Assignment 4 – Linux Firewall

Task 1. Find IP addresses

a) Find the IP address of the client and the firewall.
b) Show the addresses in screenshots.

```
ubuntu@firewall: ~                    ubuntu@firewall: ~
ubuntu@firewall:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.24.0.4  netmask 255.255.255.0  broadcast 172.24.0.255
        ether 02:42:ac:18:00:04  txqueuelen 0  (Ethernet)
        RX packets 51  bytes 6815 (6.8 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.25.0.4  netmask 255.255.255.0  broadcast 172.25.0.255
        ether 02:42:ac:19:00:04  txqueuelen 0  (Ethernet)
        RX packets 36  bytes 4553 (4.5 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ubuntu@firewall:~$
```

```
ubuntu@client:~$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.24.0.3  netmask 255.255.255.0  broadcast 172.24.0.255
        ether 02:42:ac:18:00:03  txqueuelen 0  (Ethernet)
        RX packets 54  bytes 7093 (7.0 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

ubuntu@client:~$
```

Task 2. Nmap scan

a) Perform a nmap scan on the client for open ports on the server. Show the output in a screenshot.

```
ubuntu@client:~$ nmap server
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-24 22:50 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.00022s latency).
Not shown: 997 closed ports
PORT   STATE SERVICE
22/tcp open  ssh
23/tcp open  telnet
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
ubuntu@client:~$
```

b) Run *wget* and report captured packets on Wireshark in a screenshot. To capture packets for a new command, you need to stop/start capturing without exiting Wireshark.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.24.0.101	224.0.0.251	MDNS	87	Standard query 0x0000
2	17.085913580	fe80::8d6:b6ff:fe5f…	ff02::fb	MDNS	107	Standard query 0x0000
3	17.415476121	fe80::42:7aff:fedd:…	ff02::fb	MDNS	107	Standard query 0x0000
4	61.966814006	172.24.0.3	172.25.0.3	TCP	74	59860 → 80 [SYN] Seq=0
5	61.966900592	172.25.0.3	172.24.0.3	TCP	74	80 → 59860 [SYN, ACK]
6	61.966943177	172.24.0.3	172.25.0.3	TCP	66	59860 → 80 [ACK] Seq=1
7	61.969744528	172.24.0.3	172.25.0.3	HTTP	199	GET / HTTP/1.1
8	61.969790541	172.25.0.3	172.24.0.3	TCP	66	80 → 59860 [ACK] Seq=1
9	61.972267825	172.25.0.3	172.24.0.3	TCP	83	80 → 59860 [PSH, ACK]
10	61.972468480	172.25.0.3	172.24.0.3	HTTP	1078	HTTP/1.0 200 OK (text
11	61.972561583	172.24.0.3	172.25.0.3	TCP	66	59860 → 80 [ACK] Seq=1
12	61.972596220	172.24.0.3	172.25.0.3	TCP	66	59860 → 80 [ACK] Seq=1
13	61.973645032	172.24.0.3	172.25.0.3	TCP	66	59860 → 80 [FIN, ACK]
14	61.973699551	172.25.0.3	172.24.0.3	TCP	66	80 → 59860 [ACK] Seq=1
15	64.404396408	fe80::8d6:b6ff:fe5f…	ff02::2	ICMPv6	70	Router Solicitation fr
16	67.229410784	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	Who has 172.24.0.3? Te
17	67.230075870	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	Who has 172.24.0.4? Te
18	67.230149073	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	172.24.0.4 is at 02:42
19	67.230157797	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	172.24.0.3 is at 02:42

c) Run *ssh* and report captured packets on Wireshark in a screenshot.

20	129.334460116	172.24.0.3	172.25.0.3	TCP	74	36294 → 22 [SYN] Sec
21	129.334514792	172.25.0.3	172.24.0.3	TCP	74	22 → 36294 [SYN, ACK
22	129.334541110	172.24.0.3	172.25.0.3	TCP	66	36294 → 22 [ACK] Sec
23	129.343186496	172.24.0.3	172.25.0.3	SSHv2	107	Client: Protocol (SS
24	129.343242185	172.25.0.3	172.24.0.3	TCP	66	22 → 36294 [ACK] Sec
25	129.368225186	172.25.0.3	172.24.0.3	SSHv2	107	Server: Protocol (SS
26	129.368275663	172.24.0.3	172.25.0.3	TCP	66	36294 → 22 [ACK] Sec
27	129.368643973	172.24.0.3	172.25.0.3	SSHv2	1578	Client: Key Exchange
28	129.368716156	172.25.0.3	172.24.0.3	TCP	66	22 → 36294 [ACK] Sec
29	129.370941365	172.25.0.3	172.24.0.3	SSHv2	1122	Server: Key Exchange
30	129.375443466	172.24.0.3	172.25.0.3	SSHv2	114	Client: Diffie-Hellm
31	129.398215595	172.25.0.3	172.24.0.3	SSHv2	574	Server: Diffie-Hellm
32	129.439817505	172.24.0.3	172.25.0.3	TCP	66	36294 → 22 [ACK] Sec
33	132.592458234	172.24.0.3	172.25.0.3	TCP	66	36294 → 22 [FIN, ACK
34	132.596718168	172.25.0.3	172.24.0.3	TCP	66	22 → 36294 [FIN, ACK
35	132.596767826	172.24.0.3	172.25.0.3	TCP	66	36294 → 22 [ACK] Sec

TCP	74	36294 → 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
TCP	74	22 → 36294 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA…
TCP	66	36294 → 22 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=2025395967…
SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
TCP	66	22 → 36294 [ACK] Seq=1 Ack=42 Win=29056 Len=0 TSval=501631280…
SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
TCP	66	36294 → 22 [ACK] Seq=42 Ack=42 Win=29312 Len=0 TSval=20253960…
SSHv2	1578	Client: Key Exchange Init
TCP	66	22 → 36294 [ACK] Seq=42 Ack=1554 Win=32000 Len=0 TSval=501631…
SSHv2	1122	Server: Key Exchange Init
SSHv2	114	Client: Diffie-Hellman Key Exchange Init
SSHv2	574	Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypte…
TCP	66	36294 → 22 [ACK] Seq=1602 Ack=1606 Win=33536 Len=0 TSval=2025…
TCP	66	36294 → 22 [FIN, ACK] Seq=1602 Ack=1606 Win=33536 Len=0 TSval…
TCP	66	22 → 36294 [FIN, ACK] Seq=1606 Ack=1603 Win=32000 Len=0 TSval…
TCP	66	36294 → 22 [ACK] Seq=1603 Ack=1607 Win=33536 Len=0 TSval=2025…

d) Run *telnet* and report captured packets on Wireshark in a screenshot.

36	251.098401228	172.24.0.3	172.25.0.3	TCP	74	42040 → 23 [SYN] Sec
37	251.098555473	172.25.0.3	172.24.0.3	TCP	74	23 → 42040 [SYN, ACK
38	251.098628353	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
39	251.105181051	172.24.0.3	172.25.0.3	TELNET	93	Telnet Data ...
40	251.105308994	172.25.0.3	172.24.0.3	TCP	66	23 → 42040 [ACK] Sec
41	251.209326303	172.25.0.3	172.24.0.3	TELNET	78	Telnet Data ...
42	251.209365582	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
43	251.209442913	172.25.0.3	172.24.0.3	TELNET	105	Telnet Data ...
44	251.209469075	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
45	251.209697533	172.24.0.3	172.25.0.3	TELNET	140	Telnet Data ...
46	251.209748250	172.25.0.3	172.24.0.3	TCP	66	23 → 42040 [ACK] Sec
47	251.210343675	172.25.0.3	172.24.0.3	TELNET	69	Telnet Data ...
48	251.210427426	172.24.0.3	172.25.0.3	TELNET	69	Telnet Data ...
49	251.210694853	172.25.0.3	172.24.0.3	TELNET	69	Telnet Data ...
50	251.210771689	172.24.0.3	172.25.0.3	TELNET	69	Telnet Data ...
51	251.210829112	172.25.0.3	172.24.0.3	TELNET	86	Telnet Data ...
52	251.253897853	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
53	251.254060650	172.25.0.3	172.24.0.3	TELNET	80	Telnet Data ...
54	251.254088389	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
55	253.106078750	172.24.0.3	172.25.0.3	TELNET	67	Telnet Data ...
56	253.106492157	172.25.0.3	172.24.0.3	TELNET	67	Telnet Data ...[Malf

57	253.106568262	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
58	253.106662273	172.25.0.3	172.24.0.3	TELNET	67	Telnet Data ...
59	253.106706949	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
60	253.106797489	172.25.0.3	172.24.0.3	TELNET	68	Telnet Data ...
61	253.106835465	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [ACK] Sec
62	253.115758611	172.25.0.3	172.24.0.3	TCP	66	23 → 42040 [FIN, ACK
63	253.115921106	172.24.0.3	172.25.0.3	TCP	66	42040 → 23 [FIN, ACK
64	253.115996653	172.25.0.3	172.24.0.3	TCP	66	23 → 42040 [ACK] Sec

Task 3. Use iptables to limit traffic to the server

a) Show that ssh traffic is allowed. On the client, run ssh while capturing traffic on the firewall. Report these two activities in two screenshots. Explain how you know ssh traffic is allowed.

Wireshark View:

1	0.000000000	172.24.0.3	172.25.0.3	TCP	74	40384 → 22 [SYN] Seq=0 Win=2
2	0.000090810	172.25.0.3	172.24.0.3	TCP	74	22 → 40384 [SYN, ACK] Seq=0
3	0.000123436	172.24.0.3	172.25.0.3	TCP	66	40384 → 22 [ACK] Seq=1 Ack=1
4	0.012057082	172.24.0.3	172.25.0.3	SSHv2	107	Client: Protocol (SSH-2.0-Op
5	0.012114062	172.25.0.3	172.24.0.3	TCP	66	22 → 40384 [ACK] Seq=1 Ack=4
6	0.037288138	172.25.0.3	172.24.0.3	SSHv2	107	Server: Protocol (SSH-2.0-Op
7	0.037325752	172.24.0.3	172.25.0.3	TCP	66	40384 → 22 [ACK] Seq=42 Ack=
8	0.037689212	172.24.0.3	172.25.0.3	SSHv2	1578	Client: Key Exchange Init
9	0.040652928	172.25.0.3	172.24.0.3	TCP	66	22 → 40384 [ACK] Seq=42 Ack=
10	0.100613993	172.25.0.3	172.24.0.3	SSHv2	1122	Server: Key Exchange Init
11	0.106191904	172.24.0.3	172.25.0.3	SSHv2	114	Client: Diffie-Hellman Key E
12	0.116074124	172.25.0.3	172.24.0.3	SSHv2	574	Server: Diffie-Hellman Key E
13	0.158644483	172.24.0.3	172.25.0.3	TCP	66	40384 → 22 [ACK] Seq=1602 Ac
14	41.836630667	172.24.0.3	172.25.0.3	TCP	66	40384 → 22 [FIN, ACK] Seq=16
15	41.838829590	172.25.0.3	172.24.0.3	TCP	66	22 → 40384 [FIN, ACK] Seq=16
16	41.838923047	172.24.0.3	172.25.0.3	TCP	66	40384 → 22 [ACK] Seq=1603 Ac
17	47.078830203	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	Who has 172.24.0.3? Tell 172
18	47.079438216	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	Who has 172.24.0.4? Tell 172
19	47.079991099	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	172.24.0.4 is at 02:42:ac:18
20	47.080018167	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	172.24.0.3 is at 02:42:ac:18

Firewall IPTABLES View:

```
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere

Chain FORWARD (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  anywhere             anywhere             ctstate RELATED,E
STABLISHED
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:ssh
ACCEPT     tcp  --  anywhere             anywhere             tcp dpt:http
NFLOG      all  --  anywhere             anywhere             limit: avg 2/min
burst 5 nflog-prefix  "IPTABLES DROPPED"

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ubuntu@firewall:~$
```

Client View:

```
ubuntu@client:~$ ssh server
The authenticity of host 'server (172.25.0.3)' can't be established.
ECDSA key fingerprint is SHA256:ZtE8xi5Y50aUktZ/XtgjIs1c5jxYQB84Vq5ofmlgGng.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'server,172.25.0.3' (ECDSA) to the list of known hos
ts.
ubuntu@server's password:
```

I have given multiple screenshots displaying rules and an established connection with ssh. To begin, we see completed three-way handshakes in wireshark. This shows that nothing is blocking the client from ssh. Additionally, we see a successful ssh connection with the client terminal. If ssh wasn't allowed, then wireshark would show unfinished handshakes and connections with the client terminal.

b) Show that HTTP traffic is allowed. Report the same as you did for ssh traffic.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	02:42:ac:18:00:03	Broadcast	ARP	42	Who has 172.24.0.101
2	2173.1683454…	172.24.0.3	172.25.0.3	TCP	74	35972 → 80 [SYN] Sec
3	2173.1689697…	172.25.0.3	172.24.0.3	TCP	74	80 → 35972 [SYN, ACK
4	2173.1690579…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
5	2173.1711179…	172.24.0.3	172.25.0.3	HTTP	199	GET / HTTP/1.1
6	2173.1712115…	172.25.0.3	172.24.0.3	TCP	66	80 → 35972 [ACK] Sec
7	2173.1944531…	172.25.0.3	172.24.0.3	TCP	83	80 → 35972 [PSH, ACK
8	2173.1945028…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
9	2173.1946617…	172.25.0.3	172.24.0.3	TCP	104	80 → 35972 [PSH, ACK
10	2173.1946816…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
11	2173.1949138…	172.25.0.3	172.24.0.3	TCP	103	80 → 35972 [PSH, ACK
12	2173.1949362…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
13	2173.1950031…	172.25.0.3	172.24.0.3	TCP	106	80 → 35972 [PSH, ACK
14	2173.1950198…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
15	2173.1950687…	172.25.0.3	172.24.0.3	TCP	87	80 → 35972 [PSH, ACK
16	2173.1950848…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
17	2173.1951300…	172.25.0.3	172.24.0.3	TCP	68	80 → 35972 [PSH, ACK
18	2173.1951461…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
19	2173.1966884…	172.25.0.3	172.24.0.3	HTTP	940	HTTP/1.0 200 OK (te
20	2173.1967371…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
21	2173.1975988…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [FIN, ACK

14	2173.1950198…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
15	2173.1950687…	172.25.0.3	172.24.0.3	TCP	87	80 → 35972 [PSH, ACK
16	2173.1950848…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
17	2173.1951300…	172.25.0.3	172.24.0.3	TCP	68	80 → 35972 [PSH, ACK
18	2173.1951461…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
19	2173.1966884…	172.25.0.3	172.24.0.3	HTTP	940	HTTP/1.0 200 OK (te
20	2173.1967371…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
21	2173.1975988…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [FIN, ACK
22	2173.2020987…	172.25.0.3	172.24.0.3	TCP	66	80 → 35972 [FIN, ACK
23	2173.2021641…	172.24.0.3	172.25.0.3	TCP	66	35972 → 80 [ACK] Sec
24	2178.3089032…	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	Who has 172.24.0.3?
25	2178.3091712…	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	Who has 172.24.0.4?
26	2178.3092080…	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	172.24.0.4 is at 02:
27	2178.3092114…	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	172.24.0.3 is at 02:
28	2220.4963001…	172.24.0.101	224.0.0.251	MDNS	87	Standard query 0x000
29	2237.3703296…	fe80::8d6:b6ff:fe5f…	ff02::fb	MDNS	107	Standard query 0x000
30	2237.7018484…	fe80::42:7aff:fedd:…	ff02::fb	MDNS	107	Standard query 0x000

```
ubuntu@client:~$ wget server
--2023-10-25 02:51:10--  http://server/
Resolving server (server)... 172.25.0.3
Connecting to server (server)|172.25.0.3|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 874 [text/html]
Saving to: 'index.html.1'

index.html.1                0%[                                    ]       0  --.
index.html.1              100%[===================================>]     874  --.
-KB/s    in 0.001s

2023-10-25 02:51:10 (627 KB/s) - 'index.html.1' saved [874/874]

ubuntu@client:~$
```

The screenshots above show that http connections are allowed. Wireshark is accepting packets and has completed three-way handshakes without any issues. Additionally, the client terminal shows a successful connection using http.

c) Show that telnet traffic is blocked. Report the same as you did for ssh traffic.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.24.0.3	172.25.0.3	TCP	74	46134 → 23 [SYN] Seq=0
2	1.005821393	172.24.0.3	172.25.0.3	TCP	74	[TCP Retransmission] 4
3	3.020192009	172.24.0.3	172.25.0.3	TCP	74	[TCP Retransmission] 4
4	5.004277895	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	Who has 172.24.0.4? Te
5	5.004310619	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	172.24.0.4 is at 02:42
6	7.052558268	172.24.0.3	172.25.0.3	TCP	74	[TCP Retransmission] 4
7	15.244462920	172.24.0.3	172.25.0.3	TCP	74	[TCP Retransmission] 4
8	31.372811548	172.24.0.3	172.25.0.3	TCP	74	[TCP Retransmission] 4

Protocol	Length	Info
TCP	74	46134 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
TCP	74	[TCP Retransmission] 46134 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
TCP	74	[TCP Retransmission] 46134 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
ARP	42	Who has 172.24.0.4? Tell 172.24.0.3
ARP	42	172.24.0.4 is at 02:42:ac:18:00:04
TCP	74	[TCP Retransmission] 46134 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
TCP	74	[TCP Retransmission] 46134 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
TCP	74	[TCP Retransmission] 46134 → 23 [SYN] Seq=0 Win=29200 Len=0 M…

```
ubuntu@client:~$ telnet server
Trying 172.25.0.3...
^C
ubuntu@client:~$
```

Looking over wireshark, we see that it is attempting to establish a connection with the beginning SYN packet. However, the firewall has blocked any traffic from telnet, leading to TCP retransmissions.

Multiple packets have sent with no output from the client terminal, which shows that it is blocked. Additionally, the iptables rules above show that any traffic is blocked with the exception of HTTP and SSH.

d) At the end, perform a nmap scan on the client for open ports on the server. Show the output in a screenshot.

```
ubuntu@client:~$ nmap server
Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-25 02:57 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.0043s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 4.54 seconds
ubuntu@client:~$
```

Task 4. Open a new service port

a) Show that wizbang traffic is allowed. On the client, run wizbang while capturing traffic on the firewall. Report these two activities in two screenshots. Explain how you know wizbang traffic is allowed.

```
ubuntu@client:~$ sudo ./wizbang Good Morning
^Cubuntu@client:~$ Interrupted, exiting
⋆n
sudo ./wizbang Good Morning
Sending instruction Good Morning
bye
ubuntu@client:~$
```

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.24.0.3	172.25.0.3	TCP	74	36586 → 10063 [SYN] Se
2	0.000242525	172.25.0.3	172.24.0.3	TCP	74	10063 → 36586 [SYN, AC
3	0.000295587	172.24.0.3	172.25.0.3	TCP	66	36586 → 10063 [ACK] Se
4	0.008139333	172.24.0.3	172.25.0.3	TCP	79	36586 → 10063 [PSH, AC
5	0.008204026	172.25.0.3	172.24.0.3	TCP	66	10063 → 36586 [ACK] Se
6	0.008715019	172.24.0.3	172.25.0.3	TCP	66	36586 → 10063 [FIN, AC
7	0.009358604	172.25.0.3	172.24.0.3	TCP	66	10063 → 36586 [FIN, AC
8	0.009400641	172.24.0.3	172.25.0.3	TCP	66	36586 → 10063 [ACK] Se
9	5.128532870	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	Who has 172.24.0.3? Te
10	5.128673965	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	Who has 172.24.0.4? Te
11	5.128710256	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	172.24.0.4 is at 02:42
12	5.128714202	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	172.24.0.3 is at 02:42

Wizbang runs off port 10063, so I had to create a rule to allow traffic with that port. After running the client command, a connection was established and responded with “bye”. This shows that traffic was established. Furthermore, we see that Wireshark has completed three-way handshakes with Wizbang.

b) At the end, perform a nmap scan on the client for open ports on the server. Show the output in a screenshot.

```
Nmap done: 1 IP address (1 host up) scanned in 5.13 seconds
ubuntu@client:~$ Starting Nmap 7.80 ( https://nmap.org ) at 2023-10-25 07:50 UT
C
Nmap scan report for server (172.25.0.3)
Host is up (0.0017s latency).
Not shown: 10061 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
10063/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 25.07 seconds
ubuntu@client:~$
```