

Universidad Nacional Autónoma de Nicaragua

UNAN-León

Facultad de Ciencias y Tecnología

Departamento de Computación

Ingeniería en Telemática

V año



Componente: Redes de Area Extensa

Tema: Practica de Iptables

Realizado por:

Br. Jhonatan Uziel Espinoza Ortega

Carnet: 15-00737-0

Dirigido a:

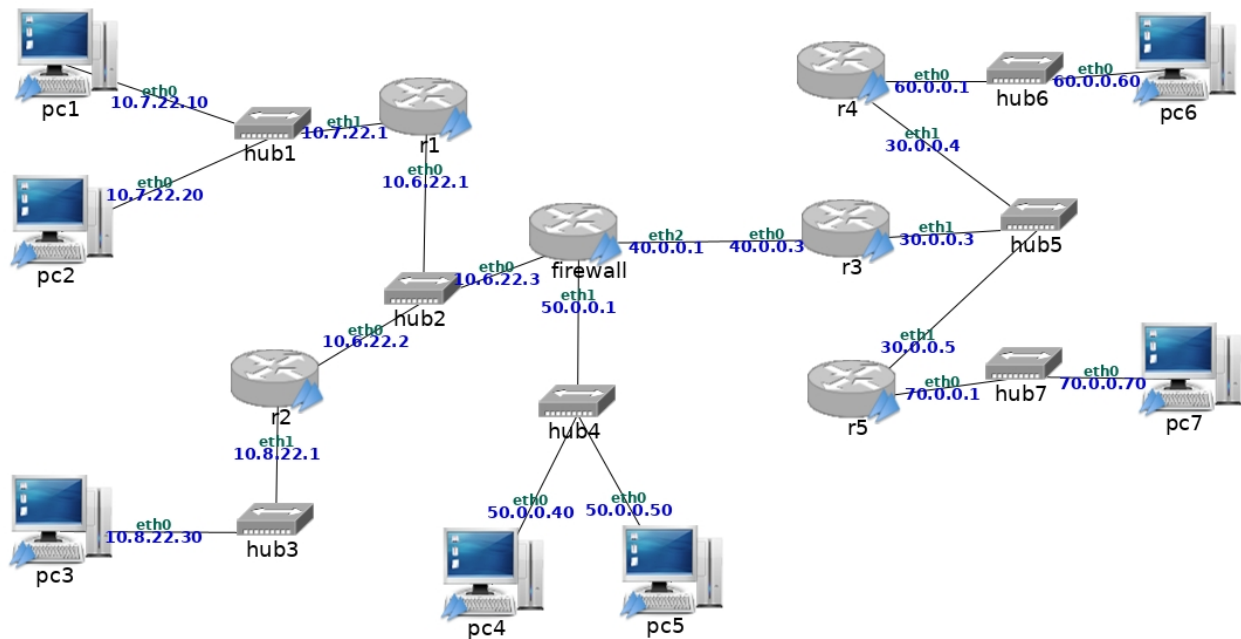
MSc. Aldo Martinez

León, Nicaragua lunes 8 de Julio del 2019.

1. Escenario para la configuración de un firewall

Arranca de una en una todas las máquinas de la figura.

Configura las direcciones IP en cada una de las máquinas, asignándoles una dirección IP válida en la subred a la que pertenecen. Configura las rutas que sean necesarias en cada uno de los routers para que todas las máquinas de las subredes privadas se puedan comunicar entre ellas y todas las máquinas de las subredes públicas se puedan comunicar entre ellas. Hasta que no se configuren las reglas NAT en el firewall no se podrán comunicar las máquinas de las subredes privadas con las de Internet. El router r3 sólo puede tener rutas a las subredes públicas: subred 1, subred 2, subred 3, subred 4 y subred 5. No puedes configurarle una ruta por defecto.



2. Traducción de direcciones y puertos en el firewall: tabla nat

Comprueba que no funciona un ping desde las máquinas internas de las redes privadas (pc1, pc2 y pc3) a destinos de Internet como pc6 o pc7.

```
pc1
pc1:~# ping 60.0.0.60
PING 60.0.0.60 (60.0.0.60) 56(84) bytes of data.

--- 60.0.0.60 ping statistics ---
73 packets transmitted, 0 received, 100% packet loss, time 72296ms

pc1:~#
```

```
pc3
pc3:~# ping 70.0.0.70
PING 70.0.0.70 (70.0.0.70) 56(84) bytes of data.

--- 70.0.0.70 ping statistics ---
21 packets transmitted, 0 received, 100% packet loss, time 20092ms

pc3:~#
```

1. Configura un script fw1.sh en el firewall para que primero borre las reglas que hubiera configuradas previamente en la tabla nat y reinicie los contadores de dicha tabla, y a continuación realice la traducción de direcciones en el tráfico saliente de las redes privadas (SNAT) y en su correspondiente tráfico de respuesta. Explica para qué subredes has tenido que realizar la configuración de SNAT. Incluye el script fw1.sh en la memoria y explícalo.

```
firewall
GNU nano 2.0.7      File: fw1.sh

#!/bin/bash

iptables -t nat -F
echo "Reglas borradas"
iptables -t nat -Z
echo "Contadores reiniciados"

iptables -t nat -A POSTROUTING -s 10.7.22.0/24 -o eth2 \
-j SNAT --to-source 40.0.0.1
echo "Traducción de direccion para subred 7 definida exitosamente"

iptables -t nat -A POSTROUTING -s 10.8.22.0/24 -o eth2 \
-j SNAT --to-source 40.0.0.1
echo "Traducción de direccion para subred 8 definida exitosamente"
```

Se ha realizado la configuracion de SNAT para la subred 7 en las cuales se encuentran la pc1 y la pc2, tambien para la subred 8 en la cual se encuentra la maquina 3.

2.1.1. ICMP

Ejecuta el script fw1.sh de 2.1.

```
firewall
firewall:~# nano fw1.sh
firewall:~# ./fw1.sh
Reglas borradas
Contadores reiniciados
Traducción de direccion para subred 7 definida exitosamente
Traducción de direccion para subred 8 definida exitosamente
firewall:~#
```

1. Realiza una captura de tráfico en r3 (iptables-01.cap). Ejecuta un ping desde pc1 a pc6 con la opción que permite enviar sólo 2 paquetes ICMP echo request (-c 2).

```
pc1
pc1:~# ping 60.0.0.60 -c 2
PING 60.0.0.60 (60.0.0.60) 56(84) bytes of data.
64 bytes from 60.0.0.60: icmp_seq=1 ttl=60 time=47.7 ms
64 bytes from 60.0.0.60: icmp_seq=2 ttl=60 time=1.58 ms

--- 60.0.0.60 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1000ms
rtt min/avg/max/mdev = 1.588/24.650/47.713/23.063 ms
pc1:~#
```

Interrumpe la captura de tráfico. Explica las direcciones IP que se usan en la captura.

```
r3
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-01.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
8 packets captured
8 packets received by filter
0 packets dropped by kernel
r3:~#
```

Se usa la ip del firewall de la interfaz eth2 (que tiene direccionamiento publico) y la ip de la pc6, que es a la que desde pc1 se le ha hecho ping.

2. Explica qué significa el resultado de la ejecución del siguiente comando en firewall:

```
firewall
firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 2 packets, 168 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
    2  168 SNAT      all  --  *      eth2    10.7.22.0/24    0.0.0.0/0
      to:40.0.0.1
    0    0 SNAT      all  --  *      eth2    10.8.22.0/24    0.0.0.0/0
      to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target    prot opt in     out     source         destination
firewall:~#
```

Qué regla/s está/n cumpliendo los paquetes ICMP echo request e ICMP echo response y cuántas veces se cumple/n. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla nat y cuantos paquetes las han cumplido.

- Se cumple la regla que fue definida para la subred 7 y se cumple 2 veces debido a que fueron 2 ping los que se le hicieron a la pc6,

3. Consulta la información de seguimiento de conexiones del módulo ip_conntrack del firewall y explica el resultado.

```
firewall
firewall:~# cat /proc/net/ip_conntrack
firewall:~#
```

2.1.2. UDP

Ejecuta el script fw1.sh de 2.1 para que reinicie los contadores de paquetes de iptables.

```
firewall
firewall:~# ./fw1.sh
Reglas borradas
Contadores reiniciados
Traduccion de direccion para subred 7 definida exitosamente
Traduccion de direccion para subred 8 definida exitosamente
firewall:~#
```

1. Ejecuta nc en modo servidor UDP en pc6 y nc en modo cliente UDP en pc2. Simultáneamente realiza una captura en r3 (iptables-02.cap) y consulta la información ip_conntrack de firewall.

Escribe 5 líneas en el terminal de pc2 para que se las envíe a pc6 (con cada línea, es decir cada vez que pulsas una cadena de caracteres y <Enter>, se envía un paquete UDP nuevo). Observa el estado de ip_conntrack. Escribe una línea en pc6 para que se la envíe a pc2. Observa el estado de ip_conntrack.

```
pc2
pc2:~# nc -u -p 6666 60.0.0.60 7777
Linea 1
Linea 2
Linea 3
Linea 4
Linea 5
Linea 6
pc2:~#
```

← Enviada por servidor

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack      Sun Apr  7 18:08:26 2019

udp      17 28 src=10.7.22.20 dst=60.0.0.60 sport=6666 dport=7777 packets=5 byte
s=180 [UNREPLIED] src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=6666 packets=0 byt
es=0 mark=0 use=1
```

```
pc6
pc6:~# nc -u -l -p 7777
Linea 1
Linea 2
Linea 3
Linea 4
Linea 5
Linea 6
```

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack      Sun Apr  7 18:08:35 2019

udp      17 27 src=10.7.22.20 dst=60.0.0.60 sport=6666 dport=7777 packets=5 byte
s=180 src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=6666 packets=1 bytes=36 mark=0
use=1
```

Interrumpe la captura y las ejecuciones de nc, explica la captura y cómo ésta se relaciona con la información que has visto en ip_conntrack.

```
r3
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-02.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
10 packets captured
10 packets received by filter
0 packets dropped by kernel
r3:~#
```

Se reflejan los paquetes udp enviados (las líneas ingresadas) desde el cliente al servidor y viceversa cuando del servidor se envía una línea al cliente, también se puede ver cómo se traduce la ip de pc1 a la ip del firewall eth2 y pc6 no sabe que en realidad es de pc1 que vienen esos mensajes, así mismo es por ello que pc6 a quien envía la línea cuando se realiza el envío de servidor a cliente lo hace a la ip de eth2 del firewall.

2. Explica lo que muestra el contenido de la tabla nat del firewall. Indica qué regla/s están cumpliendo los paquetes y cuántas veces se cumple/n. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla nat y cuántos paquetes las han cumplido.

```
firewall
firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 1 packets, 36 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

    1    36 SNAT          all  --  *       eth2    10.7.22.0/24  0.0.0.0/0
      to:40.0.0.1
    0     0 SNAT          all  --  *       eth2    10.8.22.0/24  0.0.0.0/0
      to:40.0.0.1

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

firewall:~#
```

Se cumple la regla definida para la subred 7 y la cumple un paquete.

3. Vuelve a repetir la misma prueba anterior pero iniciando el servidor UDP en pc6 y el cliente UDP en pc3. Escribe 5 líneas en el terminal de pc3 para que se las envíe a pc6. Observa el estado de ip_conntrack. Escribe una línea en pc6 para que se la envíe a pc3. Observa el estado de ip_conntrack.

```
pc3:~# nc -u -p 6666 60.0.0.60 7777
linea 1 desde pc3
linea 2 desde pc3
linea 3 desde pc3
linea 4 desde pc3
linea 5 desde pc3
linea 1 desde pc6 a pc3
pc3:~#
```

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack          Sun Apr  7 18:21:50 2019

udp      17 27 src=10.8.22.30 dst=60.0.0.60 sport=6666 dport=7777 packets=5 byte
s=230 [UNREPLIED] src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=6666 packets=0 byt
es=0 mark=0 use=1
```

```
pc6:~# nc -u -l -p 7777
linea 1 desde pc3
linea 2 desde pc3
linea 3 desde pc3
linea 4 desde pc3
linea 5 desde pc3
linea 1 desde pc6 a pc3
pc6:~#
```

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack          Sun Apr  7 18:22:12 2019

udp      17 20 src=10.8.22.30 dst=60.0.0.60 sport=6666 dport=7777 packets=5 byte
s=230 src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=6666 packets=1 bytes=52 mark=0
use=1
```

Interrumpe las ejecuciones de nc, explica lo que muestra el contenido de la tabla nat del firewall. Indica qué regla/s están cumpliendo los paquetes y cuántas veces se cumple/n. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla nat y cuantos paquetes las han cumplido.

```
firewall
firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 2 packets, 72 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
    1    36 SNAT          all  --  *      eth2    10.7.22.0/24  0.0.0.0/0
    1    36 SNAT          all  --  *      eth2    10.8.22.0/24  0.0.0.0/0
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source      destination
firewall:~#
```

Esta vez se cumplió la regla que fue definida para la subred 8, y se cumple una vez.

4. Primero inicia una captura en r3 (iptables-03.cap) para capturar todo el tráfico que atraviese este router e inicia otra captura en r1-eth0 (iptables-04.cap).

```
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-03.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
```

```
r1:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-04.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
```

Ejecuta una aplicación servidor UDP escuchando en el puerto 7777 en pc7 con el comando nc.

```
pc7:~# nc -u -l -p 7777
```

Ejecuta en pc1 una aplicación cliente UDP que utilice localmente el puerto 6666 y que se comunique con ese servidor anterior de la siguiente manera:

```
pc1:~# nc -u -p 6666 70.0.0.70 7777
```

Y ejecuta en pc2 una aplicación cliente UDP que utilice localmente el puerto 6666 y que se comunique con ese servidor anterior de la siguiente manera:

```
pc2:~# nc -u -p 6666 70.0.0.70 7777
```

Consulta la información de ip_conntrack en firewall, dado que todavía no se han enviado datos, no debería aparecer nada.

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack      Sun Apr  7 18:35:36 2019
```

Escribe una cadena de caracteres a través de la entrada estándar de pc1 y pulsa <Enter>. A continuación introduce una cadena de caracteres a través de la entrada estándar de pc2 y pulsa <Enter>. Interrumpe las dos capturas y explica qué ocurre con la traducción de direcciones y puertos.

```
pc1:~# nc -u -p 6666 70.0.0.70 7777
Linea 1 desde pc1
```

```
pc2:~# nc -u -p 6666 70.0.0.70 7777
Linea 1 desde pc2
pc2:~#
```

```
pc7:~# nc -u -l -p 7777
Linea 1 desde pc1
```

```

firewall
Every 0.5s: cat /proc/net/ip_conntrack          Sun Apr  7 18:42:30 2019
udp      17 17 src=10.7.22.10 dst=70.0.0.70 sport=6666 dport=7777 packets=1 byte
s=46 [UNREPLIED] src=70.0.0.70 dst=40.0.0.1 sport=7777 dport=6666 packets=0 byte
s=0 mark=0 use=1
udp      17 22 src=10.7.22.20 dst=70.0.0.70 sport=6666 dport=7777 packets=1 byte
s=46 [UNREPLIED] src=70.0.0.70 dst=40.0.0.1 sport=7777 dport=1024 packets=0 byte
s=0 mark=0 use=1

r3
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-03.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
7 packets captured
7 packets received by filter
0 packets dropped by kernel
r3:~# █

r1
r1:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-04.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
7 packets captured
7 packets received by filter
0 packets dropped by kernel
r1:~# █

```

En la captura del router de la red privada que ambos mensajes de las pc de esta red van con la ip privada y con los puertos especificados como clientes, y en la captura realizada en la zona publica el puerto con el que se realiza el envio del mensaje desde el cliente ya no es el mismo que la captura en la zona privada, esto es porque se ha hecho una traduccion y en este flujo de datos se tomo un puerto que estuviera libre en el firewall para hacer el envio como cliente, ya que el 6666 especificado por pc2 a la hora de hacer el envio ya habia sido ocupado primero por pc1 al haber establecido una conexi3n primero ellos.

5. Consulta la tabla nat del firewall y explica cu2ntas veces se han cumplido las reglas de traduccion de direcciones.

```

firewall
firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 4 packets, 144 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
    3  108 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0
      to:40.0.0.1
    1   36 SNAT      all  --  *      eth2    10.8.22.0/24  0.0.0.0/0
      to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target      prot opt in      out     source      destination
firewall:~# █

```

Se han cumplido hasta el momento 4 veces.

2.1.3. TCP

Ejecuta el script fw1.sh de 2.1 para que reinicie los contadores de paquetes de iptables.

```
firewall
firewall:~# ./fw1.sh
Reglas borradas
Contadores reiniciados
Traduccion de direccion para subred 7 definida exitosamente
Traduccion de direccion para subred 8 definida exitosamente
firewall:~#
```

1. Para este apartado vamos a usar nc en modo TCP.

Primero inicia una captura en r3 (iptables-05.cap) para capturar todo el tráfico que atraviese este router.

```
r3
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-05.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
```

Ejecuta una aplicación servidor TCP escuchando en el puerto 7777 en pc6 con el comando nc.

```
pc6
pc6:~# nc -l -p 7777
```

Y ejecuta en pc1 una aplicación cliente TCP que se comunique con el servidor anterior.

```
pc1
pc1:~# nc -p 6666 60.0.0.60 7777
```

Simultáneamente consulta ip_conntrack del firewall cada medio segundo. Explica el número de paquetes que se han observado en cada sentido, razonando la respuesta.

```
firewall
Every 0,5s: cat /proc/net/ip_conntrack Sun Apr 7 18:58:55 2019

tcp      6 431942 ESTABLISHED src=10.7.22.10 dst=60.0.0.60 sport=6666 dport=7777
packets=2 bytes=112 src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=6666 packets=1
bytes=60 [ASSURED] mark=0 use=1
```

2. Introduce una palabra en la entrada estándar de pc1, pulsa <Enter> y explica razonadamente lo que observas en ip_conntrack.

```
pc1
pc1:~# nc -p 6666 60.0.0.60 7777
Palabra

firewall
Every 0,5s: cat /proc/net/ip_conntrack Sun Apr 7 18:59:58 2019

tcp      6 431997 ESTABLISHED src=10.7.22.10 dst=60.0.0.60 sport=6666 dport=7777
packets=3 bytes=172 src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=6666 packets=2
bytes=112 [ASSURED] mark=0 use=1
```

3. Realiza un Ctrl+C en el terminal de pc1 para interrumpir la ejecución de nc. Interrumpe la captura en r3 y contrasta lo que observas en la captura con lo que muestra ip_conntrack.

```

pc1
pc1:~# nc -p 6666 60.0.0.60 7777
Palabra
pc1:~#

firewall
Every 0.5s: cat /proc/net/ip_conntrack          Sun Apr  7 19:00:55 2019

tcp        6 110 TIME_WAIT src=10.7.22.10 dst=60.0.0.60 sport=6666 dport=7777 pack
ets=5 bytes=276 src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=6666 packets=3 bytes
=164 [ASSURED] mark=0 use=1

r3
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-05.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
14 packets captured
14 packets received by filter
0 packets dropped by kernel
r3:~#

```

En el ip_conntrack se muestran las ip origen y destino primero antes y despues de la traduccion realizada por el firewall y esto mismo es comprobado en la captura de trafico, como efectivamente en la zona publica las ip reales de la zona privada no se conocen.

4. Consulta la tabla nat del firewall y explica cuántas veces se han cumplido las reglas de traducción de direcciones. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla nat y cuantos paquetes las han cumplido.

```

firewall
firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 1 packets, 60 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

    1    60 SNAT          all  --  *      eth2    10.7.22.0/24  0.0.0.0/0
      to:40.0.0.1
    0     0 SNAT          all  --  *      eth2    10.8.22.0/24  0.0.0.0/0
      to:40.0.0.1

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

firewall:~#

```

Se ha cumplido una las reglas definidas y la política aplicada es la que fue definida para la subred 7 y el paquete tcp enviado es el que la ha cumplido.

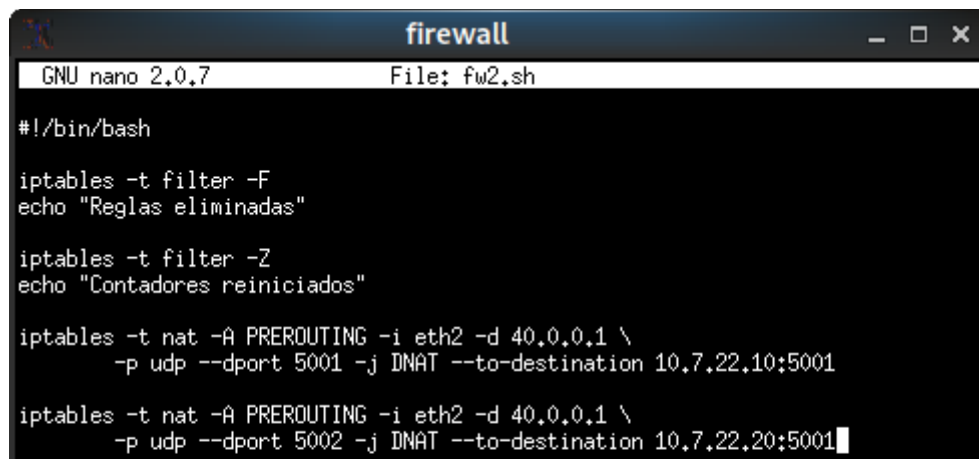
2.2. Servidor en la red privada, cliente externo

2.2.1. UDP

Realiza un nuevo script de iptables fw2.sh en firewall que primero borre las reglas que hubiera configuradas previamente en la tabla nat y reinicie los contadores de dicha tabla, y a continuación realice la siguiente traducción de direcciones:

El tráfico de entrada al firewall destinado al puerto UDP 5001 debe ser redirigido a pc1, puerto 5001.

El tráfico de entrada al firewall destinado al puerto UDP 5002 debe ser redirigido a pc2, puerto 5001.



```
firewall
GNU nano 2.0.7 File: fw2.sh

#!/bin/bash

iptables -t filter -F
echo "Reglas eliminadas"

iptables -t filter -Z
echo "Contadores reiniciados"

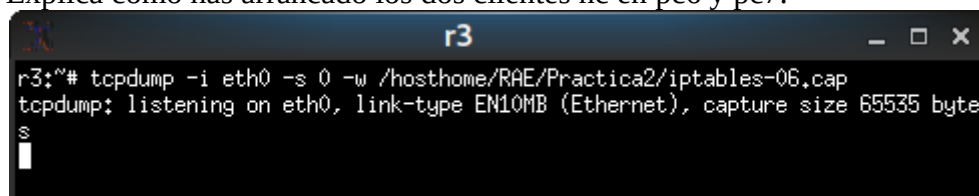
iptables -t nat -A PREROUTING -i eth2 -d 40.0.0.1 \
-p udp --dport 5001 -j DNAT --to-destination 10.7.22.10:5001

iptables -t nat -A PREROUTING -i eth2 -d 40.0.0.1 \
-p udp --dport 5002 -j DNAT --to-destination 10.7.22.20:5001
```

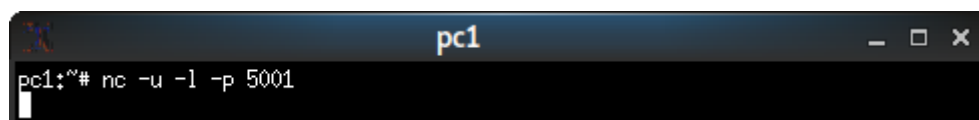
1. Explica el nuevo script.

En el shellscript fw2.sh se ejecutan 4 ordenes en las cuales primeramente se eliminan todas las reglas antes definidas que puede haber tenido el firewall, en la siguiente reinicia los contadores del uso de las reglas que se eliminaron con la primer orden. En la tercer orden se direcciona todo el trafico entrante a la red privada a traves del firewall por el puerto 5001 de udp a la pc 1 a su puerto 5001 udp y en la ultima se hace lo mismo. El trafico entrante esta vez por el puerto 5002 udp del firewall se direcciona al puerto 5001 udp de la pc2.

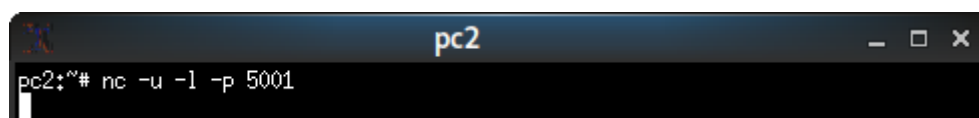
2. Inicia una captura de tráfico en r3 (iptables-06.cap). Lanza nc en modo servidor UDP en pc1 y pc2, escuchando en ambos casos en el puerto 5001. Lanza nc en modo cliente UDP en pc6 y pc7 de tal forma que el tráfico generado en pc6 lo reciba pc1 y el tráfico generado en pc2 lo reciba pc7. Explica cómo has arrancado los dos clientes nc en pc6 y pc7.



```
r3
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-06.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
```



```
pc1
pc1:~# nc -u -l -p 5001
```



```
pc2
pc2:~# nc -u -l -p 5001
```

```
pc6:~# nc -u -p 7777 40.0.0.1 5001
```

```
pc7:~# nc -u -p 7777 40.0.0.1 5002
```

3. Escribe una línea en cada uno de los terminales involucrados (pc1, pc2, pc6 y pc7). Interrumpe los clientes y servidor con Ctrl+C. Interrumpe la captura de tráfico. Explica el resultado observado en ip_conntrack y la traducción de direcciones IP y puertos realizada.

```
pc6:~# nc -u -p 7777 40.0.0.1 5001
Linea desde cliente udp
Linea desde servidor udp
```

```
pc1:~# nc -u -l -p 5001
Linea desde cliente udp
Linea desde servidor udp
```

```
pc7:~# nc -u -p 7777 40.0.0.1 5002
Linea desde cliente udp
Linea desde servidor udp
```

```
pc2:~# nc -u -l -p 5001
Linea desde cliente udp
Linea desde servidor udp
```

```
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-06.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
8 packets captured
8 packets received by filter
0 packets dropped by kernel
r3:~#
```

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack      Sun Apr  7 21:06:01 2019

udp      17 28 src=70.0.0.70 dst=40.0.0.1 sport=7777 dport=5002 packets=1 bytes=
52 src=10.7.22.20 dst=70.0.0.70 sport=5001 dport=7777 packets=1 bytes=53 mark=0
use=1
udp      17 23 src=60.0.0.60 dst=40.0.0.1 sport=7777 dport=5001 packets=1 bytes=
52 src=10.7.22.10 dst=60.0.0.60 sport=5001 dport=7777 packets=1 bytes=53 mark=0
use=1
```

En la captura en el ip_conntrack se puede ver que los paquetes llevan como destino la eth2 del firewall y los puertos destinos son del firewall y el firewall es el encargo de hacer la traducción y el que sabe cual es la correspondiente a cada una.

4. Consulta la tabla nat del firewall y explica cuántas veces se han cumplido las reglas de traducción de direcciones. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla nat y cuantos paquetes las han cumplido.

```

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 2 packets, 97 bytes)
num  pkts bytes target    prot opt in     out     source         destination
1      4   204 DNAT      udp  --  eth2    *       0.0.0.0/0      40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
2      4   205 DNAT      udp  --  eth2    *       0.0.0.0/0      40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
3      0     0 DNAT      udp  --  eth2    *       0.0.0.0/0      40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
4      0     0 DNAT      udp  --  eth2    *       0.0.0.0/0      40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001

Chain POSTROUTING (policy ACCEPT 12 packets, 649 bytes)
num  pkts bytes target    prot opt in     out     source         destination

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target    prot opt in     out     source         destination
firewall:~#

```

Se han cumplido 2 veces las reglas y han sido las definidas para las traducciones de direcciones y puertos del trafico entrante desde internet a la red privada.

2.2.2. TCP

Añade la siguiente configuración de traducción de direcciones al script fw2.sh de iptables de firewall:

El tráfico de entrada al firewall destinado al puerto TCP 80 debe ser redirigido a pc3, puerto 80.

```

GNU nano 2.0.7      File: fw2.sh
#!/bin/bash

iptables -t filter -F
echo "Reglas eliminadas"

iptables -t filter -Z
echo "Contadores reiniciados"

iptables -t nat -A PREROUTING -i eth2 -d 40.0.0.1 \
-p udp --dport 5001 -j DNAT --to-destination 10.7.22.10:5001

iptables -t nat -A PREROUTING -i eth2 -d 40.0.0.1 \
-p udp --dport 5002 -j DNAT --to-destination 10.7.22.20:5001

iptables -t nat -A PREROUTING -i eth2 -d 40.0.0.1 \
-p tcp --dport 80 -j DNAT --to-destination 10.8.22.30:80

```

1. Explica las modificaciones del script.

Hemos agregado una nueva orden en la cual el trafico que entra a la red privada a través del puerto 80 tcp del firewall sea direccionado al puerto 80 de la pc3.

2. Inicia una captura de tráfico en r3 (iptables-07.cap). Lanza nc en modo servidor TCP en pc3 escuchando en el puerto 80. Lanza nc en modo cliente TCP en pc6 de tal forma que el tráfico generado en pc6 lo reciba pc3. Explica cómo has arrancado el cliente de nc en pc6.

```
r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-07.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s

pc3:~# nc -l -p 80

pc6:~# nc -p 8080 40.0.0.1 80
```

Se ha lanzado el cliente con la ip del firewall y especificando el puerto, esto es porque en internet no se conoce la existencia de la red privada, el firewall es el encargado de hacer la traducción con la regla que se ha definido en el.

3. Interrumpe el cliente y el servidor con Ctrl+C. Interrumpe la captura de tráfico. Explica el resultado observado en ip_conntrack y la traducción de direcciones IP y puertos realizada.

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack                               Sun Apr  7 21:21:02 2019
tcp        6 431877 ESTABLISHED src=60.0.0.60 dst=40.0.0.1 sport=8080 dport=80 packets
=2 bytes=112 src=10.8.22.30 dst=60.0.0.60 sport=80 dport=8080 packets=1 bytes=60 [AS
SURED] mark=0 use=1

r3:~# tcpdump -i eth0 -s 0 -w /hosthome/RAE/Practica2/iptables-07.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
12 packets captured
12 packets received by filter
0 packets dropped by kernel
r3:~#

firewall
Every 0.5s: cat /proc/net/ip_conntrack                               Sun Apr  7 21:21:28 2019
tcp        6 110 TIME_WAIT src=60.0.0.60 dst=40.0.0.1 sport=8080 dport=80 packets=3 by
tes=164 src=10.8.22.30 dst=60.0.0.60 sport=80 dport=8080 packets=3 bytes=164 [ASSURE
D] mark=0 use=1
```

Efectivamente a como se observa en el ip_conntrack tambien se ve en la captura como aparentemente la conexión cliente-servidor se establece utilizando la ip del firewall como si estuviera ejecutandose el servidor en el.

4. Consulta la tabla nat del firewall y explica cuántas veces se han cumplido las reglas de traducción de direcciones. Indica qué políticas por defecto se están cumpliendo de las cadenas de la tabla nat y cuantos paquetes las han cumplido.

```

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 2 packets, 97 bytes)
num  pkts bytes target    prot opt in     out     source        destination
1      4   204 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
2      4   205 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
3      0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
4      0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
5      0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
6      0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
7      3   180 DNAT      tcp  --  eth2   *      0.0.0.0/0     40.0.0.1
    tcp dpt:80 to:10.8.22.30:80
8      0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
9      0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
10     0     0 DNAT      tcp  --  eth2   *      0.0.0.0/0     40.0.0.1
    tcp dpt:80 to:10.8.22.30:80
11     0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
12     0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
13     0     0 DNAT      tcp  --  eth2   *      0.0.0.0/0     40.0.0.1
    tcp dpt:80 to:10.8.22.30:80
14     0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
15     0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
16     0     0 DNAT      tcp  --  eth2   *      0.0.0.0/0     40.0.0.1
    tcp dpt:80 to:10.8.22.30:80
17     0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5001 to:10.7.22.10:5001
18     0     0 DNAT      udp  --  eth2   *      0.0.0.0/0     40.0.0.1
    udp dpt:5002 to:10.7.22.20:5001
19     0     0 DNAT      tcp  --  eth2   *      0.0.0.0/0     40.0.0.1
    tcp dpt:80 to:10.8.22.30:80

Chain POSTROUTING (policy ACCEPT 15 packets, 829 bytes)
num  pkts bytes target    prot opt in     out     source        destination

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target    prot opt in     out     source        destination

firewall:~#

```

3. Filtrado en el firewall: tabla filter

3.2. Configuración de las reglas de filtrado en el firewall

1. Crea un script en el firewall fw3.sh partiendo de la configuración de traducción de direcciones IP y puertos realizada en fw1.sh que añada la siguiente configuración:

- a) Reiniciar la tabla filter: borrar su contenido y reiniciar sus contadores.
- b) Fijar las políticas por defecto de las cadenas de la tabla filter, haciendo que por defecto se descarte todo el tráfico en el firewall excepto los paquetes de salida.
- c) Permitir el tráfico de entrada dirigido a las aplicaciones que se están ejecutando en firewall únicamente si este tráfico tiene su origen en las subredes privadas de la empresa.
- d) Permitir todo el tráfico saliente desde las subredes privadas hacia Internet y el tráfico de respuesta al saliente. Ten en cuenta que como has partido del script fw1.sh, en dicho script ya tenías las reglas de la tabla nat de modificación de la dirección IP de origen de los paquetes que reenvía el firewall y los paquetes del tráfico entrante de respuesta al saliente.
- e) Permitir desde Internet únicamente el tráfico entrante nuevo hacia la zona DMZ según las siguientes reglas y su correspondiente tráfico de salida: un servidor echo instalado en pc4 (UDP, puerto 7). Debes configurar inetd en pc4 para que arranque este servidor. Utiliza nc para probar la comunicación como cliente desde una máquina de Internet y el tráfico de respuesta. un servidor daytime instalado en pc5 (UDP, puerto 13). Debes configurar inetd en pc5 para que arranque este servidor. Utiliza nc para probar la comunicación como cliente desde una máquina de Internet y el tráfico de respuesta.
- f) Permitir únicamente la comunicación entre la red privada y la zona DMZ de la siguiente forma: Conexión de telnet (TCP, puerto 23) desde pc1 a pc5. Debes configurar inetd en pc5 para que arranque este servidor. Para poder probar esta comunicación, desde pc1 ejecuta: telnet <dir_IP_pc5> Podrás entrar de forma remota en pc5 utilizando usuario: root, clave: root. Conexión al servidor de echo (TCP, puerto 7) desde pc1 a pc4. Debes configurar inetd en pc4 para que arranque este servidor. Utiliza nc para probar la comunicación como cliente desde pc1.
- g) Desde la zona DMZ no se puede iniciar ninguna comunicación con la red privada, ni con el firewall.

Incluye el script fw3.sh en la memoria y explícalo.

Este script tiene como base el fw1.sh en el cual estan configuradas las traducciones nat y snat. Se añadieron nuevas reglas en las cuales lo que hace es permitir o negar trafico. Primero se reiniciaron los contadores de la tabla filter y fueron definidas las politicas por defecto de la tabla filter donde se descarta cualquier cosa execto los paquetes de salida. Despues se le da acceso a la red privada al firewall, se permite el trafico de la red privada a internet y su trafico de respuesta correspondiente. A la DMZ se le niega el acceso al firewall y se le niega el trafico a la red privada, se permite conexiones desde red privada a servidores de la zona DMZ con ip y puertos especificos, asi como tambien desde internet a la DMZ.



firewall

```
#!/bin/bash
```

```
echo "Borrado de reglas y reiniciando los contadores"
```

```
iptables -t nat -F
```

```
iptables -t nat -Z
```

```
#iptables -X
```

```
iptables -t filter -F
```

```
iptables -t filter -Z
```

```
echo "Políticas por defecto"
```

```
iptables -t filter -P INPUT DROP
```

```
iptables -t filter -P OUTPUT ACCEPT
```

```
iptables -t filter -P FORWARD DROP
```

```
echo "Traducciones"
```

```
iptables -t nat -P POSTROUTING ACCEPT
```

```
iptables -t nat -A POSTROUTING -s 10.7.22.0/24 -o eth2 \  
-j SNAT --to-source 40.0.0.1
```

```
iptables -t nat -A POSTROUTING -s 10.7.22.0/24 -o eth2 \  
-j SNAT --to-source 40.0.0.1
```

```
echo "Permitir todo el trafico de entrada al firewall proveniente de las redes privadas"
```

```
iptables -t filter -A INPUT -s 10.7.22.0/24 -i eth0 -j ACCEPT
```

```
iptables -t filter -A INPUT -s 10.8.22.0/24 -i eth0 -j ACCEPT
```

```
echo "Permitir el trafico saliente proveniente de las redes privadas"
```

```
iptables -t filter -A FORWARD -i eth0 -o eth2 -j ACCEPT
```

```
iptables -t filter -A FORWARD -i eth2 -o eth0 -m state \  
--state RELATED,ESTABLISHED -j ACCEPT
```

```
echo "Permitir el trafico desde Internet a la DMZ"
```

```
echo "Conexion servidor echo en pc4 puerto 7 UDP"
```

```
iptables -t filter -A FORWARD -s 60.0.0.60/24 -d 50.0.0.40 -p udp --dport 7 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 70.0.0.70/24 -d 50.0.0.40 -p udp --dport 7 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.40 -d 60.0.0.60/24 -p udp --sport 7 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.40 -d 70.0.0.70/24 -p udp --sport 7 -j ACCEPT
```

```
echo "Conexion servidor daytime en pc5 puerto 13 UDP"
```

```
iptables -t filter -A FORWARD -s 70.0.0.70/24 -d 50.0.0.50 -p udp --dport 13 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 70.0.0.70/24 -d 50.0.0.50 -p tcp --dport 13 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 60.0.0.60/24 -d 50.0.0.50 -p tcp --dport 13 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 60.0.0.60/24 -d 50.0.0.50 -p udp --dport 13 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.50 -d 70.0.0.70/24 -p udp --sport 13 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.50 -d 70.0.0.70/24 -p tcp --sport 13 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.50 -d 60.0.0.60/24 -p tcp --sport 13 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.50 -d 60.0.0.60/24 -p udp --sport 13 -j ACCEPT
```

```
echo "Permitir el trafico desde la red privada a la DMZ"
```

```
echo "Conexion Telnet, puerto 23 TCP, pc1 a pc5"
```

```
iptables -t filter -A FORWARD -s 10.7.22.10 -d 50.0.0.50 -p tcp --dport 23 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.50 -d 10.7.22.10 -p tcp --sport 23 -j ACCEPT
```

```
echo "Conexion servidor echo puerto 7 TCP pc1 a pc4"
```

```
iptables -t filter -A FORWARD -s 10.7.22.10 -d 50.0.0.40 -p tcp --dport 7 -j ACCEPT
```

```
iptables -t filter -A FORWARD -s 50.0.0.40 -d 10.7.22.10 -p tcp --sport 7 -j ACCEPT
```

```
echo "Desde la DMZ no se puede iniciar ninguna conexion"
```

```
iptables -t filter -A FORWARD -s 50.0.0.0/24 -d 10.7.22.0/24 -j DROP
```

```
iptables -t filter -A FORWARD -s 50.0.0.0/24 -d 10.8.22.0/24 -j DROP
```

```
iptables -t filter -A FORWARD -s 50.0.0.0/24 -d 60.0.0.0/24 -j DROP
```

```
iptables -t filter -A FORWARD -s 50.0.0.0/24 -d 70.0.0.0/24 -j DROP
```

```
iptables -t filter -A INPUT -i eth1 -j DROP
```

```
firewall:~# █
```

3.3. Pruebas de la configuración del firewall

Para poder comprobar qué reglas se están aplicando a cada caso que pruebas, añade a cada regla otra regla con las mismas condiciones y acción LOG de forma que quede una anotación en el fichero de log cada vez que se cumpla cada condición.

Pruebas

a) Si se arranca una aplicación servidor (TCP o UDP) en la máquina firewall sólo podrá aceptar tráfico de un cliente que envíe mensajes desde una de las máquinas de las subredes privadas. Asegúrate de que antes de lanzar cliente y servidor has ejecutado fw3.sh para que reinicie los contadores de iptables. Por ejemplo arranca un servidor UDP en firewall y arranca un cliente UDP en pc1 que se comuniquen con dicho servidor (escribe alguna línea en cada uno de los terminales para que haya tráfico UDP).

Explica en la memoria: las reglas en las tablas nat y filter que se han cumplido y el número de veces. las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```
firewall
Politicas por defecto
Traducciones
Permitir todo el trafico de entrada al firewall proveniente de las redes privadas
Permitir el trafico saliente proveniente de las redes privadas
Permitir el trafico desde Internet a la DMZ
Conexion servidor echo en pc4 puerto 7 UDP
Conexion servidor daytime en pc5 puerto 13 UDP
Permitir el trafico desde la red privada a la DMZ
Conexion Telnet, puerto 23 TCP, pc1 a pc5
Conexion servidor echo puerto 7 TCP pc1 a pc4
Desde la DMZ no se puede iniciar ninguna conexion
firewall:~#
```

```
pc1
pc1:~# nc -u -p 6666 10.6.22.3 7777
Linea desde cliente
Linea desde servidor
```

```
firewall
firewall:~# nc -u -l -p 7777
Linea desde cliente
Linea desde servidor
```

```
firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 48 bytes)
num  pkts bytes target    prot opt in     out     source    destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
1    0    0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
2    0    0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
firewall:~#
```

En la tabla NAT se puede ver que no se han requerido ninguna de estas reglas debido a que ningún paquete ha salido del firewall.

En la tabla filter si se puede ver que 2 paquetes han llegado efectivamente al firewall y fueron aceptados.

```

firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      2    97 ACCEPT    all  --  eth0   *       10.7.22.0/24         0.0.0.0/0
2      0     0 ACCEPT    all  --  eth0   *       10.8.22.0/24         0.0.0.0/0
3      0     0 DROP     all  --  eth1   *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0     0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
2      0     0 ACCEPT    all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0
3      0     0 ACCEPT    udp  --  *      *       60.0.0.0/24         50.0.0.40      state RELATED,ESTABLISHED
4      0     0 ACCEPT    udp  --  *      *       70.0.0.0/24         50.0.0.40      udp dpt:7
5      0     0 ACCEPT    udp  --  *      *       50.0.0.40           60.0.0.0/24    udp spt:7
6      0     0 ACCEPT    udp  --  *      *       50.0.0.40           70.0.0.0/24    udp spt:7
7      0     0 ACCEPT    udp  --  *      *       70.0.0.0/24         50.0.0.50      udp dpt:13
8      0     0 ACCEPT    tcp  --  *      *       70.0.0.0/24         50.0.0.50      tcp dpt:13
9      0     0 ACCEPT    tcp  --  *      *       60.0.0.0/24         50.0.0.50      tcp dpt:13
10     0     0 ACCEPT    udp  --  *      *       60.0.0.0/24         50.0.0.50      udp dpt:13
11     0     0 ACCEPT    udp  --  *      *       50.0.0.50           70.0.0.0/24    udp spt:13
12     0     0 ACCEPT    tcp  --  *      *       50.0.0.50           70.0.0.0/24    tcp spt:13
13     0     0 ACCEPT    tcp  --  *      *       50.0.0.50           60.0.0.0/24    tcp spt:13
14     0     0 ACCEPT    udp  --  *      *       50.0.0.50           60.0.0.0/24    udp spt:13
15     0     0 ACCEPT    tcp  --  *      *       10.7.22.10          50.0.0.50      tcp dpt:23
16     0     0 ACCEPT    tcp  --  *      *       50.0.0.50           10.7.22.10     tcp spt:23
17     0     0 ACCEPT    tcp  --  *      *       10.7.22.10          50.0.0.40      tcp dpt:7
18     0     0 ACCEPT    tcp  --  *      *       50.0.0.40           10.7.22.10     tcp spt:7
19     0     0 DROP     all  --  *      *       50.0.0.0/24         10.7.22.0/24
20     0     0 DROP     all  --  *      *       50.0.0.0/24         10.8.22.0/24
21     0     0 DROP     all  --  *      *       50.0.0.0/24         60.0.0.0/24
22     0     0 DROP     all  --  *      *       50.0.0.0/24         70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#

```

b) No podrá aceptar tráfico desde aplicaciones cliente lanzadas en otras subredes diferentes. Asegúrate de que antes de lanzar cliente y servidor has ejecutado fw3.sh para que reinicie los contadores de iptables. Por ejemplo, arranca un servidor UDP en firewall y arranca un cliente UDP en pc6 que se comunique con dicho servidor (escribe alguna línea en cada uno de los terminales para que haya tráfico UDP).

```

firewall
firewall:~# nc -u -l -p 7777

pc6
pc6:~# nc -u 6666 40.0.0.1 7777
pc6:~#

```

Explica en la memoria:

Las reglas en las tablas nat y filter que se han cumplido y el número de veces las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0     0 SNAT     all  --  *      eth2    10.7.22.0/24         0.0.0.0/0      to:40.0.0.1
2      0     0 SNAT     all  --  *      eth2    10.7.22.0/24         0.0.0.0/0      to:40.0.0.1

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#

```

```
firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   *       10.7.22.0/24          0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *       10.8.22.0/24          0.0.0.0/0
3      0      0 DROP      all  --  eth1   *       0.0.0.0/0             0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0
3      0      0 ACCEPT    udp  --  *      *       60.0.0.0/24          50.0.0.40
4      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24          50.0.0.40
5      0      0 ACCEPT    udp  --  *      *       50.0.0.40            60.0.0.0/24
6      0      0 ACCEPT    udp  --  *      *       50.0.0.40            70.0.0.0/24
7      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24          50.0.0.50
8      0      0 ACCEPT    tcp  --  *      *       70.0.0.0/24          50.0.0.50
9      0      0 ACCEPT    tcp  --  *      *       60.0.0.0/24          50.0.0.50
10     0      0 ACCEPT    udp  --  *      *       60.0.0.0/24          50.0.0.50
11     0      0 ACCEPT    udp  --  *      *       50.0.0.50            70.0.0.0/24
12     0      0 ACCEPT    tcp  --  *      *       50.0.0.50            70.0.0.0/24
13     0      0 ACCEPT    tcp  --  *      *       50.0.0.50            60.0.0.0/24
14     0      0 ACCEPT    udp  --  *      *       50.0.0.50            60.0.0.0/24
15     0      0 ACCEPT    tcp  --  *      *       10.7.22.10           50.0.0.50
16     0      0 ACCEPT    tcp  --  *      *       50.0.0.50            10.7.22.10
17     0      0 ACCEPT    tcp  --  *      *       10.7.22.10           50.0.0.40
18     0      0 ACCEPT    tcp  --  *      *       50.0.0.40            10.7.22.10
19     0      0 DROP      all  --  *      *       50.0.0.0/24          10.7.22.0/24
20     0      0 DROP      all  --  *      *       50.0.0.0/24          10.8.22.0/24
21     0      0 DROP      all  --  *      *       50.0.0.0/24          60.0.0.0/24
22     0      0 DROP      all  --  *      *       50.0.0.0/24          70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

2. Permitir todo el tráfico saliente desde las subredes privadas hacia Internet, modificando la dirección IP de origen de los paquetes que reenvía el firewall, y el tráfico entrante de respuesta al saliente.

Pruebas:

a) Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de Internet y se arranca una aplicación cliente para que se comunice con ese servidor en una de las máquinas de las subredes internas, el tráfico debe poder enviarse del cliente al servidor y del servidor al cliente, observando que el tráfico que sale del firewall con destino a la máquina de Internet no tiene como dirección IP origen la dirección de la máquina que pertenece a la subred privada, sino que lleva la dirección pública del firewall de la interfaz que le conecta con Internet. Ejecuta la misma prueba que en el apartado 2.1.3. Asegúrate de que antes de lanzar cliente y servidor has ejecutado fw3.sh para que reinicie los contadores de iptables. Explica en la memoria:

Las reglas en las tablas nat y filter que se han cumplido y el número de veces. las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```
pc1
pc1:~# nc -p 6666 60.0.0.60 7777
Enviado desde PC1
Enviado a PC1

pc6
pc6:~# nc -l -p 7777
Enviado desde PC1
Enviado a PC1
```

En este caso se puede ver en ambas tablas que los paquetes han cruzado el firewall en ambos sentidos y se ha hecho uso de las reglas definidas tanto para la traducción así como para el respectivo enrutamiento.

```

firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   *       10.7.22.0/24         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *       10.8.22.0/24         0.0.0.0/0
3      0      0 DROP      all  --  eth1   *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      4    234 ACCEPT    all  --  eth0   eth2    0.0.0.0/0           0.0.0.0/0
2      3    178 ACCEPT    all  --  eth2   eth0    0.0.0.0/0           0.0.0.0/0      state RELATED,ESTABLISHED
3      0      0 ACCEPT    udp  --  *      *       60.0.0.0/24         50.0.0.40      udp dpt:7
4      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24         50.0.0.40      udp dpt:7
5      0      0 ACCEPT    udp  --  *      *       50.0.0.40           60.0.0.0/24    udp spt:7
6      0      0 ACCEPT    udp  --  *      *       50.0.0.40           70.0.0.0/24    udp spt:7
7      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24         50.0.0.50      udp spt:13
8      0      0 ACCEPT    tcp  --  *      *       70.0.0.0/24         50.0.0.50      tcp dpt:13
9      0      0 ACCEPT    tcp  --  *      *       60.0.0.0/24         50.0.0.50      tcp dpt:13
10     0      0 ACCEPT    udp  --  *      *       60.0.0.0/24         50.0.0.50      udp dpt:13
11     0      0 ACCEPT    udp  --  *      *       50.0.0.50           70.0.0.0/24    udp spt:13
12     0      0 ACCEPT    tcp  --  *      *       50.0.0.50           70.0.0.0/24    tcp spt:13
13     0      0 ACCEPT    tcp  --  *      *       50.0.0.50           60.0.0.0/24    tcp spt:13
14     0      0 ACCEPT    udp  --  *      *       50.0.0.50           60.0.0.0/24    udp spt:13
15     0      0 ACCEPT    tcp  --  *      *       10.7.22.10          50.0.0.50      tcp dpt:23
16     0      0 ACCEPT    tcp  --  *      *       50.0.0.50           10.7.22.10     tcp spt:23
17     0      0 ACCEPT    tcp  --  *      *       10.7.22.10          50.0.0.40      tcp dpt:7
18     0      0 ACCEPT    tcp  --  *      *       50.0.0.40           10.7.22.10     tcp spt:7
19     0      0 DROP      all  --  *      *       50.0.0.0/24         10.7.22.0/24
20     0      0 DROP      all  --  *      *       50.0.0.0/24         10.8.22.0/24
21     0      0 DROP      all  --  *      *       50.0.0.0/24         60.0.0.0/24
22     0      0 DROP      all  --  *      *       50.0.0.0/24         70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#

```

```

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      1     60 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0      to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0      to:40.0.0.1

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#

```

b) Si se arranca una aplicación cliente en pc4 o pc5 para comunicarse con el servidor que se haya arrancado en una de las máquinas de Internet, el firewall no debería permitir reenviar ese tráfico hacia Internet. Asegúrate de que antes de lanzar cliente y servidor has ejecutado fw3.sh para que reinicie los contadores de iptables. Explica en la memoria:

Las reglas en las tablas nat y filter que se han cumplido y el número de veces. las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```

pc4
pc4:~# nc -p 6666 60.0.0.60 7777
Probando conexion

pc6
pc6:~# nc -l -p 7777

```

A como se puede observar la conexión se realiza debido a que los paquetes del establecimiento de la conexión se eliminan al llegar al router a como se ve en la tabla filter.

```

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target    prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0           to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0           to:40.0.0.1

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#

```

```

firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   *      10.7.22.0/24         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *      10.8.22.0/24         0.0.0.0/0
3      0      0 DROP      all  --  eth1   *      0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0
3      0      0 ACCEPT    udp  --  *      *       60.0.0.0/24          50.0.0.40           udp dpt:7
4      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24          50.0.0.40           udp dpt:7
5      0      0 ACCEPT    udp  --  *      *       50.0.0.40            60.0.0.0/24         udp spt:7
6      0      0 ACCEPT    udp  --  *      *       50.0.0.40            70.0.0.0/24         udp spt:7
7      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24          50.0.0.50           udp dpt:13
8      0      0 ACCEPT    tcp  --  *      *       70.0.0.0/24          50.0.0.50           tcp dpt:13
9      0      0 ACCEPT    tcp  --  *      *       60.0.0.0/24          50.0.0.50           tcp dpt:13
10     0      0 ACCEPT    udp  --  *      *       60.0.0.0/24          50.0.0.50           udp dpt:13
11     0      0 ACCEPT    tcp  --  *      *       50.0.0.50            70.0.0.0/24         tcp spt:13
12     0      0 ACCEPT    tcp  --  *      *       50.0.0.50            60.0.0.0/24         tcp spt:13
13     0      0 ACCEPT    tcp  --  *      *       50.0.0.50            60.0.0.0/24         tcp spt:13
14     0      0 ACCEPT    udp  --  *      *       50.0.0.50            60.0.0.0/24         udp spt:13
15     0      0 ACCEPT    tcp  --  *      *       10.7.22.10           50.0.0.50           tcp dpt:23
16     0      0 ACCEPT    tcp  --  *      *       50.0.0.50            10.7.22.10          tcp spt:23
17     0      0 ACCEPT    tcp  --  *      *       10.7.22.10           50.0.0.40           tcp dpt:7
18     0      0 ACCEPT    tcp  --  *      *       50.0.0.40            10.7.22.10          tcp spt:7
19     0      0 DROP      all  --  *      *       50.0.0.0/24          10.7.22.0/24
20     0      0 DROP      all  --  *      *       50.0.0.0/24          10.8.22.0/24
21     5    300 DROP      all  --  *      *       50.0.0.0/24          60.0.0.0/24
22     0      0 DROP      all  --  *      *       50.0.0.0/24          70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#

```

3. Permitir desde Internet únicamente el tráfico entrante nuevo hacia la zona DMZ según las siguientes reglas:

- Un servidor echo instalado en pc4 (UDP, puerto 7).

Pruebas:

a) Desde una máquina de Internet se debería poder acceder a ese servidor de echo de pc4. Ejecuta el siguiente comando desde una máquina de Internet:

```
nc -u <dir_IP_pc4> 7
```

Asegúrate de que antes de lanzar el cliente desde una máquina de Internet has ejecutado fw3.sh para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas nat y filter que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```

pc4
pc4:~# /etc/init.d/inetd start
Starting internet superserver: inetd.
pc4:~#

```



```
pc6:~# nc -u 50.0.0.40 7
Esto es una prueba del servidor echo
Esto es una prueba del servidor echo
```

```
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0    *       10.7.22.0/24          0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0    *       10.8.22.0/24          0.0.0.0/0
3      0      0 DROP      all  --  eth1    *       0.0.0.0/0             0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination      state
1      0      0 ACCEPT    all  --  eth0    eth2    0.0.0.0/0            0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2    eth0    0.0.0.0/0            0.0.0.0/0
3      1      65 ACCEPT    udp  --  *       *       60.0.0.0/24          50.0.0.40        udp dpt:7
4      0      0 ACCEPT    udp  --  *       *       70.0.0.0/24          50.0.0.40        udp dpt:7
5      1      65 ACCEPT    udp  --  *       *       50.0.0.40            60.0.0.0/24      udp spt:7
6      0      0 ACCEPT    udp  --  *       *       50.0.0.40            70.0.0.0/24      udp spt:7
7      0      0 ACCEPT    udp  --  *       *       70.0.0.0/24          50.0.0.50        udp dpt:13
8      0      0 ACCEPT    tcp  --  *       *       70.0.0.0/24          50.0.0.50        tcp dpt:13
9      0      0 ACCEPT    tcp  --  *       *       60.0.0.0/24          50.0.0.50        tcp dpt:13
10     0      0 ACCEPT    udp  --  *       *       60.0.0.0/24          50.0.0.50        udp dpt:13
11     0      0 ACCEPT    udp  --  *       *       50.0.0.50            70.0.0.0/24      udp spt:13
12     0      0 ACCEPT    tcp  --  *       *       50.0.0.50            70.0.0.0/24      tcp spt:13
13     0      0 ACCEPT    tcp  --  *       *       50.0.0.50            60.0.0.0/24      tcp spt:13
14     0      0 ACCEPT    udp  --  *       *       50.0.0.50            60.0.0.0/24      udp spt:13
15     0      0 ACCEPT    tcp  --  *       *       10.7.22.10           50.0.0.50        tcp dpt:23
16     0      0 ACCEPT    tcp  --  *       *       50.0.0.50            10.7.22.10       tcp spt:23
17     0      0 ACCEPT    tcp  --  *       *       10.7.22.10           50.0.0.40        tcp dpt:7
18     0      0 ACCEPT    tcp  --  *       *       50.0.0.40            10.7.22.10       tcp spt:7
19     0      0 DROP      all  --  *       *       50.0.0.0/24          10.7.22.0/24
20     0      0 DROP      all  --  *       *       50.0.0.0/24          10.8.22.0/24
21     0      0 DROP      all  --  *       *       50.0.0.0/24          60.0.0.0/24
22     0      0 DROP      all  --  *       *       50.0.0.0/24          70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

```
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 65 bytes)
num  pkts bytes target    prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 1 packets, 65 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 SNAT      all  --  *       eth2    10.7.22.0/24          0.0.0.0/0        to:40.0.0.1
2      0      0 SNAT      all  --  *       eth2    10.7.22.0/24          0.0.0.0/0        to:40.0.0.1

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

A como se ve en la tabla filter se cumplen 2 reglas en las cuales esta definido el permitir ese tipo de conexión y su trafico de respuesta. En la tabla nat no se refleja ningun cambio por lo que la conexión es desde Internet a la DMZ y no es necesario ningun tipo de traduccion.

b) Si se prueba lo mismo arrancando el comando anterior desde pc3 y se manda una cadena de caracteres, no se debería obtener respuesta. Asegúrate de que antes de lanzar el cliente de pc3 has ejecutado fw3.sh para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas nat y filter que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas nat y

filter y el número de veces que se ejecutan.

```
pc4:~# /etc/init.d/inetd start
Starting internet superserver: inetd.
pc4:~#

pc3:~# nc -u 50.0.0.40 7
Probando servidor echo en DMZ

firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 58 bytes)
num  pkts bytes target    prot opt in     out     source    destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
1    0    0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
2    0    0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
firewall:~#

firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
1    0    0 ACCEPT    all  --  eth0    *      10.7.22.0/24  0.0.0.0/0
2    0    0 ACCEPT    all  --  eth0    *      10.8.22.0/24  0.0.0.0/0
3    0    0 DROP      all  --  eth1    *      0.0.0.0/0    0.0.0.0/0
Chain FORWARD (policy DROP 1 packets, 58 bytes)
num  pkts bytes target    prot opt in     out     source    destination
1    0    0 ACCEPT    all  --  eth0    eth2    0.0.0.0/0  0.0.0.0/0
2    0    0 ACCEPT    all  --  eth2    eth0    0.0.0.0/0  0.0.0.0/0      state RELATED,ESTABLISHED
3    0    0 ACCEPT    udp  --  *      *      60.0.0.0/24  50.0.0.40      udp dpt:7
4    0    0 ACCEPT    udp  --  *      *      70.0.0.0/24  50.0.0.40      udp dpt:7
5    0    0 ACCEPT    udp  --  *      *      50.0.0.40    60.0.0.0/24    udp spt:7
6    0    0 ACCEPT    udp  --  *      *      50.0.0.40    70.0.0.0/24    udp spt:7
7    0    0 ACCEPT    udp  --  *      *      70.0.0.0/24  50.0.0.50      udp dpt:13
8    0    0 ACCEPT    tcp  --  *      *      70.0.0.0/24  50.0.0.50      tcp dpt:13
9    0    0 ACCEPT    tcp  --  *      *      60.0.0.0/24  50.0.0.50      tcp dpt:13
10   0    0 ACCEPT    udp  --  *      *      60.0.0.0/24  50.0.0.50      udp dpt:13
11   0    0 ACCEPT    udp  --  *      *      50.0.0.50    70.0.0.0/24    udp spt:13
12   0    0 ACCEPT    tcp  --  *      *      50.0.0.50    70.0.0.0/24    tcp spt:13
13   0    0 ACCEPT    tcp  --  *      *      50.0.0.50    60.0.0.0/24    tcp spt:13
14   0    0 ACCEPT    udp  --  *      *      50.0.0.50    60.0.0.0/24    udp spt:13
15   0    0 ACCEPT    tcp  --  *      *      10.7.22.10   50.0.0.50      tcp dpt:23
16   0    0 ACCEPT    tcp  --  *      *      50.0.0.50    10.7.22.10     tcp spt:23
17   0    0 ACCEPT    tcp  --  *      *      10.7.22.10   50.0.0.40      tcp dpt:7
18   0    0 ACCEPT    tcp  --  *      *      50.0.0.40    10.7.22.10     tcp spt:7
19   0    0 DROP      all  --  *      *      50.0.0.0/24  10.7.22.0/24
20   0    0 DROP      all  --  *      *      50.0.0.0/24  10.8.22.0/24
21   0    0 DROP      all  --  *      *      50.0.0.0/24  60.0.0.0/24
22   0    0 DROP      all  --  *      *      50.0.0.0/24  70.0.0.0/24
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
firewall:~#
```

Como vemos no se tiene respuesta del servidor por lo que la conexión no se realiza debido a que en el firewall no hay definida una regla para permitir esta conexión, sin embargo tampoco hay una que la niegue, pero por defecto es eliminada en la configuracion de las politicas por defecto.

- Un servidor daytime instalado en pc5 (UDP, puerto 13). El servidor daytime es un servidor que al enviarle algo, devuelve la fecha y hora de la máquina donde está instalado.

Pruebas:

a) Desde una máquina de Internet se debería poder obtener la hora de pc5. Ejecuta el siguiente comando desde una máquina de Internet:

```
nc -u <dir_IP_pc5> 13
```

Pulsa < Enter > en el terminal de nc y debería obtenerse la hora que le envía pc5.

Asegúrate de que antes de lanzar el cliente en pc5 has ejecutado fw3.sh para que reinicie los contadores de iptables.

Explica en la memoria:

- las reglas en las tablas nat y filter que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan..

```
pc5
pc5:~# /etc/init.d/inetd restart
Restarting internet superserver: inetd.
pc5:~#

pc6
pc6:~# nc 50.0.0.50 13
Sat Apr 27 05:33:26 2019
pc6:~#
```

Nota: HE REALIZADO LA PRUEBA DEL SERVIDOR DAYTIME CON EL PUERTO TCP PORQUE CON UDP NO ME CONECTA AUN SIN HABER APLICADO ALGUNA RESTRICCION EN EL FIREWALL

```
firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
1      0      0 ACCEPT    all  --  eth0   *       10.7.22.0/24      0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *       10.8.22.0/24      0.0.0.0/0
3      0      0 DROP      all  --  eth1   *       0.0.0.0/0         0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination      state
1      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2   eth0    0.0.0.0/0         0.0.0.0/0
3      0      0 ACCEPT    udp  --  *      *       60.0.0.0/24       50.0.0.40        udp dpt:7
4      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24       50.0.0.40        udp dpt:7
5      0      0 ACCEPT    udp  --  *      *       50.0.0.40         60.0.0.0/24      udp spt:7
6      0      0 ACCEPT    udp  --  *      *       50.0.0.40         70.0.0.0/24      udp spt:7
7      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24       50.0.0.50        udp dpt:13
8      0      0 ACCEPT    tcp  --  *      *       70.0.0.0/24       50.0.0.50        tcp dpt:13
9      4    216 ACCEPT    tcp  --  *      *       60.0.0.0/24       50.0.0.50        tcp dpt:13
10     0      0 ACCEPT    udp  --  *      *       60.0.0.0/24       50.0.0.50        udp dpt:13
11     0      0 ACCEPT    udp  --  *      *       50.0.0.50         70.0.0.0/24      udp spt:13
12     0      0 ACCEPT    tcp  --  *      *       50.0.0.50         70.0.0.0/24      tcp spt:13
13     4    242 ACCEPT    tcp  --  *      *       50.0.0.50         60.0.0.0/24      tcp spt:13
14     0      0 ACCEPT    udp  --  *      *       50.0.0.50         60.0.0.0/24      udp spt:13
15     0      0 ACCEPT    tcp  --  *      *       10.7.22.10        50.0.0.50        tcp dpt:23
16     0      0 ACCEPT    tcp  --  *      *       50.0.0.50         10.7.22.10       tcp spt:23
17     0      0 ACCEPT    tcp  --  *      *       10.7.22.10        50.0.0.40        tcp dpt:7
18     0      0 ACCEPT    tcp  --  *      *       50.0.0.40         10.7.22.10       tcp spt:7
19     0      0 DROP      all  --  *      *       50.0.0.0/24       10.7.22.0/24
20     0      0 DROP      all  --  *      *       50.0.0.0/24       10.8.22.0/24
21     0      0 DROP      all  --  *      *       50.0.0.0/24       60.0.0.0/24
22     0      0 DROP      all  --  *      *       50.0.0.0/24       70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
firewall:~#

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source            destination
Chain POSTROUTING (policy ACCEPT 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source            destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24      0.0.0.0/0        to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24      0.0.0.0/0        to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
firewall:~#
```

b) No se debe permitir otro tipo de tráfico desde Internet a DMZ. Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de DMZ y se arranca una aplicación cliente para que se comunice con ese servidor en una de las máquinas de Internet, el tráfico no debería poder enviarse del cliente al servidor ni del servidor al cliente.

Haz una prueba para este tipo de tráfico y explica que prueba estás haciendo.

Asegúrate de que antes de lanzar el cliente has ejecutado fw3.sh para que reinicie los contadores de iptables. Explica en la memoria:

- las reglas en las tablas nat y filter que se han cumplido y el número de veces.
- las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```
pc5
pc5:~# nc -l -p 7777

pc6
pc6:~# nc -p 8888 50.0.0.50 7777
Probando conexion

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 6 packets, 360 bytes)
num  pkts bytes target    prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0           to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0           to:40.0.0.1

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#

firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0    *        10.7.22.0/24         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0    *        10.8.22.0/24         0.0.0.0/0
3      0      0 DROP      all  --  eth1    *        0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 6 packets, 360 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0    eth2     0.0.0.0/0            0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2    eth0     0.0.0.0/0            0.0.0.0/0
3      0      0 ACCEPT    udp  --  *       *        60.0.0.0/24         50.0.0.40           udp dpt:7
4      0      0 ACCEPT    udp  --  *       *        70.0.0.0/24         50.0.0.40           udp dpt:7
5      0      0 ACCEPT    udp  --  *       *        50.0.0.40            60.0.0.0/24         udp spt:7
6      0      0 ACCEPT    udp  --  *       *        50.0.0.40            70.0.0.0/24         udp spt:7
7      0      0 ACCEPT    udp  --  *       *        70.0.0.0/24         50.0.0.50           udp dpt:13
8      0      0 ACCEPT    tcp  --  *       *        70.0.0.0/24         50.0.0.50           tcp dpt:13
9      0      0 ACCEPT    tcp  --  *       *        50.0.0.0/24         50.0.0.50           tcp dpt:13
10     0      0 ACCEPT    udp  --  *       *        60.0.0.0/24         50.0.0.50           udp dpt:13
11     0      0 ACCEPT    udp  --  *       *        50.0.0.50            70.0.0.0/24         udp spt:13
12     0      0 ACCEPT    tcp  --  *       *        50.0.0.50            70.0.0.0/24         tcp spt:13
13     0      0 ACCEPT    tcp  --  *       *        50.0.0.50            60.0.0.0/24         tcp spt:13
14     0      0 ACCEPT    udp  --  *       *        50.0.0.50            60.0.0.0/24         udp spt:13
15     0      0 ACCEPT    tcp  --  *       *        10.7.22.10           50.0.0.50           tcp dpt:23
16     0      0 ACCEPT    tcp  --  *       *        50.0.0.50            10.7.22.10          tcp spt:23
17     0      0 ACCEPT    tcp  --  *       *        10.7.22.10           50.0.0.40           tcp dpt:7
18     0      0 ACCEPT    tcp  --  *       *        50.0.0.40            10.7.22.10          tcp spt:7
19     0      0 DROP      all  --  *       *        50.0.0.0/24         10.7.22.0/24
20     0      0 DROP      all  --  *       *        50.0.0.0/24         10.8.22.0/24
21     0      0 DROP      all  --  *       *        50.0.0.0/24         60.0.0.0/24
22     0      0 DROP      all  --  *       *        50.0.0.0/24         70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

La conexión no se ha realizado, los paquetes fueron eliminados esto debido al establecer las políticas por defecto y no tener ninguna regla adicional que permita la conexión.

4. Permitir únicamente la comunicación entre la red privada y la zona DMZ de la siguiente forma:

a) Conexión de telnet (TCP, puerto 23) desde pc1 a pc5. La conexión de telnet permite a un usuario conectarse de forma remota a otra máquina.

Pruebas

1) Asegúrate de que antes de lanzar el cliente en pc1 has ejecutado fw3.sh para que reinicie los contadores de iptables. Desde pc1 ejecuta el cliente de telnet:

```
telnet <dir_IP_pc5>
```

podrás entrar de forma remota en pc5 utilizando usuario: root, clave: root. Explica en la memoria: Las reglas en las tablas nat y filter que se han cumplido y el número de veces.

Las políticas por defecto se ejecutan en las cadenas de las tablas nat y filter y el número de veces.

```
pc5:~# /etc/init.d/inetd start
Starting internet superserver: inetd.
pc5:~#

pc1:~# telnet 50.0.0.50
Trying 50.0.0.50...
Connected to 50.0.0.50.
Escape character is '^]'.
Debian GNU/Linux 5.0
pc5 login:

pc5:~# telnet 50.0.0.50
Trying 50.0.0.50...
Connected to 50.0.0.50.
Escape character is '^]'.
Debian GNU/Linux 5.0
pc5 login: root
Password:
Last login: Sat Apr 27 06:27:42 UTC 2019 on pts/0
pc5:~#
```

```
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
1      0      0 ACCEPT    all  --  eth0   *      10.7.22.0/24      0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *      10.8.22.0/24      0.0.0.0/0
3      0      0 DROP      all  --  eth1   *      0.0.0.0/0         0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination      state
1      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2   eth0    0.0.0.0/0         0.0.0.0/0
3      0      0 ACCEPT    udp  --  *      *       60.0.0.0/24      50.0.0.40        udp dpt:7
4      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24      50.0.0.40        udp dpt:7
5      0      0 ACCEPT    udp  --  *      *       50.0.0.40        60.0.0.0/24      udp spt:7
6      0      0 ACCEPT    udp  --  *      *       50.0.0.40        70.0.0.0/24      udp spt:7
7      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24      50.0.0.50        udp dpt:13
8      0      0 ACCEPT    tcp  --  *      *       70.0.0.0/24      50.0.0.50        tcp dpt:13
9      0      0 ACCEPT    tcp  --  *      *       60.0.0.0/24      50.0.0.50        tcp dpt:13
10     0      0 ACCEPT    udp  --  *      *       60.0.0.0/24      50.0.0.50        udp dpt:13
11     0      0 ACCEPT    udp  --  *      *       50.0.0.50        70.0.0.0/24      udp spt:13
12     0      0 ACCEPT    tcp  --  *      *       50.0.0.50        70.0.0.0/24      tcp spt:13
13     0      0 ACCEPT    tcp  --  *      *       50.0.0.50        60.0.0.0/24      tcp spt:13
14     0      0 ACCEPT    udp  --  *      *       50.0.0.50        60.0.0.0/24      udp spt:13
15    29    1604 ACCEPT    tcp  --  *      *       10.7.22.10       50.0.0.50        tcp dpt:23
16    23   1372 ACCEPT    tcp  --  *      *       50.0.0.50        10.7.22.10       tcp spt:23
17     0      0 ACCEPT    tcp  --  *      *       10.7.22.10       50.0.0.40        tcp dpt:7
18     0      0 ACCEPT    tcp  --  *      *       50.0.0.40        10.7.22.10       tcp spt:7
19     0      0 DROP      all  --  *      *       50.0.0.0/24      10.7.22.0/24
20     0      0 DROP      all  --  *      *       50.0.0.0/24      10.8.22.0/24
21     0      0 DROP      all  --  *      *       50.0.0.0/24      60.0.0.0/24
22     0      0 DROP      all  --  *      *       50.0.0.0/24      70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
firewall:~#
```

```
firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source      destination
Chain POSTROUTING (policy ACCEPT 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source      destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source      destination
firewall:~#
```

2) Si se prueba lo mismo arrancando el cliente de telnet desde pc2 o pc3 o cualquier máquina de Internet no debería permitir la conexión. Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado fw3.sh para que reinicie los contadores de iptables.

Explica en la memoria:

Las reglas en las tablas nat y filter que se han cumplido y el número de veces.

las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```
pc2
pc2:~# telnet 50.0.0.50
Trying 50.0.0.50...
```

```
firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 3 packets, 180 bytes)
num  pkts bytes target    prot opt in     out     source      destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source      destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source      destination
firewall:~#
```

```
firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source      destination
1      0      0 ACCEPT    all  --  eth0   *      10.7.22.0/24  0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *      10.8.22.0/24  0.0.0.0/0
3      0      0 DROP      all  --  eth1   *      0.0.0.0/0    0.0.0.0/0
Chain FORWARD (policy DROP 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source      destination
1      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0    0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0    0.0.0.0/0
3      0      0 ACCEPT    udp  --  *      *      60.0.0.0/24  50.0.0.40      udp dpt:7
4      0      0 ACCEPT    udp  --  *      *      70.0.0.0/24  50.0.0.40      udp dpt:7
5      0      0 ACCEPT    udp  --  *      *      50.0.0.40    60.0.0.0/24    udp spt:7
6      0      0 ACCEPT    udp  --  *      *      50.0.0.40    70.0.0.0/24    udp spt:7
7      0      0 ACCEPT    udp  --  *      *      70.0.0.0/24  50.0.0.50      udp dpt:13
8      0      0 ACCEPT    tcp  --  *      *      70.0.0.0/24  50.0.0.50      tcp dpt:13
9      0      0 ACCEPT    tcp  --  *      *      60.0.0.0/24  50.0.0.50      tcp dpt:13
10     0      0 ACCEPT    udp  --  *      *      60.0.0.0/24  50.0.0.50      udp dpt:13
11     0      0 ACCEPT    udp  --  *      *      50.0.0.50    70.0.0.0/24    udp spt:13
12     0      0 ACCEPT    tcp  --  *      *      50.0.0.50    70.0.0.0/24    tcp spt:13
13     0      0 ACCEPT    tcp  --  *      *      50.0.0.50    60.0.0.0/24    tcp spt:13
14     0      0 ACCEPT    udp  --  *      *      50.0.0.50    60.0.0.0/24    udp spt:13
15     0      0 ACCEPT    tcp  --  *      *      10.7.22.10   50.0.0.50      tcp dpt:23
16     0      0 ACCEPT    tcp  --  *      *      50.0.0.50    10.7.22.10     tcp spt:23
17     0      0 ACCEPT    tcp  --  *      *      10.7.22.10   50.0.0.40      tcp dpt:7
18     0      0 ACCEPT    tcp  --  *      *      50.0.0.40    10.7.22.10     tcp spt:7
19     0      0 DROP      all  --  *      *      50.0.0.0/24  10.7.22.0/24
20     0      0 DROP      all  --  *      *      50.0.0.0/24  10.8.22.0/24
21     0      0 DROP      all  --  *      *      50.0.0.0/24  60.0.0.0/24
22     0      0 DROP      all  --  *      *      50.0.0.0/24  70.0.0.0/24
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source      destination
firewall:~#
```

b) Conexión al servidor de echo (TCP, puerto 7) desde pc1 a pc4.

Si se arranca cualquier otra aplicación servidor (TCP o UDP) en una de las máquinas de la DMZ y se arranca una aplicación cliente para que se comunice con ese servidor en una de las máquinas de las subredes privadas, el tráfico no debería poder enviarse del cliente al servidor ni del servidor al cliente.

Pruebas

1) Asegúrate de que antes de lanzar el cliente en pc1 has ejecutado fw3.sh para que reinicie los contadores de iptables. Desde pc1 se debería poder conectarse al servidor de echo de pc4:

```
nc <dir_IP_pc4> 7
```

Explica en la memoria:

Las reglas en las tablas nat y filter que se han cumplido y el número de veces.

Las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```
pc1
pc1:~# nc 50.0.0.40 7
Probando servidor echo de la DMZ
Probando servidor echo de la DMZ
```

```
firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   *      10.7.22.0/24         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *      10.8.22.0/24         0.0.0.0/0
3      0      0 DROP      all  --  eth1   *      0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0
3      0      0 ACCEPT    udp  --  *      *       60.0.0.0/24         50.0.0.40
4      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24         50.0.0.40
5      0      0 ACCEPT    udp  --  *      *       50.0.0.40           60.0.0.0/24
6      0      0 ACCEPT    udp  --  *      *       50.0.0.40           70.0.0.0/24
7      0      0 ACCEPT    udp  --  *      *       70.0.0.0/24         50.0.0.50
8      0      0 ACCEPT    tcp  --  *      *       70.0.0.0/24         50.0.0.50
9      0      0 ACCEPT    tcp  --  *      *       60.0.0.0/24         50.0.0.50
10     0      0 ACCEPT    udp  --  *      *       60.0.0.0/24         50.0.0.50
11     0      0 ACCEPT    udp  --  *      *       50.0.0.50           70.0.0.0/24
12     0      0 ACCEPT    tcp  --  *      *       50.0.0.50           70.0.0.0/24
13     0      0 ACCEPT    tcp  --  *      *       50.0.0.50           60.0.0.0/24
14     0      0 ACCEPT    udp  --  *      *       60.0.0.0/24         50.0.0.13
15     0      0 ACCEPT    tcp  --  *      *       10.7.22.10          50.0.0.50
16     0      0 ACCEPT    tcp  --  *      *       50.0.0.50           10.7.22.10
17     6    353 ACCEPT    tcp  --  *      *       10.7.22.10          50.0.0.40
18     4   249 ACCEPT    tcp  --  *      *       50.0.0.40           10.7.22.10
19     0      0 DROP      all  --  *      *       50.0.0.0/24         10.7.22.0/24
20     0      0 DROP      all  --  *      *       50.0.0.0/24         10.8.22.0/24
21     0      0 DROP      all  --  *      *       50.0.0.0/24         60.0.0.0/24
22     0      0 DROP      all  --  *      *       50.0.0.0/24         70.0.0.0/24

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

```
firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source               destination
Chain POSTROUTING (policy ACCEPT 1 packets, 60 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

2) Si se prueba lo mismo arrancando nc desde pc2 o pc3 no debería conectarse.

Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado fw3.sh para que reinicie los contadores de iptables.

Explica en la memoria:

las reglas en las tablas nat y filter que se han cumplido y el número de veces.

las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```
pc2:~# nc 50.0.0.40 7
Probando servidor echo de la DMZ

firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 3 packets, 180 bytes)
num  pkts bytes target    prot opt in     out     source    destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24  0.0.0.0/0      to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
firewall:~#

firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
1      0      0 ACCEPT    all  --  eth0    *      10.7.22.0/24  0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0    *      10.8.22.0/24  0.0.0.0/0
3      0      0 DROP      all  --  eth1    *      0.0.0.0/0    0.0.0.0/0
Chain FORWARD (policy DROP 3 packets, 180 bytes)
num  pkts bytes target    prot opt in     out     source    destination
1      0      0 ACCEPT    all  --  eth0    eth2    0.0.0.0/0    0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2    eth0    0.0.0.0/0    0.0.0.0/0      state RELATED,ESTABLISHED
3      0      0 ACCEPT    udp  --  *        *      60.0.0.0/24  50.0.0.40      udp dpt:7
4      0      0 ACCEPT    udp  --  *        *      70.0.0.0/24  50.0.0.40      udp dpt:7
5      0      0 ACCEPT    udp  --  *        *      50.0.0.40    60.0.0.0/24    udp spt:7
6      0      0 ACCEPT    udp  --  *        *      50.0.0.40    70.0.0.0/24    udp spt:7
7      0      0 ACCEPT    udp  --  *        *      70.0.0.0/24  50.0.0.50      udp dpt:13
8      0      0 ACCEPT    tcp  --  *        *      70.0.0.0/24  50.0.0.50      tcp dpt:13
9      0      0 ACCEPT    tcp  --  *        *      60.0.0.0/24  50.0.0.50      tcp dpt:13
10     0      0 ACCEPT    udp  --  *        *      60.0.0.0/24  50.0.0.50      udp dpt:13
11     0      0 ACCEPT    udp  --  *        *      50.0.0.50    70.0.0.0/24    udp spt:13
12     0      0 ACCEPT    tcp  --  *        *      50.0.0.50    70.0.0.0/24    tcp spt:13
13     0      0 ACCEPT    tcp  --  *        *      50.0.0.50    60.0.0.0/24    tcp spt:13
14     0      0 ACCEPT    udp  --  *        *      50.0.0.50    60.0.0.0/24    udp spt:13
15     0      0 ACCEPT    tcp  --  *        *      10.7.22.10   50.0.0.50      tcp dpt:23
16     0      0 ACCEPT    tcp  --  *        *      50.0.0.50    10.7.22.10     tcp spt:23
17     0      0 ACCEPT    tcp  --  *        *      10.7.22.10   50.0.0.40      tcp dpt:7
18     0      0 ACCEPT    tcp  --  *        *      50.0.0.40    10.7.22.10     tcp spt:7
19     0      0 DROP      all  --  *        *      50.0.0.0/24  10.7.22.0/24
20     0      0 DROP      all  --  *        *      50.0.0.0/24  10.8.22.0/24
21     0      0 DROP      all  --  *        *      50.0.0.0/24  60.0.0.0/24
22     0      0 DROP      all  --  *        *      50.0.0.0/24  70.0.0.0/24
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source    destination
firewall:~#
```

5. Desde la zona DMZ no se puede iniciar ninguna comunicación con la red privada, ni con el firewall.

Pruebas

a) Si se arranca una aplicación servidor (TCP o UDP) en una de las máquinas de las subredes privadas y se arranca una aplicación cliente para que se comuniquen con ese servidor en una de las máquinas de DMZ, el tráfico no debería poder enviarse del cliente al servidor ni del servidor al cliente.

Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado fw3.sh para que reinicie los contadores de iptables.

Explica en la memoria:

Las reglas en las tablas nat y filter que se han cumplido y el número de veces.

```
pc4:~# nc -u -p 6666 10.22.7.10 7777
Probando conexion servidor/cliente con red privada desde DMZ

pc1:~# nc -u -l -p 7777
```

```
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   *      10.7.22.0/24         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0   *      10.8.22.0/24         0.0.0.0/0
3      0      0 DROP      all  --  eth1   *      0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 ACCEPT    all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0
3      0      0 ACCEPT    udp  --  *      *      60.0.0.0/24         50.0.0.40
4      0      0 ACCEPT    udp  --  *      *      70.0.0.0/24         50.0.0.40
5      0      0 ACCEPT    udp  --  *      *      50.0.0.40            60.0.0.0/24
6      0      0 ACCEPT    udp  --  *      *      50.0.0.40            70.0.0.0/24
7      0      0 ACCEPT    udp  --  *      *      70.0.0.0/24         50.0.0.50
8      0      0 ACCEPT    tcp  --  *      *      70.0.0.0/24         50.0.0.50
9      0      0 ACCEPT    tcp  --  *      *      60.0.0.0/24         50.0.0.50
10     0      0 ACCEPT    udp  --  *      *      60.0.0.0/24         50.0.0.50
11     0      0 ACCEPT    udp  --  *      *      50.0.0.50            70.0.0.0/24
12     0      0 ACCEPT    tcp  --  *      *      50.0.0.50            70.0.0.0/24
13     0      0 ACCEPT    tcp  --  *      *      50.0.0.50            60.0.0.0/24
14     0      0 ACCEPT    udp  --  *      *      50.0.0.50            60.0.0.0/24
15     0      0 ACCEPT    tcp  --  *      *      10.7.22.10           50.0.0.50
16     0      0 ACCEPT    tcp  --  *      *      50.0.0.50            10.7.22.10
17     0      0 ACCEPT    tcp  --  *      *      10.7.22.10           50.0.0.40
18     0      0 ACCEPT    tcp  --  *      *      50.0.0.40            10.7.22.10
19     0      0 DROP      all  --  *      *      50.0.0.0/24         10.7.22.0/24
20     0      0 DROP      all  --  *      *      50.0.0.0/24         10.8.22.0/24
21     0      0 DROP      all  --  *      *      50.0.0.0/24         60.0.0.0/24
22     0      0 DROP      all  --  *      *      50.0.0.0/24         70.0.0.0/24

Chain OUTPUT (policy ACCEPT 1 packets, 117 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

```
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 89 bytes)
num  pkts bytes target    prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24         0.0.0.0/0

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source               destination
firewall:~#
```

b) Si se arranca una aplicación servidor (TCP o UDP) en el firewall y se arranca una aplicación cliente para que se comunice con ese servidor en una de las máquinas de DMZ, el tráfico no debería poder enviarse del cliente al servidor ni del servidor al cliente.

Haz una prueba para este tipo de tráfico y explica qué prueba estás haciendo.

Asegúrate de que antes de lanzar el cliente has ejecutado fw3.sh para que reinicie los contadores de iptables.

Explica en la memoria:

Las reglas en las tablas nat y filter que se han cumplido y el número de veces.

las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el número de veces que se ejecutan.

```
pc4
pc4:~# nc -u -p 6666 50.0.0.1 7777
Probando servidor del firewall

firewall
firewall:~# nc -u -l -p 7777
```

```
firewall
firewall:~# iptables -t nat -L -v --line-numbers -n
Chain PREROUTING (policy ACCEPT 1 packets, 59 bytes)
num  pkts bytes target    prot opt in     out     source            destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
1      0      0 SNAT      all  --  *      eth2    10.7.22.0/24      0.0.0.0/0          to:40.0.0.1
2      0      0 SNAT      all  --  *      eth2    10.7.22.0/24      0.0.0.0/0          to:40.0.0.1
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
firewall:~#
```

```
firewall
firewall:~# iptables -t filter -L -v --line-numbers -n
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
1      0      0 ACCEPT    all  --  eth0    *      10.7.22.0/24      0.0.0.0/0
2      0      0 ACCEPT    all  --  eth0    *      10.8.22.0/24      0.0.0.0/0
3      1      59 DROP      all  --  eth1    *      0.0.0.0/0         0.0.0.0/0
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
1      0      0 ACCEPT    all  --  eth0    eth2    0.0.0.0/0         0.0.0.0/0
2      0      0 ACCEPT    all  --  eth2    eth0    0.0.0.0/0         0.0.0.0/0          state RELATED,ESTABLISHED
3      0      0 ACCEPT    udp  --  *      *      60.0.0.0/24       50.0.0.40          udp dpt:7
4      0      0 ACCEPT    udp  --  *      *      70.0.0.0/24       50.0.0.40          udp dpt:7
5      0      0 ACCEPT    udp  --  *      *      50.0.0.40         60.0.0.0/24       udp spt:7
6      0      0 ACCEPT    udp  --  *      *      50.0.0.40         70.0.0.0/24       udp spt:7
7      0      0 ACCEPT    udp  --  *      *      70.0.0.0/24       50.0.0.50          udp dpt:13
8      0      0 ACCEPT    tcp  --  *      *      70.0.0.0/24       50.0.0.50          tcp dpt:13
9      0      0 ACCEPT    tcp  --  *      *      60.0.0.0/24       50.0.0.50          tcp dpt:13
10     0      0 ACCEPT    udp  --  *      *      60.0.0.0/24       50.0.0.50          udp dpt:13
11     0      0 ACCEPT    udp  --  *      *      50.0.0.50         70.0.0.0/24       udp spt:13
12     0      0 ACCEPT    tcp  --  *      *      50.0.0.50         70.0.0.0/24       tcp spt:13
13     0      0 ACCEPT    tcp  --  *      *      50.0.0.50         60.0.0.0/24       tcp spt:13
14     0      0 ACCEPT    udp  --  *      *      50.0.0.50         60.0.0.0/24       udp spt:13
15     0      0 ACCEPT    tcp  --  *      *      10.7.22.10        50.0.0.50          tcp dpt:23
16     0      0 ACCEPT    tcp  --  *      *      50.0.0.50         10.7.22.10         tcp spt:23
17     0      0 ACCEPT    tcp  --  *      *      10.7.22.10        50.0.0.40          tcp dpt:7
18     0      0 ACCEPT    tcp  --  *      *      50.0.0.40         10.7.22.10         tcp spt:7
19     0      0 DROP      all  --  *      *      50.0.0.0/24       10.7.22.0/24
20     0      0 DROP      all  --  *      *      50.0.0.0/24       10.8.22.0/24
21     0      0 DROP      all  --  *      *      50.0.0.0/24       60.0.0.0/24
22     0      0 DROP      all  --  *      *      50.0.0.0/24       70.0.0.0/24
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out     source            destination
firewall:~#
```