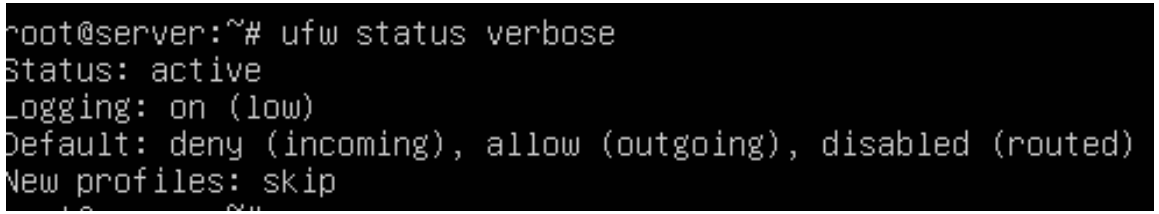
```
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
    pkts      bytes target     prot opt in     out     source               destination
```

2. CORTAFUEGOS PERSONAL CON UFW

e) Configuración de la política por defecto de UFW (permitir/denegar tráfico entrante/saliente)

Usar el comando ufw default allow|deny incoming|outgoing para definir la configuración por defecto del cortafuegos personal. Comprobar el efecto de estos comandos intentando conexiones hacia o desde el equipo.

ufw status verbose



Cambiar las políticas por defecto:

- Permitir tráfico entrante: # ufw default allow incoming
- Permitir tráfico saliente: # ufw default allow outgoing
- Denegar tráfico entrante: # ufw default deny incoming
- Denegar tráfico saliente: # ufw default deny outgoing

2. CORTAFUEGOS PERSONAL CON UFW

e) Configuración de la política por defecto de UFW (permitir/denegar tráfico entrante/saliente)

ufw default deny incoming (Bloquear todo el tráfico entrante)

ufw default allow outgoing (Permitir todo el tráfico saliente)

```
root@server:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

```
root@server:~# ufw default deny incoming
Default incoming policy changed to 'deny'
(be sure to update your rules accordingly)
root@server:~# ufw default allow outgoing
Default outgoing policy changed to 'allow'
(be sure to update your rules accordingly)
```

Desde otro equipo intento SSH (no debería funcionar)

```
┌──(kali㉿kali)-[~]
└─$ curl 192.168.64.129
Maquina Ubuntu, IP 192.168.64.129

┌──(kali㉿kali)-[~]
└─$ curl 192.168.64.129
^C
```

ANTES de las reglas

DESPUÉS de las reglas

Desde mi Ubuntu puedo hacer ping a Google (debería funcionar)

```
root@server:~# ping www.google.com
PING www.google.com (216.58.215.164) 56(84) bytes of dat
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164):
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164):
64 bytes from mad41s07-in-f4.1e100.net (216.58.215.164):
^C
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time
rtt min/avg/max/mdev = 7.427/9.647/11.701/1.748 ms
```

2. CORTAFUEGOS PERSONAL CON UFW

f) Establecer una política por defecto que permita conexiones salientes e impida conexiones entrantes.

ufw default deny incoming (Bloquear todo el tráfico entrante)

ufw default allow outgoing (Permitir todo el tráfico saliente)

Demostración en el apartado anterior

g) Sobre la configuración anterior permitir conexiones entrantes a los servicios arrancados en el apartado b) (ufw allow ...)

ufw allow 80/tcp

```
root@server:~# ufw allow 80/tcp
Rule added
Rule added (v6)
```

```
┌──(kali㉿kali)-[~]
└─$ curl 192.168.64.129
^C

┌──(kali㉿kali)-[~]
└─$ curl 192.168.64.129
Maquina Ubuntu, IP 192.168.64.129
```

Apartado f)

DESPUÉS de la regla

2. CORTAFUEGOS PERSONAL CON UFW

h) También es posible limitar las conexiones realizadas desde una IP origen (ufw limit ...), para mitigar los ataques de diccionario. Activar el acceso limitado al servidor SSH de nuestro equipo. ¿Qué límite se ha establecido para las conexiones SSH entrantes?

```
root@server:~# ufw allow 22/tcp
Rule added
Rule added (v6)
root@server:~# ufw limit 22/tcp
Rule updated
Rule updated (v6)
root@server:~# ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ------      ----
80/tcp                     ALLOW IN    Anywhere
22/tcp                     LIMIT IN    Anywhere
80/tcp (v6)                ALLOW IN    Anywhere (v6)
22/tcp (v6)                LIMIT IN    Anywhere (v6)
```

```
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
©Permission denied, please try again.
root@192.168.64.129's password:


┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
ssh: connect to host 192.168.64.129 port 22: Connection refused
```

Bloquea al 6 intento por defecto

2. CORTAFUEGOS PERSONAL CON UFW

h) También es posible limitar las conexiones realizadas desde una IP origen (ufw limit ...), para mitigar los ataques de diccionario. Activar el acceso limitado al servidor SSH de nuestro equipo. ¿Qué límite se ha establecido para las conexiones SSH entrantes?

Comentario Teo: Se establece un límite de 6 conexiones cada 30 segundos ¿cómo podemos obtener estos valores?

cat /etc/ssh/sshd_config
Buscamos “MaxSessions”

```
root@server:~# cat /etc/ssh/sshd_config | grep "MaxSessions"
#MaxSessions 10
root@server:~#
```

```
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
Permission denied, please try again.
root@192.168.64.129's password:


┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
root@192.168.64.129's password:
^C

┌──(kali㉿kali)-[~]
└─$ ssh root@192.168.64.129
ssh: connect to host 192.168.64.129 port 22: Connection refused
```

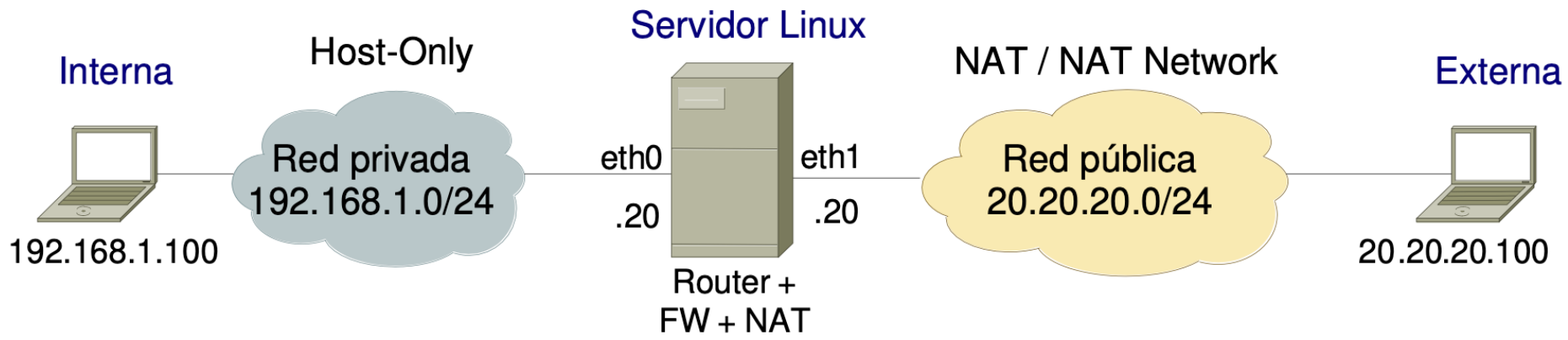
Bloquea al 6 intento por defecto

ÍNDICE

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

Vamos a preparar una maqueta como la reflejada en el gráfico adjunto. Añadiremos dos nuevas máquinas virtuales (interna y externa) y asociaremos un segundo interface (eth1) a la máquina que actúa como cortafuegos. La red exterior (puede usar el modo NAT en *VMWare* o NAT Network en *VirtualBox*) tendrá un direccionamiento público (20.20.20.0/24) y la privada (en modo Host-Only) tendrá direccionamiento privado (192.168.1.0/24). Las nuevas máquinas tendrán como *router* por defecto al equipo que actúa como firewall.

NOTA: Verificar el nombre de los interfaces asignados, y la red a la que están conectados.



3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

Ubuntu Interna
192.168.1.100
(192.168.64.130)

vim /etc/netplan/50-cloud-init.yaml
systemctl netplan apply

```
version: 2
 ethernets:
  ens160:
   dhcp4: false
   addresses:
    - 192.168.1.100/24
   routes:
    - to: default
      via: 192.168.1.20
   nameservers:
    addresses:
     - 8.8.8.8
  ens256:
   dhcp4: false
   addresses:
    - 192.168.64.130/24
   nameservers:
    addresses:
     - 8.8.8.8
```

Ubuntu Externa
20.20.20.100
(192.168.64.131)

vim /etc/netplan/50-cloud-init.yaml
systemctl netplan apply

```
version: 2
 ethernets:
  ens160:
   dhcp4: false
   addresses:
    - 20.20.20.100/24
   routes:
    - to: default
      via: 20.20.20.20
   nameservers:
    addresses:
     - 8.8.8.8
  ens256:
   addresses:
    - 192.168.64.131/24
   nameservers:
    addresses:
     - 8.8.8.8
```

Nota: Para añadir más de un interfaz, recuerda añadirlo manualmente con VMWare o VirtualBox

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

Ubuntu Servidor
192.168.64.132

```
# vim /etc/netplan/00-installer-config.yaml
# netplan apply
# vim /etc/sysctl.conf
```

```
# Uncommment the next line to enabl
net.ipv4.ip_forward=1
```

```
# sysctl -p
```

```
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: false
      addresses:
        - 192.168.1.20/24
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
```

```
    eth1:
      dhcp4: false
      addresses:
        - 20.20.20.20/24
      routes:
        - to: default
          via: 20.20.20.1
      nameservers:
        addresses:
          - 8.8.8.8
          - 8.8.4.4
    eth2:
      dhcp4:true
```

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

Ubuntu Servidor
192.168.64.132

```
root@server:~# 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
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c4:ce:ed brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    altname ens160
    inet 192.168.1.20/24 brd 192.168.1.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fec4:ceed/64 scope link
       valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c4:ce:01 brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    altname ens161
    inet 20.20.20.20/24 brd 20.20.20.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fec4:ce01/64 scope link
       valid_lft forever preferred_lft forever
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:c4:ce:f7 brd ff:ff:ff:ff:ff:ff
    altname enp26s0
    altname ens256
    inet 192.168.64.134/24 metric 100 brd 192.168.64.255 scope global dynamic eth2
       valid_lft 1666sec preferred_lft 1666sec
    inet6 fe80::20c:29ff:fec4:cef7/64 scope link
       valid_lft forever preferred_lft forever
```

Nota: le hemos añadido una interfaz eth2 para poder conectarnos por ssh y que sea más fácil trabajar

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

Ubuntu Interna
192.168.1.100
(192.168.64.130)

ens160

```
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:57:70:65 brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 192.168.1.100/24 brd 192.168.1.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:fe57:7065/64 scope link
       valid_lft forever preferred_lft forever
```

Ubuntu Externa
20.20.20.100
(192.168.64.131)

ens160

```
2: ens160: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP
group default qlen 1000
    link/ether 00:0c:29:ec:b1:7c brd ff:ff:ff:ff:ff:ff
    altname enp2s0
    inet 20.20.20.100/24 brd 20.20.20.255 scope global ens160
       valid_lft forever preferred_lft forever
    inet6 fe80::20c:29ff:feec:b17c/64 scope link
       valid_lft forever preferred_lft forever
```

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

9. Reiniciar el servicio de red y comprobar que somos capaces de alcanzar las máquinas interna y externa desde el cortafuegos (fw). Habilitar el encaminamiento en el cortafuegos (añadir net.ipv4.ip_forward=1 en el archivo /etc/sysctl.conf y a continuación ejecutar sysctl -p; puede comprobarse el nuevo valor con sysctl net.ipv4.ip_forward). Verificar que es posible comunicar las máquinas interna y externa.

```
root@server:~# ping 20.20.20.100
PING 20.20.20.100 (20.20.20.100) 56(84) bytes of data.
64 bytes from 20.20.20.100: icmp_seq=1 ttl=64 time=0.416 ms
64 bytes from 20.20.20.100: icmp_seq=2 ttl=64 time=1.03 ms
64 bytes from 20.20.20.100: icmp_seq=3 ttl=64 time=1.13 ms
^C
--- 20.20.20.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2026ms
rtt min/avg/max/mdev = 0.416/0.858/1.133/0.315 ms
root@server:~# ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=0.845 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=0.781 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=0.796 ms
^C
--- 192.168.1.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.781/0.807/0.845/0.027 ms
```

Tenemos conexión desde el FW al resto de Hosts

Ubuntu Servidor (FW)

vim /etc/sysctl.conf

```
# Uncomment the next li
net.ipv4.ip_forward=1
```

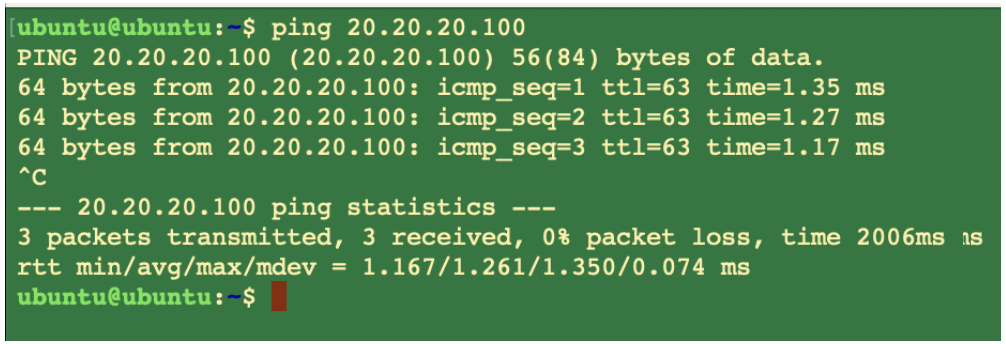
sysctl –p

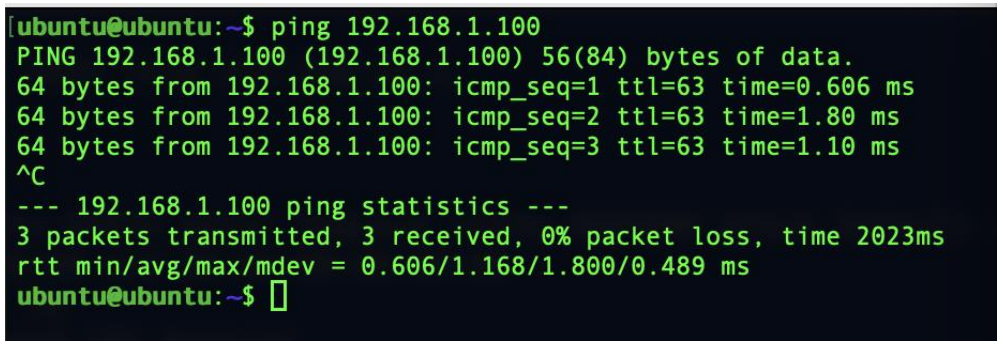
Reiniciar servicio red: # netplan apply

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

9. Reiniciar el servicio de red y comprobar que somos capaces de alcanzar las máquinas interna y externa desde el cortafuegos (fw). Habilitar el encaminamiento en el cortafuegos (añadir net.ipv4.ip_forward=1 en el archivo /etc/sysctl.conf y a continuación ejecutar sysctl -p; puede comprobarse el nuevo valor con sysctl net.ipv4.ip_forward). Verificar que es posible comunicar las máquinas interna y externa.

Ubuntu Interna 192.168.1.100	Ubuntu Externa 20.20.20.100





3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

10. Instalar la herramienta shorewall (los paquetes rpm se encuentran en el directorio /usr/local/src/shorewall).

Ubuntu Servidor

```
# cd /usr/local/src/shorewall
# rpm -ivh shorewall*.rpm
# shorewall version
```

```
root@server:~# apt-get install shorewall
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
shorewall ya está en su versión más reciente (5.2.3.4-1).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
root@server:~# shorewall version
5.2.3.4
```

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

11. Describir unas políticas básicas en el cortafuegos: se acepta el tráfico desde la red interna al exterior (saliente), se impide tráfico entrante desde el exterior, se acepta tráfico saliente desde el cortafuegos hacia cualquier sitio. Se acepta tráfico entrante al cortafuegos desde la red interna. Activar las reglas (systemctl start shorewall) y verificar su correcto funcionamiento.

Políticas

- Se acepta el tráfico desde la red interna (LAN) al exterior (saliente).
- Se impide el tráfico entrante desde el exterior hacia la red interna.
- Se acepta el tráfico saliente desde el cortafuegos hacia cualquier sitio.
- Se acepta el tráfico entrante al cortafuegos desde la red interna.

1) Definir archivo Zones:

```
# vim  /etc/shorewall/zones

fw      firewall
int     ipv4                    # loc (local)
ext     ipv4                    # net (network)
```

2) Asignar interfaces a las zonas:

```
# vim  /etc/shorewall/interfaces

ext     eth1    20.20.20.255 # Red externa (Internet)
int     eth0    192.168.1.255  # Red interna (LAN)
```

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

11. Describir unas políticas básicas en el cortafuegos: se acepta el tráfico desde la red interna al exterior (saliente), se impide tráfico entrante desde el exterior, se acepta tráfico saliente desde el cortafuegos hacia cualquier sitio. Se acepta tráfico entrante al cortafuegos desde la red interna. Activar las reglas (systemctl start shorewall) y verificar su correcto funcionamiento.

3) Definir las políticas por defecto

```
# vim  /etc/shorewall/policy
# Fuente    Destino    Política     Registro
      int       ext       ACCEPT      # Permitir tráfico saliente de int a ext
      ext       all       DROP        # Bloquear tráfico entrante desde ext
      fw        all       ACCEPT      # Permitir tráfico saliente desde el cortafuegos
      int       fw        ACCEPT      # Permitir tráfico entrante al cortafuegos desde la red interna
      all       all       REJECT      # Rechazar cualquier otra cosa
```

```
# shorewall check
# systemctl start shorewall
# systemctl enable shorewall
```

Comprobación # sudo iptables -L -n -v

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

11. Describir unas políticas básicas en el cortafuegos: se acepta el tráfico desde la red interna al exterior (saliente), se impide tráfico entrante desde el exterior, se acepta tráfico saliente desde el cortafuegos hacia cualquier sitio. Se acepta tráfico entrante al cortafuegos desde la red interna. Activar las reglas (systemctl start shorewall) y verificar su correcto funcionamiento.

```
root@server:~# systemctl start shorewall
root@server:~# systemctl enable shorewall
Synchronizing state of shorewall.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable shorewall
root@server:~# shorewall check
Checking using Shorewall 5.2.3.4...
Processing /etc/shorewall/params ...
Processing /etc/shorewall/shorewall.conf...
Loading Modules...
Checking /etc/shorewall/zones...
Checking /etc/shorewall/interfaces...
   WARNING: Shorewall no longer uses broadcast addresses in rule generation when Address Type Match is availa
interfaces (line 1)
   WARNING: Shorewall no longer uses broadcast addresses in rule generation when Address Type Match is availa
interfaces (line 2)
Determining Hosts in Zones...
Locating Action Files...
Checking /etc/shorewall/policy...
Checking TCP Flags filtering...
Checking Kernel Route Filtering...
Checking Martian Logging...
Checking MAC Filtration -- Phase 1...
Checking /etc/shorewall/conntrack...
Checking MAC Filtration -- Phase 2...
Applying Policies...
Shorewall configuration verified
```

```
Chain OUTPUT (policy DROP 4 packets, 312 bytes)
 pkts bytes target     prot opt in     out     source

Chain dynamic (0 references)
 pkts bytes target     prot opt in     out     source

Chain loc-fw (0 references)
 pkts bytes target     prot opt in     out     source

Chain loc_frwd (0 references)
 pkts bytes target     prot opt in     out     source

Chain logdrop (0 references)
 pkts bytes target     prot opt in     out     source

Chain logflags (0 references)
 pkts bytes target     prot opt in     out     source

Chain logreject (0 references)
 pkts bytes target     prot opt in     out     source

Chain net-fw (0 references)
 pkts bytes target     prot opt in     out     source

Chain net-loc (0 references)
 pkts bytes target     prot opt in     out     source

Chain net_frwd (0 references)
 pkts bytes target     prot opt in     out     source

Chain reject (0 references)
 pkts bytes target     prot opt in     out     source
```

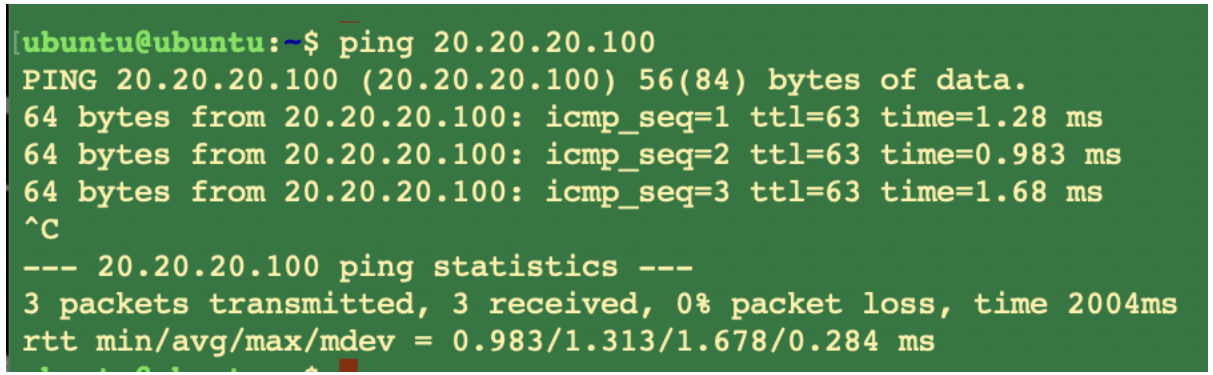
Iptables –L -v

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

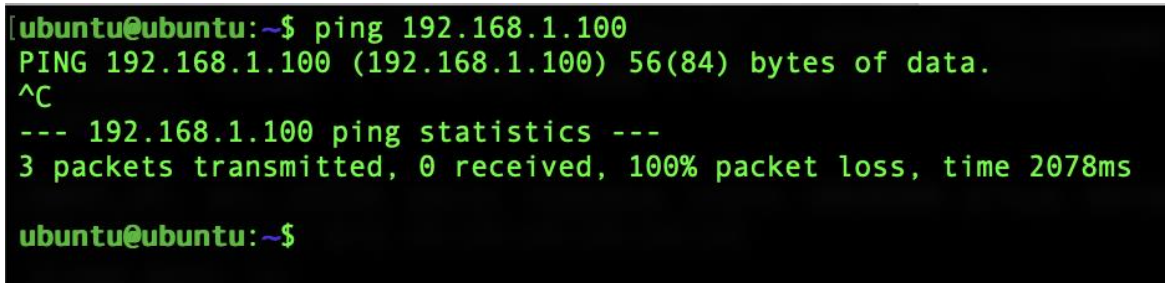
11. Describir unas políticas básicas en el cortafuegos: se acepta el tráfico desde la red interna al exterior (saliente), se impide tráfico entrante desde el exterior, se acepta tráfico saliente desde el cortafuegos hacia cualquier sitio. Se acepta tráfico entrante al cortafuegos desde la red interna. Activar las reglas (systemctl start shorewall) y verificar su correcto funcionamiento.

Verificación

1.Red interna a red externa:
Desde la **red interna**, # ping 20.20.20.100 (PERMITIDO)
Resultado esperado: El tráfico saliente debe ser permitido.



2. Red externa a red interna:
Desde la **red externa**, # ping 192.168.1.100 (BLOQUEADO).



3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

11. Describir unas políticas básicas en el cortafuegos: se acepta el tráfico desde la red interna al exterior (saliente), se impide tráfico entrante desde el exterior, se acepta tráfico saliente desde el cortafuegos hacia cualquier sitio. Se acepta tráfico entrante al cortafuegos desde la red interna. Activar las reglas (systemctl start shorewall) y verificar su correcto funcionamiento.

3. Cortafuegos a cualquier destino:

```
root@server:~# ping 20.20.20.100
PING 20.20.20.100 (20.20.20.100) 56(84) bytes of data.
64 bytes from 20.20.20.100: icmp_seq=1 ttl=64 time=0.580 ms
64 bytes from 20.20.20.100: icmp_seq=2 ttl=64 time=0.836 ms
64 bytes from 20.20.20.100: icmp_seq=3 ttl=64 time=0.669 ms
^C
--- 20.20.20.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2005ms
rtt min/avg/max/mdev = 0.580/0.695/0.836/0.106 ms
root@server:~# ping 192.168.1.100
PING 192.168.1.100 (192.168.1.100) 56(84) bytes of data.
64 bytes from 192.168.1.100: icmp_seq=1 ttl=64 time=0.624 ms
64 bytes from 192.168.1.100: icmp_seq=2 ttl=64 time=0.874 ms
64 bytes from 192.168.1.100: icmp_seq=3 ttl=64 time=0.727 ms
^C
--- 192.168.1.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2004ms
rtt min/avg/max/mdev = 0.624/0.741/0.874/0.102 ms
```

4. Red interna a cortafuegos:
Desde la **red interna**, haz un ping a la IP del cortafuegos. (PERMITIDO)
Resultado esperado: El tráfico hacia el cortafuegos debe ser permitido.

```
ubuntu@ubuntu:~$ ping 192.168.1.20
PING 192.168.1.20 (192.168.1.20) 56(84) bytes of data.
64 bytes from 192.168.1.20: icmp_seq=1 ttl=64 time=0.641 ms
64 bytes from 192.168.1.20: icmp_seq=2 ttl=64 time=0.976 ms
64 bytes from 192.168.1.20: icmp_seq=3 ttl=64 time=0.821 ms
^C
--- 192.168.1.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2053ms
rtt min/avg/max/mdev = 0.641/0.812/0.976/0.136 ms
```

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

12. Habilitar el enmascaramiento de direcciones desde la red interior hacia la red exterior. Verificar que todo el tráfico saliente se enmascara (puede usar el sniffer tcpdump)

```
# vim /etc/shorewall/masq
# Todas las ips de eth0 (192.168.1.0/24) –traduces--> eth1 (20.20.20.0/24)
    eht1     eth0
```

Enmascara cualquier dirección interna (eth0) con la IP que tenga asociada el interface externo eth1

```
root@server:~# tcpdump -i eth0 -n
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:32:59.261983 ARP, Request who-has 192.168.64.130 tell 192.168.64.1, length 46
17:32:59.838882 IP 192.168.1.100 > 20.20.20.100: ICMP echo request, id 4287, seq 1, length 64
17:32:59.839822 IP 20.20.20.100 > 192.168.1.100: ICMP echo reply, id 4287, seq 1, length 64
17:33:00.840302 IP 192.168.1.100 > 20.20.20.100: ICMP echo request, id 4287, seq 2, length 64
17:33:00.841048 IP 20.20.20.100 > 192.168.1.100: ICMP echo reply, id 4287, seq 2, length 64
```

Ping desde la máquina en la red **interna** a la **externa**

tcpdump -i eth0 –n (antes de la “traducción”)

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

12. Habilitar el enmascaramiento de direcciones desde la red interior hacia la red exterior. Verificar que todo el tráfico saliente se enmascara (puede usar el sniffer tcpdump)

```
ubuntu@ubuntu:~$ ping 20.20.20.100
PING 20.20.20.100 (20.20.20.100) 56(84) bytes of data.
64 bytes from 20.20.20.100: icmp_seq=1 ttl=63 time=0.741 ms
64 bytes from 20.20.20.100: icmp_seq=2 ttl=63 time=0.902 ms
64 bytes from 20.20.20.100: icmp_seq=3 ttl=63 time=0.641 ms
^C
--- 20.20.20.100 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2040ms
rtt min/avg/max/mdev = 0.641/0.761/0.902/0.107 ms
```

```
inet 192.168.1.100/24 brd 192.168.1.255 scope global ens160
```

```
17:09:59.900990 IP 20.20.20.20 > 20.20.20.100: ICMP echo request, id 4245, seq 1, length 64
17:09:59.901335 IP 20.20.20.100 > 20.20.20.20: ICMP echo reply, id 4245, seq 1, length 64
17:10:00.916153 IP 20.20.20.20 > 20.20.20.100: ICMP echo request, id 4245, seq 2, length 64
17:10:00.916590 IP 20.20.20.100 > 20.20.20.20: ICMP echo reply, id 4245, seq 2, length 64
17:10:01.940767 IP 20.20.20.20 > 20.20.20.100: ICMP echo request, id 4245, seq 3, length 64
17:10:01.941036 IP 20.20.20.100 > 20.20.20.20: ICMP echo reply, id 4245, seq 3, length 64
```

tcpdump -i eth1 -n

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

13. Implantar una política más restrictiva para el tráfico que atraviesa el FW (todo el tráfico será rechazado). A continuación, se habilitará la salida a servicios concretos (http, https, SMTP, ...). Configurar algún servicio de port forwarding que será visible desde el exterior y ofrecido por la máquina de la red interna. Configurar un acceso al cortafuegos por SSH desde una dirección externa concreta. se
Para realizar este apartado se recomienda estudiar los ejemplos de reglas shorewall presentes en el manual de esta aplicación (man shorewall-rules)

a) Cambiar las políticas → Rechazar todo el tráfico que atraviese el FW
b) Añadir reglas → Habilitar servicios de http, https y SMTP
c) Máquina Interna habilitar servicio apache
d) Port forwarding: Redirigir tráfico desde una IP externa al servidor Apache en **192.168.1.100**.
e) Permitir el acceso SSH al cortafuegos desde **20.20.20.100** únicamente
f) Redirigir todas las solicitudes de DNS salientes desde la red interna hacia el servidor DNS externo 20.20.20.50

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

a) Cambiar las políticas → Rechazar todo el tráfico que atraviese el FW

```
# Permitir tráfico dentro de la misma red (no pasa por el FW)
int          int          ACCEPT
ext          ext          ACCEPT


# Rechazar tráfico entre zonas externas e internas
int          all          DROP       info
ext          all          DROP       info

# Permitir tráfico interno del firewall hacia el exterior
fw           all          DROP       info
```

```
# Permitir tráfico dentro de la misma red (no pasa por el FW)
int          int          ACCEPT
ext          ext          ACCEPT


# Rechazar tráfico entre zonas externas e internas
int          all          REJECT     info
ext          all          REJECT     info

# Permitir tráfico interno del firewall hacia el exterior
fw           all          REJECT     info
```

Comentario Teo: es mejor usar DROP, no REJECT, para el tráfico externo (se eliminarán los datagramas sin informar al origen)

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

b) Añadir reglas → Habilitar servicios de http, https y SMTP

vim /etc/shorewall/rules

```
# Permitir tráfico saliente de HTTP/HTTPS/SMTP desde la red interna
ACCEPT   int   ext   tcp   80,443,25
```

c) Máquina Interna habilitar servicio apache

d) Port forwarding: Redirigir tráfico desde una IP externa al servidor Apache en **192.168.1.100**.

vim /etc/shorewall/rules

```
# Port forwarding de  Apache
DNAT    ext   int:192.168.1.100   tcp   80
```

```
ubuntu@ubuntu:~$ sudo su
[sudo] password for ubuntu:
root@ubuntu:/home/ubuntu# curl 20.20.20.20
<h1> Máquina interna </h1>
<p> Con la IP: 192.168.1.100 </p>
root@ubuntu:/home/ubuntu#
```

Ubuntu Externa

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

e) Permitir el acceso SSH al cortafuegos desde **20.20.20.100** únicamente

vim /etc/shorewall/rules

```
# Permitir tráfico SSH desde una IP externa al FW
ACCEPT   ext:20.20.20.100   fw   tcp   22
```

f) Redirigir todas las solicitudes de DNS salientes desde la red interna hacia el servidor DNS externo 20.20.20.50

vim /etc/shorewall/rules

```
# Redirigir el tráfico UDP que llega al puerto 53 (DNS) en una interfaz de red externa (ext) hacia la dirección IP interna 20.20.20.50 puerto 53.

DNAT  int  ext:20.20.20.50:53  udp  53
```

Ubuntu Externa

3. ROUTER CON FUNCIÓN DE CORTAFUEGOS Y NAT

Servidor DNS
20.20.20.50

```
root@ubuntu:/home/ubuntu# dig @localhost mi-dominio.com

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> @localhost mi-dominio.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 32657
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 1232
; COOKIE: 73228531b864725c01000000676d4a6e832a81227170824d (good)
;; QUESTION SECTION:
;mi-dominio.com.                        IN      A

;; ANSWER SECTION:
mi-dominio.com.         604800  IN      A       20.20.20.110

;; Query time: 0 msec
;; SERVER: 127.0.0.1#53(localhost) (UDP)
;; WHEN: Thu Dec 26 12:22:06 UTC 2024
;; MSG SIZE  rcvd: 87
```

Ubuntu Interna
192.168.1.100

```
root@ubuntu:/home/ubuntu# dig 20.20.20.20 mi-dominio.com

; <<>> DiG 9.18.28-0ubuntu0.24.04.1-Ubuntu <<>> 20.20.20.20 mi-dominio.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: SERVFAIL, id: 64750
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;20.20.20.20.                   IN      A

;; Query time: 690 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu Dec 26 12:37:15 UTC 2024
;; MSG SIZE  rcvd: 40

;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 9442
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 65494
;; QUESTION SECTION:
;mi-dominio.com.                        IN      A

;; ANSWER SECTION:
mi-dominio.com.         604800  IN      A       20.20.20.110

;; Query time: 0 msec
;; SERVER: 127.0.0.53#53(127.0.0.53) (UDP)
;; WHEN: Thu Dec 26 12:37:15 UTC 2024
;; MSG SIZE  rcvd: 59
```

GRACIAS

Inés del Río y Paloma Pérez de Madrid

COMENTARIOS TEO

Apartado 1. En las conexiones “salientes” http, https y dns (nosotros somos el lado cliente) y ssh, http (“entrantes”, nosotros somos el servidor), al igual que en icmp, hay que establecer reglas en ambos sentidos (INPUT y OUTPUT) que deben ser complementarias.
Por ejemplo, para ssh entrante (nuestra máquina actúa de servidor) las reglas serán:
iptables -A INPUT -p tcp --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
CORREGIDO

Apartado 5. En estas reglas con estado de TCP y UDP, lo que buscamos es que en OUTPUT se permita NEW y ESTABLISHED, y en INPUT sólo ESTABLISHED (podemos iniciar conexiones hacia el exterior, pero desde el exterior no pueden iniciarlas contra nosotros). En la memoria está al revés. **CORREGIDO**

Apartado 6. Las reglas son correctas, aunque hay que limitar a 2 conexiones cada 180 segundos, no a 3.
SIMPLEMENTE CAMBIAR DE “3” A “2”

Apartado 7. h) Se establece un límite de 6 conexiones cada 30 segundos ¿cómo podemos obtener estos valores? **CORREGIDO**

Apartado 13. En la nueva política es mejor usar DROP, no REJECT, para el tráfico externo (se eliminarán los datagramas sin informar al origen). La política quedaría:
net all DROP
all all REJECT

CORREGIDO