

**Universidad del País Vasco**

FACULTAD DE INFORMÁTICA

**SEGURIDAD RENDIMIENTO Y  
DISPONIBILIDAD DE SERVICIOS EN  
INFRAESTRUCTURAS**

*Firewall con IPTABLES*

Autores:  
*Grupo 3*

Donostia, a 4 de Mayo de 2021

## Índice

<b>1. Descripción de la práctica</b>	<b>3</b>
<b>2. Desarrollo de la práctica</b>	<b>3</b>
2.1. Ejercicios Previos . . . . .	3
2.2. Ejercicio 1 . . . . .	3
2.3. Ejercicio 2 . . . . .	9
2.4. Ejercicio 3 . . . . .	10
2.5. Ejercicio 4 . . . . .	12
2.6. Ejercicio 5 . . . . .	12
2.7. Ejercicio 6 . . . . .	14
2.8. Ejercicio 7 . . . . .	16
2.9. Ejercicio 8 . . . . .	18
<b>Anexos</b>	<b>19</b>
<b>A. fw-rules.sh</b>	<b>20</b>

## Lista de reglas

1. Reglas para permitir un comando <i>ping</i> desde la LAN a la DMZ . . . . .	11
2. Regla de redireccionamiento de paquetes icmp entre la LAN y la DMZ . . . . .	11
3. Regla de traducción de direcciones entre la LAN y la DMZ . . . . .	13
4. Reglas que permitan hacer <i>ping</i> desde LAN a Internet . . . . .	14
5. Traducción de direcciones salientes a Internet desde la LAN . . . . .	15
6. Reglas de permisión de conexión al servidor de la DMZ desde el exterior . . . . .	16

## Índice de figuras

1. Comando para la configuración de las reglas . . . . .	3
2. Ejecución del comando <i>route</i> en la máquina FW . . . . .	3
3. Ejecución del comando <i>route</i> en la máquina PC LAN . . . . .	4
4. Ejecución del comando <i>route</i> en la máquina PC DMZ . . . . .	4
5. Ejecución del comando <i>route</i> en la máquina PC EXT . . . . .	4
6. Ejecución del comando <i>ifconfig</i> en la máquina FW . . . . .	4
7. Ejecución del comando <i>ifconfig</i> en la máquina PC LAN . . . . .	5
8. Ejecución del comando <i>ifconfig</i> en la máquina PC DMZ . . . . .	5
9. Ejecución del comando <i>ifconfig</i> en la máquina PC EXT . . . . .	5
10. Ejecución del comando <i>ping</i> en la máquina PC EXT . . . . .	6
11. Ejecución del comando <i>ping</i> en la máquina PC DMZ . . . . .	6
12. Ejecución del comando <i>ping</i> en la máquina PC LAN . . . . .	7
13. Ejecución del comando <i>wget</i> en la máquina PC LAN . . . . .	7
14. Ejecución del comando <i>wget</i> al servidor LAN . . . . .	7
15. Ejecución del comando <i>nc</i> en la máquina PC DMZ . . . . .	8
16. Ejecución del comando <i>nc</i> en la máquina PC LAN . . . . .	8
17. Capturas <i>wireshark</i> . . . . .	8
18. Captura entre el <i>firewall</i> y la LAN . . . . .	8
19. Captura entre el <i>firewall</i> e Internet . . . . .	9
20. Intento de conexión al servidor de la LAN desde el PC externo . . . . .	10
21. Intento de conexión al servidor de la LAN desde la DMZ . . . . .	10
22. Comando <i>ping</i> desde la LAN a la interfaz LAN del <i>firewall</i> . . . . .	10
23. Comando <i>ping</i> a la interfaz de la DMZ . . . . .	11

24.	Prueba de conexión desde la LAN a la DMZ . . . . .	11
25.	Ping exitoso al servidor de la DMZ desde la LAN . . . . .	12
26.	Intento de <i>ping</i> desde la DMZ al servidor de la LAN . . . . .	12
27.	Conexión al PC de la DMZ desde la LAN . . . . .	12
28.	Acceso a la red LAN desde la DMZ . . . . .	13
29.	Contenido inicial de la tabla NAT . . . . .	13
30.	Intento de acceso desde la DMZ a la LAN . . . . .	13
31.	Nueva regla en la tabla NAT . . . . .	14
32.	Ping desde maquina en red LAN a Internet . . . . .	15
33.	Ubicación de captura de Wireshark . . . . .	15
34.	Ping desde la red LAN a Internet con traducción de IPs . . . . .	16
35.	Intento de conexión al servidor de la DMZ desde un PC externo . . . . .	16
36.	Conexión correcta el servidor de la DMZ desde el exterior . . . . .	17
37.	Representación de los lugares de captura de paquetes TCP . . . . .	17
38.	Captura de paquetes TCP antes del <i>firewall</i> . . . . .	17
39.	Captura de paquetes TCP después del <i>firewall</i> . . . . .	18
40.	Acceso al servidor correcto utilizando su IP privada . . . . .	18
41.	Error de acceso al servidor utilizando su IP privada . . . . .	18

## 1. Descripción de la práctica

El objetivo de esta actividad es comprender el funcionamiento de un *firewall* en una red con DMZ<sup>1</sup>, estudiando las reglas de filtrado y traducción entre redes. Para ello, se utilizará el software *Netfilter*, en concreto con el comando *iptables*. En cuanto al tiempo total de realización de la práctica ha sido de **15 horas**.

## 2. Desarrollo de la práctica

En esta sección se describirán cada uno de los apartados de la práctica así como la resolución de cada uno de ellos. Así mismo, en el anexo A se encuentra el fichero **fw-rules.sh** que contiene la configuración completa del *firewall* realizada durante la práctica.

### 2.1. Ejercicios Previos

Antes de empezar a hacer el primer ejercicio, hemos tenido que realizar unos pasos previos en la máquina virtual. Dicho pasos han sido ejecutar el programa GNS3, para ello hemos ido al menú principal y dentro del apartado de Educación, se encuentra el programa GNS3. Una vez abierto, hemos creado un proyecto con un nombre aleatorio con el fin de que se nos cree la carpeta GNS3 dentro de nuestro *home* del *user1*.

Ahora que ya está creada la carpeta hay que descomprimir el proyecto que se va a utilizar para esta práctica. El archivo a descomprimir se llama *srdsi-firewall.tar.gz* y se encuentra dentro de la carpeta *GNS3-images* que está en *home* del *user1*. Lo descomprimimos dentro de la carpeta */home/user1/GNS3/projects*. Una vez hecho eso, abrimos desde el programa *GNS3* el proyecto que acabamos de descomprimir. Una vez abierto tarda un rato en cargar, y después hemos puesto en marcha todas las máquinas pulsando el botón verde de *play* que se encuentra en la parte superior. Ahora ya podemos empezar con los ejercicios propuestos.

### 2.2. Ejercicio 1

En el primer apartado de este ejercicio se pide cargar la configuración inicial de las reglas de *iptables* en FW. Para ello se debe hacer doble *click* sobre la máquina FW y se abrirá una terminal. En dicha terminal, se escribirá el siguiente comando *sh ./fw-rules.sh*. Hecho esto se pasará al segundo apartado.

```
root@FW:~# sh ./fw-rules.sh
root@FW:~#
```

Figura 1: Comando para la configuración de las reglas

En este segundo apartado se pide comprobar la configuración de las interfaces ejecutando el comando *ifconfig*, y el contenido de las tablas de encaminamiento ejecutando el comando *route -n* de los PC's y de FW. Para ello, se realizará doble *click* sobre cada uno de los PC's y FW para que se abran las terminales correspondientes, se abrirá una ventana nueva en la terminal por cada máquina seleccionada. Una vez dentro se ejecutarán los dos comando indicados anteriormente. Una vez hecho esto, los resultados obtenidos han sido los siguientes.

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	203.0.113.1	0.0.0.0	UG	0	0	0	eth0
10.10.10.0	0.0.0.0	255.255.255.0	U	0	0	0	eth2
192.168.1.0	0.0.0.0	255.255.255.0	U	0	0	0	eth1
203.0.113.0	0.0.0.0	255.255.255.0	U	0	0	0	eth0

Figura 2: Ejecución del comando *route* en la máquina FW

<sup>1</sup>Demilitarized Zone o Zona desmilitarizada

```
root@PC_LAN:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         192.168.1.1   0.0.0.0       UG    0      0      0 eth0
192.168.1.0    0.0.0.0       255.255.255.0  U      0      0      0 eth0
root@PC_LAN:~#
```

Figura 3: Ejecución del comando *route* en la máquina PC LAN

```
root@PC_DMZ:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         10.10.10.1   0.0.0.0       UG    0      0      0 eth0
10.10.10.0     0.0.0.0       255.255.255.0  U      0      0      0 eth0
root@PC_DMZ:~#
```

Figura 4: Ejecución del comando *route* en la máquina PC DMZ

```
root@PC_EXT:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask        Flags Metric Ref  Use Iface
0.0.0.0         172.16.0.1   0.0.0.0       UG    0      0      0 eth0
172.16.0.0     0.0.0.0       255.255.0.0   U      0      0      0 eth0
root@PC_EXT:~#
```

Figura 5: Ejecución del comando *route* en la máquina PC EXT

```
root@FW:~# ifconfig
eth0      Link encap:Ethernet HWaddr 62:2c:e9:5d:1e:d1
          inet addr:203.0.113.10  Bcast:203.0.113.255 Mask:255.255.255.0
          inet6 addr: fe80::602c:e9ff:fe5d:1ed1/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:11 errors:0 dropped:0 overruns:0 frame:0
            TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:866 (866.0 B)  TX bytes:866 (866.0 B)

eth1      Link encap:Ethernet HWaddr 9a:de:70:41:c1:d9
          inet addr:192.168.1.1  Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::98de:70ff:fe41:c1d9/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:11 errors:0 dropped:0 overruns:0 frame:0
            TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:866 (866.0 B)  TX bytes:866 (866.0 B)

eth2      Link encap:Ethernet HWaddr 3a:16:11:89:1e:30
          inet addr:10.10.10.1  Bcast:10.10.10.255 Mask:255.255.255.0
          inet6 addr: fe80::3816:11ff:fe89:1e30/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:11 errors:0 dropped:0 overruns:0 frame:0
            TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:866 (866.0 B)  TX bytes:866 (866.0 B)

lo       Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

root@FW:~#
```

Figura 6: Ejecución del comando *ifconfig* en la máquina FW

```
root@PC_LAN:~# ifconfig
eth0      Link encap:Ethernet HWaddr f6:28:7a:fc:7d:fc
          inet addr:192.168.1.10 Bcast:192.168.1.255 Mask:255.255.255.0
          inet6 addr: fe80::f428:7aff:fe:fc/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:8 errors:0 dropped:0 overruns:0 frame:0
            TX packets:11 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:600 (600.0 B) TX bytes:866 (866.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@PC_LAN:~#
```

Figura 7: Ejecución del comando *ifconfig* en la máquina PC LAN

```
root@PC_DMZ:~# ifconfig
eth0      Link encap:Ethernet HWaddr 92:bb:d6:0e:0c:a3
          inet addr:10.10.10.30 Bcast:10.10.10.255 Mask:255.255.255.0
          inet6 addr: fe80::90bb:d6ff:fe0e:ca3/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:11 errors:0 dropped:1 overruns:0 frame:0
            TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:846 (846.0 B) TX bytes:936 (936.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@PC_DMZ:~#
```

Figura 8: Ejecución del comando *ifconfig* en la máquina PC DMZ

```
root@PC_EXT:~# ifconfig
eth0      Link encap:Ethernet HWaddr 62:ec:ab:cc:0e:cc
          inet addr:172.16.0.20 Bcast:172.16.255.255 Mask:255.255.0.0
          inet6 addr: fe80::60ec:abff:fecc:0e/64 Scope:Link
            UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
            RX packets:12 errors:0 dropped:0 overruns:0 frame:0
            TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:936 (936.0 B) TX bytes:936 (936.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1 Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
            UP LOOPBACK RUNNING MTU:65536 Metric:1
            RX packets:0 errors:0 dropped:0 overruns:0 frame:0
            TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

root@PC_EXT:~#
```

Figura 9: Ejecución del comando *ifconfig* en la máquina PC EXT

Como se puede observar en las capturas, las máquinas no tienen ninguna regla de filtrado/traducción definidas, como ya se adelantaba en el enunciado del ejercicio. Solo dispone reglas básicas para dejar pasar los

paquetes que vayan a direcciones específicas, es decir, al interior o al exterior de la red.

En el tercer apartado se pide verificar que desde cualquier PC se puede acceder al resto de PC's. Para ello nos disponemos a ejecutar el comando *ping/traceroute -I IP\_host* desde cada PC a cada PC. Las direcciones IP ya son conocidas, ya que se han cogido de las capturas realizadas de ejecutar el comando *ifconfig* a las máquinas en el apartado anterior. Generalmente, para realizar los intentos de conexión a otras máquinas, se ha utilizado el comando *ping -c 3 IP\_host*. La opción *-c 3* es para enviar únicamente 3 paquetes desde la máquina en la que estamos a la máquina con la IP indicada con el parámetro *IP\_host*.

```
root@PC_EXT:~# ping -c 3 10.10.10.30
PING 10.10.10.30 (10.10.10.30) 56(84) bytes of data.
64 bytes from 10.10.10.30: icmp_seq=1 ttl=61 time=3.48 ms
64 bytes from 10.10.10.30: icmp_seq=2 ttl=61 time=1.39 ms
64 bytes from 10.10.10.30: icmp_seq=3 ttl=61 time=1.38 ms

--- 10.10.10.30 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.389/2.090/3.484/0.986 ms
root@PC_EXT:~# ping -c 3 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=61 time=2.29 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=61 time=1.21 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=61 time=0.423 ms

--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.423/1.308/2.292/0.766 ms
root@PC_EXT:~#
```

Figura 10: Ejecución del comando *ping* en la máquina PC EXT

```
root@PC_DMZ:~# ping -c 3 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=63 time=1.35 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=63 time=1.12 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=63 time=0.979 ms

--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.979/1.152/1.356/0.160 ms
root@PC_DMZ:~# ping -c 3 172.16.0.20
PING 172.16.0.20 (172.16.0.20) 56(84) bytes of data.
64 bytes from 172.16.0.20: icmp_seq=1 ttl=61 time=1.81 ms
64 bytes from 172.16.0.20: icmp_seq=2 ttl=61 time=1.35 ms
64 bytes from 172.16.0.20: icmp_seq=3 ttl=61 time=1.46 ms

--- 172.16.0.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.357/1.545/1.815/0.198 ms
root@PC_DMZ:~#
```

Figura 11: Ejecución del comando *ping* en la máquina PC DMZ

```
root@PC_LAN:~# ping -c 3 172.16.0.20
PING 172.16.0.20 (172.16.0.20) 56(84) bytes of data.
64 bytes from 172.16.0.20: icmp_seq=1 ttl=61 time=1.94 ms
64 bytes from 172.16.0.20: icmp_seq=2 ttl=61 time=1.38 ms
64 bytes from 172.16.0.20: icmp_seq=3 ttl=61 time=1.60 ms

--- 172.16.0.20 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 1.385/1.645/1.946/0.235 ms
root@PC_LAN:~# ping -c 3 10.10.10.30
PING 10.10.10.30 (10.10.10.30) 56(84) bytes of data.
64 bytes from 10.10.10.30: icmp_seq=1 ttl=63 time=0.947 ms
64 bytes from 10.10.10.30: icmp_seq=2 ttl=63 time=1.71 ms
64 bytes from 10.10.10.30: icmp_seq=3 ttl=63 time=1.19 ms

--- 10.10.10.30 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 0.947/1.284/1.710/0.320 ms
root@PC_LAN:~#
```

Figura 12: Ejecución del comando *ping* en la máquina PC LAN

Una vez hecho esto, se puede confirmar que la configuración de las máquinas *router* y FW es “correcta”, ya que se puede acceder desde cualquier PC al resto de PC’s.

En el cuarto apartado se pide probar el acceso a los servidores web desde el PC en la red LAN. Para ello se ha ejecutado el comando `wget 10.10.10.30`. Al ejecutarlo se obtiene la siguiente salida.

```
root@PC_LAN:~# wget 10.10.10.30
converted 'http://10.10.10.30' (ANSI_X3.4-1968) -> 'http://10.10.10.30' (UTF-8)
--2021-05-02 19:49:21-- http://10.10.10.30/
Connecting to 10.10.10.30:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 117 [text/html]
Saving to: 'index.html.1'

index.html.1          100%[=====]      117  --.-KB/s   in 0s
2021-05-02 19:49:21 (25.8 MB/s) - 'index.html.1' saved [117/117]
root@PC_LAN:~#
```

Figura 13: Ejecución del comando *wget* en la máquina PC LAN

Se ha accedido correctamente al servidor. La IP utilizada en el comando anterior es la del servidor web de la DMZ. Sin embargo, el propio PC LAN también dispone de un servicio web, así que se ha accedido desde el PC DMZ esta vez al PC LAN para ver si era accesible. Para ello se ha ejecutado el comando `wget 192.168.1.10`.

```
root@PC_DMZ:~# wget 192.168.1.10
converted 'http://192.168.1.10' (ANSI_X3.4-1968) -> 'http://192.168.1.10' (UTF-8)
--2021-05-03 14:56:52-- http://192.168.1.10/
Connecting to 192.168.1.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 117 [text/html]
Saving to: 'index.html'

index.html          100%[=====]      117  --.-KB/s   in 0s
2021-05-03 14:56:52 (10.1 MB/s) - 'index.html' saved [117/117]
root@PC_DMZ:~#
```

Figura 14: Ejecución del comando *wget* al servidor LAN

En el quinto apartado se pide examinar el acceso a otros puertos TCP/UDP haciendo uso de la herramienta *nc*. De nuevo, como en el apartado anterior, se está trabajando sobre el PC LAN. En el se ha ejecutado el comando `nc -zv 10.10.10.30 1-65535` para visualizar si la máquina PC DMZ disponía de algún puerto más abierto. Lo que nos se ha encontrado es que únicamente tiene abierto el puerto 80, que era al que se ha accedido en el apartado anterior. Se ha realizado lo mismo pero esta vez desde el PC DMZ al PC LAN ejecutando el comando `nc -zv 192.168.1.10`. Y al igual que en el caso anterior, únicamente está abierto el puerto 80.

```
root@PC_DMZ:~# nc -zv 192.168.1.10 1-65535
PC_LAN [192.168.1.10] 80 (http) open
```

Figura 15: Ejecución del comando *nc* en la máquina PC DMZ

```
root@PC_LAN:~# nc -zv 10.10.10.30 1-65535
PC_DMZ [10.10.10.30] 80 (http) open
```

Figura 16: Ejecución del comando *nc* en la máquina PC LAN

En este último apartado se pide analizar las tramas enviadas y recibidas por la red haciendo uso de la herramienta *Wireshark*. Haciendo captura desde la LAN hasta el FW y desde INTERNET a FW, no se ha visualizado ningún tráfico. Era de esperar ya que de momento no estamos haciendo uso de los mismo.

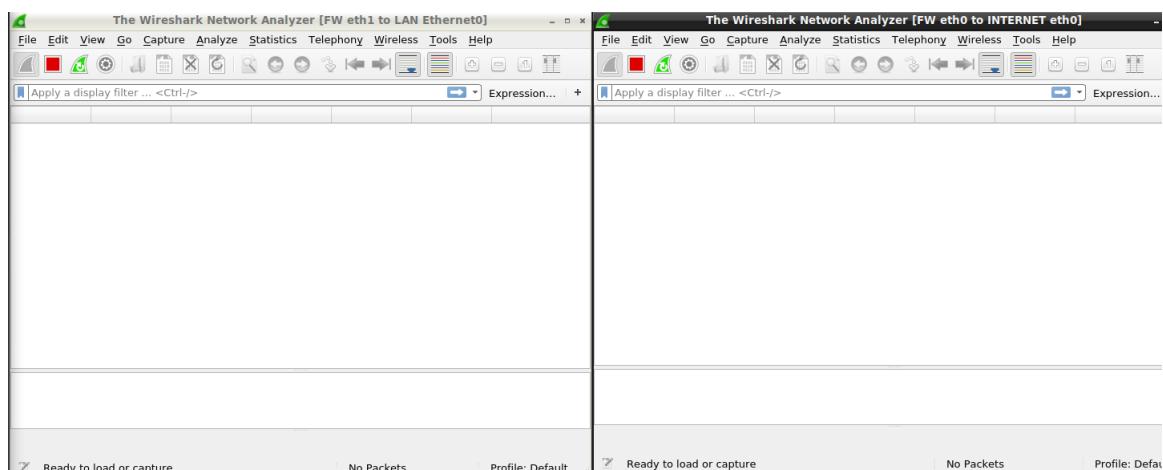


Figura 17: Capturas *wireshark*

A continuación, se va a probar ha hacer un *ping* desde la PC EXT hasta la PC LAN y se va a analizar en tráfico en los puntos antes comentados.

Source	Destination	Protocol	Length	Info
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) request
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) reply
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) request
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) reply
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) request
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) reply

Figura 18: Captura entre el *firewall* y la LAN

Source	Destination	Protocol	Length	Info
2e:b8:79:4a:1c:c8	Broadcast	ARP	42	Who has 203.0.113.10?
26:5f:99:1a:7c:c4	2e:b8:79:4a:1c:c8	ARP	42	203.0.113.10 is at 26
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) request
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) reply
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) request
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) reply
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) request
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) reply

Figura 19: Captura entre el *firewall* e Internet

Ahora sí se puede ver que se ha creado tráfico. Características a destacar, que al hacer un *ping* solo se han podido observar mensajes que utilizan protocolo ARP e ICMP. Dentro de las mismas se puede observar que dentro de los mensajes ICMP hay dos tipos: *Echo request* y *Echo reply*. Analizando la traza de FW a INTERNET, ver figura 19, se ve que los mensajes de tipo *request* se corresponden con los mensajes de entrada al *firewall*, es decir, los mensajes que tienen como destino la máquina a la que se hace el *ping*, y los de tipo *reply* se corresponden con los mensajes de respuesta a la máquina origen.

### 2.3. Ejercicio 2

En este segundo ejercicio se pide añadir en la máquina FW el contenido de *fw-ejerc2.txt* al fichero *fw-rules.sh*. Una vez hecho eso se volverá a ejecutar el comando *sh ./fw-rules.sh* para aplicar los cambios realizados.

En el primer apartado de este ejercicio se pide suponer en base a las reglas añadidas el nuevo comportamiento del *firewall*. Las tres primeras reglas añadidas van a hacer que todos los paquetes que le lleguen, tenga que salir o entrar por el *firewall*, los deseche. Esto se debe a que haciendo uso de la opción *-P* se aplica por política que se desechen los paquetes de *INPUT/OUTPUT/FORWARD*. Por lo que probablemente ya no se consiga ponerse en contacto desde fuera de la red a máquinas que estén en Internet. Así mismo tampoco permitirá las conexiones entre máquinas que tengan que pasar por el *firewall* en ambas direcciones. Es decir, desde el PC DMZ al PC LAN y viceversa.

Seguidamente las siguientes reglas auxiliares van a limitar las máquinas y en qué direcciones pueden ponerte en contacto. Concretamente las primeras cinco reglas auxiliares de tipo *INPUT*, van a permitir que todos aquellos paquetes que vengan de cualquier puerto desde las *ip's LO\_IFACE\_IP, FW\_IFACE\_IP, LAN\_IFACE\_IP, DMZ\_IFACE\_IP, INTENET\_IFACE\_IP*, desde la interfaz *localhost*, e interfaz *LO\_IFACE* puedan pasar.

Las siguientes cinco reglas de tipo *OUTPUT* van a permitir que todos aquellos paquetes que salgan del *firewall* con las *ip's LO\_IFACE\_IP, FW\_IFACE\_IP, LAN\_IFACE\_IP, DMZ\_IFACE\_IP* e *INTENET\_IFACE\_IP* puedan pasar.

Antes de pasar el siguiente apartado, se va a realizar una pequeña conclusión. La conclusión es que no se va a poder realizar ninguna conexión entre ninguna máquina ya que no está configurada la regla de *FORWARD*. Es decir, si no se observa el esquema que se tiene en el programa *GNS3*, falta la regla que permita pasar los paquetes que entran por *eth0* y tengan que salir por *eth1*. Y lo mismo ocurre con *eth2* al resto. De manera que aunque estén las reglas que permitan enviar paquetes de una IP específica a otra no significa que vayan a dejarlas pasar por el *firewall*. Realmente no es que no les deje pasar, si no que una vez que entran, al no haber una regla que especifique qué hacer con ellos, no sabe qué hacer con ellos. Y como ya se ha comentado anteriormente la política por defecto es desechar todos los paquetes, por lo que eso es lo que hará.

Para comprobar las suposiciones anteriores, a continuación, se va a tratar de realizar un *ping* desde el PC de la DMZ al servidor ubicado en la LAN. Así mismo, también se tratará de acceder desde un PC situado en Internet al servidor ubicado en la LAN.

```
root@PC_EXT:~# ping -c 3 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=61 time=3.10 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=61 time=2.38 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=61 time=2.36 ms

--- 192.168.1.10 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 2.361/2.618/3.109/0.347 ms
root@PC_EXT:~# ping -c 3 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.

--- 192.168.1.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2038ms

root@PC_EXT:~#
```

Figura 20: Intento de conexión al servidor de la LAN desde el PC externo

```
root@PC_DMZ:~# ping -c 3 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.

--- 192.168.1.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2039ms

root@PC_DMZ:~#
```

Figura 21: Intento de conexión al servidor de la LAN desde la DMZ

Al realizar esta prueba se ha podido ver claramente el comportamiento del *firewall*. Ahora rechaza completamente todas las conexiones que llegan a él.

## 2.4. Ejercicio 3

En el tercer ejercicio se nos pide que insertemos nuevas reglas dentro del fichero *fw-rules.sh*.

Una vez hecho esto, en el primer apartado se nos pide comprobar el funcionamiento de las dos primera reglas haciendo uso del comando *ping*. Para ello se abrirá la terminal del PC LAN y se ejecutará el siguiente comando *ping -c 3 192.168.1.1*.

```
root@PC_LAN:~# ping -c 3 192.168.1.1
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=0.736 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.212 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.226 ms

--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2017ms
rtt min/avg/max/mdev = 0.212/0.391/0.736/0.244 ms
root@PC_LAN:~#
```

Figura 22: Comando *ping* desde la LAN a la interfaz LAN del *firewall*

En el segundo apartado se pide comprobar el funcionamiento de las siguientes dos reglas. Se sabe que la función de dichas reglas es la misma que las dos anteriores salvo que en este caso permiten aceptar el comando *ping* para la interfaz de la DMZ en lugar de la de la LAN.

```
root@PC_DMZ:~# ping -c 3 10.10.10.1
PING 10.10.10.1 (10.10.10.1) 56(84) bytes of data.
64 bytes from 10.10.10.1: icmp_seq=1 ttl=64 time=1.01 ms
64 bytes from 10.10.10.1: icmp_seq=2 ttl=64 time=0.297 ms
64 bytes from 10.10.10.1: icmp_seq=3 ttl=64 time=0.482 ms

--- 10.10.10.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2018ms
rtt min/avg/max/mdev = 0.297/0.599/1.018/0.305 ms
root@PC_DMZ:~#
```

Figura 23: Comando *ping* a la interfaz de la DMZ

En el tercer apartado se pide implementar unas reglas que permitan un comando *ping* de la red LAN a la DMZ. Para ello se ha necesitado de la implementación de varias reglas y no solo una. Las reglas han sido las siguientes.

```
$IPTABLES -A INPUT -i $LAN_IFACE -s $LAN_IP -d $DMZ_IP \
          -p icmp -m icmp --icmp-type echo-request \
          -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A INPUT -i $DMZ_IFACE -s $DMZ_IP -d $LAN_IP \
          -p icmp -m icmp --icmp-type echo-reply \
          -m state --state ESTABLISHED,RELATED -j ACCEPT
```

**Esta cadena se aplica a datagramas que llegan a procesos locales del FW**

Regla 1: Reglas para permitir un comando *ping* desde la LAN a la DMZ

```
root@PC_LAN:~# ping -c 3 10.10.10.30
PING 10.10.10.30 (10.10.10.30) 56(84) bytes of data.

--- 10.10.10.30 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2025ms
root@PC_LAN:~#
```

Figura 24: Prueba de conexión desde la LAN a la DMZ

Como era de esperar no se permite realizar el *ping* sobre la máquina de la DMZ. Esto se debe a lo explicado en el ejercicio anterior, es decir, todas estas reglas no sirven si no se implementa una regla que permita pasar los paquetes de *eht1* (*LAN\_IFACE*) a *eht2* (*DMZ\_IFACE*) y viceversa.

En el último apartado se pide comprobar si es necesario añadir alguna otra regla. Como se acaba de comentar en el apartado anterior, la respuesta es sí. Se tiene que implementar una regla que nos permita redirigir los paquetes que entran por *eht1* a *eht2* y viceversa para el correcto funcionamiento del comando *ping*. Para ello las reglas que se han implementado han sido las siguientes.

```
$IPTABLES -A FORWARD -i $LAN_IFACE -o $DMZ_IFACE \
          -s $LAN_IP -d $DMZ_IP \
          -p icmp -m icmp --icmp-type echo-request \
          -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -i $DMZ_IFACE -o $LAN_IFACE \
          -p icmp -m icmp --icmp-type echo-reply \
          -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Regla 2: Regla de redireccionamiento de paquetes icmp entre la LAN y la DMZ

Gracias a estas traducciones se permite el transito de mensajes *ping* desde la LAN a la DMZ. Es decir, ahora se puede realizar un *ping* desde un pc que se encuentra en la LAN a la DMZ pero no al revés.

```
root@PC_LAN:~# ping -c 3 10.10.10.30
PING 10.10.10.30 (10.10.10.30) 56(84) bytes of data.
64 bytes from 10.10.10.30: icmp_seq=1 ttl=63 time=0.569 ms
64 bytes from 10.10.10.30: icmp_seq=2 ttl=63 time=0.491 ms
64 bytes from 10.10.10.30: icmp_seq=3 ttl=63 time=0.485 ms

--- 10.10.10.30 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2049ms
rtt min/avg/max/mdev = 0.485/0.515/0.569/0.038 ms
root@PC_LAN:~#
```

Figura 25: *Ping* exitoso al servidor de la DMZ desde la LAN

```
root@PC_DMZ:~# ping -c 3 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.

--- 192.168.1.10 ping statistics ---
3 packets transmitted, 0 received, 100% packet loss, time 2024ms

root@PC_DMZ:~#
```

Figura 26: Intento de *ping* desde la DMZ al servidor de la LAN

## 2.5. Ejercicio 4

En este ejercicio, lo primero a realizar es añadir el contenido necesario al *script* que contiene el conjunto de reglas realizadas hasta ahora.

Una vez hecho esto, en el primer apartado se pide describir la finalidad de dichas reglas. Las reglas permiten, por un lado, el establecimiento de una nueva conexión tcp, desde la LAN a la DMZ, a través de los puertos 80 y 443. La segunda regla, permite el paso de estas respuestas tcp desde la DMZ a la LAN.

*¿al 443?*

```
root@PC_LAN:~# wget 10.10.10.30
converted 'http://10.10.10.30' (ANSI_X3.4-1968) -> 'http://10.10.10.30' (UTF-8)
--2021-05-02 23:31:02-- http://10.10.10.30/
Connecting to 10.10.10.30:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 117 [text/html]
Saving to: 'index.html.3'

index.html.3          100%[=====]      117  --.-KB/s   in 0s
2021-05-02 23:31:02 (8.28 MB/s) - 'index.html.3' saved [117/117]
```

Figura 27: Conexión al PC de la DMZ desde la LAN

## 2.6. Ejercicio 5

La idea de tener una red DMZ es evitar, que en caso de sufrir un ataque, este se propague por el resto de nuestra red. Por tanto, el objetivo de este ejercicio es evitar que esta zona tenga acceso a la red LAN de la organización pero permitiendo la entrada de tráfico proveniente de la red LAN.

Por tanto, el primer paso a realizar será comprobar que actualmente la DMZ tiene acceso a la red LAN. Para ello, simplemente se hará un *ping* a la máquina de la red LAN para comprobar que efectivamente se reciben paquetes.

```
root@PC_DMZ:~# ping -c 4 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.
64 bytes from 192.168.1.10: icmp_seq=1 ttl=63 time=1.97 ms
64 bytes from 192.168.1.10: icmp_seq=2 ttl=63 time=0.653 ms
64 bytes from 192.168.1.10: icmp_seq=3 ttl=63 time=0.518 ms
64 bytes from 192.168.1.10: icmp_seq=4 ttl=63 time=0.455 ms

--- 192.168.1.10 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3028ms
rtt min/avg/max/mdev = 0.455/0.900/1.974/0.624 ms
root@PC_DMZ:~#
```

Figura 28: Acceso a la red LAN desde la DMZ

Así mismo, la figura 29 muestra el contenido inicial de la tabla NAT antes de añadir las reglas correspondientes al ejercicio.

```
root@PC_DMZ:~# iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
root@PC_DMZ:~#
```

Figura 29: Contenido inicial de la tabla NAT

A continuación se añadirá la regla 3 la cual se encarga de, antes de enviar el paquete o paquetes, *POSTROUTING*, que vienen de la dirección *LAN\_IP* y van a la dirección *DMZ\_IP* saliendo por la interfaz *DMZ\_IFACE*, cambia la dirección de origen por la *DMZ\_IFACE\_IP*.

```
$IPTABLES -t nat -A POSTROUTING -o $DMZ_IFACE -p all \
-s $LAN_IP -d $DMZ_IP -j SNAT --to $DMZ_IFACE_IP
```

Regla 3: Regla de traducción de direcciones entre la LAN y la DMZ

Si ahora se comprueba el contenido de la tabla nat, ver figura 31, se puede observar que se ha aplicado la traducción antes explicada. Así mismo, si se intenta hacer ahora un *ping* desde la DMZ a la red LAN. El resultado será muy diferente al mostrado en la figura 28.

```
root@PC_DMZ:~# ping -c 4 192.168.1.10
PING 192.168.1.10 (192.168.1.10) 56(84) bytes of data.

--- 192.168.1.10 ping statistics ---
4 packets transmitted, 0 received, 100% packet loss, time 3049ms
root@PC_DMZ:~#
```

Figura 30: Intento de acceso desde la DMZ a la LAN

**El objetivo era ocultar las direcciones de la LAN**

```
root@FW:~# iptables -t nat -L -n -v
Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out    source               destination
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out    source               destination
Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out    source               destination
Chain POSTROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out    source               destination
      0      0 SNAT      all   -- *      eth2    192.168.1.0/24        10.10.10.0/24      to:10.10.10.1
root@FW:~#
```

Figura 31: Nueva regla en la tabla NAT

## 2.7. Ejercicio 6

Puesto que las máquinas de la red LAN deben tener acceso a Internet pero no pueden ser directamente visibles desde Internet se deben realizar un par de ajustes. En primer lugar, se deben agregar al *firewall* las reglas correspondientes para garantizar el acceso a Internet desde las máquinas de la LAN. Para ello, se añadirán las siguientes reglas.

```
$IPTABLES -A FORWARD -i $LAN_IFACE -o $INTERNET_IFACE \
          -p icmp -m icmp --icmp-type echo-request \
          -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNET_IFACE -o $LAN_IFACE \
          -p icmp -m icmp --icmp-type echo-reply \
          -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -I OUTPUT 1 -o $LAN_IFACE -s $LAN_IP -d $ANY_IP \
          -p icmp -m icmp --icmp-type echo-request \
          -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -I INPUT 1 -i $INTERNET_IFACE -s $ANY_IP -d $LAN_IP \
          -p icmp -m icmp --icmp-type echo-reply \
          -m state --state ESTABLISHED,RELATED -j ACCEPT
```

faltan reglas para 80 y 443

Regla 4: Reglas que permitan hacer *ping* desde LAN a Internet

Las dos primeras reglas, las que se agregan a la cadena *FORWARD*, se encargan de redireccionar los paquetes ICMP de la interfaz LAN a la interfaz de INTERNET y viceversa, es decir, se redireccionarán los paquetes ICMP de tipo *echo-request* de la interfaz de la LAN a Internet, y los de tipo *echo-reply*, los de respuesta, se redireccionarán de la interfaz de Internet a la interfaz LAN.

Los otras dos reglas simplemente permiten la entrada y salida de los paquetes ICMP de tipo *echo-request* y *echo-reply*, respectivamente. **al FW!!!**

Para comprobar que efectivamente las nuevas reglas añadidas funcionan correctamente simplemente se hará un *ping* desde una máquina en la red LAN a otra en Internet, por ejemplo PC\_EXT con IP 172.16.0.20. La captura que se muestra en la figura 32 se ha realizado en la salida de la interfaz de internet del *firewall*, ver figura 33.

Source	Destination	Protocol	Length	Info
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) request id: 1000000000000000
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) reply id: 1000000000000000
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) request id: 1000000000000000
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) reply id: 1000000000000000
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) request id: 1000000000000000
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) reply id: 1000000000000000
192.168.1.10	172.16.0.20	ICMP	98	Echo (ping) request id: 1000000000000000
172.16.0.20	192.168.1.10	ICMP	98	Echo (ping) reply id: 1000000000000000

Figura 32: Ping desde maquina en red LAN a Internet

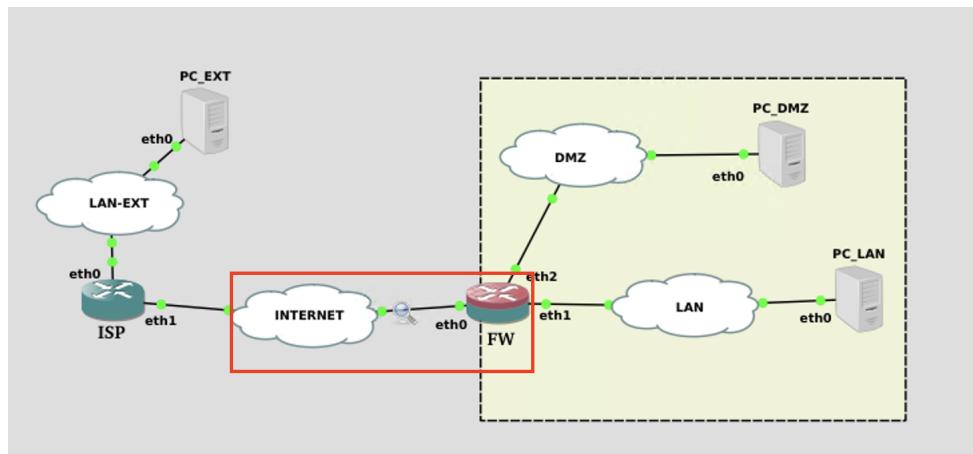


Figura 33: Ubicación de captura de Wireshark

No obstante, como ve se en la captura de Wireshark anterior la dirección de origen del paquete es la IP privada de la red LAN, en su lugar, debería ser la IP pública asignada, en este caso, a la interfaz 0, o Interfaz de Internet. Para que se haga este cambio se deberá añadir una nueva regla a la tabla nat, la cual traducirá la IP privada de la máquina de la red LAN a la IP pública antes comentada. Para ello, se introducirá la siguiente regla.

```
$IPTABLES -t nat -A POSTROUTING -o $INTERNET_IFACE -p all \
-s $LAN_IP -d $ANY_IP -j SNAT --to $INTERNET_IFACE_IP
```

Regla 5: Traducción de direcciones salientes a Internet desde la LAN

Si ahora se vuelve a realizar una captura en la misma ubicación que la mostrada en la figura 33 el resultado es diferente. Ahora la IP de origen de los paquetes es 203.0.113.10, la IP de la interfaz de Internet del firewall.

Source	Destination	Protocol	Length	Info
203.0.113.10	172.16.0.20	ICMP	98	Echo (ping) request
172.16.0.20	203.0.113.10	ICMP	98	Echo (ping) reply
203.0.113.10	172.16.0.20	ICMP	98	Echo (ping) request
172.16.0.20	203.0.113.10	ICMP	98	Echo (ping) reply
203.0.113.10	172.16.0.20	ICMP	98	Echo (ping) request
172.16.0.20	203.0.113.10	ICMP	98	Echo (ping) reply
203.0.113.10	172.16.0.20	ICMP	98	Echo (ping) request
172.16.0.20	203.0.113.10	ICMP	98	Echo (ping) reply

Figura 34: Ping desde la red LAN a Internet con traducción de IPs

## 2.8. Ejercicio 7

En este apartado se pide realizar la configuración necesaria para que un PC externo, en Internet, se pueda conectar al servidor alojado en la DMZ. No obstante, la dirección de acceso al servidor será la IP 203.0.113.10 y no la dirección IP privada 10.10.10.30.

Por tanto, el primer paso a realizar será agregar las reglas proporcionadas para este ejercicio (*fw-ejer7.txt*). Estas reglas se encargan de realizar la traducción de la dirección y el puerto públicas a privadas. No obstante, únicamente con estas traducciones un pc externo no se podrá conectar a nuestro servidor, ya que aún no se han habilitado las reglas correspondientes en el *firewall*. Para comprobarlo se tratará de realizar una petición al servidor utilizando `wget`.

```
root@PC_EXT:~# wget -v -T 5 -t 1 203.0.113.10
converted 'http://203.0.113.10' (ANSI X3.4-1968) -> 'http://203.0.113.10' (UTF-8)
--2021-05-01 13:22:36-- http://203.0.113.10/
Connecting to 203.0.113.10:80... failed: Connection timed out.
Giving up.

root@PC_EXT:~#
```

Figura 35: Intento de conexión al servidor de la DMZ desde un PC externo

Para permitir la conexión al servidor se añadirán las siguientes reglas.

```
$IPTABLES -A INPUT -i $INTERNET_IFACE -d $INTERNET_IFACE_IP \
          -p tcp -j ACCEPT
$IPTABLES -A OUTPUT -o $DMZ_IFACE -d $DMZ_IFACE_IP \
          -p tcp -j ACCEPT                                ¿al fw?

$IPTABLES -A FORWARD -i $INTERNET_IFACE -o $DMZ_IFACE \
          -p tcp -j ACCEPT
$IPTABLES -A FORWARD -i $DMZ_IFACE -o $INTERNET_IFACE \
          -p tcp -j ACCEPT                                ¿a cualquier puerto?
```

Regla 6: Reglas de permisión de conexión al servidor de la DMZ desde el exterior

Las dos primeras reglas mostradas en el listing 6 permiten las conexiones *tcp* entrantes, por la interfaz de Internet, y salientes por la interfaz de la DMZ. Así mismo, las dos últimas reglas, permiten el redirecccionamiento de las conexiones *tcp* de la interfaz de Internet a la de la DMZ y viceversa. Si ahora se prueba a realizar de nuevo la conexión al servidor utilizando el comando `wget` el resultado es el siguiente.

```

root@PC_EXT:# wget -v -T 5 -t 1 203.0.113.10
converted 'http://203.0.113.10' (ANSI X3.4-1968) -> 'http://203.0.113.10' (UTF-8)
--2021-05-01 13:28:07-- http://203.0.113.10/
Connecting to 203.0.113.10:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 117 [text/html]
Saving to: 'index.html'

index.html                                100%[=====] 9.70 MB/s

2021-05-01 13:28:07 (9.70 MB/s) - 'index.html' saved [117/117]

root@PC_EXT:~# 

```

Figura 36: Conexión correcta al servidor de la DMZ desde el exterior

Para asegurar que la traducción de IPs se han realizado correctamente se han comprobado capturas en dos lugares, la primera, figura 38, captura el punto anterior al paso por el *firewall*, ver cuadrado rojo en la figura 37. La segunda, figura 39, captura los paquetes después de su paso por el *firewall*, ver cuadrado verde en la figura 37.

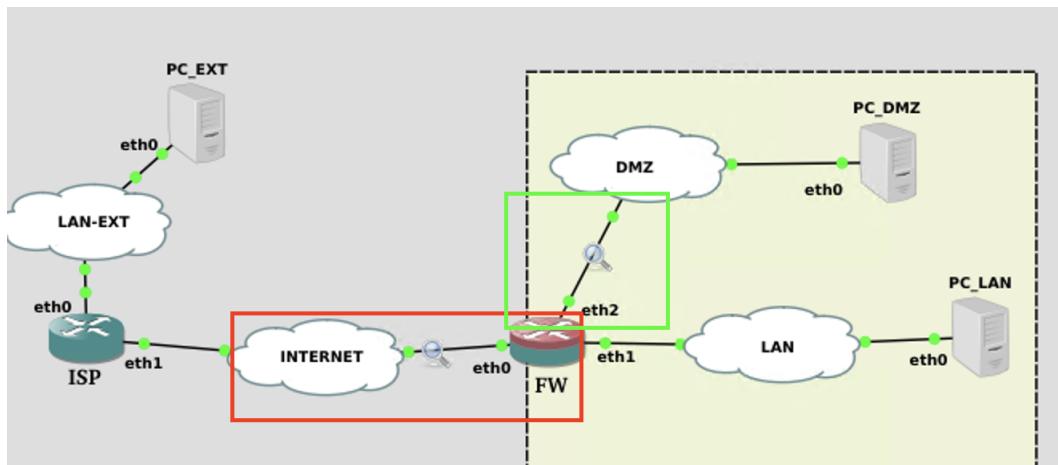


Figura 37: Representación de los lugares de captura de paquetes TCP

Source	Destination	Protocol	Length	Info
172.16.0.20	203.0.113.10	TCP	74	42510 → 80 [SYN] Seq=0 Win=16
203.0.113.10	172.16.0.20	TCP	74	80 → 42510 [SYN, ACK] Seq=1 Win=16
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [ACK] Seq=1 Win=16
172.16.0.20	203.0.113.10	HTTP	176	GET / HTTP/1.1
203.0.113.10	172.16.0.20	TCP	66	80 → 42510 [ACK] Seq=1 Win=16
203.0.113.10	172.16.0.20	HTTP	490	HTTP/1.1 200 OK (text/html)
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [ACK] Seq=11 Win=16
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [FIN, ACK] Seq=11 Win=16
203.0.113.10	172.16.0.20	TCP	66	80 → 42510 [FIN, ACK] Seq=11 Win=16
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [ACK] Seq=11 Win=16

Figura 38: Captura de paquetes TCP antes del *firewall*

Source	Destination	Protocol	Length	Info
172.16.0.20	203.0.113.10	TCP	74	42510 → 80 [SYN] Seq=0 Win=1460
203.0.113.10	172.16.0.20	TCP	74	80 → 42510 [SYN, ACK] Seq=1 Ack=1 Win=1460
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [ACK] Seq=1 Ack=1 Win=1460
172.16.0.20	203.0.113.10	HTTP	176	GET / HTTP/1.1
203.0.113.10	172.16.0.20	TCP	66	80 → 42510 [ACK] Seq=1 Ack=1 Win=1460
203.0.113.10	172.16.0.20	HTTP	490	HTTP/1.1 200 OK (text/html)
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [ACK] Seq=111 Ack=1 Win=1460
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [FIN, ACK] Seq=111 Ack=1 Win=1460
203.0.113.10	172.16.0.20	TCP	66	80 → 42510 [FIN, ACK] Seq=112 Ack=1 Win=1460
172.16.0.20	203.0.113.10	TCP	66	42510 → 80 [ACK] Seq=112 Ack=1 Win=1460

Figura 39: Captura de paquetes TCP después del *firewall*

## 2.9. Ejercicio 8

En el ejercicio anterior se ha dado acceso al servidor de la DMZ desde Internet, este acceso se hace desde la IP 203.0.113.10. No obstante, con las reglas actuales del *firewall* también se puede acceder al servidor utilizando su IP privada, 10.10.10.30.

En la figura 40 se puede ver que efectivamente es posible el acceso al servidor de la DMZ utilizando su IP privada.

```
root@PC_EXT:~# wget -v -T 5 -t 1 10.10.10.30
converted 'http://10.10.10.30' (ANSI_X3.4-1968) -> 'http://10.10.10.30' (UTF-8)
--2021-05-01 15:09:47-- http://10.10.10.30/
Connecting to 10.10.10.30:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 117 [text/html]
Saving to: 'index.html'

index.html                                              100%[=====] 117/117
2021-05-01 15:09:47 (11.5 MB/s) - 'index.html' saved [117/117]
```

Figura 40: Acceso al servidor correcto utilizando su IP privada

Para evitar el acceso utilizando IPs privadas se añadirán las reglas del fichero *fw-ejer8.txt*. Tras recargar la configuración del *firewall*, se obtiene el resultado esperado.

```
root@PC_EXT:~# wget -v -T 5 -t 1 10.10.10.30
converted 'http://10.10.10.30' (ANSI_X3.4-1968) -> 'http://10.10.10.30' (UTF-8)
--2021-05-01 15:16:40-- http://10.10.10.30/
Connecting to 10.10.10.30:80... failed: Connection timed out.
Giving up.

root@PC_EXT:~#
```

Figura 41: Error de acceso al servidor utilizando su IP privada.

## **Anexos**

## A. fw-rules.sh

```
#!/bin/sh
# FW con tres patas (DMZ, LAN, INTERNET)
#
#
# Configuración
# Interfaces
LO_IFACE=lo
LAN_IFACE=eth1
DMZ_IFACE=eth2
INTERNET_IFACE=eth0
#
# @IP interfaces
LO_IFACE_IP=127.0.0.1
FW_IFACE_IP=127.0.1.1
LAN_IFACE_IP=192.168.1.1
DMZ_IFACE_IP=10.10.10.1
INTERNET_IFACE_IP=203.0.113.10
#
# @IP redes
LAN_IP=192.168.1.0/24
DMZ_IP=10.10.10.0/24
ANY_IP=0.0.0.0/0
#
# Servicios
LAN_WEB_IP=192.168.1.10
DMZ_WEB_IP=10.10.10.30
#
# Comando
IPTABLES="/sbin/iptables"
#
##### Reglas iptables
#
### FILTER
# Vaciar reglas
$IPTABLES -F
$IPTABLES -X
$IPTABLES -Z
#
# INSERTAR AQUÍ fw-ejerc2.txt
#
$IPTABLES -P INPUT DROP
$IPTABLES -P OUTPUT DROP
$IPTABLES -P FORWARD DROP
#
# Reglas auxiliares
# desde la interface localhost a las @IP's locales
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LO_IFACE_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $FW_IFACE_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $LAN_IFACE_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $DMZ_IFACE_IP -j ACCEPT
$IPTABLES -A INPUT -p ALL -i $LO_IFACE -s $INTERNET_IFACE_IP -j ACCEPT
# Permitir salida solo por @IP's locales
```

```

$IPTABLES -A OUTPUT -p ALL -s $LO_IFACE_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $FW_IFACE_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $LAN_IFACE_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $DMZ_IFACE_IP -j ACCEPT
$IPTABLES -A OUTPUT -p ALL -s $INTERNET_IFACE_IP -j ACCEPT
#
##### INSERTAR AQU ^ fw-ejerc3.txt
$IPTABLES -A INPUT -i $LAN_IFACE -s $LAN_IP \
    -p icmp -m icmp --icmp-type echo-request \
    -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT -o $LAN_IFACE -d $LAN_IP \
    -p icmp -m icmp --icmp-type echo-reply \
    -m state --state RELATED,ESTABLISHED -j ACCEPT
$IPTABLES -A INPUT -i $DMZ_IFACE -s $DMZ_IP \
    -p icmp -m icmp --icmp-type echo-request \
    -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A OUTPUT -o $DMZ_IFACE -d $DMZ_IP \
    -p icmp -m icmp --icmp-type echo-reply \
    -m state --state RELATED,ESTABLISHED -j ACCEPT
#
#Reglas 3 C
$IPTABLES -A INPUT -i $LAN_IFACE -s $LAN_IP -d $DMZ_IP \
    -p icmp -m icmp --icmp-type echo-request \
    -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A INPUT -i $DMZ_IFACE -s $DMZ_IP -d $LAN_IP \
    -p icmp -m icmp --icmp-type echo-reply \
    -m state --state ESTABLISHED,RELATED -j ACCEPT

#Reglas 3 D

$IPTABLES -A FORWARD -i $LAN_IFACE -o $DMZ_IFACE \
    -p icmp -m icmp --icmp-type echo-request \
    -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -i $DMZ_IFACE -o $LAN_IFACE \
    -p icmp -m icmp --icmp-type echo-reply \
    -m state --state ESTABLISHED,RELATED -j ACCEPT

#####
##### INSERTAR AQU ^ fw-ejerc4.txt
$IPTABLES -A FORWARD -i $LAN_IFACE -s $LAN_IP \
    -o $DMZ_IFACE -d $DMZ_WEB_IP \
    -p tcp -m multiport --dport 80,443 -j ACCEPT
$IPTABLES -A FORWARD -o $LAN_IFACE -d $LAN_IP \
    -i $DMZ_IFACE -s $DMZ_WEB_IP \
    -p tcp -m multiport --sport 80,443 -j ACCEPT
#
## log all dropped packets in /var/log/messages
$IPTABLES -A INPUT -j LOG --log-prefix "INPUT DROPPED:"
$IPTABLES -A OUTPUT -j LOG --log-prefix "OUTPUT DROPPED:"
$IPTABLES -A FORWARD -j LOG --log-prefix "FORWARD DROPPED:"
#
#
#

```

```

#
#Insertar Aquí reglas ejercicio 6a

$IPTABLES -A FORWARD -i $LAN_IFACE -o $INTERNET_IFACE \
           -p icmp --icmp-type echo-request \
           -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNET_IFACE -o $LAN_IFACE \
           -p icmp --icmp-type echo-reply \
           -m state --state ESTABLISHED,RELATED -j ACCEPT

$IPTABLES -I OUTPUT 1 -o $LAN_IFACE -s $LAN_IP -d $ANY_IP \
           -p icmp -m icmp --icmp-type echo-request \
           -m state --state NEW,ESTABLISHED,RELATED -j ACCEPT
$IPTABLES -I INPUT 1 -i $INTERNET_IFACE -s $ANY_IP -d $LAN_IP \
           -p icmp -m icmp --icmp-type echo-reply \
           -m state --state ESTABLISHED,RELATED -j ACCEPT

#
### NAT
# Vaciar reglas
$IPTABLES -t nat -F
$IPTABLES -t nat -X
$IPTABLES -t nat -Z
#
# INSERTAR AQUÍ reglas auxiliares ejercicio 7a
$IPTABLES -A INPUT -i $INTERNET_IFACE -d $INTERNET_IFACE_IP \
           -p tcp -j ACCEPT
$IPTABLES -A OUTPUT -o $DMZ_IFACE -s $DMZ_IFACE_IP \
           -p tcp -j ACCEPT

$IPTABLES -A FORWARD -i $INTERNET_IFACE -o $DMZ_IFACE \
           -p tcp -j ACCEPT
$IPTABLES -A FORWARD -i $DMZ_IFACE -o $INTERNET_IFACE \
           -p tcp -j ACCEPT

# Reglas
#
# INSERTAR AQUÍ regla ejercicio 6b
#
$IPTABLES -t nat -A POSTROUTING -o $INTERNET_IFACE -p all \
           -s $LAN_IP -d $ANY_IP -j SNAT --to $INTERNET_IFACE_IP

#
# INSERTAR AQUÍ regla ejercicio 5
$IPTABLES -t nat -A POSTROUTING -o $DMZ_IFACE -p all \
           -s $LAN_IP -d $DMZ_IP -j SNAT --to $DMZ_IFACE_IP
# INSERTAR AQUÍ fw-ejerc7.txt
#
#
#
# Forwarding al servicio web en la DMZ
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
           -d $INTERNET_IFACE_IP -p tcp --dport 80 \

```

```

--sport 1024:65535 -j DNAT --to-destination $DMZ_WEB_IP:80
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
           -d $INTERNET_IFACE_IP -p tcp --dport 443 \
           --sport 1024:65535 -j DNAT --to-destination $DMZ_WEB_IP:443
#
##### INSERTAR AQUI fw-ejerc8.txt
# No permitir paquetes con @IPs privadas como destino
# aunque nunca deberian ser ruteables por Internet
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
           -d $DMZ_IP -j DNAT --to $LO_IFACE_IP
$IPTABLES -t nat -A PREROUTING -i $INTERNET_IFACE \
           -d $LAN_IP -j DNAT --to $LO_IFACE_IP
#
# Reglas auxiliares
# Validar que en loopback solo llegan paquetes de las @IP's locales
$IPTABLES -t nat -A PREROUTING -i $LO_IFACE -s $LO_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $LO_IFACE -s $FW_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $LO_IFACE -s $LAN_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $LO_IFACE -s $DMZ_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $LO_IFACE -s $INTERNET_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -s $LO_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -s $FW_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -s $LAN_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -s $DMZ_IFACE_IP -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -s $INTERNET_IFACE_IP -j ACCEPT
#
# Validar que en las interfaces llegan paquetes de la red correspondiente
$IPTABLES -t nat -A PREROUTING -i $LAN_IFACE -s $LAN_IP -j ACCEPT
$IPTABLES -t nat -A PREROUTING -i $DMZ_IFACE -s $DMZ_IP -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -o $LAN_IFACE -d $LAN_IP -j ACCEPT
$IPTABLES -t nat -A POSTROUTING -o $DMZ_IFACE -d $DMZ_IP -j ACCEPT
#
#

```