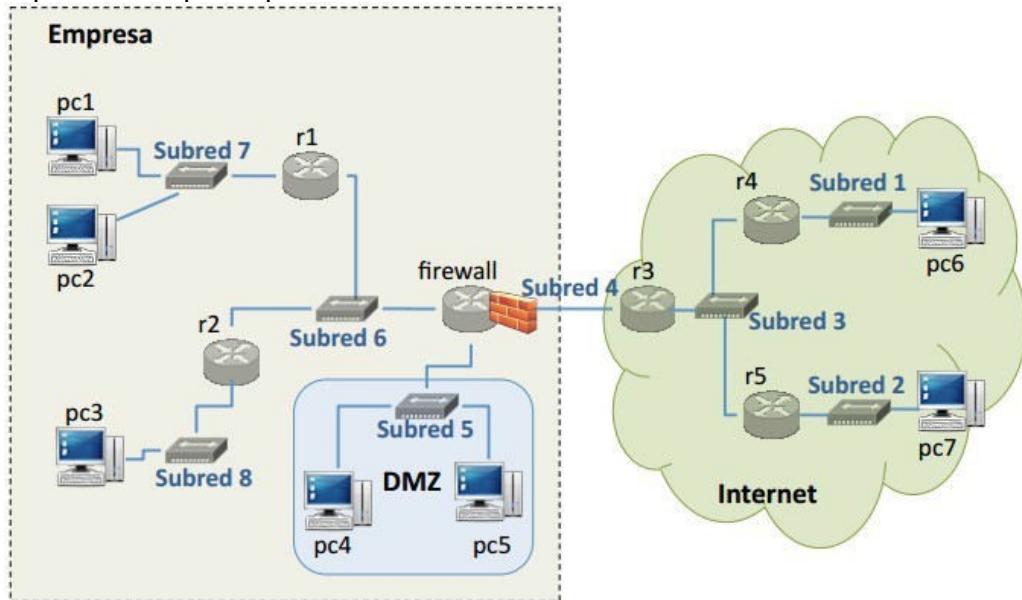


## Planificación y Gestión de Redes de Ordenadores Practica 2: Red frontera – (firewalls)

### 1. Escenario para la configuración de un firewall

En la figura 1 se muestra una red empresarial (pc1, pc2, pc3, pc4, pc5, r1, r2 y firewall) formada por una zona con direccionamiento privado y una zona DMZ con direccionamiento publico. La empresa se conecta a Internet a través de un router firewall que sera necesario configurarlo atendiendo a los requisitos impuestos por la empresa.

Escenario proporcionado por la practica.



Escenario montado en NetGui.

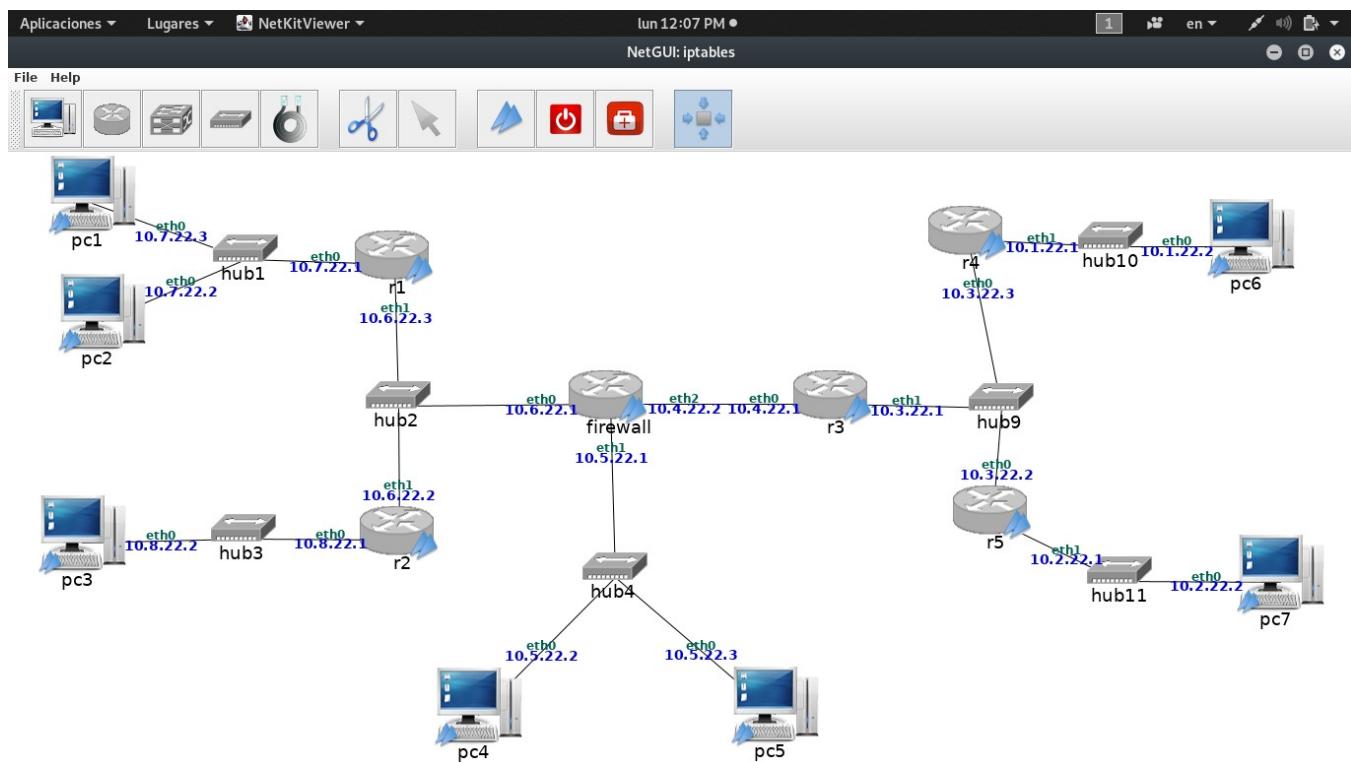


Figura 1: Escenario de red para los ejercicios de configuración de firewall

Deberás primeramente asignar direcciones IP a las maquinas que aparecen en la figura, utilizando las subredes que tenías asignadas en la práctica 1: subred 1, subred 2, subred 3, subred 4 y subred 5. Para las subredes de ámbito privado, calcula las direcciones IP teniendo en cuenta la siguiente fórmula:

- Subred 6: 10.6.22.0/24
  - Subred 7: 10.7.22.0/24
  - Subred 8: 10.8.22.0/24

La empresa tiene las maquinas pc4 y pc5 que se encuentran en una subred publica: subred5. Estas maquinas proporcionan servicios basicos de la empresa, como por ejemplo un servidor de fecha y hora. A este tipo de configuración, donde la empresa tiene una o varias subredes publicas para ofrecer servicios a Internet se le denomina zona desmilitarizada o DMZ (DeMilitarized Zone).

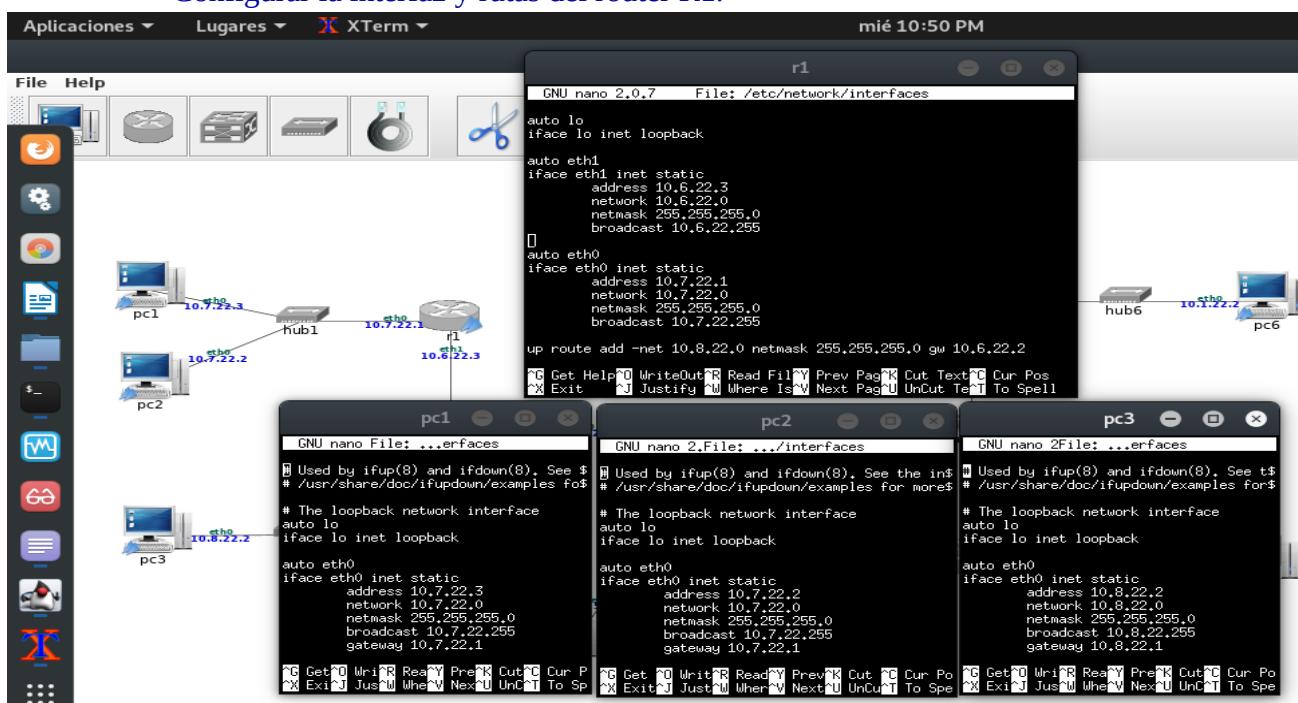
Todas las maquinas de la empresa se conectan a Internet a través de la maquina firewall. El firewall deberá aplicar reglas de traducción de direcciones para tráfico de las subredes privadas, no siendo necesaria la traducción de direcciones para el tráfico de la zona DMZ.

En este escenario, se considera que Internet esta formado por las siguientes maquinas: r3, r4, r5, pc6 y pc7.

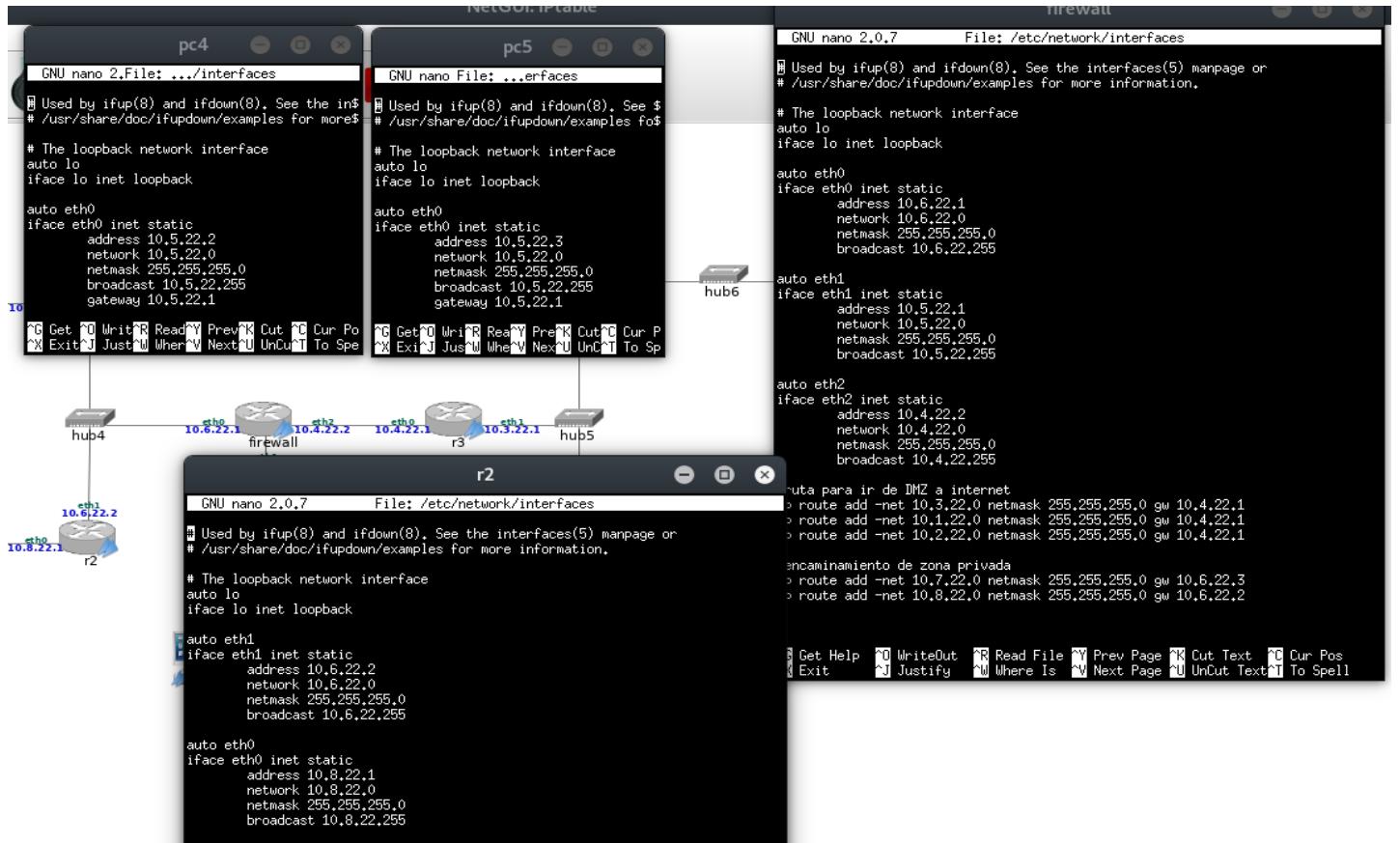
Arranca de una en una todas las maquinas de la figura.

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

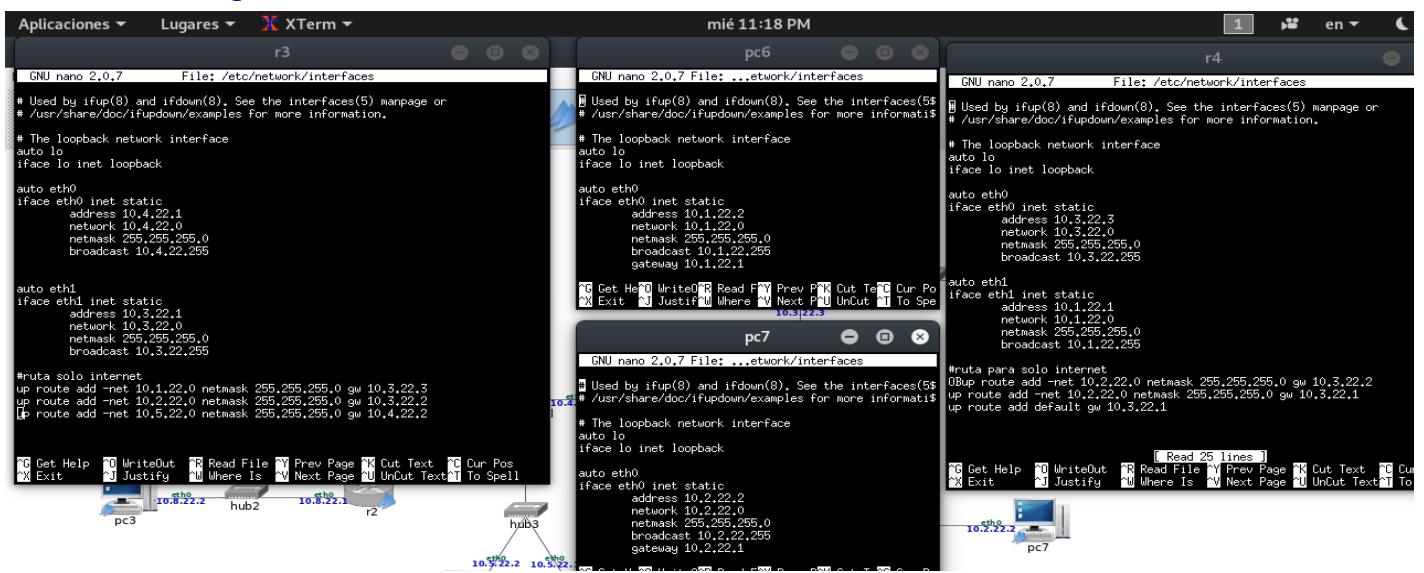
- Configuración de las interfaces y el encaminamiento de los dispositivos del escenario.
  - Configurar la interfaz de la PC1.
  - Configurar la interfaz de la PC2.
  - Configurar la interfaz de la PC3.
  - Configurar la interfaz y rutas del router R1.



- Configurar la interfaz y rutas del router R2.
- Configurar la interfaz y rutas del router firewall.
- Configurar la interfaz de PC4.
- Configurar la interfaz de PC5.



- Configurar la interfaz y rutas del router R3.
- Configurar la interfaz de PC6.
- Configurar la interfaz y rutas del router R4.
- Configurar la interfaz de PC7.



- Configurar la interfaz y rutas del router R5.

Aplicaciones ▾ Lugares ▾ XTerm ▾ mié 11:20 NetGUI: IP

File r5

GNU nano 2.0.7 File: /etc/network/interfaces

```
# Used by ifup(8) and ifdown(8). See the interfaces(5) manpage or
# /usr/share/doc/ifupdown/examples for more information.

# The loopback network interface
auto lo
iface lo inet loopback

auto eth0
iface eth0 inet static
    address 10.3.22.2
    network 10.3.22.0
    netmask 255.255.255.0
    broadcast 10.3.22.255

auto eth1
iface eth1 inet static
    address 10.2.22.1
    network 10.2.22.0
    netmask 255.255.255.0
    broadcast 10.2.22.255

#ruta para solo internet
up route add -net 10.1.22.0 netmask 255.255.255.0 gw 10.2.22.3
up route add -net 10.1.22.0 netmask 255.255.255.0 gw 10.3.22.1
#p route add default gw 10.3.22.1
```

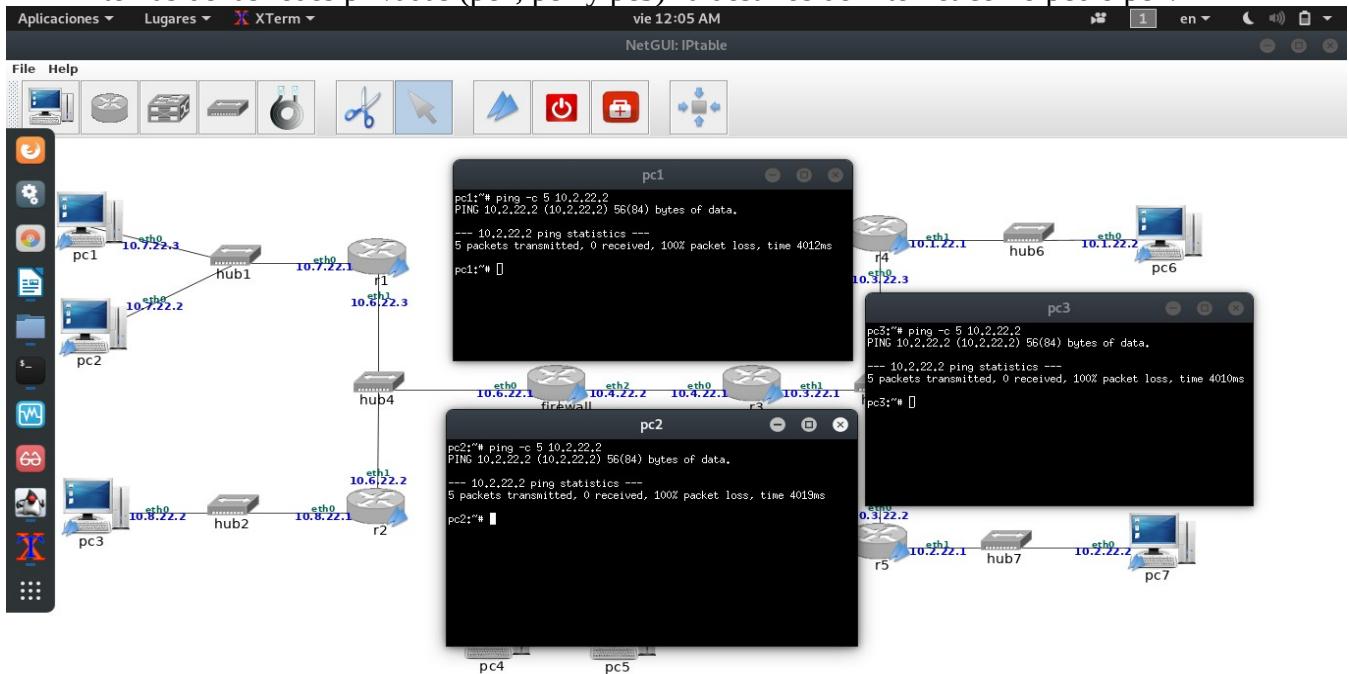
Get Help ⌘G WriteOut ⌘O Read File ⌘R Prev Page ⌘Y Cut Text ⌘C Cur Pos  
Exit ⌘X Justify ⌘J Where Is ⌘W Next Page ⌘V UnCut Text ⌘U To Spell ⌘T

Incluye en la memoria una imagen del escenario de NetGUI donde se muestren las direcciones IP que has configurado y los nombres de las interfaces (eth0, eth1, etc).

En esta practica se configurara la maquina firewall para que actué como traductor de direcciones y como cortafuegos. Habrá que definir varias reglas utilizando iptables. Por este motivo, es recomendable guardar dichas reglas en un fichero script de shell. Para esta practica se hará uso de la herramienta nc que permite arrancar aplicaciones TCP y UDP en modo cliente/servidor. Consulta el anexo de la secciona 4 para ver como se utilizan.

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

2.1. Cliente en la red privada, servidor externo Comprueba que no funciona un ping desde las maquinas internas de las redes privadas (pc1, pc2 y pc3) a destinos de Internet como pc6 o pc7.



Se puede ver que no hay conectividad de la red privada a la red publica. Las tres PC de la red interna tratan de enviar dos paquetes icmp a las computadoras en la red publica.

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 trafico saliente de las redes privadas (SNAT) y en su correspondiente trafico de respuesta. Explica para que subredes has tenido que realizar la configuración de SNAT. Incluye el script fw1.sh en la memoria y expícalo.

```
#script fw1.sh
#borar la tabla filter
iptables -t filter -F
iptables -t filter -Z
#Reglas para aplicar nateo en firewall

iptables -t nat -A POSTROUTING -s 10.6.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2
iptables -t nat -A POSTROUTING -s 10.7.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2
iptables -t nat -A POSTROUTING -s 10.8.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2
```

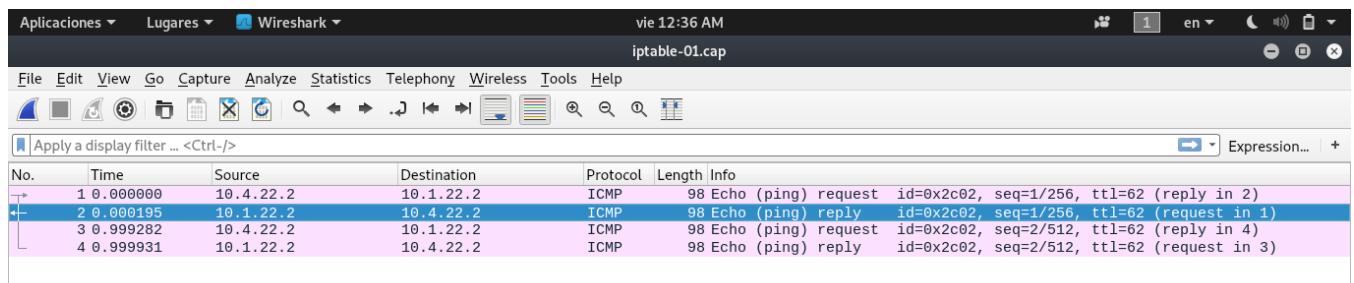
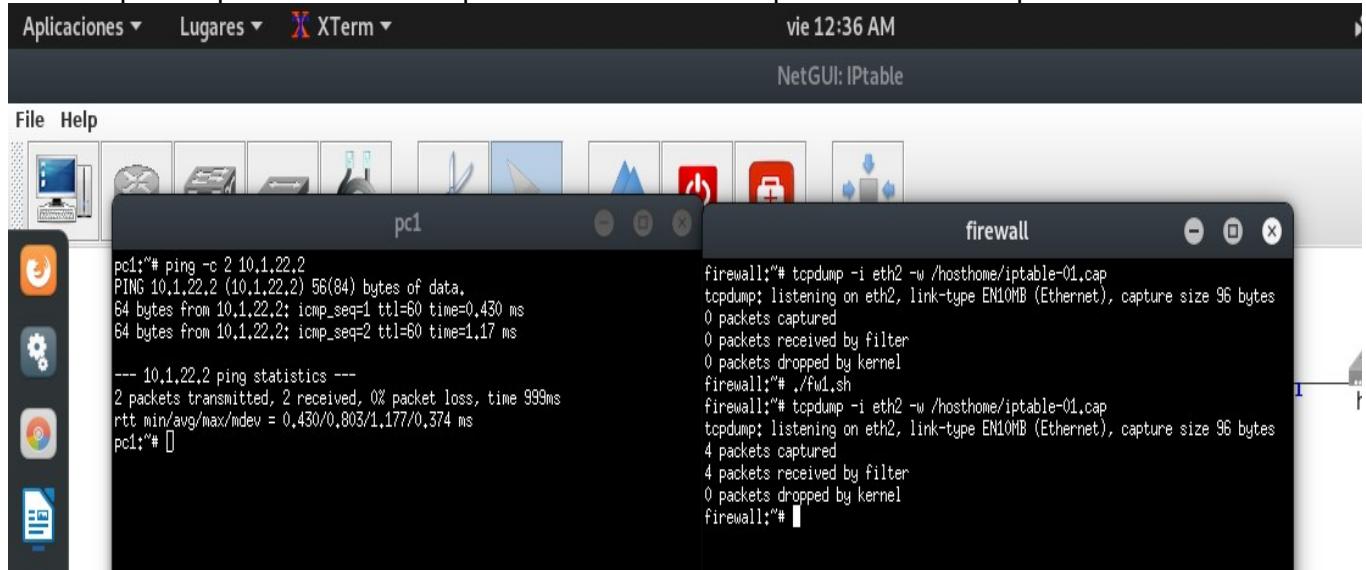
La configuración de SNAT en el router Firewall se hizo para cambiar las ip de las subredes 10.6.22.0/24, 10.7.22.0/24 y 10.8.22.0/24 por la ip de la interfaz eth2 del Firewall 10.4.22.2. La explicación de lo que hice en el script: -t [Table], -A [Append] (Agrega una regla a una cadena), -s[Source] (IP origen), -o [out-interface] (Interfaz de destino), -j [Acción] –to-source [con el nuevo origen, la IP de salida].

### 2.1.1. ICMP

Ejecuta el script fw1.sh de 2.1.

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

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



Se puede observar que efectivamente se puede hacer ping desde la PC1 hasta la PC6, en la captura de trafico que se hizo en la interfaz eth2 del router Firewall se puede ver que el SNAT esta haciendo su trabajo, la ip de la PC1 al salir del router Firewall se esta reemplazando por la ip que se indico en el script (10.4.22.2).

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

```
firewall:~# iptables -t nat -L -v -n
```

```
firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 4 packets, 336 bytes)
pkts bytes target    prot opt in     out    source          destination
      0   0 SNAT      all  --  eth2    10.6.22.0/24  0.0.0.0/0        to:10.4.22.2
      2  168 SNAT     all  --  eth2    10.7.22.0/24  0.0.0.0/0        to:10.4.22.2
      0   0 SNAT      all  --  eth2    10.8.22.0/24  0.0.0.0/0        to:10.4.22.2

Chain POSTROUTING (policy ACCEPT 6 packets, 408 bytes)
pkts bytes target    prot opt in     out    source          destination
      0   0 SNAT      all  --  eth2    10.6.22.0/24  0.0.0.0/0        to:10.4.22.2
      2  168 SNAT     all  --  eth2    10.7.22.0/24  0.0.0.0/0        to:10.4.22.2
      0   0 SNAT      all  --  eth2    10.8.22.0/24  0.0.0.0/0        to:10.4.22.2

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

- Se puede ver que aparece la cadena PREROUTING la cual ha sido configurada en el script. Las configuraciones SNAT, nos indica que han sido 12 paquetes enviados desde una de las redes configuradas en el SNAT. Los parámetros del comando: -L (lista de reglas), -v (información de conexiones), -n (devuelve direcciones IP y puertos correspondientes).
- Que reglas estan cumpliendo los paquetes ICMP echo request e ICMP echo response y cuantas veces se cumple/n. Indica que políticas por defecto se están cumpliendo de las cadenas de la tabla nat y cuantos paquetes las han cumplido.
- Las reglas que se están cumpliendo para los paquetes ICMP echo request e ICMP echo response a las subredes origen 10.6.22.0/24, 10.7.22.0/24, 10.8.22.0/24 se cambiaron las direcciones IPs por la IP 10.4.22.2, se esta cumpliendo una veces. La política que se esta cumpliendo por defecto es ACCEPT(acepta paquetes) y fueron 12 paquetes los que la ha cumplido.

3. Consulta la información de seguimiento de conexiones del modulo ip\_conntrack del firewall y explica el resultado.

→ Para consultar esta información, antes tengo que instalar el software. Debido a que el escenario esta montado en NETGUI no pude obtener resultado.

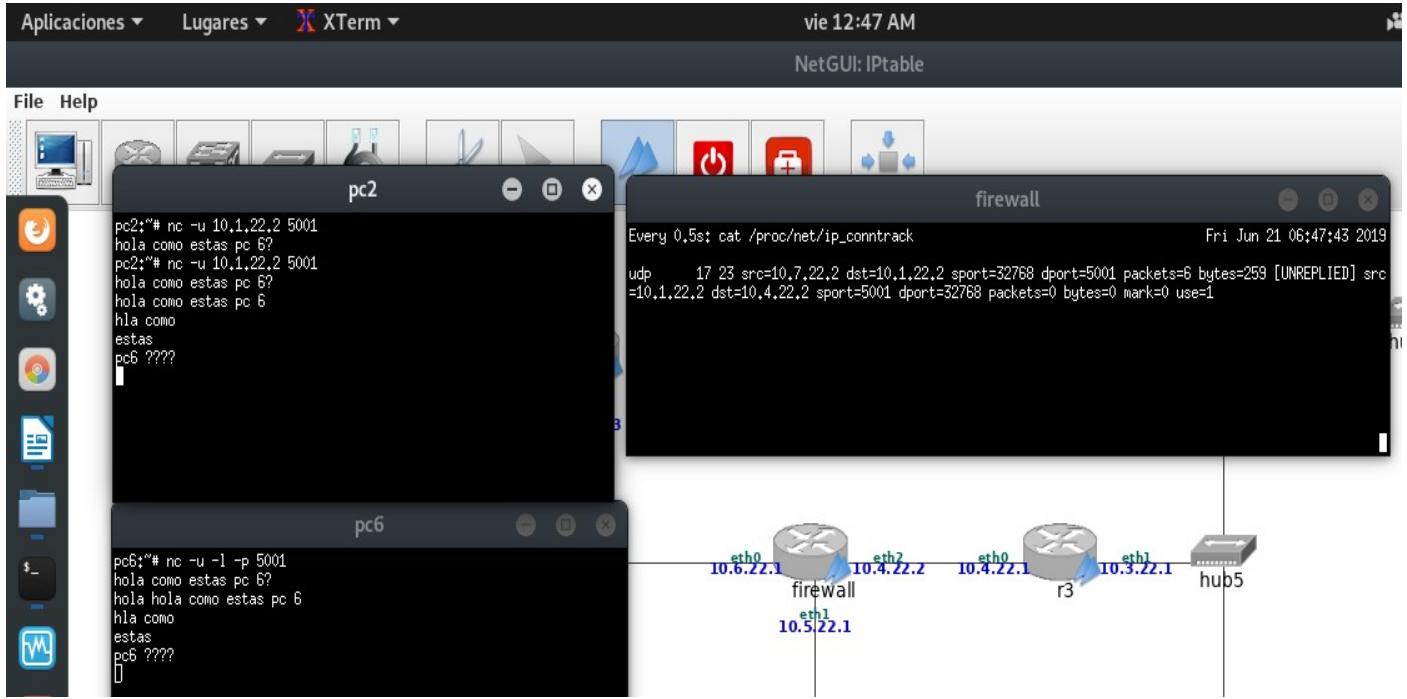
#### 2.1.1. UDP

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

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 con el comando:

```
firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack.
```

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. Interrumpe la captura y las ejecuciones de nc, explica la captura y como esta se relaciona con la información que has visto en ip\_conntrack.



Se puede ver en el estado de ip\_conntrack que los paquetes UDP origen de la PC2 (10.7.22.2) con destino 10.1.22.2, cuando los paquetes salen del router Firewall, la IP cambia por 10.4.22.2 y eso se puede notar en la respuesta del server UDP PC6, la IP origen en el sentido contrario es 10.1.22.2 y la IP destino es la IP de la interfaz eth2 del Firewall 10.4.22.2. También están los puertos de origen y destino, la cantidad de paquetes que se mandan del server UDP y la cantidad de paquetes que manda el server UDP.

firewall

```
Every 0.5s: cat /proc/net/ip_conntrack
Fri Jun 21 06:53
udp      17 10 src=10.7.22.2 dst=10.1.22.2 sport=32768 dport=5001 packets=6 bytes=247 [UNREPL
=10.1.22.2 dst=10.4.22.2 sport=5001 dport=32768 packets=0 bytes=0 mark=0 use=1
```

r3:~# tcpdump -i eth0 -w /hosthome/iptables-02.cap
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
8 packets captured
8 packets received by filter
0 packets dropped by kernel
r3:~#

Aplicaciones ▾ Lugares ▾ Wireshark ▾ vie 12:54 AM
iptables-02.cap

| No. | Time      | Source            | Destination       | Protocol | Length | Info                              |
|-----|-----------|-------------------|-------------------|----------|--------|-----------------------------------|
| 1   | 0.000000  | 10.4.22.2         | 10.1.22.2         | UDP      | 63     | 32768 → 5001 Len=21               |
| 2   | 4.977683  | 2a:4d:61:4e:27:ac | 12:db:da:c1:cb:1b | ARP      | 42     | Who has 10.4.22.1? Tell 10.4.22.2 |
| 3   | 4.977716  | 12:db:da:c1:cb:1b | 2a:4d:61:4e:27:ac | ARP      | 42     | 10.4.22.1 is at 12:db:da:c1:cb:1b |
| 4   | 10.777677 | 10.4.22.2         | 10.1.22.2         | UDP      | 64     | 32768 → 5001 Len=22               |
| 5   | 14.969655 | 10.4.22.2         | 10.1.22.2         | UDP      | 59     | 32768 → 5001 Len=17               |
| 6   | 16.631219 | 10.4.22.2         | 10.1.22.2         | UDP      | 48     | 32768 → 5001 Len=6                |
| 7   | 18.281454 | 10.4.22.2         | 10.1.22.2         | UDP      | 48     | 32768 → 5001 Len=6                |
| 8   | 25.279467 | 10.4.22.2         | 10.1.22.2         | UDP      | 49     | 32768 → 5001 Len=7                |

Frame 4: 64 bytes on wire (512 bits), 64 bytes captured (512 bits)
Ethernet II, Src: 2a:4d:61:4e:27:ac (2a:4d:61:4e:27:ac), Dst: 12:db:da:c1:cb:1b (12:db:da:c1:cb:1b)
Internet Protocol Version 4, Src: 10.4.22.2, Dst: 10.1.22.2
User Datagram Protocol, Src Port: 32768, Dst Port: 5001
Data (22 bytes)

|      |   |                 |
|------|---|-----------------|
| 0000 | 12 db da c1 cb 1b 2a 4d 61 4e 27 ac 08 00 45 00 | .....*M aN'...E |
| 0010 | 00 32 2e 42 40 00 3e 11 ce 70 0a 04 16 02 0a 01 | .2.B@>..p.....  |

iptable-02.cap

Packets: 8 · Displayed: 8 (100.0%)

Profile: Default

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

Aplicaciones ▾ Lugares ▾ XTerm ▾ vie 12:56  
NetGUI: IP

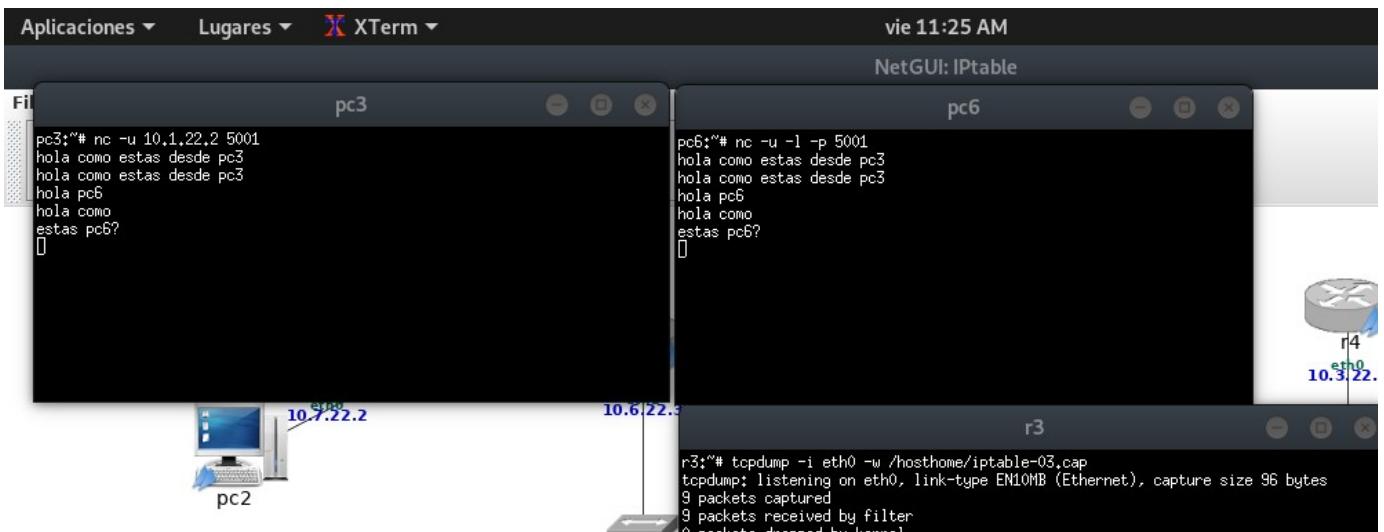
```
File Help
firewall
Chain PREROUTING (policy ACCEPT 7 packets, 484 bytes)
pkts bytes target prot opt in out source destination
Chain POSTROUTING (policy ACCEPT 6 packets, 408 bytes)
pkts bytes target prot opt in out source destination
  0    0 SNAT    all -- any eth2  10.6.22.0/24 anywhere      to:10.4.22.2
  5  316 SNAT    all -- any eth2  10.7.22.0/24 anywhere      to:10.4.22.2
  0    0 SNAT    all -- any eth2  10.8.22.0/24 anywhere      to:10.4.22.2
Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
pkts bytes target prot opt in out source destination
firewall:~#
```

**La regla SNAT se está cumpliendo para la subred 10.7.22.0/24. Se envió un paquete de 316 bytes.**

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

**Cuando se envían los mensajes de PC3 a PC6 se ve la IP origen (10.8.22.2) y esta se cambia cuando sale del Firewall (10.4.22.2).**

**Cuando el server PC6 manda el mensaje, este lo hace a la IP del Firewall y esto pasa por la restricción declarada en el nat.**



Aplicaciones ▾ Lugares ▾ XTerm ▾ vie 11:27 AM

```
File Help
firewall
Chain PREROUTING (policy ACCEPT 9 packets, 592 bytes)
pkts bytes target prot opt in out source destination
Chain POSTROUTING (policy ACCEPT 7 packets, 476 bytes)
pkts bytes target prot opt in out source destination
  0    0 SNAT    all -- any eth2  10.6.22.0/24 anywhere      to:10.4.22.2
  5  316 SNAT    all -- any eth2  10.7.22.0/24 anywhere      to:10.4.22.2
  2  108 SNAT    all -- any eth2  10.8.22.0/24 anywhere      to:10.4.22.2
Chain OUTPUT (policy ACCEPT 5 packets, 308 bytes)
pkts bytes target prot opt in out source destination
firewall:~#
```

**La política por defecto que se esta cumpliendo es ACCEPT en la cadena POSTROUTING de la tabla nat y la esta cumpliendo dos paquetes.**

4. Primero inicia una captura en r3 (iptables-03.cap) para capturar todo el tráfico que atravesese este router e inicia otra captura en r1-eth0 (iptables-04.cap). Ejecuta una aplicación servidor UDP escuchando en el puerto 7777 en pc7 con el comando nc.

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:

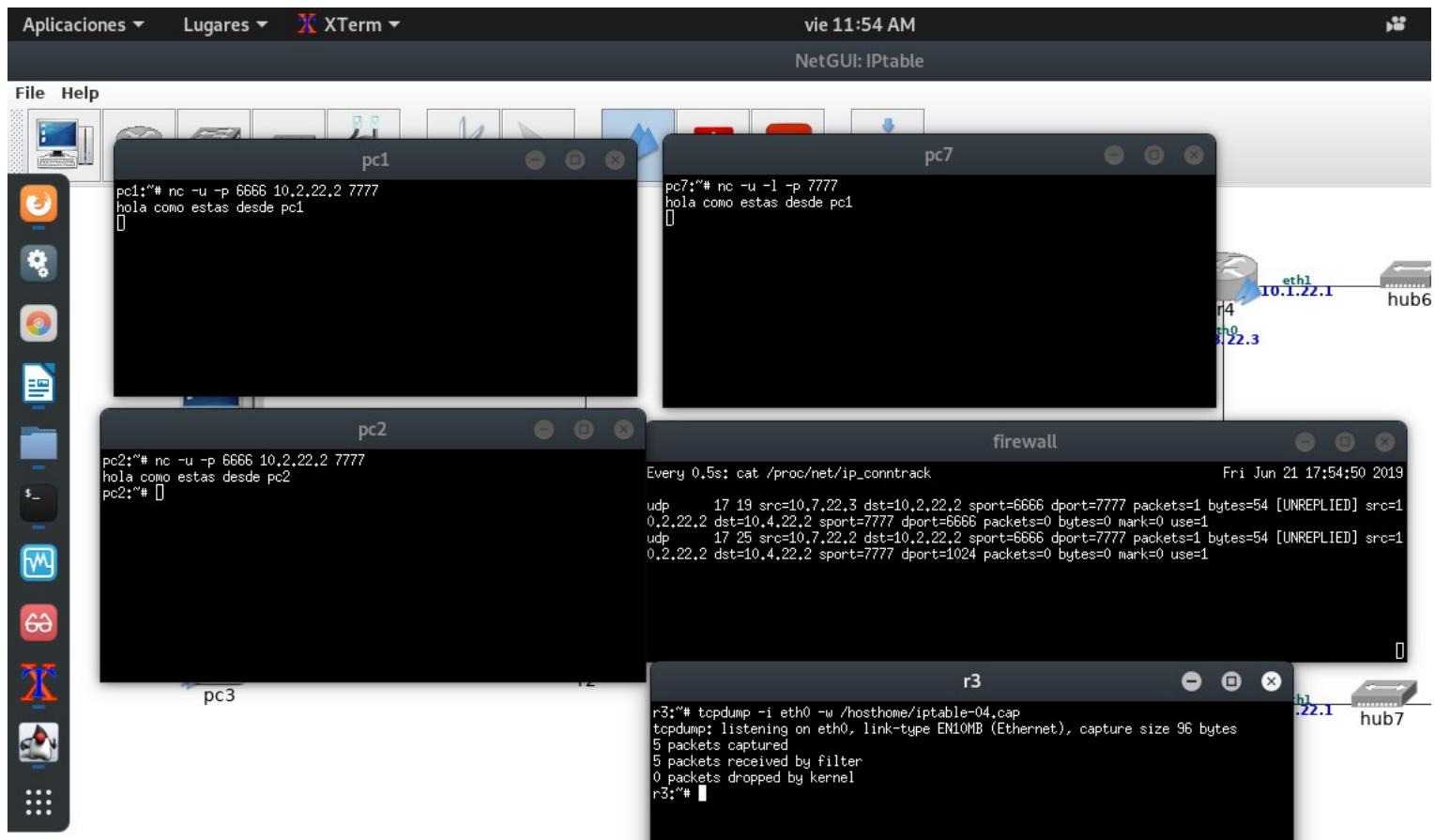
```
pc1:~# nc -u -p 6666 <dirIP_de_pc7> 7777
```

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:

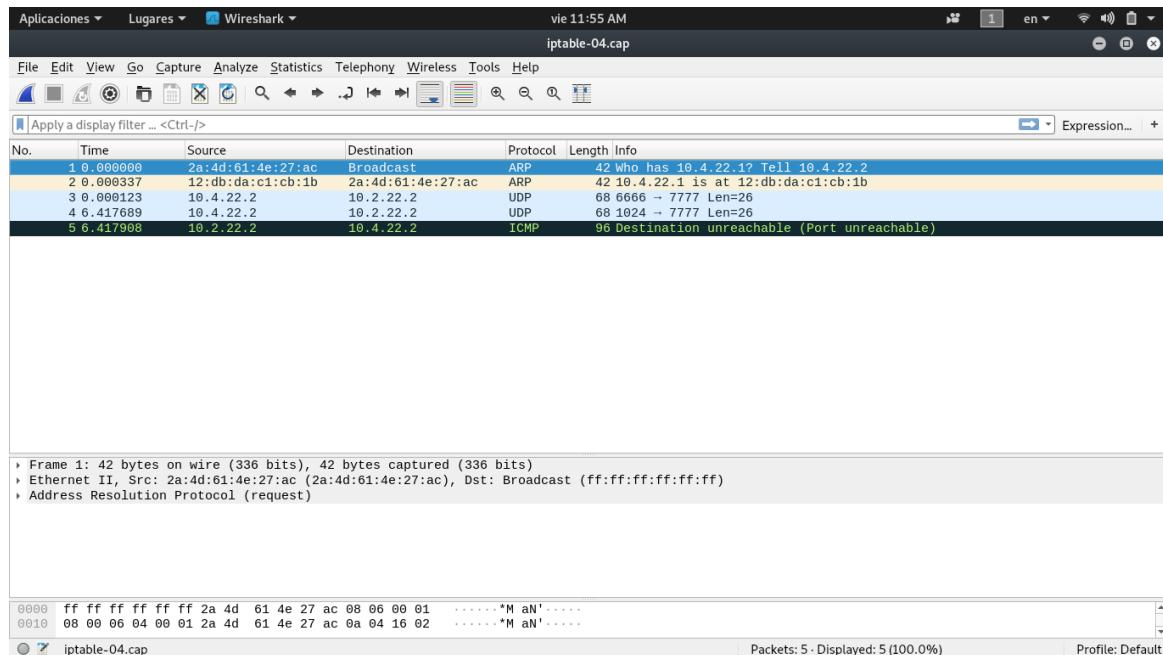
```
pc2:~# nc -u -p 6666 <dirIP_de_pc7> 7777
```

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

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 que ocurre con la traducción de direcciones y puertos.

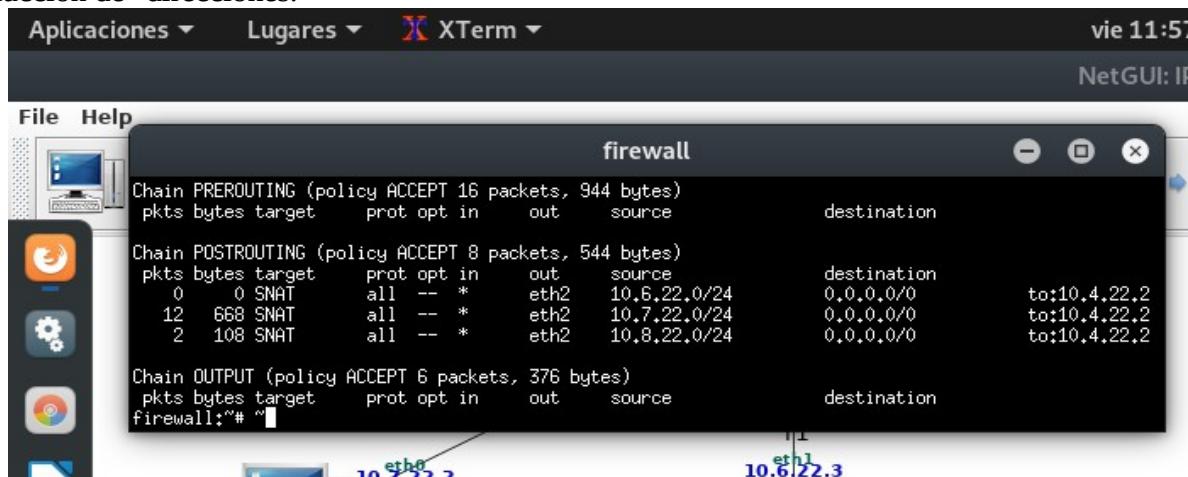


**Se mando un mensaje de PC1 y PC2 al server en PC7. En el ip\_conntrack se puede ver que la IP origen(10.7.22.3) y (10.7.22.2), ambas han mandado un paquete al server(10.2.22.2). A la PC1 y PC2 se les cambia la IP al salir del router Firewall, se puede ver que la PC2 cambia el puerto a 1024.**



**El mensaje de PC2 no llega al server en PC7, eso pasa porque el puerto no se puede conectar dos host al mismo puerto.**

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



**En la tabla nat del router Firewall se ha cumplido 2 veces la regla de traducción de PREROUTING snat.**

### 2.1.1. TCP

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

- 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.

Ejecuta una aplicación servidor TCP escuchando en el puerto 7777 en pc6 con el comando nc. Y ejecuta en pc1 una aplicación cliente TCP que se comunique con el servidor anterior. No introduzcas nada por la entrada estándar, ni en pc1 ni en pc6.

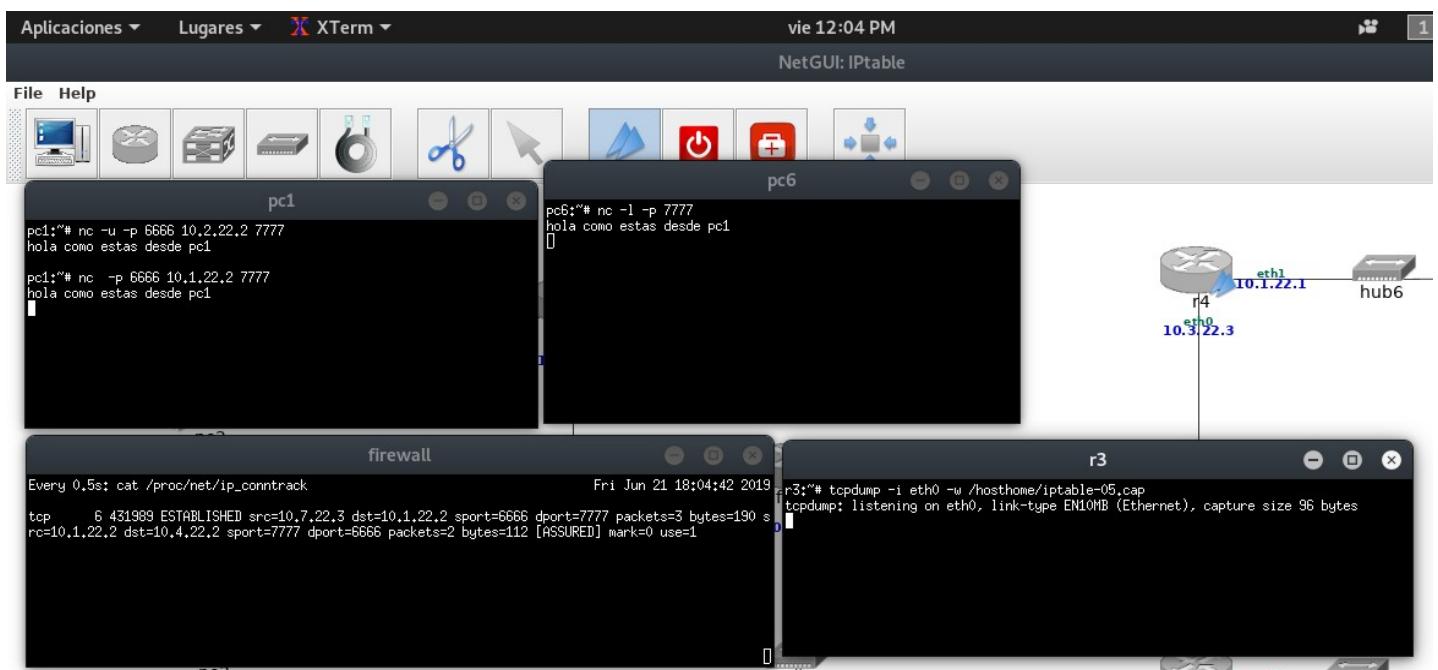
Simultáneamente consulta ip\_conntrack del firewall cada medio segundo. Explica el número de paquetes que se han observado en cada sentido, razonando la respuesta.

```
firewall
Every 0.5s: cat /proc/net/ip_conntrack      Fri Jun 21 17:54:50 2019

udp      17 19 src=10.7.22.3 dst=10.2.22.2 sport=6666 dport=7777 packets=1 bytes=54 [UNREPLIED] src=1
0.2.22.2 dst=10.4.22.2 sport=7777 dport=6666 packets=0 bytes=0 mark=0 use=1
udp      17 25 src=10.7.22.2 dst=10.2.22.2 sport=6666 dport=7777 packets=1 bytes=54 [UNREPLIED] src=1
0.2.22.2 dst=10.4.22.2 sport=7777 dport=1024 packets=0 bytes=0 mark=0 use=1
```

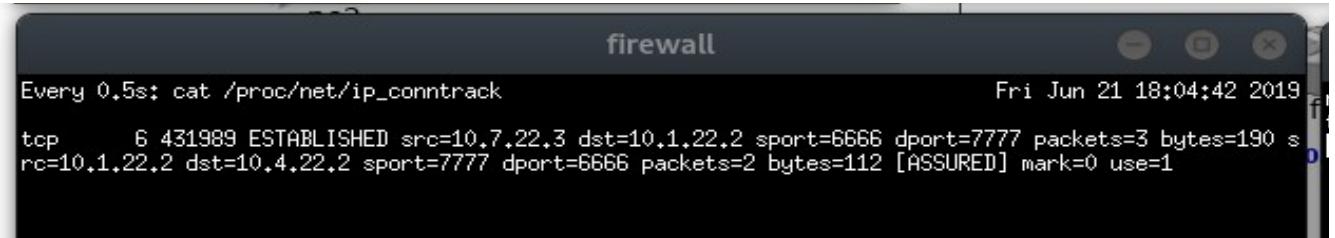
Se puede ver en el ip\_conntrack el estado de la conexión TCP ESTABLISHED, esto es sin haber metido la palabra desde el cliente. Se ve que se han enviado 2 paquetes, eso pasa porque el protocolo TCP esta orientado a la conexión, por eso es que hay tres paquetes sin antes haber mandado mensajes desde. Esos mensajes son cuando se establece la conexión.

- Introduce una palabra en la entrada estándar de pc1, pulsa <Enter> y explica razonadamente lo que observas en ip\_conntrack.



Se puede ver que al enviar la palabra desde PC1, el numero de paquetes aumenta a 3 desde ip\_conntrack.

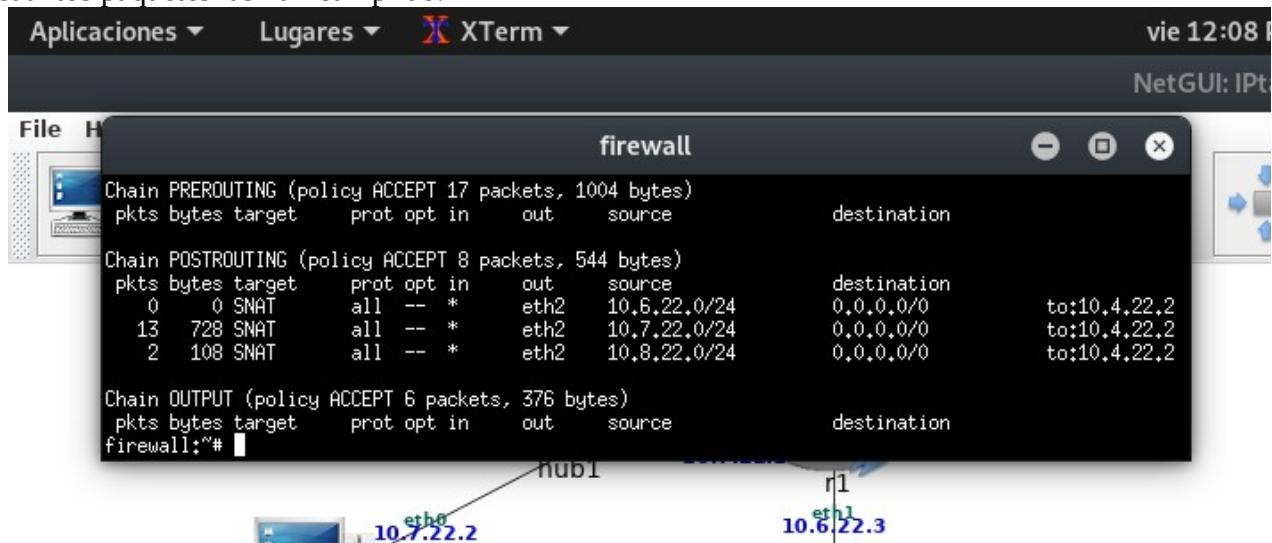
3. Realiza un Ctrl+C en el terminal de pc1 para interrumpir la ejecución de nc. Interrumpe la captura en r3 y contrasta lo que observas en la captura con lo que muestra ip\_conntrack.



```
firewall
Every 0.5s: cat /proc/net/ip_conntrack
Fri Jun 21 18:04:42 2019
tcp      6 431989 ESTABLISHED src=10.7.22.3 dst=10.1.22.2 sport=6666 dport=7777 packets=3 bytes=190 s
rc=10.1.22.2 dst=10.4.22.2 sport=7777 dport=6666 packets=2 bytes=112 [ASSURED] mark=0 use=1
```

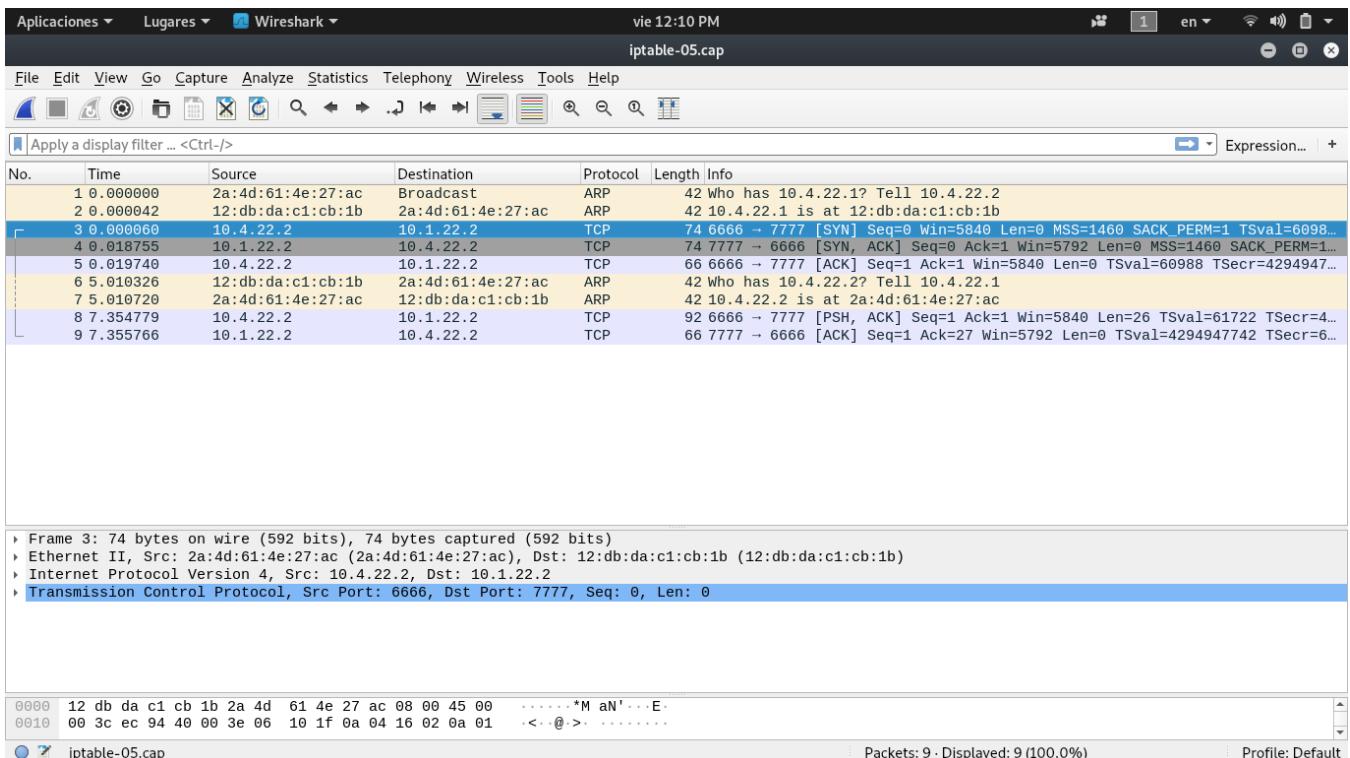
**Cuando interrumpo la ejecución nc en PC1, la conexión pasa a ser de ESTABLISHED a TIME\_WAIT.**

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



```
firewall
Chain PREROUTING (policy ACCEPT 17 packets, 1004 bytes)
pkts bytes target     prot opt in     out     source               destination
Chain POSTROUTING (policy ACCEPT 8 packets, 544 bytes)
pkts bytes target     prot opt in     out     source               destination
      0    0 SNAT       all   --  *      eth2    10.6.22.0/24      0.0.0.0/0          to:10.4.22.2
     13   728 SNAT      all   --  *      eth2    10.7.22.0/24      0.0.0.0/0          to:10.4.22.2
      2   108 SNAT      all   --  *      eth2    10.8.22.0/24      0.0.0.0/0          to:10.4.22.2
Chain OUTPUT (policy ACCEPT 6 packets, 376 bytes)
pkts bytes target     prot opt in     out     source               destination
firewall:~#
```

**Se ha cumplido una vez la regla de traducción de direcciones, se aplicó la política ACCEPT por defecto en la cadena PREROUTING para solo un paquete.**



**En esta captura de trafico que se hizo en el router R3 se puede ver hay mensajes que se originan de la IP 10.4.22.2 y se dirigen al servidor 10.1.22.2 mediante una conexión TCP.**

## 2.2. Servidor en la red privada, cliente externo

Aunque en una red como la que aparece en la figura, lo habitual es colocar los servidores accesibles desde el exterior en la zona DMZ, para ver como funciona DNAT, vamos permitir que haya servidores accesibles desde el exterior en la red privada interna.

### 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: → Creación del script, permisos y ejecución.

El trafico de entrada al firewall destinado al puerto UDP 5001 debe ser redirigido a pc1, puerto 5001.  
El trafico de entrada al firewall destinado al puerto UDP 5002 debe ser redirigido a pc2, puerto 5001.

1. Explica el nuevo script.

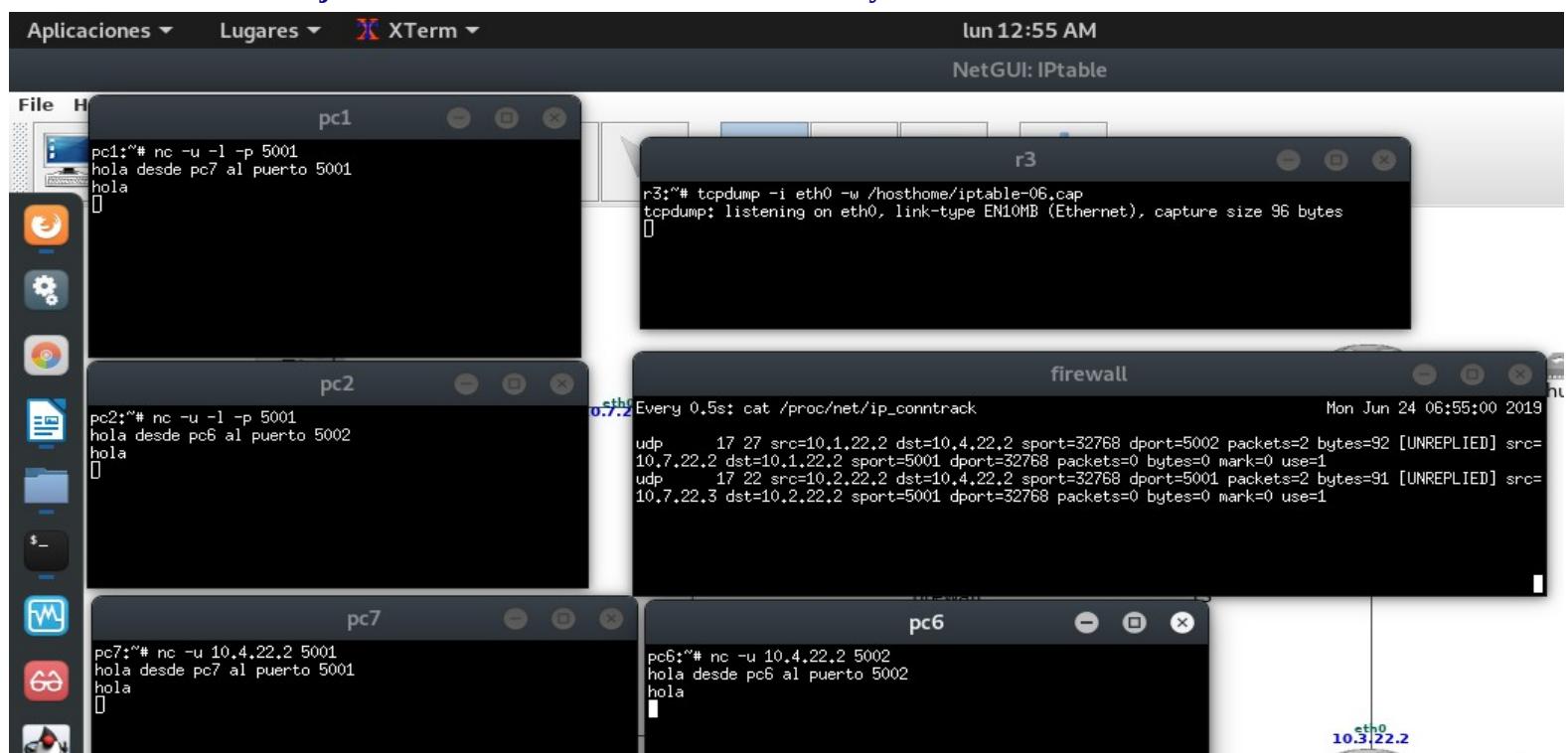
```
#script fw2.sh
#borrar la tabla filter
iptables -t filter -F
iptables -t filter -Z
#se esta aplicando nuevastraducciones DNAT
iptables -t nat -A PREROUTING -p udp -i eth2 --dport 5001 -j DNAT --to-destination 10.7.22.3:5001
iptables -t nat -A PREROUTING -p udp -i eth2 --dport 5002 -j DNAT --to-destination 10.7.22.2:5001
```

- Se borran las reglas configuradas con anterioridad y se reinician a cero los contadores de las cadenas.
- El trafico de entrada al Firewall por la interfaz eth2 destinado al puerto 5001 se redireccionara a la PC1 con puerto 5001.
- El trafico de entrada al Firewall por la interfaz eth2 destinado al puerto 5002 se redireccionara a la PC2 con puerto 5001.

2. Inicia una captura de trafico 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 trafico generado en pc6 lo reciba pc1 y el trafico generado en pc2 lo reciba pc7. Explica como has arrancado los dos clientes nc en pc6 y pc7.

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

- Los server UDP escuchan en el mismo puerto.
- Los clientes UDP los he arrancado indicándole que todo los mensajes sean dirigido a la IP 10.4.22.2, esta es la IP borde del Firewall, estos mensajes se dirigen a los puertos 5001 y 5002.
- La captura de trafico se realizara en el router R3.
- Se lanza ip\_conntrack.
- Mensajes intercambiados entre los server UDP y los clientes UDP.



- Se puede observar las IP origen y su correspondiente destino para cada flujo.
- En el primer flujo la IP origen 10.2.22.2 y su IP destino 10.4.22.2 (esta IP esta en la interfaz frontera del router Firewall), esto va dirigido al puerto destino 5002, cuando llega al Firewall la IP destino (10.7.22.2) cambia junto con su puerto destino(5001).
- En el segundo flujo la IP origen 10.1.22.2 y su IP destino 10.4.22.2 (esta IP esta en la interfaz frontera del router Firewall), esto va dirigido al puerto destino 5001, cuando llega al Firewall la IP destino(10.7.22.3) cambia junto con su puerto destino(5001).
- El DNAT esta dando resultado.

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

```

pkts bytes target    prot opt in   out      source          destination
  1   58 INAT      udp  --  eth2  *       0.0.0.0/0      0.0.0.0/0      udp dpt:5001 to:10.7.22.3:5001
  1   58 INAT      udp  --  eth2  *       0.0.0.0/0      0.0.0.0/0      udp dpt:5002 to:10.7.22.2:5001
  0   0  DNAT      tcp  --  eth2  *       0.0.0.0/0      0.0.0.0/0      tcp dpt:80 to:10.8.22.2:80

Chain POSTROUTING (policy ACCEPT 2 packets, 116 bytes)
pkts bytes target    prot opt in   out      source          destination

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

```

The screenshot shows the NetGUI: IPtable interface with the 'firewall' table open. It lists several rules for network translation (NAT). There are three INAT rules (entries 1, 2, and 3) that map traffic from the local network (0.0.0.0/0) to external hosts (10.7.22.3:5001, 10.7.22.2:5001, and 10.8.22.2:80 respectively). A POSTROUTING chain is also defined, and an OUTPUT chain is present but empty. The interface includes icons for various applications like a browser, file manager, and terminal.

Puedo ver que las traducciones se están cumpliendo para dos paquetes, se cumple una política por defecto que es ACCEPT la cual no se cumple para ningún paquete, la cadena que se está cumpliendo al inicio es la PREROUTING.

5. Relaciona los resultados de la captura de tráfico con la información extraída de ip\_conntrack y la tabla nat del firewall.

| No. | Time      | Source            | Destination       | Protocol | Length | Info                              |
|-----|-----------|-------------------|-------------------|----------|--------|-----------------------------------|
| 1   | 0.000000  | c6:da:50:5d:05:6f | Broadcast         | ARP      | 42     | Who has 10.4.22.2? Tell 10.4.22.1 |
| 2   | 0.000344  | da:93:d4:ca:a6:26 | c6:da:50:5d:05:6f | ARP      | 42     | 10.4.22.2 is at da:93:d4:ca:a6:26 |
| 3   | 0.000371  | 10.1.22.2         | 10.4.22.2         | UDP      | 72     | 32768 - 5002 Len=39               |
| 4   | 14.437711 | 10.2.22.2         | 10.4.22.2         | UDP      | 72     | 32768 - 5001 Len=39               |
| 5   | 19.339000 | 10.2.22.2         | 10.4.22.2         | UDP      | 47     | 32768 - 5001 Len=5                |
| 6   | 23.453771 | 10.1.22.2         | 10.4.22.2         | UDP      | 48     | 32768 - 5002 Len=6                |

The screenshot shows a Wireshark capture session titled 'iptable-06.cap'. The packet list pane shows six ARP requests and their responses. The details and bytes panes provide a detailed view of each captured frame. The status bar at the bottom indicates 'Packets: 6 · Displayed: 6 (100.0%)' and 'Profile: Default'.

### 2.2.1. 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.

1. Explica las modificaciones del script.

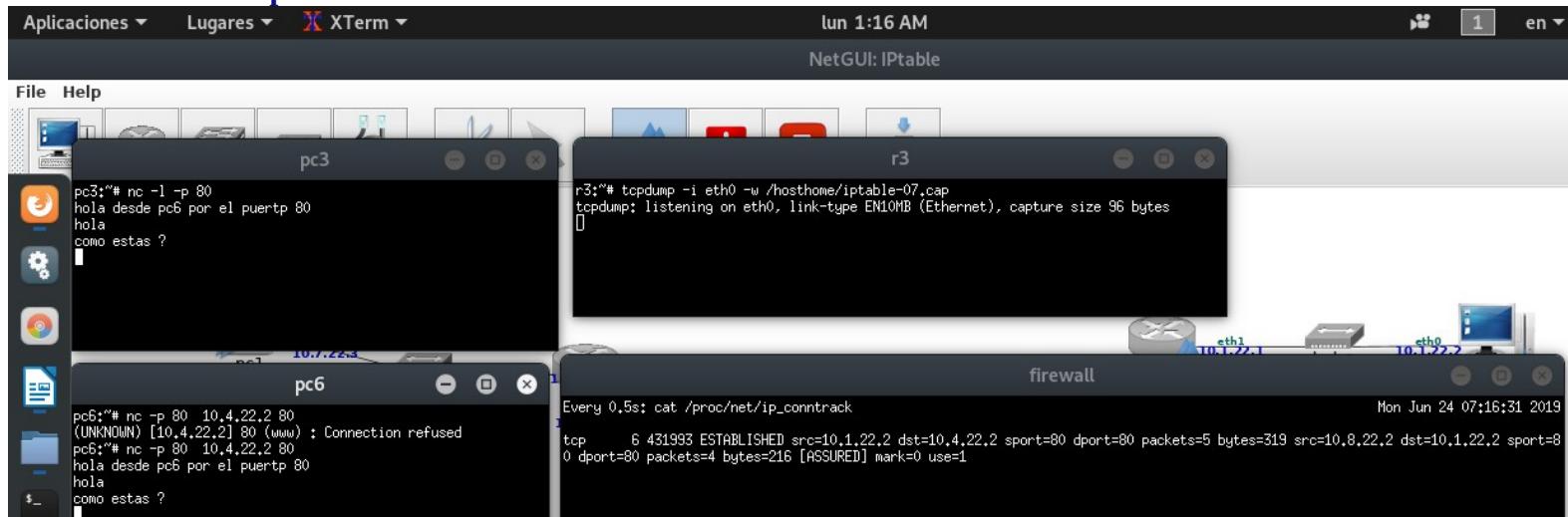
```
#!/bin/bash
# "Se borran reglas anteriores"
iptables -t nat -F
# "Reiniciar contadores de una cadena"
iptables -t nat -Z

#se estan aplicando nuevas traducciones DNAT
#estoy indicando que todo paquete que llega al Firewall a la interfaz eth2 con un puerto destino
#sera destinado a una IP y sera dirigido con otro puerto a esa IP.
iptables -t nat -A PREROUTING -p udp -i eth2 --dport 5001 -j DNAT --to-destination 10.7.22.3:5001
iptables -t nat -A PREROUTING -p udp -i eth2 --dport 5002 -j DNAT --to-destination 10.7.22.2:5001

#acá se estara aplicando un DNAT al trafico con destino al puerto 80 TCP, este sera redirigido a
#la PC3, puerto 80.
iptables -t nat -A PREROUTING -p tcp -i eth2 --dport 80 -j DNAT --to-destination 10.8.22.2:80
```

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 como has arrancado el cliente de nc en pc6.
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.

- **He agregado una nueva cadena PREROUTING para una regla de traducción de la tabla nat en este caso par el protocolo TCP en el puerto eth2 el cual lleva como destino el puerto 80, este tráfico sera retransmitido a la IP 10.8.22.2 puerto 80.**
- **La captura realizada en el router R3.**



- **Se lanza el server TCP el cual escucha en el puerto 80, el lanzamiento del cliente TCP con puerto origen y destino 80, este lleva como IP destino la interfaz eth2 del Firewall.**
- **Se puede ver que el protocolo de la conexión es TCP la cual esta en el estado ESTABLISHED, la IP origen es la PC6 10.1.22.2 y la IP destino 10.4.22.2(esta IP es la de la interfaz eth2 del Firewall), el puerto al que esta dirigida esta conexión es el 80. Cuando el mensaje entra al Firewall, la IP destino cambia por el DNAT configurado en el script, IP destino 10.8.22.0 con el puerto 80 y la IP origen es la 10.1.22.2.**

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

```

firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out    source               destination
  0   0 INNAT      udp  --  eth2   *      0.0.0.0/0          0.0.0.0/0
  0   0 INNAT      udp  --  eth2   *      0.0.0.0/0          0.0.0.0/0
  1   60 DNAT      tcp  --  eth2   *      0.0.0.0/0          0.0.0.0/0
Chain POSTROUTING (policy ACCEPT 1 packets, 60 bytes)
pkts bytes target     prot opt in     out    source               destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
pkts bytes target     prot opt in     out    source               destination
firewall:~#

```

Puedo ver que la traducción se están cumpliendo para un paquete de 60 bytes, se ve una política por defecto que es ACCEPT la cual no se cumple para ningún paquete, la cadena que se esta cumpliendo al inicio es la PREROUTING. Ademas de la cadena PREROUTING, están las cadenas POSTROUTING y OUTPUT, estas tienen la política ACCEPT y solo en la cadena POSTROUTING la política ACCEPT se cumple para un paquete.

5. Relaciona los resultados de la captura de trafico con la información extraída de ip\_conntrack y la tabla nat del firewall.

| No. | Time      | Source            | Destination       | Protocol | Length | Info   |
|-----|-----------|-------------------|-------------------|----------|--------|--|
| 1   | 0.000000  | c6:da:50:5d:05:6f | Broadcast         | ARP      | 42     | Who has 10.4.22.2 Tell 10.4.22.1   |
| 2   | 0.000264  | da:93:d4:ca:a6:26 | c6:da:50:5d:05:6f | ARP      | 42     | 10.4.22.2 is at da:93:d4:ca:a6:26  |
| 4   | 0.000539  | 10.4.22.2         | 10.1.22.2         | TCP      | 54     | 80 -- 80 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0  |
| 5   | 4.992302  | da:93:d4:ca:a6:26 | c6:da:50:5d:05:6f | ARP      | 42     | Who has 10.4.22.17 Tell 10.4.22.2  |
| 6   | 4.992337  | c6:da:50:5d:05:6f | da:93:d4:ca:a6:26 | ARP      | 42     | 10.4.22.1 is at c6:da:50:5d:05:6f  |
| 7   | 42.659410 | 10.4.22.2         | 10.4.22.2         | TCP      | 74     | [TCP Port numbers reused] 80 -- 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSecr=0 WS2=0 |
| 8   | 42.665259 | 10.4.22.2         | 10.1.22.2         | TCP      | 74     | 80 -- 80 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSecr=4294957109             |
| 9   | 42.665928 | 10.1.22.2         | 10.4.22.2         | TCP      | 66     | 80 -- 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSecr=146047   |
| 10  | 47.647074 | c6:da:50:5d:05:6f | da:93:d4:ca:a6:26 | ARP      | 42     | Who has 10.4.22.22 Tell 10.4.22.1  |
| 11  | 47.647271 | da:93:d4:ca:a6:26 | c6:da:50:5d:05:6f | ARP      | 42     | 10.4.22.22 is at da:93:d4:ca:a6:26   |
| 12  | 62.724047 | 10.1.22.2         | 10.4.22.2         | TCP      | 98     | 80 -- 80 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=32 TSecr=148057                                     |
| 13  | 62.724929 | 10.4.22.2         | 10.1.22.2         | TCP      | 66     | 80 -- 80 [ACK] Seq=1 Ack=33 Win=5792 Len=0 TSecr=4294959115                                      |

Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)  
Ethernet II, Src: c6:da:50:5d:05:6f (c6:da:50:5d:05:6f), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
Address Resolution Protocol (request)

0000 ff ff ff ff ff c6 da 50 5d 05 6f 08 06 00 01 ..... Pj:o....  
0010 08 00 06 04 00 01 c6 da 50 5d 05 6f 0a 04 16 01 ..... Pj:o....

- Se pude ver en la captura que se hizo en el router R3 que hay trafico origen de la IP

## **10.4.22.2 y destino 10.4.22.2. Este trafico va dirigido al puerto 80.**

### 3. Filtrado en el firewall: tabla filter

#### 3.1. Introducción: Servidores echo, daytime, telnet

En Linux se pueden manejar un conjunto de servicios a través del demonio inetd: Internet "superserver". Este demonio proporciona el acceso a esos servicios, en particular, en la práctica utilizaremos los servicios: echo, daytime y telnet. **Para activar los servicios que queremos utilizar, editaremos el fichero /etc/inetd.conf.** En este fichero encontraremos comentados algunos de los servicios que queremos usar en la práctica. Para usarlos, habrá que descomentar dichas líneas. Si no encontramos esas líneas comentadas, escribiremos la configuración en ese fichero tal y como se indica a continuación:

-Daytime: Es un servidor que escucha conexiones TCP en el puerto daytime (en /etc/services podemos ver que el puerto daytime está asociado al puerto 13). **Cuando un proceso se conecta a este número de puerto, el servidor daytime devuelve la hora actual de la máquina.** Para activar el servidor daytime es necesario que en la máquina donde queremos ese servidor, en su fichero /etc/inetd.conf se encuentre la siguiente línea:

#### **daytime stream tcp nowait root internal**

-Echo: Es un servidor que espera paquetes UDP en el puerto echo (en /etc/services podemos ver que el puerto echo está asociado al puerto 7). **Cuando un proceso envía una cadena de caracteres al servidor echo, este servidor devuelve al cliente esa misma cadena.** Para activar el servidor echo es necesario que en la máquina donde queremos ese servidor, en su fichero /etc/inetd.conf se encuentre la siguiente línea:

#### **echo dgram udp nowait root internal**

También se puede lanzar este mismo servicio a través del protocolo TCP, para ello, la línea del fichero /etc/inetd.conf debe ser:

#### **echo stream tcp nowait root internal**

-Telnet: Es un servidor que escucha conexiones TCP en el puerto telnet (en /etc/services podemos ver que el puerto telnet está asociado al puerto 23). **Cuando un proceso se conecta a este número de puerto, se establece una conexión remota entre la máquina cliente y la máquina servidor en la que el cliente puede ejecutar comandos en el servidor.** Requiere una fase de autenticación. Para activar el servidor telnet es necesario que en la máquina donde queremos ese servidor, en su fichero /etc/inetd.conf se encuentre la siguiente línea:

#### **telnet stream tcp nowait telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd**

Para poder permitir el acceso remoto como usuario root en el servidor de telnet es necesario comentar, en la máquina donde se va a lanzar el servidor, la siguiente línea en el fichero:

```
/etc/pam.d/login  
# auth [success=ok ignore=ignore user_unknown=ignore default=die] pam_securtty.so
```

La línea es un comentario porque comienza con el carácter '#'.

Una vez configurado el servicio que queremos arrancar dentro del fichero /etc/inetd.conf es necesario rearrancar el demonio inetd en la máquina donde queremos configurar los servicios para que se cargue

la configuración de ese fichero. Para ello deberás ejecutar:

**/etc/init.d/inetd restart**

→ **Arrancando el demonio inetd**

→ **Activar los servicios que queremos utilizar, editaremos el fichero /etc/inetd.conf.**

Puedes comprobar que servicios están activos ejecutando:

**netstat -atun antes y después de activar los servicios**

The screenshot shows a terminal window titled "firewall" running in XTerm. The window displays the following text:

```
firewall:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address     State
tcp      0      0 0.0.0.0:111               0.0.0.0:*          LISTEN
udp      0      0 0.0.0.0:111               0.0.0.0:*          LISTEN
firewall:~# /etc/init.d/inetd restart
Restarting internet superserver: inetd.
firewall:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address     State
tcp      0      0 0.0.0.0:7                0.0.0.0:*          LISTEN
tcp      0      0 0.0.0.0:13               0.0.0.0:*          LISTEN
tcp      0      0 0.0.0.0:111               0.0.0.0:*          LISTEN
tcp      0      0 0.0.0.0:23               0.0.0.0:*          LISTEN
udp      0      0 0.0.0.0:7                0.0.0.0:*          LISTEN
udp      0      0 0.0.0.0:69               0.0.0.0:*          LISTEN
udp      0      0 0.0.0.0:111               0.0.0.0:*          LISTEN
firewall:~# cat -n /etc/inetd.conf | more
 1
 2 # /etc/inetd.conf: see inetd(8) for further informations.
 3 #
 4 # Internet superserver configuration database
 5 daytime    stream  tcp    nowait  root    internal
 6 echo       dgram   udp    nowait  root    internal
 7 echo       stream  tcp    nowait  root    internal
 8 telnet     stream  tcp    nowait  telnetd /usr/sbin/tcpd /usr/sbin/in.telnetd
 9
10 # Lines starting with "#;LABEL:" or "#<off>" should not
11 # be changed unless you know what you are doing!
12 #
13 # If you want to disable an entry so it isn't touched during
```

Los servicios que se activaron en el Firewall.

### 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.**

#A)

#borrar la tabla filter

iptables -t filter -F

iptables -t filter -Z

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.**

#B)

#políticas por defecto descarta todo el tráfico al firewall y acepta todo tráfico saliente

iptables -t filter -P INPUT DROP

iptables -t filter -P FORWARD DROP

iptables -t filter -P OUTPUT ACCEPT

#PARTIMOS DEL FW1.SH

#Reglas para aplicar nateo en firewall

```
iptables -t nat -A POSTROUTING -s 10.6.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2  
iptables -t nat -A POSTROUTING -s 10.7.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2  
iptables -t nat -A POSTROUTING -s 10.8.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2
```

c) Permitir el trafico de entrada dirigido a las aplicaciones que se están ejecutando en firewall

‘unicamente si este trafico tiene su origen en las subredes privadas de la empresa.

```
#C  
#permitiendo trafico proveniente de las redes privadas  
iptables -t filter -A INPUT -s 10.6.22.0/24 -j ACCEPT  
iptables -t filter -A INPUT -s 10.7.22.0/24 -j ACCEPT  
iptables -t filter -A INPUT -s 10.8.22.0/24 -j ACCEPT
```

d) Permitir todo el trafico saliente desde las subredes privadas hacia Internet y el trafico de respuesta al saliente. Ten en cuenta que como has partido del script fw1.sh, en dicho script ya tenias las reglas de latabla nat de modificación de la dirección IP de origen de los paquetes que reenvía el firewall y los paquetes del trafico entrante de respuesta al saliente.

#D)  
#se permite el reenvio de paquetes entrante perteneciente a una conexion ya establecida

```
iptables -t filter -A FORWARD -i eth2 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT  
#conexion de la red privada
```

```
iptables -t filter -A FORWARD -i eth1 -o eth2 -m state --state RELATED,ESTABLISHED -j ACCEPT  
#conexion de la DMZ
```

```
iptables -t filter -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

e) Permitir desde Internet únicamente el trafico entrante nuevo hacia la zona DMZ según las siguientes reglas y su correspondiente trafico 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 maquina de Internet y el trafico de respuesta.

#E)  
#permitiendo el trafico de internet hacia la DMZ cuando se dirige a un puerto y pc con ip especifica  
#el server echo instalado en PC4(UDP, port 7)

```
iptables -t filter -A FORWARD -i eth2 -d 10.5.22.2 -o eth1 -p udp --dport 7 -j ACCEPT  
#el server daytime instalado en la PC5 (TCP , port 13)  
iptables -t filter -A FORWARD -i eth1 -d 10.5.22.3 -o eth1 -p udp --dport 13 -j ACCEPT
```

-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 maquina de Internet y el trafico 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.**

```
#F  
#permitir conexion de la red privada a la dmz:  
#conexion telnet (TCP, port 13) desde PC1 -> PC5
```

```
iptables -t filter -A FORWARD -i eth0 -s 10.7.22.3 -d 10.5.22.3 -p tcp --dport 23 -j ACCEPT
```

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.**

```
#conexion ECHO (TCP, port 7) desde PC1 -> PC4  
iptables -t filter -A FORWARD -i eth0 -s 10.7.22.3 -d 10.5.22.2 -p tcp --dport 7 -j ACCEPT
```

**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 expícalo.**

```
#G  
#eliminando comunicacion de la DMZ a la red privada y al firewall.  
iptables -t filter -A FORWARD -s 10.5.22.2 -d 10.7.22.0/24 -j DROP  
iptables -t filter -A FORWARD -s 10.5.22.3 -d 10.7.22.0/24 -j DROP
```

```
iptables -t filter -A FORWARD -s 10.5.22.2 -d 10.8.22.0/24 -j DROP  
iptables -t filter -A FORWARD -s 10.5.22.3 -d 10.8.22.0/24 -j DROP
```

```
iptables -t filter -A FORWARD -s 10.5.22.2 -d 10.6.22.0/24 -j DROP  
iptables -t filter -A FORWARD -s 10.5.22.3 -d 10.6.22.0/24 -j DROP
```

### 3.1. Pruebas de la configuración del firewall

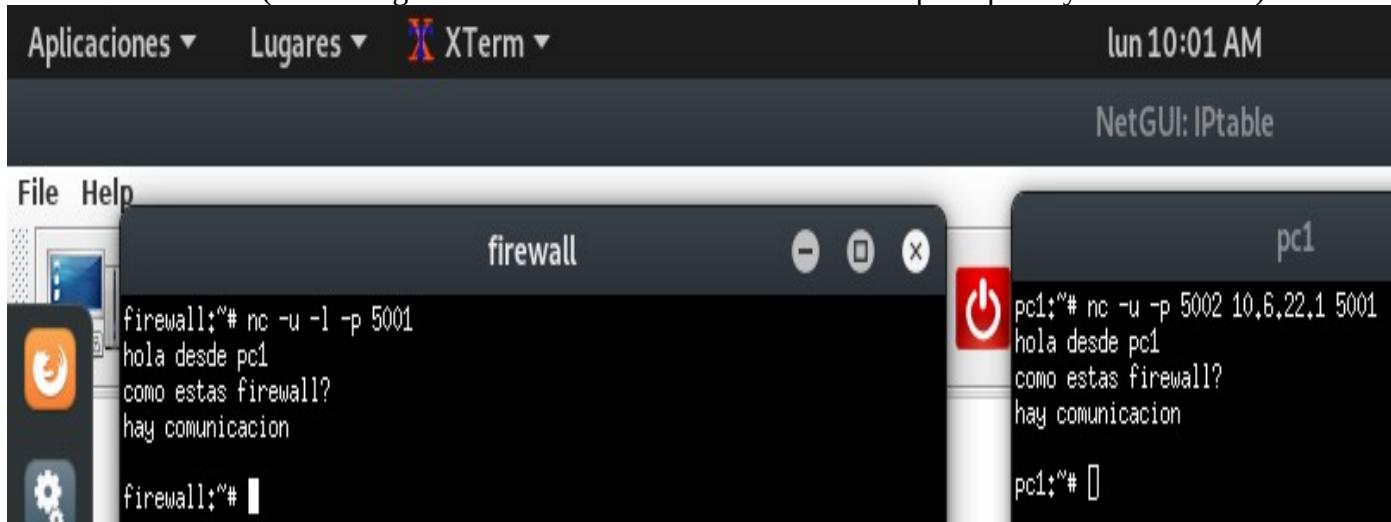
Para poder comprobar que reglas se están aplicando a cada caso que pruebas, añade a cada regla otra regla con las misma condiciones y acción LOG de forma que quede una anotación en el fichero de log cada vez que se cumpla cada condición.

A continuación se dan algunas pautas para probar cada una de las restricciones de fw3.sh:

1. Permitir el tráfico de entrada en la máquina firewall únicamente desde las subredes privadas de la empresa.

#### Pruebas

- a) Si se arranca una aplicación servidor (TCP o UDP) en la máquina firewall solo podrá aceptar tráfico de un cliente que envíe mensajes desde una de las máquinas de las subredes privadas. Asegurate 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 comunique con dicho servidor (escribe alguna línea en cada uno de los terminales para que haya tráfico UDP).



Server UDP en Firewall, cliente UDP PC1

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.

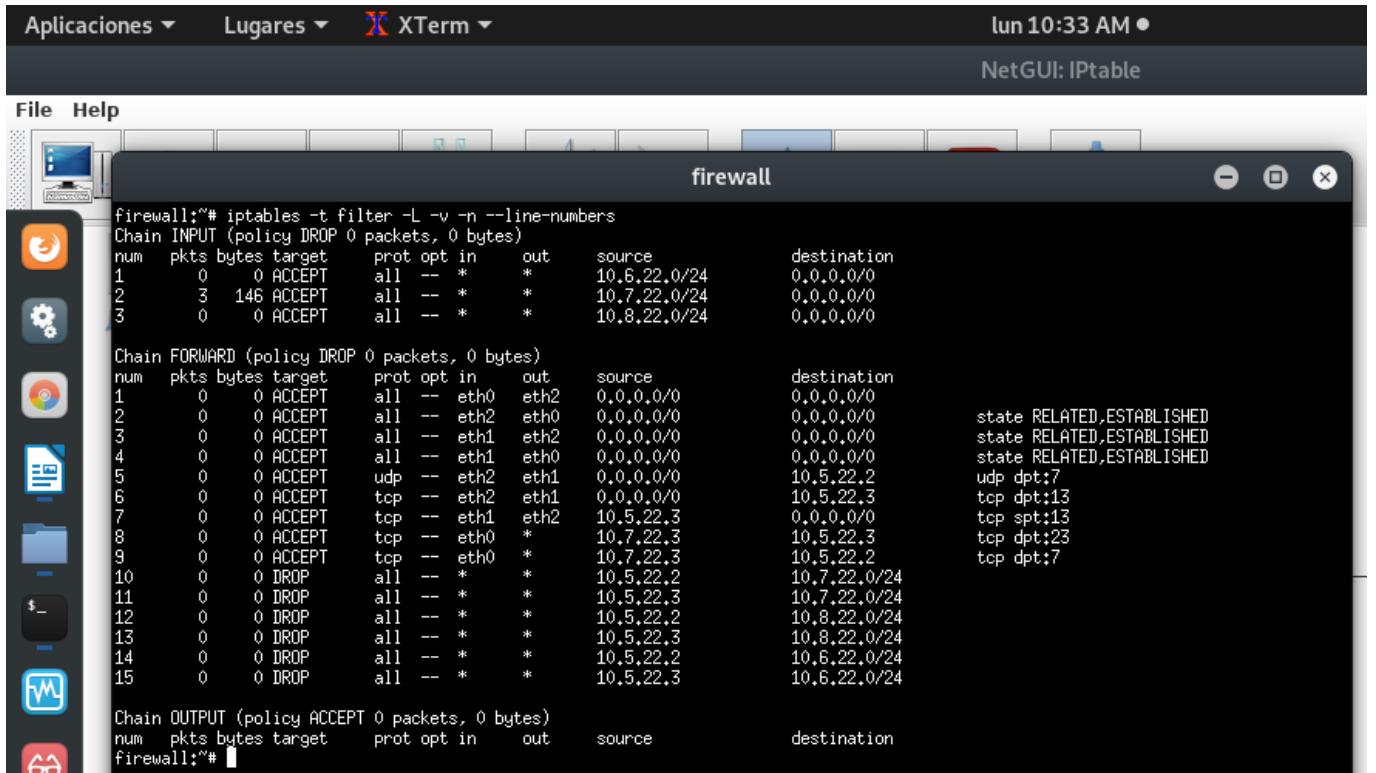
Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 10:30 AM •  
NetGUI: IPtable

File Help

firewall

```
firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 2 packets, 108 bytes)
num  pkts bytes target     prot opt in     out    source        destination
Chain POSTROUTING (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source        destination
 1      0   0 SNAT       all  --  *      eth2    10.6.22.0/24    0.0.0.0/0          to:10.4.22.2
 2      0   0 SNAT       all  --  *      eth2    10.7.22.0/24    0.0.0.0/0          to:10.4.22.2
 3      0   0 SNAT       all  --  *      eth2    10.8.22.0/24    0.0.0.0/0          to:10.4.22.2
Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source        destination
firewall:~#
```

**La tabla NAT se han cumplido tres reglas, en la primera un paquete, en la segunda para 4 paquetes y en la tercera para 4 paquetes, en cada regla esta la política por defecto ACCEPT.**



The screenshot shows a terminal window titled "firewall" running on a Linux desktop. The window displays the output of the command "iptables -t filter -L -v -n --line-numbers". The output shows three chains: INPUT, FORWARD, and OUTPUT. The INPUT chain has 3 rules, the FORWARD chain has 15 rules, and the OUTPUT chain has 2 rules. The rules are listed with their line numbers, packet counts, byte counts, target, protocol, options, and source/destination information. The policy for each chain is also indicated.

```
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target  prot opt in     out    source        destination
1    0     0 ACCEPT   all  --  *      *      10.6.22.0/24  0.0.0.0/0
2    3    146 ACCEPT   all  --  *      *      10.7.22.0/24  0.0.0.0/0
3    0     0 ACCEPT   all  --  *      *      10.8.22.0/24  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   all  --  eth1   eth2   0.0.0.0/0    0.0.0.0/0
4    0     0 ACCEPT   all  --  eth1   eth0   0.0.0.0/0    0.0.0.0/0
5    0     0 ACCEPT   udp  --  eth2   eth1   0.0.0.0/0    10.5.22.2    udp dpt:7
6    0     0 ACCEPT   tcp  --  eth2   eth1   0.0.0.0/0    10.5.22.3    tcp dpt:13
7    0     0 ACCEPT   tcp  --  eth1   eth2   10.5.22.3   0.0.0.0/0    tcp spt:13
8    0     0 ACCEPT   tcp  --  eth0   *      10.7.22.3   10.5.22.3    tcp dpt:23
9    0     0 ACCEPT   tcp  --  eth0   *      10.7.22.3   10.5.22.2    tcp dpt:7
10   0     0 DROP    all  --  *      *      10.5.22.2   10.7.22.0/24
11   0     0 DROP    all  --  *      *      10.5.22.3   10.7.22.0/24
12   0     0 DROP    all  --  *      *      10.5.22.2   10.8.22.0/24
13   0     0 DROP    all  --  *      *      10.5.22.3   10.8.22.0/24
14   0     0 DROP    all  --  *      *      10.5.22.2   10.6.22.0/24
15   0     0 DROP    all  --  *      *      10.5.22.3   10.6.22.0/24

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

**La tabla filter tiene tres reglas INPUT, FORWARD y OUTPUT, solo en la regla OUTPUT hay paquetes que se le están cumpliendo políticas por defecto el cual son dos. En la regla INPUT se cumple la política DROP a un total de 0 paquetes, en la regla FORWARD se cumple la política DROP a un total de 0 paquetes y la regla OUTPUT se cumple una política ACCEPT a un total de 2 paquetes.**

- b) No podrá aceptar tráfico desde aplicaciones cliente lanzadas en otras subredes diferentes. Asegurate 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).

### Comunicación de PC6 al Firewall.

```

firewall:~# nc -u -l -p 7777
dscsc
csadcs
fadvdavdvdvdafvfdgwefwugergw
ok , ya vemos que si tenemos
comunicacion a firewall

```

```

pc6:~# nc -u -p 6666 10.4.22.2 7777
dscsc
csadcs
fadvdavdvdvdafvfdgwefwugergw
ok , ya vemos que si tenemos
comunicacion a firewall

```

Explica en la memoria:

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

**Tabla nat:** Se puede ver que en la tabla NAT que esta la política por defecto ACCEPT para las reglas PREROUTING, POSTROUTING y OUTPUT. Previamente hice un ping de la PC6 al Firewall y se puede constatar que las reglas de la tabla NAT se esta cumpliendo, porque, no hay comunicación de la PC6 al Firewall.

```

firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 5 packets, 231 bytes)
num pkts bytes target     prot opt in     out     source         destination
Chain POSTROUTING (policy ACCEPT 4 packets, 240 bytes)
num pkts bytes target     prot opt in     out     source         destination
1    0    0 SNAT      all  --  eth2    10.6.22.0/24  0.0.0.0/0        to:10.4.22.2
2    0    0 SNAT      all  --  eth2    10.7.22.0/24  0.0.0.0/0        to:10.4.22.2
3    0    0 SNAT      all  --  eth2    10.8.22.0/24  0.0.0.0/0        to:10.4.22.2
4    0    0 SNAT      all  --  eth2    10.6.22.0/24  0.0.0.0/0        to:10.4.22.2
5    0    0 SNAT      all  --  eth2    10.7.22.0/24  0.0.0.0/0        to:10.4.22.2
6    0    0 SNAT      all  --  eth2    10.8.22.0/24  0.0.0.0/0        to:10.4.22.2

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

```

**Tabla filter:** se puede ver que hay tres reglas predefinidas(INPUT, FORWARD y OUTPUT), en la regla INPUT se puede ver que se cumplió la política DROP para 2 paquetes. Esos fueron los que se le negó la entrada al Firewall.

```
Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 10:42 AM •
firewall
```

```
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target  prot opt in     out    source        destination
1    10   399 ACCEPT  all   --   *      *      10.1.22.0/24  0.0.0.0/0
2    0    0 ACCEPT  all   --   *      *      10.6.22.0/24  0.0.0.0/0
3    0    0 ACCEPT  all   --   *      *      10.7.22.0/24  0.0.0.0/0
4    0    0 ACCEPT  all   --   *      *      10.8.22.0/24  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  all   --  eth1   eth2   0.0.0.0/0   0.0.0.0/0
4    0    0 ACCEPT  all   --  eth1   eth0   0.0.0.0/0   0.0.0.0/0
5    0    0 ACCEPT  udp   --  eth2   eth1   0.0.0.0/0   10.5.22.2  udp dpt:7
6    0    0 ACCEPT  tcp   --  eth2   eth1   0.0.0.0/0   10.5.22.3  tcp dpt:13
7    0    0 ACCEPT  tcp   --  eth1   eth2   10.5.22.3  0.0.0.0/0  tcp spt:13
8    0    0 ACCEPT  tcp   --  eth0   *      10.7.22.3  10.5.22.3  tcp dpt:23
9    0    0 ACCEPT  tcp   --  eth0   *      10.7.22.3  10.5.22.2  tcp dpt:7
10   0    0 DROP   all   --   *      *      10.5.22.2  10.7.22.0/24
11   0    0 DROP   all   --   *      *      10.5.22.3  10.7.22.0/24
12   0    0 DROP   all   --   *      *      10.5.22.2  10.8.22.0/24
13   0    0 DROP   all   --   *      *      10.5.22.3  10.8.22.0/24
14   0    0 DROP   all   --   *      *      10.5.22.2  10.6.22.0/24
15   0    0 DROP   all   --   *      *      10.5.22.3  10.6.22.0/24

Chain OUTPUT (policy ACCEPT 4 packets, 245 bytes)
num  pkts bytes target  prot opt in     out    source        destination
firewall:~#
```

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

### Pruebas

- a) Si se arranca una aplicación servidor (TCP o UDP) en una de las maquinas de Internet y se arranca una aplicación cliente para que se comunique con ese servidor en una de las maquinas de las subredes internas, el trafico debe poder enviarse del cliente al servidor y del servidor al cliente, observando que el trafico que sale del firewall con destino a la maquina de Internet no tiene como dirección IP origen la dirección de la maquina que pertenece a la subred privada, sino que lleva la dirección publica del firewall de la interfaz que le conecta con Internet. Ejecuta la misma prueba que en el apartado 2.1.3. Asegurate 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 numero de veces.-las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el numero de veces que se ejecutan.

### Tabla NAT y FILTER antes de enviar mensajes.

Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 10:48 AM •

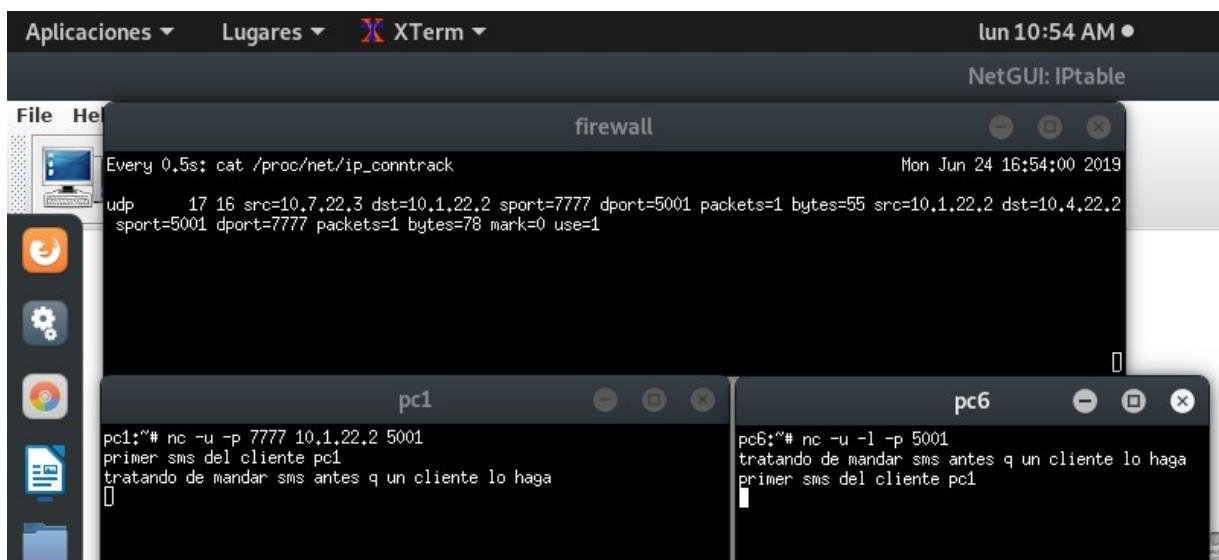
**firewall**

```
firewall:~# iptables -t nat -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 5 packets, 231 bytes)
num  pkts bytes target     prot opt in     out    source         destination
Chain POSTROUTING (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1    0    0 SNAT      all  --  *    eth2   10.6.22.0/24  0.0.0.0/0      to:10.4.22.2
2    0    0 SNAT      all  --  *    eth2   10.7.22.0/24  0.0.0.0/0      to:10.4.22.2
3    0    0 SNAT      all  --  *    eth2   10.8.22.0/24  0.0.0.0/0      to:10.4.22.2
4    0    0 SNAT      all  --  *    eth2   10.6.22.0/24  0.0.0.0/0      to:10.4.22.2
5    0    0 SNAT      all  --  *    eth2   10.7.22.0/24  0.0.0.0/0      to:10.4.22.2
6    0    0 SNAT      all  --  *    eth2   10.8.22.0/24  0.0.0.0/0      to:10.4.22.2

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1    10  398 ACCEPT      all  --  *    *    10.1.22.0/24  0.0.0.0/0
2    0    0 ACCEPT      all  --  *    *    10.6.22.0/24  0.0.0.0/0
3    0    0 ACCEPT      all  --  *    *    10.7.22.0/24  0.0.0.0/0
4    0    0 ACCEPT      all  --  *    *    10.8.22.0/24  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      all  --  eth1  eth2   0.0.0.0/0    0.0.0.0/0
4    0    0 ACCEPT      all  --  eth1  eth0   0.0.0.0/0    0.0.0.0/0
5    0    0 ACCEPT      udp   --  eth2  eth1   0.0.0.0/0    10.5.22.2
6    0    0 ACCEPT      tcp   --  eth2  eth1   0.0.0.0/0    10.5.22.3
7    0    0 ACCEPT      tcp   --  eth1  eth2   10.5.22.3   0.0.0.0/0
8    0    0 ACCEPT      tcp   --  eth0  *    10.7.22.3   10.5.22.3
9    0    0 ACCEPT      tcp   --  eth0  *    10.7.22.3   10.5.22.2
10   0    0 DROP        all  --  *    *    10.5.22.2   10.7.22.0/24
11   0    0 DROP        all  --  *    *    10.5.22.3   10.7.22.0/24
12   0    0 DROP        all  --  *    *    10.5.22.2   10.8.22.0/24
13   0    0 DROP        all  --  *    *    10.5.22.3   10.8.22.0/24
14   0    0 DROP        all  --  *    *    10.5.22.2   10.6.22.0/24
15   0    0 DROP        all  --  *    *    10.5.22.3   10.6.22.0/24

Chain OUTPUT (policy ACCEPT 4 packets, 245 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~#
```



En esta conexión UDP se puede ver que el Firewall ha recibido un paquete de respuesta, es por eso que podemos ver que presenta un mensaje [ASSURED]. Cuando en la conexión UDP solo el cliente es el que manda mensajes, este presenta un mensaje [UNREPLIED].

Tabla NAT y FILTER después de mandar los mensajes: se puede ver que en la tabla NAT se cumple una regla SNAT en la cadena POSTROUTING y también se ve que son tres cadenas que se aplican en la tabla NAT(PREROUTING, POSTROUTING y OUTPUT) la política por

**defecto que se cumple es ACCEPT. En la tabla FILTER se cumplen tres reglas FORWARD, las cadenas por defecto para las cadenas (INPUT y FORWARD) es DROP y para la cadena OUTPUT la política por defecto es ACCEPT.**

```
Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 10:54 AM •
firewall

firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 7 packets, 345 bytes)
num  pkts bytes target     prot opt in     out    source         destination
Chain POSTROUTING (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      0     0 SNAT       all  --  eth2    10.6.22.0/24  0.0.0.0/0      to:10.4.22.2
2      2   114 SNAT       all  --  eth2    10.7.22.0/24  0.0.0.0/0      to:10.4.22.2
3      0     0 SNAT       all  --  eth2    10.8.22.0/24  0.0.0.0/0      to:10.4.22.2
4      0     0 SNAT       all  --  eth2    10.6.22.0/24  0.0.0.0/0      to:10.4.22.2
5      0     0 SNAT       all  --  eth2    10.7.22.0/24  0.0.0.0/0      to:10.4.22.2
6      0     0 SNAT       all  --  eth2    10.8.22.0/24  0.0.0.0/0      to:10.4.22.2

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      10   399 ACCEPT     all  --  *      *      10.1.22.0/24  0.0.0.0/0
2      0     0 ACCEPT     all  --  *      *      10.6.22.0/24  0.0.0.0/0
3      0     0 ACCEPT     all  --  *      *      10.7.22.0/24  0.0.0.0/0
4      0     0 ACCEPT     all  --  *      *      10.8.22.0/24  0.0.0.0/0

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      2   114 ACCEPT     all  --  eth0    eth2    0.0.0.0/0      0.0.0.0/0
2      2   185 ACCEPT     all  --  eth2    eth0    0.0.0.0/0      0.0.0.0/0      state RELATED,ESTABLISHED
3      0     0 ACCEPT     all  --  eth1    eth2    0.0.0.0/0      0.0.0.0/0      state RELATED,ESTABLISHED
4      0     0 ACCEPT     all  --  eth1    eth0    0.0.0.0/0      0.0.0.0/0      state RELATED,ESTABLISHED
5      0     0 ACCEPT     udp  --  eth2    eth1    0.0.0.0/0      10.5.22.2      udp dpt:7
6      0     0 ACCEPT     tcp  --  eth2    eth1    0.0.0.0/0      10.5.22.3      tcp dpt:13
7      0     0 ACCEPT     tcp  --  eth1    eth2    10.5.22.3     0.0.0.0/0      tcp spt:13
8      0     0 ACCEPT     tcp  --  eth0    *      10.7.22.3     10.5.22.3      tcp dpt:23
9      0     0 ACCEPT     tcp  --  eth0    *      10.7.22.3     10.5.22.2      tcp dpt:7
10     0     0 DROP        all  --  *      *      10.5.22.2     10.7.22.0/24
11     0     0 DROP        all  --  *      *      10.5.22.3     10.7.22.0/24
12     0     0 DROP        all  --  *      *      10.5.22.2     10.8.22.0/24
13     0     0 DROP        all  --  *      *      10.5.22.3     10.8.22.0/24
14     0     0 DROP        all  --  *      *      10.5.22.2     10.6.22.0/24
15     0     0 DROP        all  --  *      *      10.5.22.3     10.6.22.0/24

Chain OUTPUT (policy ACCEPT 4 packets, 245 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 maquinas 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.

### Mensajes que se trataron de mandar la PC5 y la PC6.

```

lun 11:03 AM •
NetGUI: IPTable
pc5
pc5:~# nc -p 5000 10.1.22.2 5001
tratando de mandar mensajes tcp a la pc6
mensaje 2
?????
[]

pc6
pc6:~# nc -u -l -p 5001

```

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 numero de veces que se ejecutan.

**En la tabla NAT no se cumple ninguna regla, porque, el tráfico no se ha originado en la red privada, en la tabla FILTER se puede ver que se cumple la regla DROP para 5 paquetes. Y la política que se ejecuta por defecto es DROP en la tabla FILTER.**

```

Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 11:03 AM • NetGUI
firewall
firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 12 packets, 645 bytes)
num  pkts bytes target      prot opt in     out    source        destination
num  pkts bytes target      prot opt in     out    source        destination
Chain POSTROUTING (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target      prot opt in     out    source        destination
1    0     0 SNAT          all   --  *      eth2   10.6.22.0/24  0.0.0.0/0      to:10.4.22.2
2    2    114 SNAT          all   --  *      eth2   10.7.22.0/24  0.0.0.0/0      to:10.4.22.2
3    0     0 SNAT          all   --  *      eth2   10.8.22.0/24  0.0.0.0/0      to:10.4.22.2
4    0     0 SNAT          all   --  *      eth2   10.6.22.0/24  0.0.0.0/0      to:10.4.22.2
5    0     0 SNAT          all   --  *      eth2   10.7.22.0/24  0.0.0.0/0      to:10.4.22.2
6    0     0 SNAT          all   --  *      eth2   10.8.22.0/24  0.0.0.0/0      to:10.4.22.2
Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target      prot opt in     out    source        destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target      prot opt in     out    source        destination
1    10   399 ACCEPT       all   --  *      *      10.1.22.0/24  0.0.0.0/0
2    0     0 ACCEPT       all   --  *      *      10.6.22.0/24  0.0.0.0/0
3    0     0 ACCEPT       all   --  *      *      10.7.22.0/24  0.0.0.0/0
4    0     0 ACCEPT       all   --  *      *      10.8.22.0/24  0.0.0.0/0
Chain FORWARD (policy DROP 5 packets, 300 bytes)
num  pkts bytes target      prot opt in     out    source        destination
1    2    114 ACCEPT       all   --  eth0   eth2   0.0.0.0/0      0.0.0.0/0
2    2    165 ACCEPT       all   --  eth2   eth0   0.0.0.0/0      0.0.0.0/0
[LISHED]
3    0     0 ACCEPT       all   --  eth1   eth2   0.0.0.0/0      0.0.0.0/0
[LISHED]
4    0     0 ACCEPT       all   --  eth1   eth0   0.0.0.0/0      0.0.0.0/0
[LISHED]
5    0     0 ACCEPT       udp   --  eth2   eth1   0.0.0.0/0      10.5.22.2      udp dpt:7
6    0     0 ACCEPT       tcp   --  eth2   eth1   0.0.0.0/0      10.5.22.3      tcp dpt:13
7    0     0 ACCEPT       tcp   --  eth1   eth2   10.5.22.3      0.0.0.0/0      tcp spt:13
8    0     0 ACCEPT       tcp   --  eth0   *      10.7.22.3      10.5.22.3      tcp dpt:23
9    0     0 ACCEPT       tcp   --  eth0   *      10.7.22.3      10.5.22.2      tcp dpt:7
10   0     0 DROP        all   --  *      *      10.5.22.2      10.7.22.0/24
11   0     0 DROP        all   --  *      *      10.5.22.3      10.7.22.0/24
12   0     0 DROP        all   --  *      *      10.5.22.2      10.8.22.0/24
13   0     0 DROP        all   --  *      *      10.5.22.3      10.8.22.0/24
14   0     0 DROP        all   --  *      *      10.5.22.2      10.6.22.0/24
15   0     0 DROP        all   --  *      *      10.5.22.3      10.6.22.0/24
Chain OUTPUT (policy ACCEPT 4 packets, 245 bytes)
num  pkts bytes target      prot opt in     out    source        destination
firewall:~# []

```

1. 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).

The screenshot shows an XTerm window titled "pc4". The terminal output is as follows:

```
Configuring network interfaces...done.
Starting radvd;
pc4:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:111              0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:111              0.0.0.0:*               LISTEN
pc4:~# nano /etc/inetd.conf
pc4:~# /etc/init.d/inetd restart
Restarting internet superserver: inetd.
pc4:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 0.0.0.0:111              0.0.0.0:*               LISTEN
tcp        0      0 0.0.0.0:23               0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:7                0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:69               0.0.0.0:*               LISTEN
udp        0      0 0.0.0.0:111              0.0.0.0:*               LISTEN
pc4:~# cat /etc/inetd.conf
# /etc/inetd.conf: see inetd(8) for further informations.
#
# Internet superserver configuration database
#echo      stream  tcp    nowait  root    internal
#echo      dgram   udp    nowait  root    internal
```

## Pruebas

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

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

The screenshot shows several XTerm windows and a NetGUI interface. One window on pc4 shows the command "nc -u -l -p 7" being run. Another window on pc6 shows the command "nc -u 10.5.22.2 7" being run. A third window on pc4 shows the response "holo , desde pc6 a pc4 ahora si hay comunicacion". A fourth window on pc6 shows the response "holo , desde pc4 a pc6 ahora si hay comunicacion". The NetGUI interface shows a "firewall" window with the command "Every 0.5s: cat /proc/net/ip\_conntrack" running, displaying network connection logs.

Asegurate de que antes de lanzar el cliente desde una maquina 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 numero de veces.

Aplicaciones ▾ Lugares ▾ XTerm ▾

firewall

```
firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 15 packets, 811 bytes)
num  pkts bytes target     prot opt in     out    source         destination
      0    0 SNAT      all  --  * eth2    10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
      2   114 SNAT      all  --  * eth2    10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
      0    0 SNAT      all  --  * eth2    10.8.22.0/24  0.0.0.0/0          to:10.4.22.2
      0    0 SNAT      all  --  * eth2    10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
      0    0 SNAT      all  --  * eth2    10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
      0    0 SNAT      all  --  * eth2    10.8.22.0/24  0.0.0.0/0          to:10.4.22.2

Chain POSTROUTING (policy ACCEPT 5 packets, 291 bytes)
num  pkts bytes target     prot opt in     out    source         destination
      0    0 SNAT      all  --  * eth2    10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
      2   114 SNAT      all  --  * eth2    10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
      3    0 SNAT      all  --  * eth2    10.8.22.0/24  0.0.0.0/0          to:10.4.22.2
      4    0 SNAT      all  --  * eth2    10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
      5    0 SNAT      all  --  * eth2    10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
      6    0 SNAT      all  --  * eth2    10.8.22.0/24  0.0.0.0/0          to:10.4.22.2

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in     out    source         destination
      1    13  557 ACCEPT    all  --  * eth2    10.1.22.0/24  0.0.0.0/0
      2    0    0 ACCEPT    all  --  * eth2    10.6.22.0/24  0.0.0.0/0
      3    0    0 ACCEPT    all  --  * eth2    10.7.22.0/24  0.0.0.0/0
      4    0    0 ACCEPT    all  --  * eth2    10.8.22.0/24  0.0.0.0/0

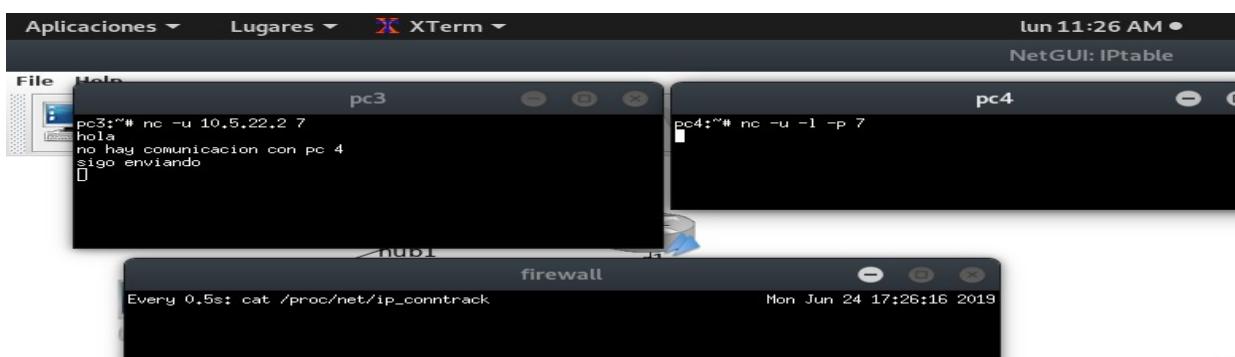
Chain FORWARD (policy DROP 6 packets, 360 bytes)
num  pkts bytes target     prot opt in     out    source         destination
      1    2   114 ACCEPT    all  --  eth0  eth2    0.0.0.0/0  0.0.0.0/0
      2    2   165 ACCEPT    all  --  eth2  eth0    0.0.0.0/0  0.0.0.0/0
      3    1   48 ACCEPT    all  --  eth1  eth2    0.0.0.0/0  0.0.0.0/0
      4    0    0 ACCEPT    all  --  eth1  eth0    0.0.0.0/0  0.0.0.0/0
      5    2   105 ACCEPT   udp  --  eth2  eth1    0.0.0.0/0  10.5.22.2 state RELATED,ESTABLISHED
      6    0    0 ACCEPT   tcp  --  eth2  eth1    0.0.0.0/0  10.5.22.3
      7    0    0 ACCEPT   tcp  --  eth1  eth2    10.5.22.3  0.0.0.0/0
      8    0    0 ACCEPT   tcp  --  eth0  *      10.7.22.3  10.5.22.3
      9    0    0 ACCEPT   tcp  --  eth0  *      10.7.22.3  10.5.22.2 state RELATED,ESTABLISHED
     10   0    0 DROP      all  --  *      *      10.5.22.2  10.7.22.0/24
     11   0    0 DROP      all  --  *      *      10.5.22.3  10.7.22.0/24
     12   0    0 DROP      all  --  *      *      10.5.22.2  10.8.22.0/24
     13   0    0 DROP      all  --  *      *      10.5.22.3  10.8.22.0/24
     14   0    0 DROP      all  --  *      *      10.5.22.2  10.6.22.0/24
     15   0    0 DROP      all  --  *      *      10.5.22.3  10.6.22.0/24

Chain OUTPUT (policy ACCEPT 7 packets, 487 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~#
```

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

Para la tabla NAT se puede ver que no se ha cumplido ninguna regla, hay tres cadenas(PREROUTING, POSTROUTING y OUTPUT), en cada cadena se cumple la política ACCEPT con diferentes numero de paquetes para cada una. En la tabla FILTER se cumple la regla ACCEPT dos veces en el puerto UDP 7, esto se da en la cadena FORWARD. Las reglas que se cumplen en la cadena FORWARD se puede ver que cuando el paquete va del cliente al servidor ECHO lo hace a través del puerto 7 UDP, pero cuando es en sentido contrario no especifica el puerto, pero si se puede ver que el estado de la conexión(RELATED, ESTABLISHED), es una conexión previamente establecida.

- Si se prueba lo mismo arrancando el comando anterior desde pc3 y se manda una cadena de caracteres, no se debería obtener respuesta.



Asegurate 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 numero de veces.

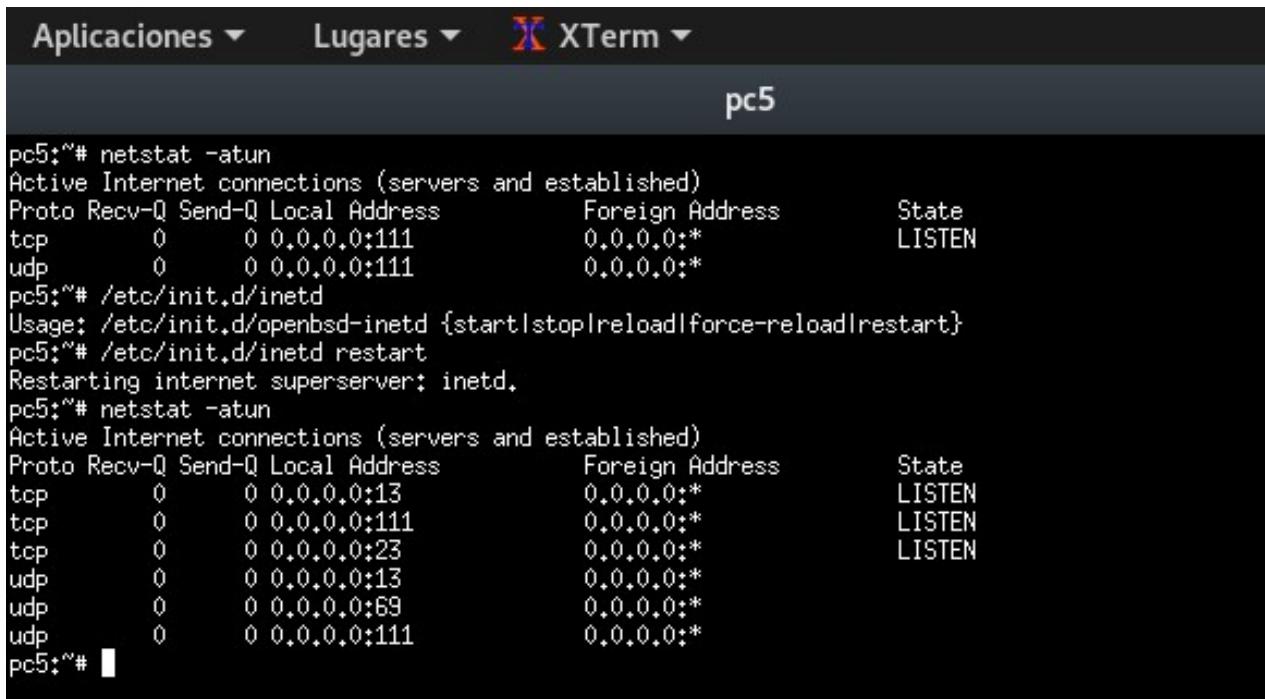
- las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el numero de veces que se ejecutan.

**En esta prueba para la tabla NAT no hay reglas que se cumplan, en la tabla FILTER tampoco hay reglas que se cumplan, pero en la cadena FORWARD para el trafico entrante se puede ver que se ejecuta la política por defecto DROP para los paquetes que tratan de llegar al server UDP en la PC4.**

```
Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 11:2
firewall
firewall:~# iptables -t nat -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 18 packets, 944 bytes)
num  pkts bytes target     prot opt in     out    source         destination
Chain POSTROUTING (policy ACCEPT 5 packets, 291 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      0     0 SNAT       all  --  eth2    10.6.22.0/24   0.0.0.0/0        to:10.4.22.2
2      2   114 SNAT       all  --  eth2    10.7.22.0/24   0.0.0.0/0        to:10.4.22.2
3      0     0 SNAT       all  --  eth2    10.8.22.0/24   0.0.0.0/0        to:10.4.22.2
4      0     0 SNAT       all  --  eth2    10.6.22.0/24   0.0.0.0/0        to:10.4.22.2
5      0     0 SNAT       all  --  eth2    10.7.22.0/24   0.0.0.0/0        to:10.4.22.2
6      0     0 SNAT       all  --  eth2    10.8.22.0/24   0.0.0.0/0        to:10.4.22.2
Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1     13  557 ACCEPT     all  --  *      *      10.1.22.0/24   0.0.0.0/0
2      0     0 ACCEPT     all  --  *      *      10.6.22.0/24   0.0.0.0/0
3      0     0 ACCEPT     all  --  *      *      10.7.22.0/24   0.0.0.0/0
4      0     0 ACCEPT     all  --  *      *      10.8.22.0/24   0.0.0.0/0
Chain FORWARD (policy DROP 9 packets, 493 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      2   114 ACCEPT     all  --  eth0   eth2    0.0.0.0/0      0.0.0.0/0
2      2   165 ACCEPT     all  --  eth2   eth0    0.0.0.0/0      0.0.0.0/0        state RELATED,ESTABL
ISHED
3      1   48 ACCEPT     all  --  eth1   eth2    0.0.0.0/0      0.0.0.0/0        state RELATED,ESTABL
4      0     0 ACCEPT     all  --  eth1   eth0    0.0.0.0/0      0.0.0.0/0        state RELATED,ESTABL
ISHED
5      2   105 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0      10.5.22.2      udp dpt:7
6      0     0 ACCEPT     tcp  --  eth2   eth1    0.0.0.0/0      10.5.22.3      tcp dpt:13
7      0     0 ACCEPT     tcp  --  eth1   eth2    10.5.22.3      0.0.0.0/0      tcp spt:13
8      0     0 ACCEPT     tcp  --  eth0   *      10.7.22.3      10.5.22.3      tcp dpt:23
9      0     0 ACCEPT     tcp  --  eth0   *      10.7.22.3      10.5.22.2      tcp dpt:7
10     0     0 DROP       all  --  *      *      10.5.22.2      10.7.22.0/24
11     0     0 DROP       all  --  *      *      10.5.22.3      10.7.22.0/24
12     0     0 DROP       all  --  *      *      10.5.22.2      10.8.22.0/24
13     0     0 DROP       all  --  *      *      10.5.22.3      10.8.22.0/24
14     0     0 DROP       all  --  *      *      10.5.22.2      10.6.22.0/24
15     0     0 DROP       all  --  *      *      10.5.22.3      10.6.22.0/24
Chain OUTPUT (policy ACCEPT 7 packets, 487 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~#
```

-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 maquina donde esta instalado.

### Se esta levantando el servicio DAYTIME el cual devolverá la hora y fecha del server.



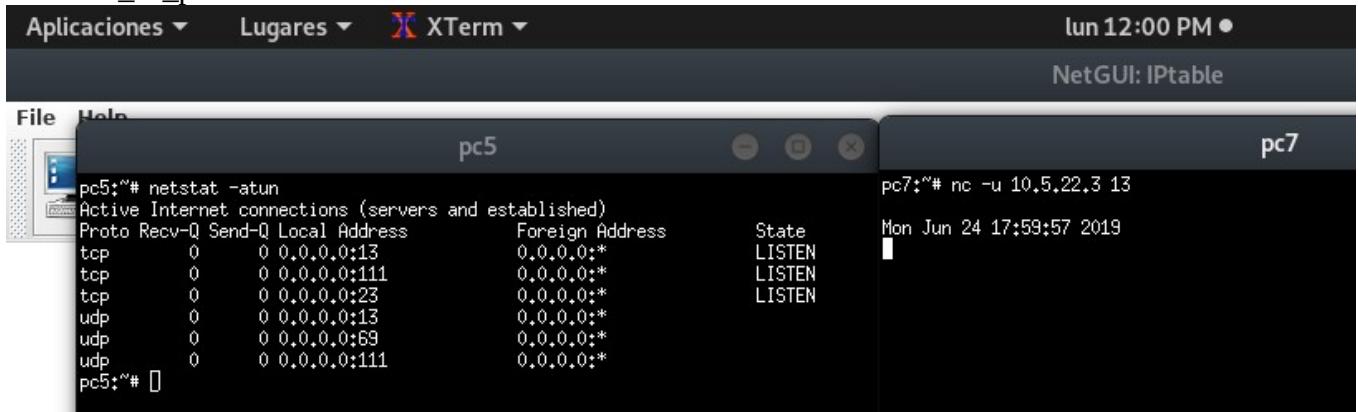
```
pc5:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:111              0.0.0.0:*              LISTEN
udp      0      0 0.0.0.0:111              0.0.0.0:*              LISTEN
pc5:~# /etc/init.d/inetd
Usage: /etc/init.d/openbsd-inetd {start|stop|reload|force-reload|restart}
pc5:~# /etc/init.d/inetd restart
Restarting internet superserver: inetd.
pc5:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:13               0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:111              0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:23               0.0.0.0:*              LISTEN
udp      0      0 0.0.0.0:13               0.0.0.0:*              LISTEN
udp      0      0 0.0.0.0:69               0.0.0.0:*              LISTEN
udp      0      0 0.0.0.0:111              0.0.0.0:*              LISTEN
pc5:~#
```

### El servidor DAYTIME del puerto 13 en PC5.

#### Pruebas

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

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



```
pc5:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:13               0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:111              0.0.0.0:*              LISTEN
tcp      0      0 0.0.0.0:23               0.0.0.0:*              LISTEN
udp      0      0 0.0.0.0:13               0.0.0.0:*              LISTEN
udp      0      0 0.0.0.0:69               0.0.0.0:*              LISTEN
udp      0      0 0.0.0.0:111              0.0.0.0:*              LISTEN
pc5:~#
```

```
pc7:~# nc -u 10.5.22.3 13
Mon Jun 24 17:59:57 2019
```

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

Asegurate 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 numero de veces.

**En este caso no se ejecutan reglas en la tabla NAT, esto se debe a que en la sección donde se natea solo se específico que se le aplicara a las PC de las redes privadas.las políticas por defecto que se ejecutan en las cadenas de las tablas nat y filter y el numero de veces que se ejecutan.**

```
Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 12:02 PM ●
firewall

firewall:~# ifconfig
firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 6 packets, 329 bytes)
num  pkts bytes target     prot opt in     out    source         destination
Chain POSTROUTING (policy ACCEPT 5 packets, 269 bytes)
num  pkts bytes target    prot opt in     out    source         destination
1      0     0 SNAT       all   --  *      eth2   10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
2      0     0 SNAT       all   --  *      eth2   10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
3      0     0 SNAT       all   --  *      eth2   10.8.22.0/24  0.0.0.0/0          to:10.4.22.2
4      0     0 SNAT       all   --  *      eth2   10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
5      0     0 SNAT       all   --  *      eth2   10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
6      0     0 SNAT       all   --  *      eth2   10.8.22.0/24  0.0.0.0/0          to:10.4.22.2

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out    source         destination
1      0     0 ACCEPT     all   --  *      *      10.1.22.0/24  0.0.0.0/0
2      0     0 ACCEPT     all   --  *      *      10.2.22.0/24  0.0.0.0/0
3      0     0 ACCEPT     all   --  *      *      10.3.22.0/24  0.0.0.0/0
4      0     0 ACCEPT     all   --  *      *      10.6.22.0/24  0.0.0.0/0
5      0     0 ACCEPT     all   --  *      *      10.7.22.0/24  0.0.0.0/0
6      0     0 ACCEPT     all   --  *      *      10.8.22.0/24  0.0.0.0/0

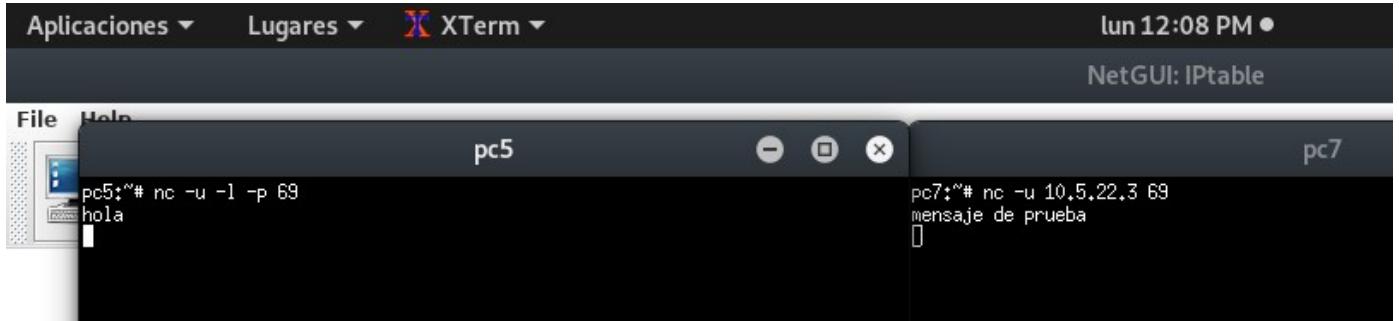
Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target    prot opt in     out    source         destination
1      0     0 ACCEPT     all   --  eth0   eth2   0.0.0.0/0    0.0.0.0/0
2      0     0 ACCEPT     all   --  eth2   eth0   0.0.0.0/0    0.0.0.0/0          state RELATED,ESTABLISHED
3      1     54 ACCEPT     all   --  eth1   eth2   0.0.0.0/0    0.0.0.0/0          state RELATED,ESTABLISHED
4      0     0 ACCEPT     all   --  eth1   eth0   0.0.0.0/0    0.0.0.0/0          state RELATED,ESTABLISHED
5      0     0 ACCEPT     udp   --  eth2   eth1   0.0.0.0/0    10.5.22.2    udp dpt:7
6      1     29 ACCEPT     udp   --  eth2   eth1   0.0.0.0/0    10.5.22.3    udp dpt:13
7      0     0 ACCEPT     tcp   --  eth0   *      10.7.22.3   10.5.22.3    tcp dpt:23
8      0     0 ACCEPT     tcp   --  eth0   *      10.7.22.3   10.5.22.2    tcp dpt:7
9      0     0 DROP       all   --  *      *      10.5.22.2   10.7.22.0/24
10     0     0 DROP       all   --  *      *      10.5.22.3   10.7.22.0/24
11     0     0 DROP       all   --  *      *      10.5.22.2   10.8.22.0/24
12     0     0 DROP       all   --  *      *      10.5.22.3   10.8.22.0/24
13     0     0 DROP       all   --  *      *      10.5.22.2   10.6.22.0/24
14     0     0 DROP       all   --  *      *      10.5.22.3   10.6.22.0/24

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

En esta captura se puede ver que se están aplicando dos reglas, se están aceptando los paquetes TCP que entran por la interfaz eth2 y salen por la interfaz eth1, que tiene como destino la IP 10.5.22.3 al puerto 13. También se puede ver otra regla ejecutada y es la numero 3, esta indica que esta permitiendo el reenvío de paquetes entrantes que pertenecen a una conexión ya existente.

- a) 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 comunique 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 qué prueba estás haciendo. Asegúrate de que antes de lanzar el cliente has ejecutado fw3.sh para que reinicie los contadores de iptables.



**Estoy lanzando un servidor UDP en la PC5, la cual esta en la zona DMZ. Y lanzo un cliente desde internet. Los mensajes que se mandaron no obtuvieron respuesta.**

Explica en la memoria:

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

```
Aplicaciones ▾ Lugares ▾ XTerm ▾ lun 12:09 PM •
NetGUI: IPtable

File Help
pc5 pc7
pc5:~# nc -u -l -p 69
holo
pc7:~# nc -u 10.5.22.3 69
mensaje de prueba
[]

firewall
firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 7 packets, 376 bytes)
num pkts bytes target prot opt in out source destination
Chain POSTROUTING (policy ACCEPT 5 packets, 269 bytes)
num pkts bytes target prot opt in out source destination
1 0 0 SNAT all -- * eth2 10.6.22.0/24 0.0.0.0/0 to:10.4.22.2
2 0 0 SNAT all -- * eth2 10.7.22.0/24 0.0.0.0/0 to:10.4.22.2
3 0 0 SNAT all -- * eth2 10.8.22.0/24 0.0.0.0/0 to:10.4.22.2
4 0 0 SNAT all -- * eth2 10.6.22.0/24 0.0.0.0/0 to:10.4.22.2
5 0 0 SNAT all -- * eth2 10.7.22.0/24 0.0.0.0/0 to:10.4.22.2
6 0 0 SNAT all -- * eth2 10.8.22.0/24 0.0.0.0/0 to:10.4.22.2

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num pkts bytes target prot opt in out source destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num pkts bytes target prot opt in out source destination
1 0 0 ACCEPT all -- * * 10.1.22.0/24 0.0.0.0/0
2 0 0 ACCEPT all -- * * 10.2.22.0/24 0.0.0.0/0
3 0 0 ACCEPT all -- * * 10.3.22.0/24 0.0.0.0/0
4 0 0 ACCEPT all -- * * 10.6.22.0/24 0.0.0.0/0
5 0 0 ACCEPT all -- * * 10.7.22.0/24 0.0.0.0/0
6 0 0 ACCEPT all -- * * 10.8.22.0/24 0.0.0.0/0

Chain FORWARD (policy DROP 1 packets, 47 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 state RELATED,ESTABLISHED
2 0 0 ACCEPT all -- eth2 eth0 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
3 1 54 ACCEPT all -- eth1 eth2 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
4 0 0 ACCEPT all -- eth1 eth0 0.0.0.0/0 0.0.0.0/0 state RELATED,ESTABLISHED
5 0 0 ACCEPT udp -- eth2 eth1 0.0.0.0/0 10.5.22.2 udp dpt:7
6 1 29 ACCEPT udp -- eth2 eth1 0.0.0.0/0 10.5.22.3 udp dpt:13
7 0 0 ACCEPT tcp -- eth0 * 10.7.22.3 10.5.22.3 tcp dpt:23
8 0 0 ACCEPT tcp -- eth0 * 10.7.22.3 10.5.22.2 tcp dpt:7
9 0 0 DROP all -- * * 10.5.22.2 10.7.22.0/24
10 0 0 DROP all -- * * 10.5.22.3 10.7.22.0/24
11 0 0 DROP all -- * * 10.5.22.2 10.8.22.0/24
12 0 0 DROP all -- * * 10.5.22.3 10.8.22.0/24
13 0 0 DROP all -- * * 10.5.22.2 10.6.22.0/24
14 0 0 DROP all -- * * 10.5.22.3 10.6.22.0/24

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

**En la tabla nat no se ejecutaron reglas.**

**En la tabla filter se puede ver en la cadena FORWARD que se ha ejecutado una política DROP, acá esta rechazando los paquetes que entran al Firewall y que van dirigido a otra maquina. Con esto se puede comprobar que el trafico hacia este servidor no esta autorizado.**

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.

The screenshot shows a desktop environment with several windows. At the top, there's a menu bar with "Aplicaciones", "Lugares", and "XTerm". The system tray shows the date "mar 10:00" and "NetGUI: II".

**Terminal pc1:**

```
pc1:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:111              0.0.0.0:*
tcp      0      0 0.0.0.0:23               0.0.0.0:*
udp     0      0 0.0.0.0:69               0.0.0.0:*
udp     0      0 0.0.0.0:111              0.0.0.0:*
```

**Terminal pc5:**

```
pc5:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:111              0.0.0.0:*
tcp      0      0 0.0.0.0:111              0.0.0.0:*
pc5:~# /etc/init.d/inetd restart
Restarting internet superserver: inetd.
pc5:~# netstat -atun
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 0.0.0.0:13               0.0.0.0:*
tcp      0      0 0.0.0.0:111              0.0.0.0:*
tcp      0      0 0.0.0.0:23               0.0.0.0:*
udp     0      0 0.0.0.0:13               0.0.0.0:*
udp     0      0 0.0.0.0:69               0.0.0.0:*
udp     0      0 0.0.0.0:111              0.0.0.0:*
```

**Terminal firewall:**

```
firewall:~# ./fw3.sh
firewall:~#
```

A vertical line on the right side of the screen has labels: "10.5.22.1" at the bottom, "eth1" above it, and "firewall" at the top.

Pruebas

- 1) Asegurate 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>

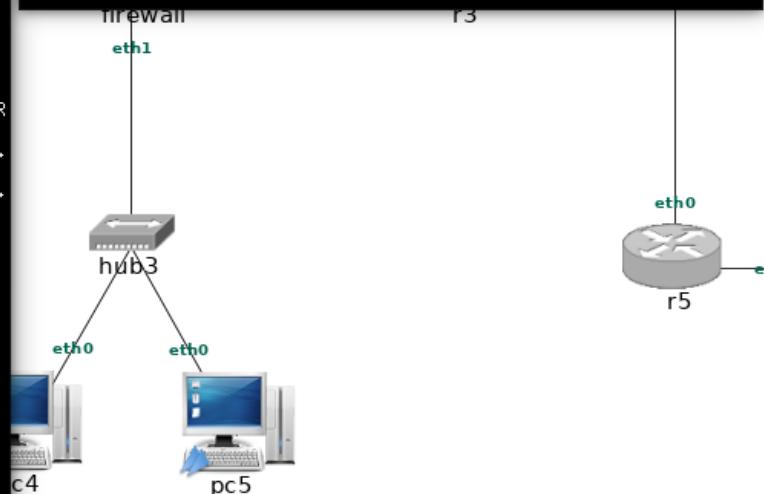
The figure shows a Linux desktop environment with two terminal windows and a network diagram.

**Terminal Windows:**

- Terminal 1 (pc5):** Shows a telnet session to 10.5.22.3 and a log of kernel boot messages from Jun 25 16:29:15 to Jun 25 16:32:08. The log includes messages about mounting filesystems (ext2, e2fsck), networking (eth0, eth1), and services (inetd, bind9, named).
- Terminal 2 (pc5):** Shows commands to restart the inetd and bind9 services, followed by a failed connection attempt to bind9rndc and the start of the domain name service.

**Network Diagram:**

- A central **firewall** device is connected to **eth1**.
- eth1** is connected to a **hub3**.
- hub3** has two **eth0** ports, each connected to a computer labeled **pc4** and **pc5**.
- pc5** also has an **eth0** port connected to a **r5** router.



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 numero de veces.

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

- **En la tabla nat no se ejecuta ninguna regla, ya que la nat configurada esta especifica para cambiar las IPs de la zona privada con dirección a la zona internet.**
- **En la tabla filter se puede ver que en la cadena OUTPUT se ejecuta una política por defecto ACCEPT, estos son los paquetes que están saliendo del Firewall. En la cadena FORWARD se ejecuta la regla ACCEPT, en una de las ejecuciones se puede ver que no se especifica el protocolo, la interfaz de entrada es la eth1 y la de salida es la eth0 y en este caso se puede ver que el reenvío de paquetes para esta conexión esta permitido. La regla ACCEPT numero 8 me indica la conexión al puerto 23 telnet la cual va de la PC origen 10.7.22.3 a la PC destino 10.5.22.3**

```
Aplicaciones ▾ Lugares ▾ XTerm ▾ mar 10:40 AM ●
firewall

firewall:~# iptables -t nat -L -v -n
Chain PREROUTING (policy ACCEPT 2 packets, 120 bytes)
pkts bytes target prot opt in     out      source          destination
Chain POSTROUTING (policy ACCEPT 6 packets, 360 bytes)
pkts bytes target prot opt in     out      source          destination
  0    0 SNAT    all  -- *      eth2    10.6.22.0/24   0.0.0.0/0      to:10.4.22.2
  0    0 SNAT    all  -- *      eth2    10.7.22.0/24   0.0.0.0/0      to:10.4.22.2
  0    0 SNAT    all  -- *      eth2    10.8.22.0/24   0.0.0.0/0      to:10.4.22.2

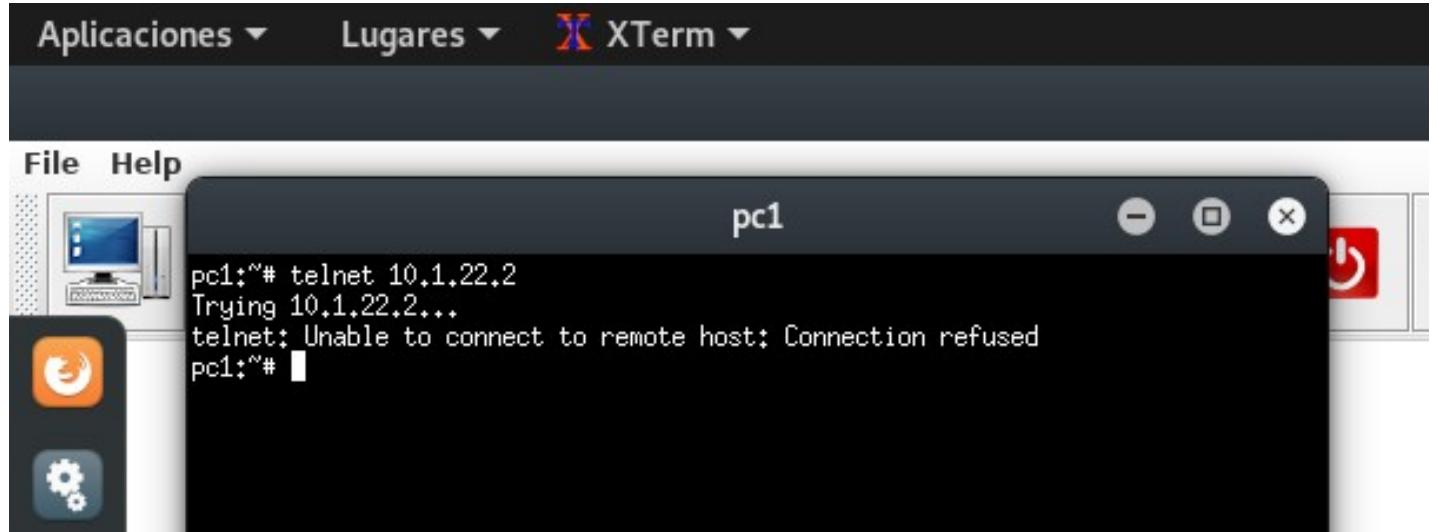
Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
pkts bytes target prot opt in     out      source          destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target prot opt in     out      source          destination
1    0    0 ACCEPT  all  -- *      *      10.1.22.0/24   0.0.0.0/0
2    0    0 ACCEPT  all  -- *      *      10.2.22.0/24   0.0.0.0/0
3    0    0 ACCEPT  all  -- *      *      10.5.22.0/24   0.0.0.0/0
4    0    0 ACCEPT  all  -- *      *      10.6.22.0/24   0.0.0.0/0
5    0    0 ACCEPT  all  -- *      *      10.7.22.0/24   0.0.0.0/0
6    0    0 ACCEPT  all  -- *      *      10.8.22.0/24   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  all  -- eth1   eth2    0.0.0.0/0      0.0.0.0/0
4    31   1802 ACCEPT  all  -- eth1   eth0    0.0.0.0/0      0.0.0.0/0
5    0    0 ACCEPT  udp   -- eth2   eth1    0.0.0.0/0      10.5.22.2
6    0    0 ACCEPT  udp   -- eth2   eth1    0.0.0.0/0      10.5.22.3
7    41   2234 ACCEPT  tcp   -- eth0   *      10.7.22.3    10.5.22.3
8    0    0 ACCEPT  tcp   -- eth0   *      10.7.22.3    10.5.22.2
9    0    0 DROP    all  -- *      *      10.5.22.2    10.7.22.0/24
10   0    0 DROP    all  -- *      *      10.5.22.3    10.7.22.0/24
11   0    0 DROP    all  -- *      *      10.5.22.2    10.8.22.0/24
12   0    0 DROP    all  -- *      *      10.5.22.3    10.8.22.0/24
13   0    0 DROP    all  -- *      *      10.5.22.2    10.6.22.0/24
14   0    0 DROP    all  -- *      *      10.5.22.3    10.6.22.0/24

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 maquina de

- Internet no debería permitir la conexión.
- Haz una prueba para este tipo de tráfico y explica que prueba estas haciendo. Asegurate de que antes de
- lanzar el cliente has ejecutado fw3.sh para que reinicie los contadores de iptables.



En esta prueba estoy tratando de hacer una conexión telnet a la PC6. Se puede ver un mensaje, “no se puede conectar el host remoto. No hay ruta para host.”

Explica en la memoria:

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

```
firewall:~# iptables -t nat -L -v --line-numbers
Chain PREROUTING (policy ACCEPT 3 packets, 180 bytes)
num  pkts bytes target     prot opt in   out    source         destination
    0    0   0 SNAT      all -- *    eth2  10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
    1    1   60 SNAT      all -- *    eth2  10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
    3    0   0 SNAT      all -- *    eth2  10.8.22.0/24  0.0.0.0/0          to:10.4.22.2

Chain POSTROUTING (policy ACCEPT 6 packets, 360 bytes)
num  pkts bytes target     prot opt in   out    source         destination
    1    0   0 SNAT      all -- *    eth2  10.1.22.0/24  0.0.0.0/0          to:10.4.22.2
    2    1   60 SNAT      all -- *    eth2  10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
    3    0   0 SNAT      all -- *    eth2  10.8.22.0/24  0.0.0.0/0          to:10.4.22.2

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in   out    source         destination
    0    0   0 ACCEPT    all -- *    *      10.1.22.0/24  0.0.0.0/0
    2    0   0 ACCEPT    all -- *    *      10.2.22.0/24  0.0.0.0/0
    3    0   0 ACCEPT    all -- *    *      10.5.22.0/24  0.0.0.0/0
    4    0   0 ACCEPT    all -- *    *      10.6.22.0/24  0.0.0.0/0
    5    0   0 ACCEPT    all -- *    *      10.7.22.0/24  0.0.0.0/0
    6    0   0 ACCEPT    all -- *    *      10.8.22.0/24  0.0.0.0/0

Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in   out    source         destination
    1    60 ACCEPT    all -- eth0  eth0  0.0.0.0/0          0.0.0.0/0
    2    1   40 ACCEPT    all -- eth2  eth0  0.0.0.0/0          0.0.0.0/0
    3    0   0 ACCEPT    all -- eth1  eth2  0.0.0.0/0          0.0.0.0/0
    4    39  2232 ACCEPT   all -- eth1  eth0  0.0.0.0/0          0.0.0.0/0
    5    0   0 ACCEPT    udp -- eth2  eth1  0.0.0.0/0          10.5.22.2
    6    0   0 ACCEPT    udp -- eth2  eth1  0.0.0.0/0          10.5.22.3
    7    53  2865 ACCEPT   tcp -- eth0  *      10.7.22.3  10.5.22.3
    8    0   0 ACCEPT    tcp -- eth0  *      10.7.22.3  10.5.22.3
    9    0   0 DROP      all -- *    *      10.5.22.2  10.5.22.2
    10   0   0 DROP      all -- *    *      10.5.22.3  10.5.22.2
    11   0   0 DROP      all -- *    *      10.5.22.2  10.5.22.3
    12   0   0 DROP      all -- *    *      10.5.22.3  10.5.22.2
    13   0   0 DROP      all -- *    *      10.5.22.2  10.5.22.3
    14   0   0 DROP      all -- *    *      10.5.22.3  10.5.22.2

Chain FORWARD (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in   out    source         destination
    1    60 ACCEPT    all -- eth0  eth0  0.0.0.0/0          state RELATED,ESTABLISHED
    2    1   40 ACCEPT    all -- eth2  eth0  0.0.0.0/0          state RELATED,ESTABLISHED
    3    0   0 ACCEPT    all -- eth1  eth2  0.0.0.0/0          state RELATED,ESTABLISHED
    4    39  2232 ACCEPT   all -- eth1  eth0  0.0.0.0/0          state RELATED,ESTABLISHED
    5    0   0 ACCEPT    udp -- eth2  eth1  0.0.0.0/0          10.5.22.2
    6    0   0 ACCEPT    udp -- eth2  eth1  0.0.0.0/0          10.5.22.3
    7    53  2865 ACCEPT   tcp -- eth0  *      10.7.22.3  10.5.22.3
    8    0   0 ACCEPT    tcp -- eth0  *      10.7.22.3  10.5.22.3
    9    0   0 DROP      all -- *    *      10.5.22.2  10.5.22.2
    10   0   0 DROP      all -- *    *      10.5.22.3  10.5.22.2
    11   0   0 DROP      all -- *    *      10.5.22.2  10.5.22.3
    12   0   0 DROP      all -- *    *      10.5.22.3  10.5.22.2
    13   0   0 DROP      all -- *    *      10.5.22.2  10.5.22.3
    14   0   0 DROP      all -- *    *      10.5.22.3  10.5.22.2

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 comunique 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

**El servidor ECHO(TCP, puerto 7) esta corriendo.**

**He lanzado ip\_conntrack para ver la información de la conexión.**

**Los mensajes que he mandado de la PC1 al server ECHO en PC4.**

The screenshot shows a desktop environment with three windows:

- Terminal window (pc4):** Shows the command `netstat -an` output, which lists several listening ports on the machine, including port 7 (tcp 0.0.0.0:7).
- Terminal window (pc1):** Shows the command `nc 10.5.22.2 7` being run, followed by a series of messages exchanged between pc1 and pc4. The messages include "holo , server de echo" and "holo, estamos conectados al servidor de echo , devuelve todo lo que estemos escribiendo".
- Firewall window:** Shows the command `cat /proc/net/ip\_conntrack` running every 0.5s. The log displays a single entry for a TCP connection: "tcp 6 431987 ESTABLISHED src=10.7.22.3 dst=10.5.22.2 sport=48545 dport=7 packets=8 bytes=539 src=10.5.22.2 dst=10.7.22.3 sport=7 dport=48545 packets=5 bytes=383 [ASSURED] mark=0 use=1".

La conexión luego que el cliente interrumpió la conexión con el server ECHO en PC4.

En la tabla NAT no se ha aplicado ninguna regla. En la tabla filter se puede ver que se ejecutaron dos reglas. En la linea 4 se ve que el reenvío de paquetes para la conexión ha sido establecida. En la linea 9 que se esta accediendo al servidor ECHO.

```
Aplicaciones ▾ Lugares ▾ XTerm ▾ mar 11:06 AM •
firewall

firewall:~# iptables -t nat -L -v -n --line-numbers
Chain PREROUTING (policy ACCEPT 10 packets, 555 bytes)
num  pkts bytes target     prot opt in     out    source         destination
Chain POSTROUTING (policy ACCEPT 11 packets, 660 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      0     0 SNAT       all  --  eth2   10.6.22.0/24  0.0.0.0/0          to:10.4.22.2
2      1     60 SNAT      all  --  eth2   10.7.22.0/24  0.0.0.0/0          to:10.4.22.2
3      0     0 SNAT      all  --  eth2   10.8.22.0/24  0.0.0.0/0          to:10.4.22.2

Chain OUTPUT (policy ACCEPT 4 packets, 240 bytes)
num  pkts bytes target     prot opt in     out    source         destination
firewall:~# iptables -t filter -L -v -n --line-numbers
Chain INPUT (policy DROP 0 packets, 0 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      0     0 ACCEPT    all  --  *      *      10.1.22.0/24  0.0.0.0/0
2      0     0 ACCEPT    all  --  *      *      10.2.22.0/24  0.0.0.0/0
3      0     0 ACCEPT    all  --  *      *      10.5.22.0/24  0.0.0.0/0
4      0     0 ACCEPT    all  --  *      *      10.6.22.0/24  0.0.0.0/0
5      0     0 ACCEPT    all  --  *      *      10.7.22.0/24  0.0.0.0/0
6      0     0 ACCEPT    all  --  *      *      10.8.22.0/24  0.0.0.0/0

Chain FORWARD (policy DROP 2 packets, 75 bytes)
num  pkts bytes target     prot opt in     out    source         destination
1      1     60 ACCEPT    all  --  eth0   eth2   0.0.0.0/0    0.0.0.0/0
2      1     40 ACCEPT    all  --  eth2   eth0   0.0.0.0/0    0.0.0.0/0
3      0     0 ACCEPT    all  --  eth1   eth2   0.0.0.0/0    0.0.0.0/0
4     48    2775 ACCEPT   all  --  eth1   eth0   0.0.0.0/0    0.0.0.0/0
5      0     0 ACCEPT    udp  --  eth2   eth1   0.0.0.0/0    10.5.22.2
6      0     0 ACCEPT    udp  --  eth2   eth1   0.0.0.0/0    10.5.22.3
7     53    2864 ACCEPT   tcp  --  eth0   *      10.7.22.3   10.5.22.3
8     12    779 ACCEPT   tcp  --  eth0   *      10.7.22.3   10.5.22.2
9      0     0 DROP      all  --  *      *      10.5.22.2   10.7.22.0/24
10     0     0 DROP      all  --  *      *      10.5.22.3   10.7.22.0/24
11     0     0 DROP      all  --  *      *      10.5.22.2   10.8.22.0/24
12     0     0 DROP      all  --  *      *      10.5.22.3   10.8.22.0/24
13     0     0 DROP      all  --  *      *      10.5.22.2   10.6.22.0/24
14     0     0 DROP      all  --  *      *      10.5.22.3   10.6.22.0/24

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

Script completo.

#a)

#borrar la tabla filter, su contenido y reiniciar sus contradores.

iptables -t filter -F

iptables -t filter -Z

#b)

#politicas por defecto descarta todo trafico al Firewall y acepta el trafico saliente

iptables -t filter -P INPUT DROP

iptables -t filter -P FORWARD DROP

iptables -t filter -P OUTPUT ACCEPT

#partiendo del script fw1.sh

#lo que se hizo en el script fw1.sh, reglas para aplicar nateo en el Firewall.

iptables -t nat -A POSTROUTING -s 10.6.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2

iptables -t nat -A POSTROUTING -s 10.7.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2

iptables -t nat -A POSTROUTING -s 10.8.22.0/24 -o eth2 -j SNAT --to-source 10.4.22.2

#c)

#con esto se esta permitiendo el trafico dirigido a las app que estan ejecutando den el Firewall, tienen que provenir de las subredes privadas.

iptables -t filter -A INPUT -s 10.1.22.0/24 -j ACCEPT

iptables -t filter -A INPUT -s 10.2.22.0/24 -j ACCEPT

iptables -t filter -A INPUT -s 10.3.22.0/24 -j ACCEPT

iptables -t filter -A INPUT -s 10.6.22.0/24 -j ACCEPT

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

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

#d)

#se permite el trafico saliente subres privada hacia internet:SNAT en scrip fw1.sh

#permite el reenvio de todos los paquetes que recibe el router atraves de la interfaz

#eth0 para que se envie por la interfaz eth2

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

#permite el reenvio paquetes entrantes que pertenecen a una conexion ya establecida

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

#antes ya se habia establecido la conexion de la red privada a internet->

iptables -t filter -A FORWARD -i eth1 -o eth2 -m state --state RELATED,ESTABLISHED -j ACCEPT

#antes ya se habia establecido la conexion de la red internet a DMZ->

iptables -t filter -A FORWARD -i eth1 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT

#antes ya se habia establecido la conexion de la red privada a DMZ<-

#e)

#permitir trafico de internet hacia la DMZ cuando se dirige a un puerto y pc destino.

#el server ECHO instalado en PC4(UDP,port 7)

#en la tabla filter le aplico la cadena FORWARD(paquetes que entran al Firewall pero que van destinado a otra maquina.se ejecuta antes de consultar la tabla de encaminamiento)

iptables -t filter -A FORWARD -i eth2 -d 10.5.22.2 -o eth1 -p udp --dport 7 -j ACCEPT

#el server daytime instalado en PC5(TCP,port 13)

iptables -t filter -A FORWARD -i eth2 -d 10.5.22.3 -o eth1 -p udp --dport 13 -j ACCEPT

#f)

#permitir conexion de la red privada a la zona DMZ:

#conexion telnet(TCP, puerto 13) desde PC1 a PC5

iptables -t filter -A FORWARD -i eth0 -s 10.7.22.3 -d 10.5.22.3 -p tcp --dport 23 -j ACCEPT

iptables -t filter -A FORWARD -i eth2 -s 10.1.22.2 -d 10.5.22.3 -p tcp --dport 23 -j ACCEPT

#conexion ECHO(TCP,port 7) desde PC1 a PC4

iptables -t filter -A FORWARD -i eth0 -s 10.7.22.3 -d 10.5.22.2 -p tcp --dport 7 -j ACCEPT

#g)

#eliminando comunicacion de la zona DMZ a la red privada y al Firewall.

iptables -t filter -A FORWARD -s 10.5.22.2 -d 10.7.22.0/24 -j DROP

iptables -t filter -A FORWARD -s 10.5.22.3 -d 10.7.22.0/24 -j DROP

iptables -t filter -A FORWARD -s 10.5.22.2 -d 10.8.22.0/24 -j DROP

iptables -t filter -A FORWARD -s 10.5.22.3 -d 10.8.22.0/24 -j DROP

iptables -t filter -A FORWARD -s 10.5.22.2 -d 10.6.22.0/24 -j DROP

iptables -t filter -A FORWARD -s 10.5.22.3 -d 10.6.22.0/24 -j DROP