ALBERTO LEÓN LUENGO

MEMORIA PRÁCTICA 3: NAT Y CORTAFUEGOS (FIREWALLS)

ÍNDICE DE CONTENIDOS

[] 1. INTRODUCCIÓN

A continuación se proporcionan algunos consejos para facilitar la realización de la práctica.

} 1.1 EDICIÓN Y EJECUCIÓN DE SCRIPTS

En esta práctica se configurará la máquina firewall para que actúe 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.
Considera la posibilidad de editar y guardar el script en el sistema de ficheros de la máquina real, ejecutándolo desde dentro de la máquina virtual. Así, si tu script fw.sh está almacenado directamente en tu HOME de la máquina real, podrías editarlo en ella con un editor gráfico (por ejemplo, gedit) y luego ejecutarlo en la máquina firewall escribiendo dentro de esa máquina virtual:

/hosthome/fw.sh

} 1.2 COMPROBACIÓN DE LA CONFIGURACIÓN DEL FIREWALL

Durante la práctica frecuentemente tendrás que ir comprobando que el firewall está correctamente configurado, es decir:

- deja pasar el tráfico que está permitido.
- impide el paso del tráfico que debe ser bloqueado.
- realiza la traducción de direcciones IP necesaria para que no aparezcan en Internet paquetes con direcciones privadas.

Para ello deberás emplear la herramienta netcat (nc) (ya utilizada en prácticas del curso pasado) que permite arrancar aplicaciones TCP y UDP en modo cliente o servidor.
El enunciado de la práctica te irá indicando cuándo y en qué máquinas debes lanzar un cliente o un servidor TCP o UDP para ir probando la configuración del firewall. Consulta la documentación adjunta para recordar la sintaxis de netcat.

[] 2. TRADUCCIÓN DE DIRECCIONES Y PUERTOS EN EL FIREWALL (TABLA NAT)

} 2.1 CLIENTES EN LA RED PRIVADA Y SERVIDORES EXTERNOS

Este es el script fw1.sh que cumple con los requisitos solicitados:
COMANDO: firewall:~# nano /hosthome/fw1.sh (mostrar contenido del fichero fw1.sh)

```
#!/bin/bash

# Borra las reglas que hubiese configuradas previamente en la tabla nat
iptables -t nat -F

# Reinicia los contadores de la tabla nat
iptables -t nat -Z

# Realiza la traducción de direcciones para el tráfico saliente de las redes
# privadas (SNAT) y su correspondiente tráfico de respuesta
iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100
```

} 2.1.1 PRUEBAS CON TCP

PREGUNTA 1
COMANDO: firewall:~# /hosthome/fw1.sh (ejecutar script fw1.sh)
COMANDO: r3:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-01.cap (arrancar una captura de tráfico)
COMANDO: firewall:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-02.cap (arrancar una captura de tráfico)
COMANDO: pc6:~# nc -l -p 7777 (arrancar una aplicación servidor TCP)
COMANDO: pc1:~# nc -p 6666 100.211.5.60 7777 (arrancar una aplicación cliente TCP)
COMANDO: firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack (mostrar cambios en /proc/net/ip_conntrack)

```
Every 0.5s: cat /proc/net/ip_conntrack                                                                                                  Sat Feb 24 10:50:30 2024

tcp      6 431914 ESTABLISHED src=10.211.0.10 dst=100.211.5.60 sport=6666 dport=7777 packets=2 bytes=112 src=100.211.5.60 dst=100.211.1.100 sport=7777 dport=6666
packets=1 bytes=60 [ASSURED] mark=0 use=1
```

Como se puede ver, inicialmente se observan 2 paquetes, correspondientes al SYN que envía el cliente al servidor y el SYN,ACK que envía el servidor al cliente.

PREGUNTA 2
COMANDO: pc1:~# hola <enter> (mandar un mensaje al servidor)

```
Every 0.5s: cat /proc/net/ip_conntrack                                                                                                  Sat Feb 24 10:51:41 2024

tcp      6 431962 ESTABLISHED src=10.211.0.10 dst=100.211.5.60 sport=6666 dport=7777 packets=3 bytes=169 src=100.211.5.60 dst=100.211.1.100 sport=7777 dport=6666
packets=2 bytes=112 [ASSURED] mark=0 use=1
```

Una vez ejecutado este comando, se observa un nuevo paquete correspondiente al PSH,ACK que contiene el mensaje que el cliente ha enviado al servidor.

PREGUNTA 3
COMANDO: pc1:~# ctrl+c (interrumpir ejecución del comando watch -n 0.5 cat /proc/net/ip_conntrack)

```
Every 0.5s: cat /proc/net/ip_conntrack                                                                                                  Sat Feb 24 10:52:06 2024

tcp      6 107 TIME_WAIT src=10.211.0.10 dst=100.211.5.60 sport=6666 dport=7777 packets=5 bytes=273 src=100.211.5.60 dst=100.211.1.100 sport=7777 dport=6666 packe
ts=3 bytes=164 [ASSURED] mark=0 use=1
```

Y por último, estos dos nuevos paquetes corresponden al FIN,ACK que el cliente envía al servidor, notificando que ha cerrado la conexión, por lo que el servidor le contesta con el mismo mensaje.

PREGUNTA 4

COMANDO: r3:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: firewall:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: user@f-lX-pcX:~$ wireshark iptables-01.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	100.211.1.100	100.211.5.60	TCP	74	6666 → 7777 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=416107 TSecr=0 WS=2
2	0.000516	100.211.5.60	100.211.1.100	TCP	74	7777 → 6666 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=409427 TSecr=416107
3	0.001498	100.211.1.100	100.211.5.60	TCP	66	6666 → 7777 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=416107 TSecr=409427
4	4.993085	12:92:fa:f2:48:77	4e:d4:93:5f:65:d8	ARP	42	Who has 100.211.1.100? Tell 100.211.1.3
5	4.993357	4e:d4:93:5f:65:d8	12:92:fa:f2:48:77	ARP	42	Who has 100.211.1.3? Tell 100.211.1.100
6	4.993380	12:92:fa:f2:48:77	4e:d4:93:5f:65:d8	ARP	42	100.211.1.3 is at 12:92:fa:f2:48:77
7	4.993402	4e:d4:93:5f:65:d8	12:92:fa:f2:48:77	ARP	42	100.211.1.100 is at 4e:d4:93:5f:65:d8
8	22.464978	100.211.1.100	100.211.5.60	TCP	71	6666 → 7777 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=5 TSval=418361 TSecr=409427
9	22.465242	100.211.5.60	100.211.1.100	TCP	66	7777 → 6666 [ACK] Seq=1 Ack=6 Win=5792 Len=0 TSval=411674 TSecr=418361
10	26.607970	100.211.1.100	100.211.5.60	TCP	66	6666 → 7777 [FIN, ACK] Seq=6 Ack=1 Win=5840 Len=0 TSval=418776 TSecr=411674
11	26.608334	100.211.5.60	100.211.1.100	TCP	66	7777 → 6666 [FIN, ACK] Seq=1 Ack=7 Win=5792 Len=0 TSval=412088 TSecr=418776
12	26.608554	100.211.1.100	100.211.5.60	TCP	66	6666 → 7777 [ACK] Seq=7 Ack=2 Win=5840 Len=0 TSval=418776 TSecr=412088

COMANDO: user@f-lX-pcX:~$ wireshark iptables-02.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.211.0.10	100.211.5.60	TCP	74	6666 → 7777 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=416107 TSecr=0 WS=2
2	0.000994	100.211.5.60	10.211.0.10	TCP	74	7777 → 6666 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=409427 TSecr=416107
3	0.001558	10.211.0.10	100.211.5.60	TCP	66	6666 → 7777 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=416107 TSecr=409427
4	4.993473	5a:00:4f:ec:68:89	26:93:a7:a6:98:1b	ARP	42	Who has 10.211.1.1? Tell 10.211.1.100
5	4.993633	26:93:a7:a6:98:1b	5a:00:4f:ec:68:89	ARP	42	10.211.1.1 is at 26:93:a7:a6:98:1b
6	22.465117	10.211.0.10	100.211.5.60	TCP	71	6666 → 7777 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=5 TSval=418361 TSecr=409427
7	22.465560	100.211.5.60	10.211.0.10	TCP	66	7777 → 6666 [ACK] Seq=1 Ack=6 Win=5792 Len=0 TSval=411674 TSecr=418361
8	26.608156	10.211.0.10	100.211.5.60	TCP	66	6666 → 7777 [FIN, ACK] Seq=6 Ack=1 Win=5840 Len=0 TSval=418776 TSecr=411674
9	26.608615	100.211.5.60	10.211.0.10	TCP	66	7777 → 6666 [FIN, ACK] Seq=1 Ack=7 Win=5792 Len=0 TSval=412088 TSecr=418776
10	26.608759	10.211.0.10	100.211.5.60	TCP	66	6666 → 7777 [ACK] Seq=7 Ack=2 Win=5840 Len=0 TSval=418776 TSecr=412088
11	27.457265	26:93:a7:a6:98:1b	5a:00:4f:ec:68:89	ARP	42	Who has 10.211.1.100? Tell 10.211.1.1
12	27.457294	5a:00:4f:ec:68:89	26:93:a7:a6:98:1b	ARP	42	10.211.1.100 is at 5a:00:4f:ec:68:89

Ambas capturas tienen el mismo número de paquetes, con la única diferencia de los mensajes de ARP, donde pc6 pregunta por pc1, y pc1 pregunta por pc6.

PREGUNTA 5

COMANDO: firewall:~# iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                Sat Feb 24 11:01:23 2024

Chain PREROUTING (policy ACCEPT 2 packets, 120 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    2   120 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

Como se puede ver, la única regla que se está cumpliendo es la misma que se estableció en el script fw1.sh, la cual se cumple 2 veces (una por cada paquete enviado), y en el caso de PREROUTING acepta 2 paquetes:

iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100

PREGUNTA 6

COMANDO: firewall:~# iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                Sat Feb 24 11:01:23 2024

Chain PREROUTING (policy ACCEPT 2 packets, 120 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    2   120 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

Al igual que en la pregunta anterior, la única regla que se está cumpliendo es la misma que se estableció en el script fw1.sh, la cual se cumple 2 veces (una por cada paquete enviado):

iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100

} 2.1.2 PRUEBAS CON UDP

PREGUNTA 1

COMANDO: firewall:~# /hosthome/fw1.sh (ejecutar script fw1.sh)
COMANDO: r3:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-03.cap (arrancar una captura de tráfico)
COMANDO: firewall:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-04.cap (arrancar una captura de tráfico)
COMANDO: pc6:~# nc -u -l -p 7777 (arrancar una aplicación servidor UDP)
COMANDO: pc2:~# nc -u -p 6666 100.211.5.60 7777 (arrancar una aplicación cliente UDP)

a)
COMANDO: firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack (mostrar cambios en /proc/net/ip_conntrack)

```
Every 0.5s: cat /proc/net/ip_conntrack                                  Sat Feb 24 11:44:29 2024
```

Como se puede ver, el fichero /proc/net/ip_conntrack está vacío, ya que en este caso, se está realizando una conexión UDP y no TCP como se había hecho anteriormente.

b)
COMANDO: pc2:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: pc2:~# que <enter> (mandar un mensaje al servidor)
COMANDO: pc2:~# tal <enter> (mandar un mensaje al servidor)
COMANDO: pc2:~# estas <enter> (mandar un mensaje al servidor)
COMANDO: pc2:~# amigo <enter> (mandar un mensaje al servidor)

```
Every 0.5s: cat /proc/net/ip_conntrack                                  Sat Feb 24 11:45:20 2024

udp      17 19 src=10.211.0.20 dst=100.211.5.60 sport=6666 dport=7777 packets=5 bytes=165 [UNREPLIED] src=100.211.5.60 dst=100.211.1.100 sport=7777 dport=6666 pac
kets=0 bytes=0 mark=0 use=1
```

Como se puede ver, aparecen 5 paquetes, correspondientes a los 5 mensajes enviados desde pc2 a pc6.

c)
COMANDO: pc6:~# mio <enter> (mandar un mensaje al servidor)

```
Every 0.5s: cat /proc/net/ip_conntrack                                  Sat Feb 24 11:50:40 2024

udp      17 19 src=10.211.0.20 dst=100.211.5.60 sport=6666 dport=7777 packets=1 bytes=32 [UNREPLIED] src=100.211.5.60 dst=100.211.1.100 sport=7777 dport=6666 pack
ets=0 bytes=0 mark=0 use=1
```

Como se puede ver, aparece un único paquete, correspondientes al mensaje enviado desde pc6 a pc2.

d)
La asociación entre cliente y servidor observada en el fichero /proc/net/ip_conntrack desaparece 30 segundos después de haberse producido el envío.

e)
COMANDO: r3:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: firewall:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: user@f-lX-pcX:~$ wireshark iptables-03.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	4e:d4:93:5f:65:d8	Broadcast	ARP	42	Who has 100.211.1.3? Tell 100.211.1.100
2	0.000450	12:92:fa:f2:48:77	4e:d4:93:5f:65:d8	ARP	42	100.211.1.3 is at 12:92:fa:f2:48:77
3	0.000197	100.211.1.100	100.211.5.60	UDP	47	6666 → 7777 Len=5
4	1.501454	100.211.1.100	100.211.5.60	UDP	46	6666 → 7777 Len=4
5	2.734209	100.211.1.100	100.211.5.60	UDP	46	6666 → 7777 Len=4
6	4.216261	100.211.1.100	100.211.5.60	UDP	48	6666 → 7777 Len=6
7	5.805138	100.211.1.100	100.211.5.60	UDP	48	6666 → 7777 Len=6
8	25.563783	100.211.5.60	100.211.1.100	UDP	46	7777 → 6666 Len=4
9	30.566220	12:92:fa:f2:48:77	4e:d4:93:5f:65:d8	ARP	42	Who has 100.211.1.100? Tell 100.211.1.3
10	30.566403	4e:d4:93:5f:65:d8	12:92:fa:f2:48:77	ARP	42	100.211.1.100 is at 4e:d4:93:5f:65:d8

COMANDO: user@f-lX-pcX:~$ wireshark iptables-04.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	26:93:a7:a6:98:1b	Broadcast	ARP	42	Who has 10.211.1.100? Tell 10.211.1.1
2	0.000449	5a:00:4f:ec:68:89	26:93:a7:a6:98:1b	ARP	42	10.211.1.100 is at 5a:00:4f:ec:68:89
3	0.000193	10.211.0.20	100.211.5.60	UDP	47	6666 → 7777 Len=5
4	1.512792	10.211.0.20	100.211.5.60	UDP	46	6666 → 7777 Len=4
5	2.745473	10.211.0.20	100.211.5.60	UDP	46	6666 → 7777 Len=4
6	4.227491	10.211.0.20	100.211.5.60	UDP	48	6666 → 7777 Len=6
7	5.816541	10.211.0.20	100.211.5.60	UDP	48	6666 → 7777 Len=6
8	25.575507	100.211.5.60	10.211.0.20	UDP	46	7777 → 6666 Len=4
9	30.577628	5a:00:4f:ec:68:89	26:93:a7:a6:98:1b	ARP	42	Who has 10.211.1.1? Tell 10.211.1.100
10	30.577789	26:93:a7:a6:98:1b	5a:00:4f:ec:68:89	ARP	42	10.211.1.1 is at 26:93:a7:a6:98:1b

Ambas capturas tienen el mismo número de paquetes, con la única diferencia de los mensajes de ARP, donde pc6 pregunta por pc2, y pc2 pregunta por pc6, y los mensajes de UDP corresponden con los mensajes intercambiados entre ambas máquinas (paquetes 3, 4, 5, 6, 7 y 8).

PREGUNTA 2

COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                                Sat Feb 24 11:54:52 2024

Chain PREROUTING (policy ACCEPT 2 packets, 65 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1    33 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

Como se puede ver, la única regla que se está cumpliendo es la misma que se estableció en el script fw1.sh, la cual se cumple una vez, y en el caso de PREROUTING acepta 2 paquetes:

iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100

PREGUNTA 3

COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                                Sat Feb 24 11:54:52 2024

Chain PREROUTING (policy ACCEPT 2 packets, 65 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1    33 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

Como se puede ver, la única regla que se está cumpliendo es la misma que se estableció en el script fw1.sh, la cual se cumple una vez, y en el caso de PREROUTING acepta 2 paquetes:

iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100

PREGUNTA 4

COMANDO: r3:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-05.cap (arrancar una captura de tráfico)
COMANDO: firewall:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-06.cap (arrancar una captura de tráfico)
COMANDO: pc7:~# nc -u -l -p 7777 (arrancar una aplicación servidor UDP)
COMANDO: pc1:~# nc -u -p 6666 100.211.6.70 7777 (arrancar una aplicación cliente UDP)
COMANDO: pc2:~# nc -u -p 6666 100.211.6.70 7777 (arrancar una aplicación cliente UDP)
COMANDO: pc1:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: pc2:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: r3:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: firewall:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: user@f-lX-pcX:~$ wireshark iptables-05.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	4e:d4:93:5f:65:d8	Broadcast	ARP	42	Who has 100.211.1.3? Tell 100.211.1.100
2	0.000180	12:92:fa:f2:48:77	4e:d4:93:5f:65:d8	ARP	42	100.211.1.3 is at 12:92:fa:f2:48:77
3	0.000078	100.211.1.100	100.211.6.70	UDP	47	6666 → 7777 Len=5
4	7.008962	100.211.1.100	100.211.6.70	UDP	47	1024 → 7777 Len=5
5	7.009332	100.211.6.70	100.211.1.100	ICMP	75	Destination unreachable (Port unreachable)
6	12.010955	12:92:fa:f2:48:77	4e:d4:93:5f:65:d8	ARP	42	Who has 100.211.1.100? Tell 100.211.1.3
7	12.011351	4e:d4:93:5f:65:d8	12:92:fa:f2:48:77	ARP	42	100.211.1.100 is at 4e:d4:93:5f:65:d8

COMANDO: user@f-lX-pcX:~$ wireshark iptables-06.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	26:93:a7:a6:98:1b	Broadcast	ARP	42	Who has 10.211.1.100? Tell 10.211.1.1
2	0.000072	5a:00:4f:ec:68:89	26:93:a7:a6:98:1b	ARP	42	10.211.1.100 is at 5a:00:4f:ec:68:89
3	0.000099	10.211.0.10	100.211.6.70	UDP	47	6666 → 7777 Len=5
4	7.019231	10.211.0.20	100.211.6.70	UDP	47	6666 → 7777 Len=5
5	7.019819	100.211.6.70	10.211.0.20	ICMP	75	Destination unreachable (Port unreachable)
6	12.021272	5a:00:4f:ec:68:89	26:93:a7:a6:98:1b	ARP	42	Who has 10.211.1.1? Tell 10.211.1.100
7	12.021627	26:93:a7:a6:98:1b	5a:00:4f:ec:68:89	ARP	42	10.211.1.1 is at 26:93:a7:a6:98:1b

Ambas capturas tienen el mismo número de paquetes, con la única diferencia de los mensajes de ARP, donde pc6 pregunta por pc1 y pc2, mientras que pc1 y pc2 preguntan por pc6, y los mensajes de UDP corresponden con los mensajes intercambiados entre ambas máquinas (paquetes 3 y 4), con la única diferencia que en la primera captura, el primer mensaje de UDP se envía desde el puerto 6666 y el segundo se envía desde el puerto 1024, mientras que en la segunda captura, los dos mensajes de UDP se envían desde el mismo puerto (6666), para después notificar que no se ha podido alcanzar dicho puerto, ya que el servidor sólo puede comunicarse con un cliente a la vez.

} 2.1.3 PRUEBAS CON ICMP

PREGUNTA 1

COMANDO: firewall:~# /hosthome/fw1.sh (ejecutar script fw1.sh)
COMANDO: pc6:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-07.cap (arrancar una captura de tráfico)
COMANDO: r1:~# tcpdump -i eth1 -s 0 -w /hosthome/iptables-08.cap (arrancar una captura de tráfico)

PREGUNTA 2

COMANDO: pc1:~# ping -c 2 100.211.5.60 (realizar un ping de 2 paquetes)

PREGUNTA 3

COMANDO: pc6:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: r1:~# ctrl+c (interrumpir una captura de tráfico)

PREGUNTA 4

COMANDO: firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack (mostrar cambios en /proc/net/ip_conntrack)

```
Every 0.5s: cat /proc/net/ip_conntrack                                  Sat Feb 24 12:58:21 2024
```

Como se puede ver, el fichero /proc/net/ip_conntrack está vacío.

PREGUNTA 5

COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                    Sat Feb 24 12:59:30 2024

Chain PREROUTING (policy ACCEPT 2 packets, 168 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    2   168 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

Como se puede ver, la única regla que se está cumpliendo es la misma que se estableció en el script fw1.sh, la cual se cumple 2 veces (una por cada paquete enviado), y en el caso de PREROUTING acepta 2 paquetes:

iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100

COMANDO: user@f-lX-pcX:~$ wireshark iptables-07.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	96:25:97:29:34:9a	Broadcast	ARP	42	Who has 100.211.5.60? Tell 100.211.5.4
2	0.000195	0e:f9:33:1e:bd:cd	96:25:97:29:34:9a	ARP	42	100.211.5.60 is at 0e:f9:33:1e:bd:cd
3	0.000083	100.211.1.100	100.211.5.60	ICMP	98	Echo (ping) request id=0x1f02, seq=1/256, ttl=60 (reply in 4)
4	0.000098	100.211.5.60	100.211.1.100	ICMP	98	Echo (ping) reply id=0x1f02, seq=1/256, ttl=64 (request in 3)
5	0.967946	100.211.1.100	100.211.5.60	ICMP	98	Echo (ping) request id=0x1f02, seq=2/512, ttl=60 (reply in 6)
6	0.967992	100.211.5.60	100.211.1.100	ICMP	98	Echo (ping) reply id=0x1f02, seq=2/512, ttl=64 (request in 5)
7	4.995792	0e:f9:33:1e:bd:cd	96:25:97:29:34:9a	ARP	42	Who has 100.211.5.4? Tell 100.211.5.60
8	4.996065	96:25:97:29:34:9a	0e:f9:33:1e:bd:cd	ARP	42	100.211.5.4 is at 96:25:97:29:34:9a

COMANDO: user@f-lX-pcX:~$ wireshark iptables-08.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ea:2c:7b:fe:13:de	Broadcast	ARP	42	Who has 10.211.0.1? Tell 10.211.0.10
2	0.000095	2a:ec:17:ba:86:98	ea:2c:7b:fe:13:de	ARP	42	10.211.0.1 is at 2a:ec:17:ba:86:98
3	0.000062	10.211.0.10	100.211.5.60	ICMP	98	Echo (ping) request id=0x1f02, seq=1/256, ttl=64 (reply in 4)
4	0.023018	100.211.5.60	10.211.0.10	ICMP	98	Echo (ping) reply id=0x1f02, seq=1/256, ttl=60 (request in 3)
5	0.989574	10.211.0.10	100.211.5.60	ICMP	98	Echo (ping) request id=0x1f02, seq=2/512, ttl=64 (reply in 6)
6	0.991265	100.211.5.60	10.211.0.10	ICMP	98	Echo (ping) reply id=0x1f02, seq=2/512, ttl=60 (request in 5)
7	5.028267	2a:ec:17:ba:86:98	ea:2c:7b:fe:13:de	ARP	42	Who has 10.211.0.10? Tell 10.211.0.1
8	5.028412	ea:2c:7b:fe:13:de	2a:ec:17:ba:86:98	ARP	42	10.211.0.10 is at ea:2c:7b:fe:13:de

Ambas capturas tienen el mismo número de paquetes, con la única diferencia de los mensajes de ARP, donde pc6 pregunta por pc1, mientras que pc1 preguntan por pc6, y los mensajes de ICMP corresponden a los mensajes de request y reply de cada uno de los paquetes enviados en el ping.

} 2.2 SERVIDORES EN LA RED PRIVADA Y CLIENTES EXTERNOS

} 2.2.1 APERTURA DE PUERTOS TCP

Este es el script fw2.sh que cumple con los requisitos solicitados:

COMANDO: firewall:~# nano /hosthome/fw2.sh (mostrar contenido del fichero fw2.sh)

```
#!/bin/sh

# Apartado 2.2.1

# Borra las reglas que hubiese configuradas previamente en la tabla nat
iptables -t nat -F

# Reinicia los contadores de la tabla nat
iptables -t nat -Z

# El tráfico de entrada al firewall destinado al puerto TCP 80
# es redirigido al puerto 80 de pc3.
iptables -t nat -A PREROUTING -i eth2 -d 100.211.1.100 -p tcp --dport 80 -j DNAT --to-destination 10.211.2.30:80
```

COMANDO: firewall:~# /hosthome/fw2.sh (ejecutar script fw2.sh)
COMANDO: r2:~# tcpdump -i eth1 -s 0 -w /hosthome/iptables-09.cap (arrancar una captura de tráfico)
COMANDO: r4:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-10.cap (arrancar una captura de tráfico)
COMANDO: pc3:~# nc -l -p 80 (arrancar una aplicación servidor TCP)
COMANDO: pc6:~# nc -p 6666 100.211.1.100 80 (arrancar una aplicación cliente TCP)
COMANDO: pc6:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: r2:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: r4:~# ctrl+c (interrumpir una captura de tráfico)

PREGUNTA 1

COMANDO: firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack (mostrar cambios en /proc/net/ip_conntrack)

```
Every 0.5s: cat /proc/net/ip_conntrack                                                    Thu Feb 29 14:58:38 2024

tcp      6 431993 ESTABLISHED src=100.211.5.60 dst=100.211.1.100 sport=6666 dport=80 packets=3 bytes=169 src=10.211.2.30 dst=100.211.5.60 sport=80 dport=6666 packets=2 bytes=112 [ASSURED] mark=0 use=1
```

Como se puede ver, aparecen 3 paquetes, correspondientes al inicio de la comunicación entre pc3 y pc6, el mensaje que envía el cliente al servidor y otro para notificar el cierre de la comunicación.

PREGUNTA 2

COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                      Thu Feb 29 15:00:48 2024

Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1    60 DNAT       tcp  --  eth2   *       0.0.0.0/0            100.211.1.100        tcp dpt:80 to:10.211.2.30:80

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
```

Como se puede ver, la única regla que se está cumpliendo es la misma que se estableció en el script fw2.sh, la cual se cumple una vez en PREROUTING, y en el caso de POSTROUTING acepta 1 paquete:

iptables -t nat -A PREROUTING -i eth2 -d 100.211.1.100 -p tcp --dport 80 -j DNAT --to-destination 10.211.2.30:80

PREGUNTA 3

COMANDO: user@f-IX-pcX:~$ wireshark iptables-09.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	be:42:68:0a:76:76	Broadcast	ARP	42	Who has 10.211.2.30? Tell 10.211.2.2
2	0.000094	9e:1e:a2:fa:26:d4	be:42:68:0a:76:76	ARP	42	10.211.2.30 is at 9e:1e:a2:fa:26:d4
3	0.000097	100.211.5.60	10.211.2.30	TCP	74	6666 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=21945 TSecr=0 WS=2
4	0.000221	10.211.2.30	100.211.5.60	TCP	74	80 → 6666 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=26950 TSecr=21945 WS=2
5	0.000775	100.211.5.60	10.211.2.30	TCP	66	6666 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=21951 TSecr=26950
6	9.125387	100.211.5.60	10.211.2.30	TCP	71	6666 → 80 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=5 TSval=22863 TSecr=26950
7	9.125583	10.211.2.30	100.211.5.60	TCP	66	80 → 6666 [ACK] Seq=1 Ack=6 Win=5792 Len=0 TSval=27863 TSecr=22863
8	35.524874	100.211.5.60	10.211.2.30	TCP	66	6666 → 80 [FIN, ACK] Seq=6 Ack=1 Win=5840 Len=0 TSval=25503 TSecr=27863
9	35.525629	10.211.2.30	100.211.5.60	TCP	66	80 → 6666 [FIN, ACK] Seq=1 Ack=7 Win=5792 Len=0 TSval=30502 TSecr=25503
10	35.527015	100.211.5.60	10.211.2.30	TCP	66	6666 → 80 [ACK] Seq=7 Ack=2 Win=5840 Len=0 TSval=25503 TSecr=30502
11	40.507133	be:42:68:0a:76:76	9e:1e:a2:fa:26:d4	ARP	42	Who has 10.211.2.30? Tell 10.211.2.2
12	40.507340	9e:1e:a2:fa:26:d4	be:42:68:0a:76:76	ARP	42	10.211.2.30 is at 9e:1e:a2:fa:26:d4

COMANDO: user@f-IX-pcX:~$ wireshark iptables-10.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	1a:6c:13:dd:5f:99	Broadcast	ARP	42	Who has 100.211.5.4? Tell 100.211.5.60
2	0.000221	ba:af:64:25:75:25	1a:6c:13:dd:5f:99	ARP	42	100.211.5.4 is at ba:af:64:25:75:25
3	0.000424	100.211.5.60	100.211.1.100	TCP	74	6666 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=21945 TSecr=0 WS=2
4	0.030119	100.211.1.100	100.211.5.60	TCP	74	80 → 6666 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=26950 TSecr=21945 WS=2
5	0.030281	100.211.5.60	100.211.1.100	TCP	66	6666 → 80 [ACK] Seq=1 Ack=1 Win=5840 Len=0 TSval=21951 TSecr=26950
6	5.020515	ba:af:64:25:75:25	1a:6c:13:dd:5f:99	ARP	42	Who has 100.211.5.60? Tell 100.211.5.4
7	5.020674	1a:6c:13:dd:5f:99	ba:af:64:25:75:25	ARP	42	100.211.5.60 is at 1a:6c:13:dd:5f:99
8	9.154698	100.211.5.60	100.211.1.100	TCP	71	6666 → 80 [PSH, ACK] Seq=1 Ack=1 Win=5840 Len=5 TSval=22863 TSecr=26950
9	9.155452	100.211.1.100	100.211.5.60	TCP	66	80 → 6666 [ACK] Seq=1 Ack=6 Win=5792 Len=0 TSval=27863 TSecr=22863
10	35.553882	100.211.5.60	100.211.1.100	TCP	66	6666 → 80 [FIN, ACK] Seq=6 Ack=1 Win=5840 Len=0 TSval=25503 TSecr=27863
11	35.555942	100.211.1.100	100.211.5.60	TCP	66	80 → 6666 [FIN, ACK] Seq=1 Ack=7 Win=5792 Len=0 TSval=30502 TSecr=25503
12	35.556231	100.211.5.60	100.211.1.100	TCP	66	6666 → 80 [ACK] Seq=7 Ack=2 Win=5840 Len=0 TSval=25503 TSecr=30502

Ambas capturas tienen el mismo número de paquetes, con la única diferencia de los mensajes de ARP, donde pc3 pregunta por pc6, y pc6 pregunta por pc3, y los mensajes de TCP corresponden a los mensajes intercambiados entre ambas máquinas indicando el inicio de la comunicación, el envío del mensaje y la finalización de la comunicación, tanto en un sentido como en el otro.

} 2.2.2 APERTURA DE PUERTOS UDP

Estas son las nuevas líneas que se han añadido al script fw2.sh que cumple con los requisitos solicitados:
COMANDO: firewall:~# nano /hosthome/fw2.sh (mostrar contenido del fichero fw2.sh)

```
# Apartado 2.2.2

# El tráfico de entrada al firewall destinado al puerto UDP 5001
# es redirigido al puerto 5001 de pc1
iptables -t nat -A PREROUTING -i eth2 -d 100.211.1.100 -p udp --dport 5001 -j DNAT --to-destination 10.211.0.10:5001

# El tráfico de entrada al firewall destinado al puerto UDP 5002
# es redirigido al puerto 5001 de pc2
iptables -t nat -A PREROUTING -i eth2 -d 100.211.1.100 -p udp --dport 5002 -j DNAT --to-destination 10.211.0.20:5001
```

COMANDO: firewall:~# /hosthome/fw2.sh (ejecutar script fw2.sh)
COMANDO: r1:~# tcpdump -i eth1 -s 0 -w /hosthome/iptables-11.cap (arrancar una captura de tráfico)
COMANDO: r4:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-12.cap (arrancar una captura de tráfico)
COMANDO: r5:~# tcpdump -i eth1 -s 0 -w /hosthome/iptables-13.cap (arrancar una captura de tráfico)
COMANDO: pc1:~# nc -u -l -p 5001 (arrancar una aplicación servidor UDP)
COMANDO: pc2:~# nc -u -l -p 5001 (arrancar una aplicación servidor UDP)
COMANDO: pc6:~# nc -u -p 6666 100.211.1.100 5001 (arrancar una aplicación cliente UDP)
COMANDO: pc7:~# nc -u -p 6666 100.211.1.100 5002 (arrancar una aplicación cliente UDP)
COMANDO: pc6:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: pc7:~# holi <enter> (mandar un mensaje al servidor)
COMANDO: r2:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: r4:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: r5:~# ctrl+c (interrumpir una captura de tráfico)

PREGUNTA 1

COMANDO: firewall:~# watch -n 0.5 cat /proc/net/ip_conntrack (mostrar cambios en /proc/net/ip_conntrack)

```
Every 0.5s: cat /proc/net/ip_conntrack                                                                    Thu Feb 29 15:17:23 2024

udp      17 13 src=100.211.5.60 dst=100.211.1.100 sport=6666 dport=5001 packets=1 bytes=33 [UNREPLIED] src=10.211.0.10 dst=100.211.5.60 sport=5001 dport=6666 pack
ets=0 bytes=0 mark=0 use=1
udp      17 22 src=100.211.6.70 dst=100.211.1.100 sport=6666 dport=5002 packets=1 bytes=33 [UNREPLIED] src=10.211.0.20 dst=100.211.6.70 sport=5001 dport=6666 pack
ets=0 bytes=0 mark=0 use=1
```

Como se puede ver, el fichero /proc/net/ip_conntrack muestra dos líneas, una por cada mensaje que ha enviado cada uno de los clientes, el que envía pc6 a pc1 y el que envía pc7 a pc2, el cual se redirige al puerto 5001, como se indica en las reglas del script fw2.sh.

PREGUNTA 2

COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                                      Thu Feb 29 15:20:59 2024

Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 DNAT       tcp  --  eth2   *       0.0.0.0/0            100.211.1.100        tcp dpt:80 to:10.211.2.30:80
    1    33 DNAT       udp  --  eth2   *       0.0.0.0/0            100.211.1.100        udp dpt:5001 to:10.211.0.10:5001
    1    33 DNAT       udp  --  eth2   *       0.0.0.0/0            100.211.1.100        udp dpt:5002 to:10.211.0.20:5001

Chain POSTROUTING (policy ACCEPT 2 packets, 66 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
```

Como se puede ver, las reglas que se está cumpliendo son las que se establecieron en el script fw2.sh inicialmente, además de las dos nuevas reglas que se han añadido, las cuales se cumplen una vez cada una en PREROUTING, y en el caso de POSTROUTING acepta 2 paquetes:

iptables -t nat -A PREROUTING -i eth2 -d 100.211.1.100 -p udp --dport 5001 -j DNAT --to-destination 10.211.0.10:5001
iptables -t nat -A PREROUTING -i eth2 -d 100.211.1.100 -p udp --dport 5002 -j DNAT --to-destination 10.211.0.20:5001

PREGUNTA 3

COMANDO: user@f-IX-pcX:~$ wireshark iptables-11.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	aa:40:80:95:58:58	Broadcast	ARP	42	Who has 10.211.0.10? Tell 10.211.0.1
2	0.000078	be:a6:e0:0d:99:07	aa:40:80:95:58:58	ARP	42	10.211.0.10 is at be:a6:e0:0d:99:07
3	0.000082	100.211.5.60	10.211.0.10	UDP	47	6666 → 5001 Len=5
4	7.539253	aa:40:80:95:58:58	Broadcast	ARP	42	Who has 10.211.0.20? Tell 10.211.0.1
5	7.539678	aa:e1:bf:54:01:36	aa:40:80:95:58:58	ARP	42	10.211.0.20 is at aa:e1:bf:54:01:36
6	7.539695	100.211.6.70	10.211.0.20	UDP	47	6666 → 5001 Len=5

COMANDO: user@f-IX-pcX:~$ wireshark iptables-12.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	100.211.5.60	100.211.1.100	UDP	47	6666 → 5001 Len=5
2	4.997518	82:d6:f3:92:bf:24	42:40:8d:45:80:15	ARP	42	Who has 100.211.5.4? Tell 100.211.5.60
3	4.997545	42:40:8d:45:80:15	82:d6:f3:92:bf:24	ARP	42	100.211.5.4 is at 42:40:8d:45:80:15

COMANDO: user@f-IX-pcX:~$ wireshark iptables-13.cap (abrir una captura de tráfico en wireshark)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	100.211.6.70	100.211.1.100	UDP	47	6666 → 5002 Len=5
2	5.000465	da:f0:76:79:00:77	96:96:be:4b:5e:3f	ARP	42	Who has 100.211.6.5? Tell 100.211.6.70
3	5.000505	96:96:be:4b:5e:3f	da:f0:76:79:00:77	ARP	42	100.211.6.5 is at 96:96:be:4b:5e:3f

Ambas capturas tienen el mismo número de paquetes, con la única diferencia de los mensajes de ARP, donde pc1 pregunta por pc6 y pc2 pregunta por pc7, pc7 pregunta por pc2 y pc6 pregunta por pc1, y los mensajes de UDP corresponden con los mensajes intercambiados entre ambas máquinas.

[] 3. FILTRADO DE TRÁFICO EN EL FIREWALL (TABLA FILTER)

Este es el script fw1.sh que se tiene inicialmente:

COMANDO: firewall:~# nano /hosthome/fw1.sh (mostrar contenido del fichero fw1.sh)

```
#!/bin/bash

# Borra las reglas que hubiese configuradas previamente en la tabla nat
iptables -t nat -F

# Reinicia los contadores de la tabla nat
iptables -t nat -Z

# Realiza la traducción de direcciones para el tráfico saliente de las redes
# privadas (SNAT) y su correspondiente tráfico de respuesta
iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100
```

Y éstas son las nuevas líneas que se han ido añadiendo a este nuevo script, ahora llamado fw3.sh:

COMANDO: firewall:~# nano /hosthome/fw3.sh (mostrar contenido del fichero fw3.sh)

PREGUNTA 1

```
# PREGUNTA 1

# Borra las reglas que hubiese configuradas previamente en la tabla filter
iptables -t filter -F

# Reinicia los contadores de la tabla nat
iptables -t filter -Z
```

PREGUNTA 2

```
# PREGUNTA 2

# Fija 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 que cree el propio firewall (configuración habitual en un firewall).
iptables -t filter -P INPUT DROP
iptables -t filter -P FORWARD DROP
iptables -t filter -P OUTPUT ACCEPT
```

PREGUNTA 3

```
# PREGUNTA 3

# Permite el tráfico de entrada dirigido a las aplicaciones que se están
# ejecutando en el propio firewall únicamente si este tráfico tiene su
# origen en las subredes privadas de la empresa.
iptables -t filter -A INPUT -i eth0 -j ACCEPT
```

PREGUNTA 4

```
# PREGUNTA 4

# Permite todo el tráfico saliente desde las subredes privadas hacia Internet
# y el tráfico de respuesta al saliente.
iptables -t filter -A FORWARD -i eth0 -o eth2 -j ACCEPT
iptables -t filter -A FORWARD -i eth2 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
```

PREGUNTA 5

```
# PREGUNTA 5

# Permite desde Internet únicamente el tráfico entrante nuevo hacia la zona
# DMZ según las siguientes reglas:

# Acceso a un servidor echo existente en pc4 (UDP, puerto 7), donde el servidor
# de echo es un servidor que al enviarle una cadena de caracteres, devuelve la
# misma cadena que se le ha enviado.
iptables -t filter -A FORWARD -i eth2 -o eth1 -d 100.211.0.40 -p udp --dport 7 -j LOG --log-prefix echo
iptables -t filter -A FORWARD -i eth2 -o eth1 -d 100.211.0.40 -p udp --dport 7 -j ACCEPT

# Acceso a un servidor daytime existente en pc5 (UDP, puerto 13), donde el
# servidor daytime es un servidor que al enviarle algo, devuelve la fecha y
# hora de la máquina donde está instalado.
iptables -t filter -A FORWARD -i eth2 -o eth1 -d 100.211.0.50 -p udp --dport 13 -j LOG --log-prefix daytime
iptables -t filter -A FORWARD -i eth2 -o eth1 -d 100.211.0.50 -p udp --dport 13 -j ACCEPT

# Para este tipo de tráfico configura además reglas/s con acción LOG para que
# cada vez que se permita el tráfico UDP descrito anteriormente, se deje un
# mensaje en el fichero de LOG del sistema.
```

PREGUNTA 6

```
# PREGUNTA 6

# Permite únicamente la comunicación entre la red privada y la zona DMZ de la
# siguiente forma:

# Acceso desde pc1 a un servidor de echo (TCP, puerto 7) existente en pc4.
iptables -t filter -A FORWARD -i eth0 -o eth1 -d 100.211.0.40 -p tcp --dport 7 -j LOG --log-prefix tcp
iptables -t filter -A FORWARD -i eth0 -o eth1 -d 100.211.0.40 -p tcp --dport 7 -j ACCEPT

# Para este tipo de tráfico configura además reglas/s con acción LOG para que
# cada vez que se permita el tráfico UDP descrito anteriormente, se deje un
# mensaje en el fichero de LOG del sistema.
```

PREGUNTA 7

```
# PREGUNTA 7

# No se debe permitir iniciar ninguna comunicación con la red privada ni con
# el propio firewall desde la zona DMZ.
iptables -t filter -A FORWARD -i eth1 -o eth0 -m state --state NEW -j DROP
iptables -t filter -A INPUT -i eth1 -m state --state NEW -j DROP
```

PREGUNTA 1

COMANDO: firewall:~# /hosthome/fw3.sh (ejecutar script fw3.sh)
COMANDO: firewall:~# nc -l -p 1111 (arrancar una aplicación servidor TCP)
COMANDO: pc1:~# nc -p 6666 100.211.1.100 1111 (arrancar una aplicación cliente TCP)
COMANDO: pc1:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                Thu Feb 29 16:10:14 2024

Chain PREROUTING (policy ACCEPT 1 packets, 60 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

COMANDO: pc6:~# nc -p 6666 100.211.1.100 1111 (arrancar una aplicación cliente TCP)

```
Every 0.5s: iptables -t nat -L -v -n                                                Thu Feb 29 16:11:58 2024

Chain PREROUTING (policy ACCEPT 5 packets, 300 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

Al lanzar un segundo cliente dirigido al mismo servidor, y si el primer cliente ya ha realizado un envío de mensajes a dicho servidor, se puede comprobar cómo al intentar enviar mensajes desde el segundo cliente, estos son descartados, quedando así en espera indefinidamente, ya que la comunicación TCP sólo permite la comunicación de un servidor con un único cliente.

PREGUNTA 2

COMANDO: firewall:~# /hosthome/fw3.sh (ejecutar script fw3.sh)
COMANDO: r3:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-14.cap (arrancar una captura de tráfico)
COMANDO: pc6:~# nc -l -p 1111 (arrancar una aplicación servidor TCP)
COMANDO: pc1:~# nc -p 6666 100.211.5.60 1111 (arrancar una aplicación cliente TCP)
COMANDO: pc1:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: r3:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                                Thu Feb 29 16:14:06 2024

Chain PREROUTING (policy ACCEPT 1 packets, 60 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    1    60 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

Como se puede ver, la única regla que se está cumpliendo es la misma que se estableció en el script fw1.sh, la cual se cumple 1 vez en el caso de POSTROUTING, y en el caso de PREROUTING acepta 1 paquete:

iptables -t nat -A POSTROUTING -s 10.211.0.0/16 -o eth2 -j SNAT --to-source 100.211.1.100

COMANDO: firewall:~# watch -n 0.5 iptables -t filter -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t filter -L -v -n                                                             Thu Feb 29 16:15:13 2024

Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            0.0.0.0/0           state NEW

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    5   273 ACCEPT     all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
    3   164 ACCEPT     all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40        udp dpt:7 LOG flags 0 level 4 prefix `echo'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40        udp dpt:7
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50        udp dpt:13 LOG flags 0 level 4 prefix `daytime'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50        udp dpt:13
    0     0 LOG        tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40        tcp dpt:7 LOG flags 0 level 4 prefix `tcp'
    0     0 ACCEPT     tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40        tcp dpt:7
    0     0 DROP       all  --  eth1   eth0    0.0.0.0/0            0.0.0.0/0           state NEW

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

Y en el caso de la tabla filter, se ve como sólo aparecen paquetes que han atravesado el firewall entrando por su interfaz eth0 y saliendo por eth2, y viceversa.

PREGUNTA 3

COMANDO: firewall:~# /hosthome/fw3.sh (ejecutar script fw3.sh)
COMANDO: pc4:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-15.cap (arrancar una captura de tráfico)
COMANDO: pc7:~# nc -u -p 6666 100.211.0.40 7 (arrancar una aplicación cliente UDP)
COMANDO: pc7:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: pc4:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                                Thu Feb 29 16:17:19 2024

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

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0           to:100.211.1.100

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

COMANDO: firewall:~# watch -n 0.5 iptables -t filter -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t filter -L -v -n                                                             Thu Feb 29 16:18:08 2024

Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            0.0.0.0/0           state NEW

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0           state RELATED,ESTABLISHED
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40        udp dpt:7 LOG flags 0 level 4 prefix `echo'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40        udp dpt:7
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50        udp dpt:13 LOG flags 0 level 4 prefix `daytime'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50        udp dpt:13
    0     0 LOG        tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40        tcp dpt:7 LOG flags 0 level 4 prefix `tcp'
    0     0 ACCEPT     tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40        tcp dpt:7
    0     0 DROP       all  --  eth1   eth0    0.0.0.0/0            0.0.0.0/0           state NEW

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

COMANDO: firewall:~# watch -n 0.5 /var/log/kern.log (mostrar cambios en /var/log/kern.log)

```
sh: /var/log/kern.log: Permission denied
```

En este caso, tanto la tabla nat como la tabla filter aparecen de la misma forma que antes de realizar la comunicación, además de no poder visualizar por temas de permisos el contenido del fichero /var/log/kern.log.

PREGUNTA 4

COMANDO: firewall:~# /hosthome/fw3.sh (ejecutar script fw3.sh)
COMANDO: pc5:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-16.cap (arrancar una captura de tráfico)
COMANDO: pc6:~# nc -u -p 6666 100.211.0.50 13 (arrancar una aplicación cliente UDP)
COMANDO: pc6:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: pc5:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t filter -L -v -n                                                              Thu Feb 29 16:22:29 2024

Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            0.0.0.0/0            state NEW

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0            state RELATED,ESTABLISHED
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40         udp dpt:7 LOG flags 0 level 4 prefix `echo'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40         udp dpt:7
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50         udp dpt:13 LOG flags 0 level 4 prefix `daytime'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50         udp dpt:13
    0     0 LOG        tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40         tcp dpt:7 LOG flags 0 level 4 prefix `tcp'
    0     0 ACCEPT     tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40         tcp dpt:7
    0     0 DROP       all  --  eth1   eth0    0.0.0.0/0            0.0.0.0/0            state NEW

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

COMANDO: firewall:~# watch -n 0.5 iptables -t filter -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                                 Thu Feb 29 16:24:26 2024

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

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0            to:100.211.1.100

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

COMANDO: firewall:~# watch -n 0.5 /var/log/kern.log (mostrar cambios en /var/log/kern.log)

```
sh: /var/log/kern.log: Permission denied
```

En este caso, tanto la tabla nat como la tabla filter aparecen de la misma forma que antes de realizar la comunicación, además de no poder visualizar por temas de permisos el contenido del fichero /var/log/kern.

PREGUNTA 5

COMANDO: firewall:~# /hosthome/fw3.sh (ejecutar script fw3.sh)
COMANDO: pc4:~# tcpdump -i eth0 -s 0 -w /hosthome/iptables-17.cap (arrancar una captura de tráfico)
COMANDO: pc1:~# nc -u -p 6666 100.211.0.40 7 (arrancar una aplicación cliente UDP)
COMANDO: pc1:~# hola <enter> (mandar un mensaje al servidor)
COMANDO: pc4:~# ctrl+c (interrumpir una captura de tráfico)
COMANDO: firewall:~# watch -n 0.5 iptables -t nat -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t nat -L -v -n                                                                 Thu Feb 29 16:24:26 2024

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

Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 SNAT       all  --  *      eth2    10.211.0.0/16        0.0.0.0/0            to:100.211.1.100

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

COMANDO: firewall:~# watch -n 0.5 iptables -t filter -L -v -n (mostrar la lista de reglas del firewall)

```
Every 0.5s: iptables -t filter -L -v -n                                                              Thu Feb 29 16:25:17 2024

Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   *       0.0.0.0/0            0.0.0.0/0
    0     0 DROP       all  --  eth1   *       0.0.0.0/0            0.0.0.0/0            state NEW

Chain FORWARD (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 ACCEPT     all  --  eth0   eth2    0.0.0.0/0            0.0.0.0/0
    0     0 ACCEPT     all  --  eth2   eth0    0.0.0.0/0            0.0.0.0/0            state RELATED,ESTABLISHED
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40         udp dpt:7 LOG flags 0 level 4 prefix `echo'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.40         udp dpt:7
    0     0 LOG        udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50         udp dpt:13 LOG flags 0 level 4 prefix `daytime'
    0     0 ACCEPT     udp  --  eth2   eth1    0.0.0.0/0            100.211.0.50         udp dpt:13
    0     0 LOG        tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40         tcp dpt:7 LOG flags 0 level 4 prefix `tcp'
    0     0 ACCEPT     tcp  --  eth0   eth1    0.0.0.0/0            100.211.0.40         tcp dpt:7
    0     0 DROP       all  --  eth1   eth0    0.0.0.0/0            0.0.0.0/0            state NEW

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

COMANDO: firewall:~# watch -n 0.5 /var/log/kern.log (mostrar cambios en /var/log/kern.log)

```
sh: /var/log/kern.log: Permission denied
```

En este caso, tanto la tabla nat como la tabla filter aparecen de la misma forma que antes de realizar la comunicación, además de no poder visualizar por temas de permisos el contenido del fichero /var/log/kern.