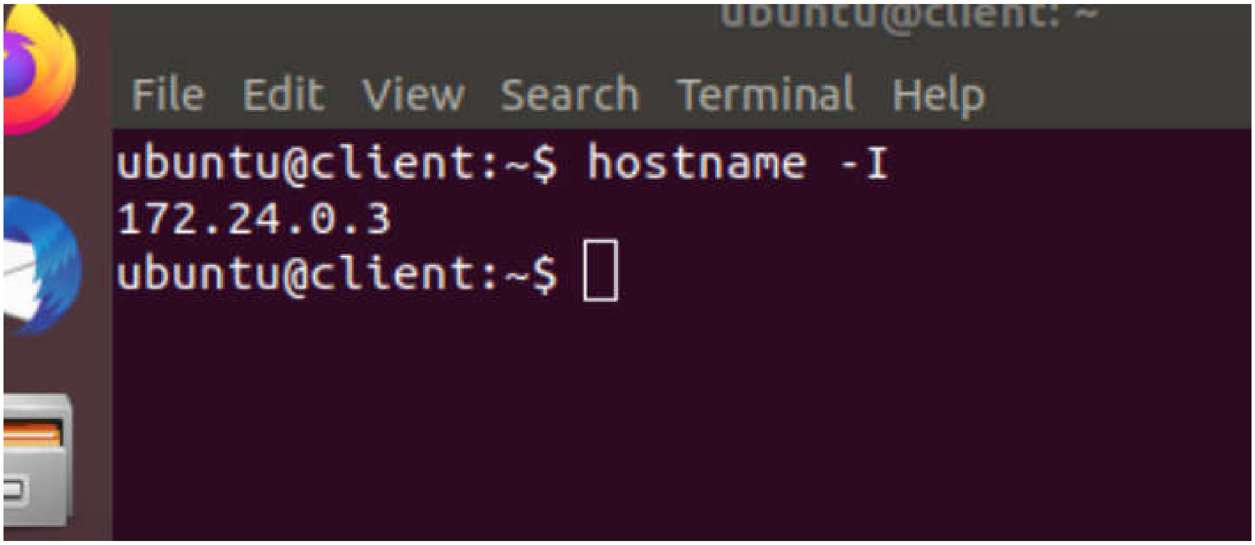
Homework 4 – Linux Firewall

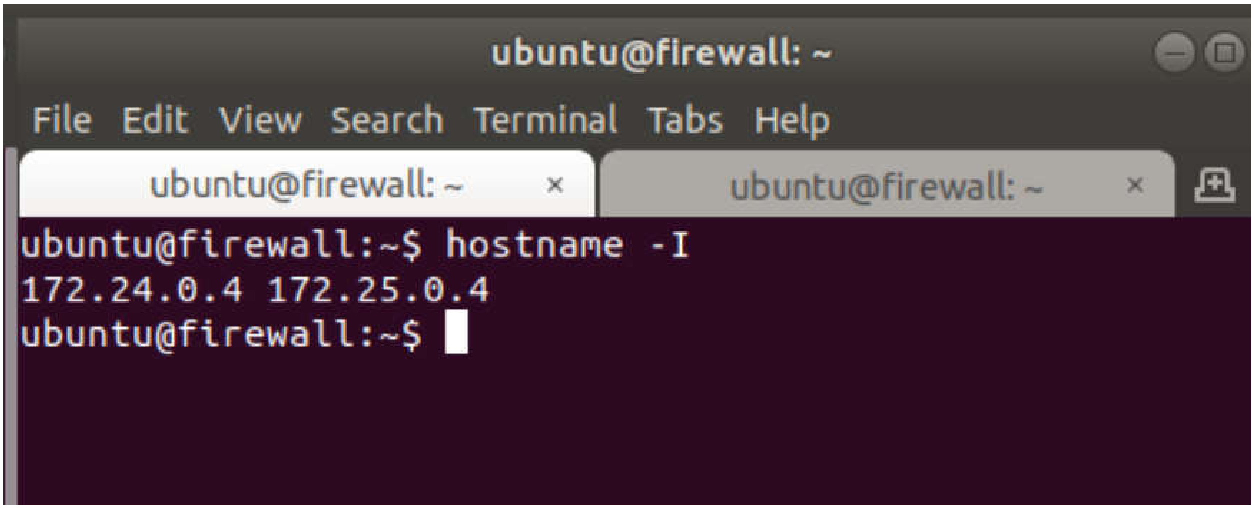
Task 1. Find IP addresses

a) Find the IP address of the client and the firewall.

b) Show the addresses in screenshots.



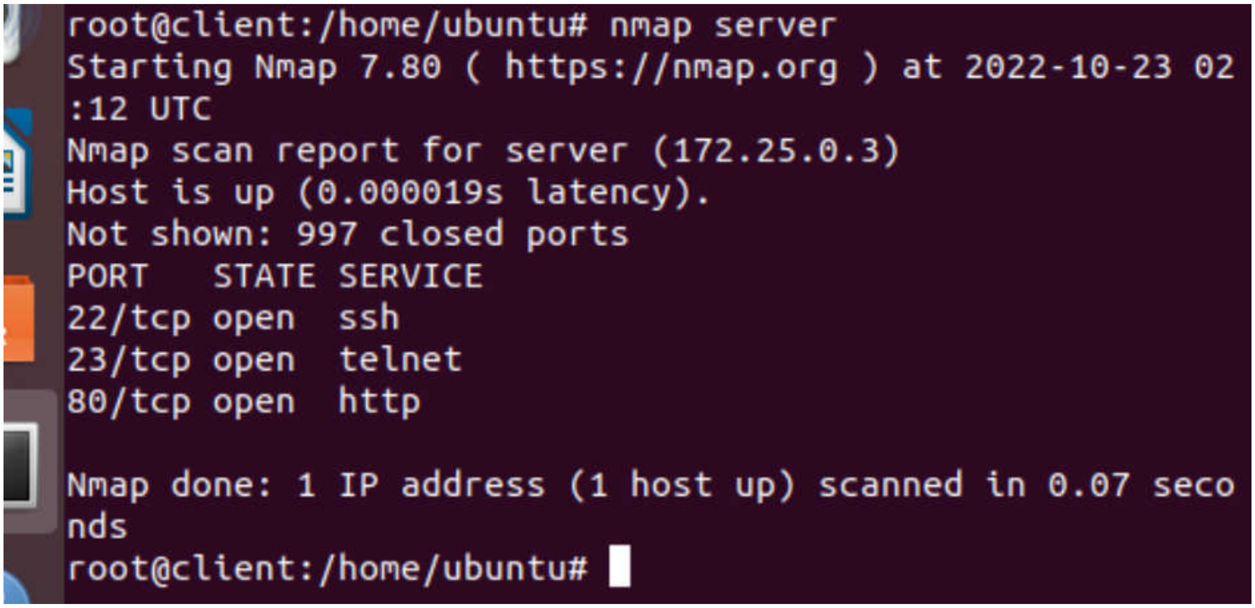
```
ubuntu@client:~$ hostname -I
172.24.0.3
ubuntu@client:~$ 
```



```
ubuntu@firewall:~$ hostname -I
172.24.0.4 172.25.0.4
ubuntu@firewall:~$ 
```

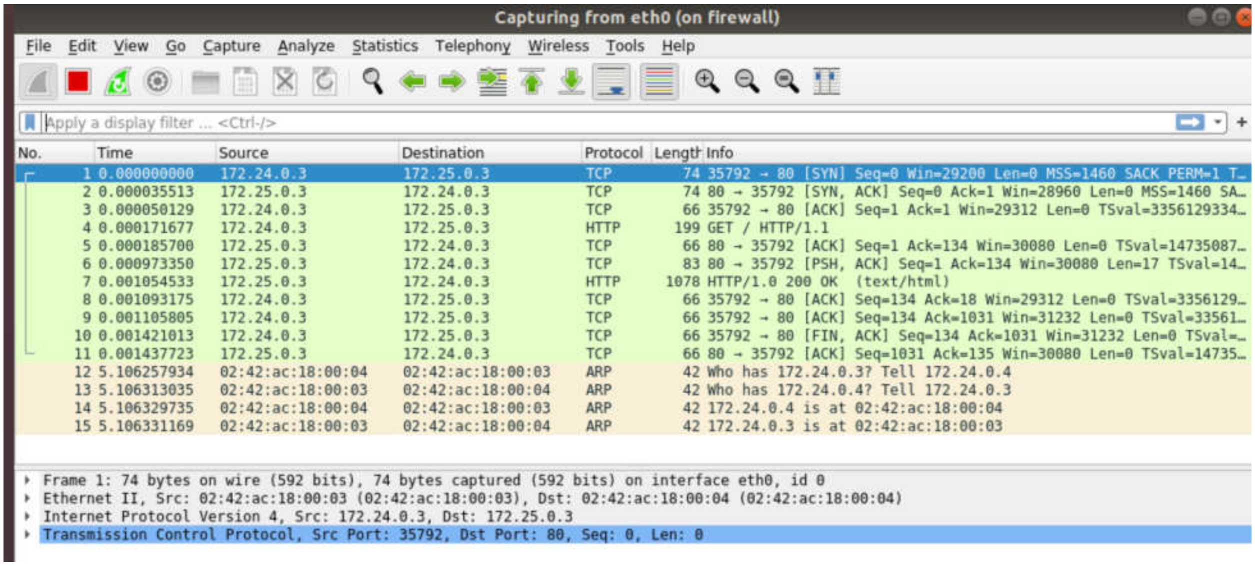
Task 2. Nmap scan

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



```
root@client:/home/ubuntu# nmap server
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-23 02
:12 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.000019s 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 0.07 seco
nds
root@client:/home/ubuntu# 
```

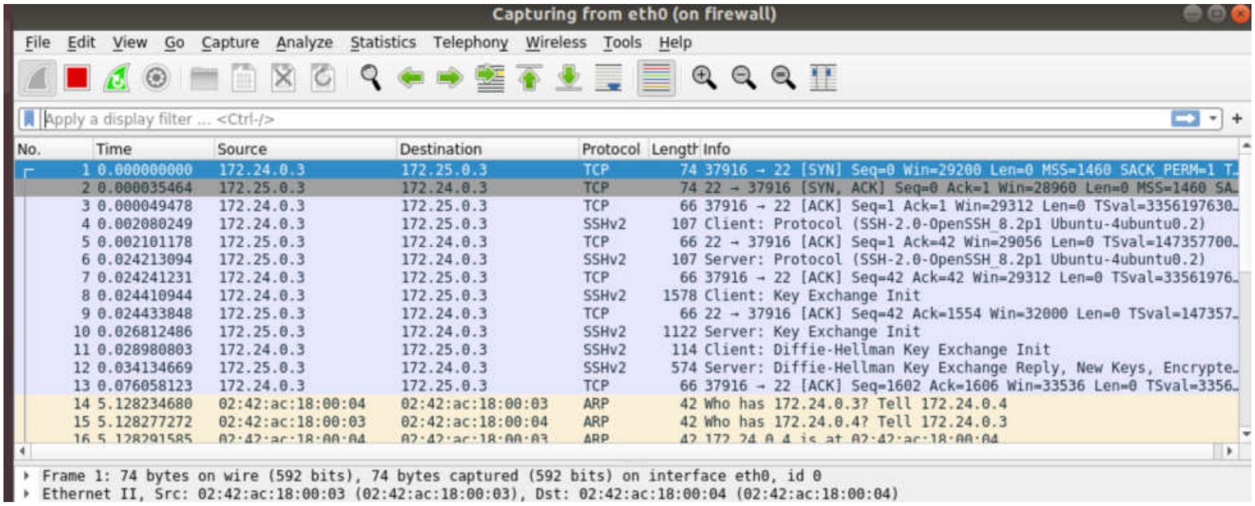
b) Run *wget* and report captured packets on wireshark in a screenshot. To capture packets for a new command, you need to stop/start capturing without exiting wireshark.


Capturing from eth0 (on firewall)
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.24.0.3	172.25.0.3	TCP	74	35792 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
2	0.000035513	172.25.0.3	172.24.0.3	TCP	74	80 → 35792 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA…
3	0.000050129	172.24.0.3	172.25.0.3	TCP	66	35792 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3356129334…
4	0.000171677	172.24.0.3	172.25.0.3	HTTP	199	GET / HTTP/1.1
5	0.000185700	172.25.0.3	172.24.0.3	TCP	66	80 → 35792 [ACK] Seq=1 Ack=134 Win=30080 Len=0 TSval=14735087…
6	0.000973350	172.25.0.3	172.24.0.3	TCP	83	80 → 35792 [PSH, ACK] Seq=1 Ack=134 Win=30080 Len=17 TSval=14…
7	0.001054533	172.25.0.3	172.24.0.3	HTTP	1078	HTTP/1.0 200 OK (text/html)
8	0.001093175	172.24.0.3	172.25.0.3	TCP	66	35792 → 80 [ACK] Seq=134 Ack=18 Win=29312 Len=0 TSval=3356129…
9	0.001105805	172.24.0.3	172.25.0.3	TCP	66	35792 → 80 [ACK] Seq=134 Ack=1031 Win=31232 Len=0 TSval=33561…
10	0.001421013	172.24.0.3	172.25.0.3	TCP	66	35792 → 80 [FIN, ACK] Seq=134 Ack=1031 Win=31232 Len=0 TSval=…
11	0.001437723	172.25.0.3	172.24.0.3	TCP	66	80 → 35792 [ACK] Seq=1031 Ack=135 Win=30080 Len=0 TSval=14735…
12	5.106257934	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	Who has 172.24.0.3? Tell 172.24.0.4
13	5.106313035	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	Who has 172.24.0.4? Tell 172.24.0.3
14	5.106329735	02:42:ac:18:00:04	02:42:ac:18:00:03	ARP	42	172.24.0.4 is at 02:42:ac:18:00:04
15	5.106331169	02:42:ac:18:00:03	02:42:ac:18:00:04	ARP	42	172.24.0.3 is at 02:42:ac:18:00:03

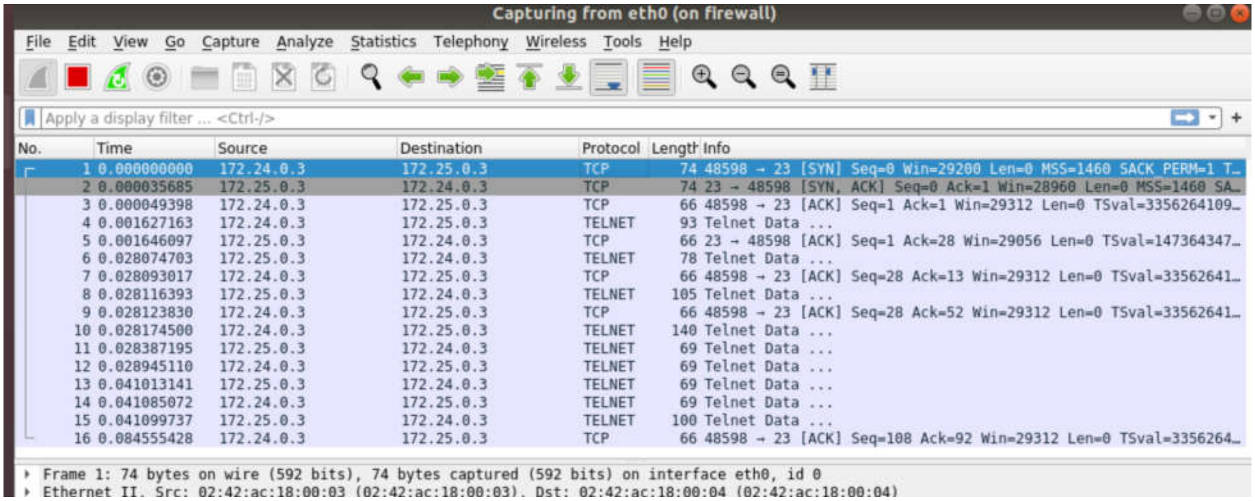
```
▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
▸ Ethernet II, Src: 02:42:ac:18:00:03 (02:42:ac:18:00:03), Dst: 02:42:ac:18:00:04 (02:42:ac:18:00:04)
▸ Internet Protocol Version 4, Src: 172.24.0.3, Dst: 172.25.0.3
▸ Transmission Control Protocol, Src Port: 35792, Dst Port: 80, Seq: 0, Len: 0
```

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


Capturing from eth0 (on firewall)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time          Source             Destination        Protocol Length Info
   1 0.000000000   172.24.0.3         172.25.0.3         TCP        74 37916 → 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
   2 0.000035464   172.25.0.3         172.24.0.3         TCP        74 22 → 37916 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA…
   3 0.000049478   172.24.0.3         172.25.0.3         TCP        66 37916 → 22 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3356197630…
   4 0.002080249   172.24.0.3         172.25.0.3         SSHv2     107 Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
   5 0.002101178   172.25.0.3         172.24.0.3         TCP        66 22 → 37916 [ACK] Seq=1 Ack=42 Win=29056 Len=0 TSval=147357700…
   6 0.024213094   172.25.0.3         172.24.0.3         SSHv2     107 Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
   7 0.024241231   172.24.0.3         172.25.0.3         TCP        66 37916 → 22 [ACK] Seq=42 Ack=42 Win=29312 Len=0 TSval=33561976…
   8 0.024410944   172.24.0.3         172.25.0.3         SSHv2    1578 Client: Key Exchange Init
   9 0.024433848   172.25.0.3         172.24.0.3         TCP        66 22 → 37916 [ACK] Seq=42 Ack=1554 Win=32000 Len=0 TSval=147357…
  10 0.026812486   172.25.0.3         172.24.0.3         SSHv2    1122 Server: Key Exchange Init
  11 0.028980803   172.24.0.3         172.25.0.3         SSHv2     114 Client: Diffie-Hellman Key Exchange Init
  12 0.034134669   172.25.0.3         172.24.0.3         SSHv2     574 Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypte…
  13 0.076058123   172.24.0.3         172.25.0.3         TCP        66 37916 → 22 [ACK] Seq=1602 Ack=1606 Win=33536 Len=0 TSval=3356…
  14 5.128234680   02:42:ac:18:00:04  02:42:ac:18:00:03  ARP        42 Who has 172.24.0.3? Tell 172.24.0.4
  15 5.128277272   02:42:ac:18:00:03  02:42:ac:18:00:04  ARP        42 Who has 172.24.0.4? Tell 172.24.0.3
  16 5.128291585   02:42:ac:18:00:04  02:42:ac:18:00:03  ARP        42 172.24.0.4 is at 02:42:ac:18:00:04

▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
▸ Ethernet II, Src: 02:42:ac:18:00:03 (02:42:ac:18:00:03), Dst: 02:42:ac:18:00:04 (02:42:ac:18:00:04)
```

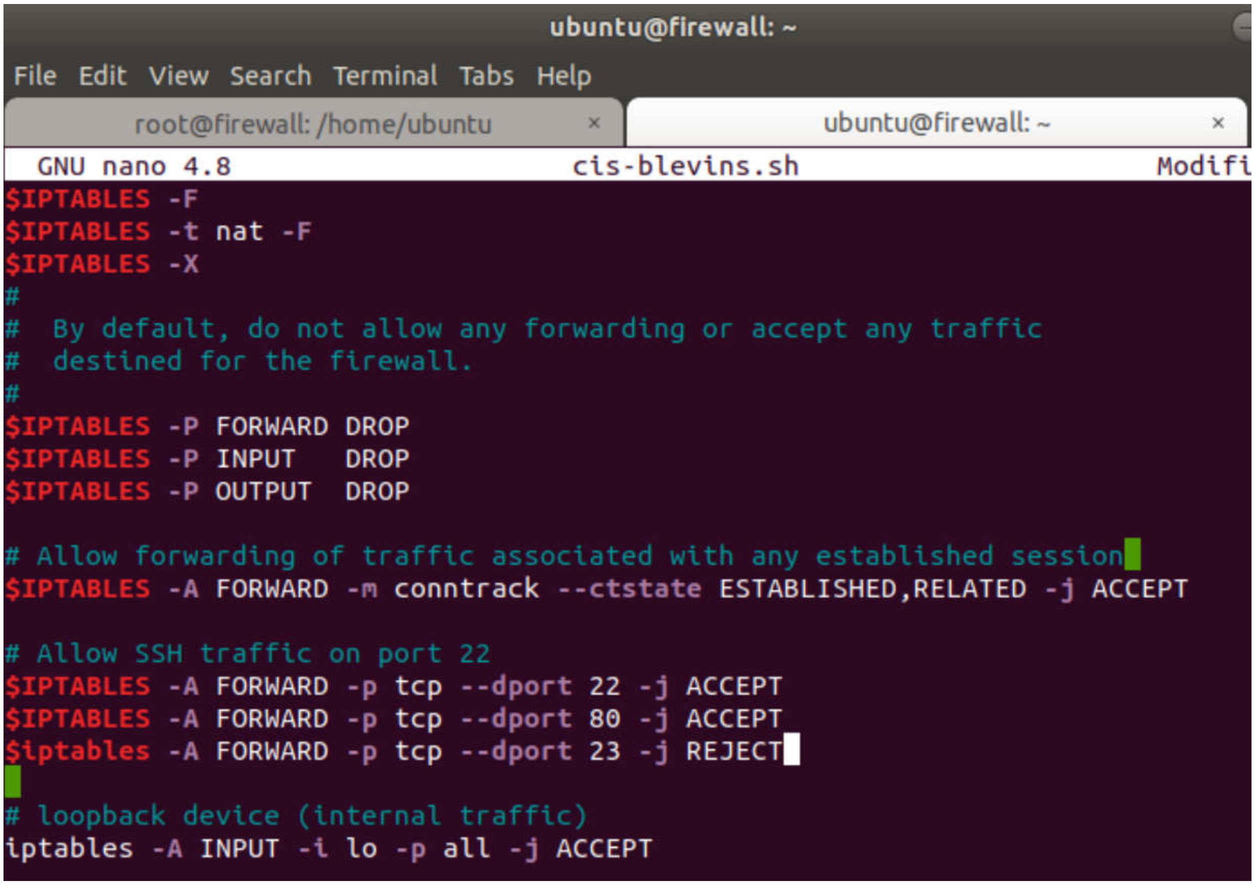
d) Run *telnet* and report captured packets on wireshark in a screenshot.


Capturing from eth0 (on firewall)
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

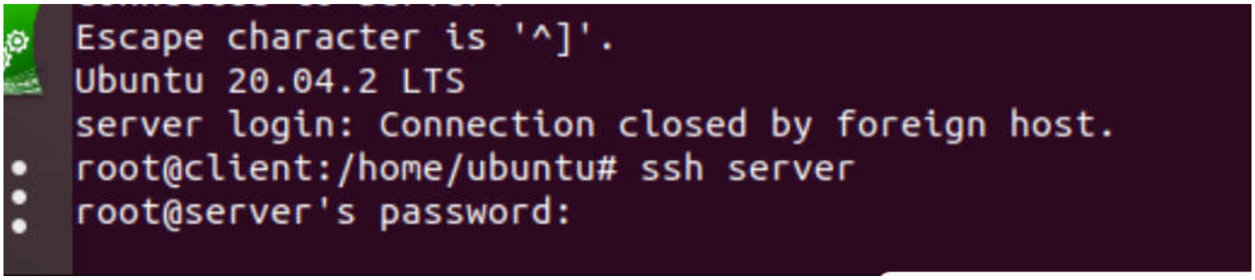
```
No.  Time          Source       Destination  Protocol Length Info
   1 0.000000000   172.24.0.3   172.25.0.3   TCP        74 48598 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
   2 0.000035685   172.25.0.3   172.24.0.3   TCP        74 23 → 48598 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA…
   3 0.000049398   172.24.0.3   172.25.0.3   TCP        66 48598 → 23 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3356264109…
   4 0.001627163   172.24.0.3   172.25.0.3   TELNET     93 Telnet Data ...
   5 0.001646097   172.25.0.3   172.24.0.3   TCP        66 23 → 48598 [ACK] Seq=1 Ack=28 Win=29056 Len=0 TSval=147364347…
   6 0.028074703   172.25.0.3   172.24.0.3   TELNET     78 Telnet Data ...
   7 0.028093017   172.24.0.3   172.25.0.3   TCP        66 48598 → 23 [ACK] Seq=28 Ack=13 Win=29312 Len=0 TSval=33562641…
   8 0.028116393   172.25.0.3   172.24.0.3   TELNET    105 Telnet Data ...
   9 0.028123830   172.24.0.3   172.25.0.3   TCP        66 48598 → 23 [ACK] Seq=28 Ack=52 Win=29312 Len=0 TSval=33562641…
  10 0.028174500   172.24.0.3   172.25.0.3   TELNET    140 Telnet Data ...
  11 0.028387195   172.25.0.3   172.24.0.3   TELNET     69 Telnet Data ...
  12 0.028945110   172.24.0.3   172.25.0.3   TELNET     69 Telnet Data ...
  13 0.041013141   172.25.0.3   172.24.0.3   TELNET     69 Telnet Data ...
  14 0.041085072   172.24.0.3   172.25.0.3   TELNET     69 Telnet Data ...
  15 0.041099737   172.25.0.3   172.24.0.3   TELNET    100 Telnet Data ...
  16 0.084555428   172.24.0.3   172.25.0.3   TCP        66 48598 → 23 [ACK] Seq=108 Ack=92 Win=29312 Len=0 TSval=3356264…

▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
▸ Ethernet II, Src: 02:42:ac:18:00:03 (02:42:ac:18:00:03), Dst: 02:42:ac:18:00:04 (02:42:ac:18:00:04)
```

Task 3. Use iptables to limit traffic to the server

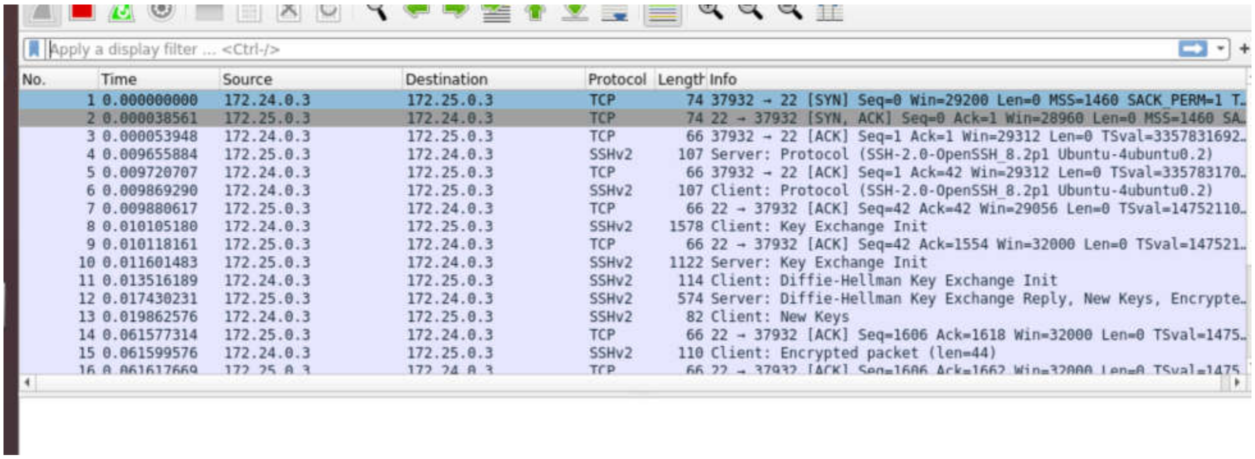
a) Show that ssh traffic is allowed. On the client, run ssh while capturing traffic on the firewall. Report these two activities in two screenshots. Explain how you know ssh traffic is allowed.


ubuntu@firewall: ~
File Edit View Search Terminal Tabs Help
root@firewall: /home/ubuntu
ubuntu@firewall: ~

```
  GNU nano 4.8                    cis-blevins.sh                    Modifi
$IPTABLES -F
$IPTABLES -t nat -F
$IPTABLES -X
#
#  By default, do not allow any forwarding or accept any traffic
#  destined for the firewall.
#
$IPTABLES -P FORWARD DROP
$IPTABLES -P INPUT   DROP
$IPTABLES -P OUTPUT  DROP

# Allow forwarding of traffic associated with any established session
$IPTABLES -A FORWARD -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT

# Allow SSH traffic on port 22
$IPTABLES -A FORWARD -p tcp --dport 22 -j ACCEPT
$IPTABLES -A FORWARD -p tcp --dport 80 -j ACCEPT
$iptables -A FORWARD -p tcp --dport 23 -j REJECT


# loopback device (internal traffic)
iptables -A INPUT -i lo -p all -j ACCEPT
```



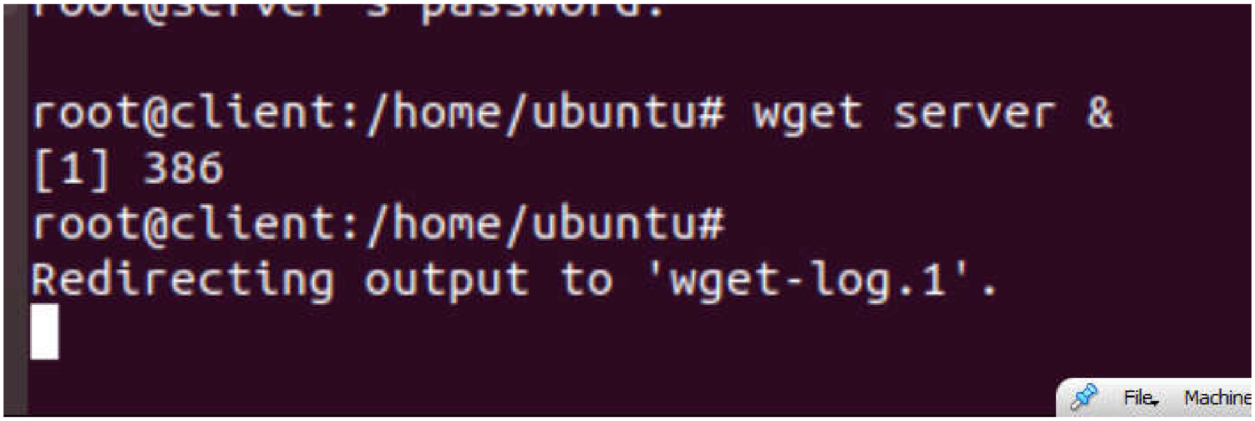
```
Escape character is '^]'.
Ubuntu 20.04.2 LTS
server login: Connection closed by foreign host.
root@client:/home/ubuntu# ssh server
root@server's password:
```



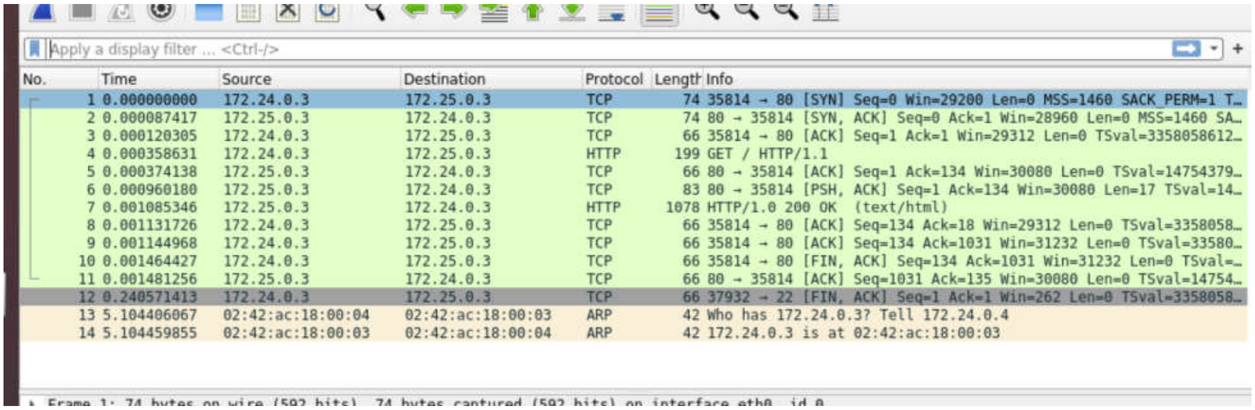
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	172.24.0.3	172.25.0.3	TCP	74	37932 → 22 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
2	0.000038561	172.25.0.3	172.24.0.3	TCP	74	22 → 37932 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA…
3	0.000053948	172.24.0.3	172.25.0.3	TCP	66	37932 → 22 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3357831692…
4	0.009655884	172.25.0.3	172.24.0.3	SSHv2	107	Server: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
5	0.009720707	172.24.0.3	172.25.0.3	TCP	66	37932 → 22 [ACK] Seq=1 Ack=42 Win=29312 Len=0 TSval=335783170…
6	0.009869290	172.24.0.3	172.25.0.3	SSHv2	107	Client: Protocol (SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.2)
7	0.009880617	172.25.0.3	172.24.0.3	TCP	66	22 → 37932 [ACK] Seq=42 Ack=42 Win=29056 Len=0 TSval=14752110…
8	0.010105180	172.24.0.3	172.25.0.3	SSHv2	1578	Client: Key Exchange Init
9	0.010118161	172.25.0.3	172.24.0.3	TCP	66	22 → 37932 [ACK] Seq=42 Ack=1554 Win=32000 Len=0 TSval=147521…
10	0.011601483	172.25.0.3	172.24.0.3	SSHv2	1122	Server: Key Exchange Init
11	0.013516189	172.24.0.3	172.25.0.3	SSHv2	114	Client: Diffie-Hellman Key Exchange Init
12	0.017430231	172.25.0.3	172.24.0.3	SSHv2	574	Server: Diffie-Hellman Key Exchange Reply, New Keys, Encrypte…
13	0.019862576	172.24.0.3	172.25.0.3	SSHv2	82	Client: New Keys
14	0.061577314	172.25.0.3	172.24.0.3	TCP	66	22 → 37932 [ACK] Seq=1606 Ack=1618 Win=32000 Len=0 TSval=1475…
15	0.061599576	172.24.0.3	172.25.0.3	SSHv2	110	Client: Encrypted packet (len=44)
16	0.061617669	172.25.0.3	172.24.0.3	TCP	66	22 → 37932 [ACK] Seq=1606 Ack=1662 Win=32000 Len=0 TSval=1475

I know ssh traffic is allowed because the client and server are successfully exchanging packets per the above wireshark screenshot. The server allows me to attempt to login. The iptables rule specifically accepts packets through port 22.

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



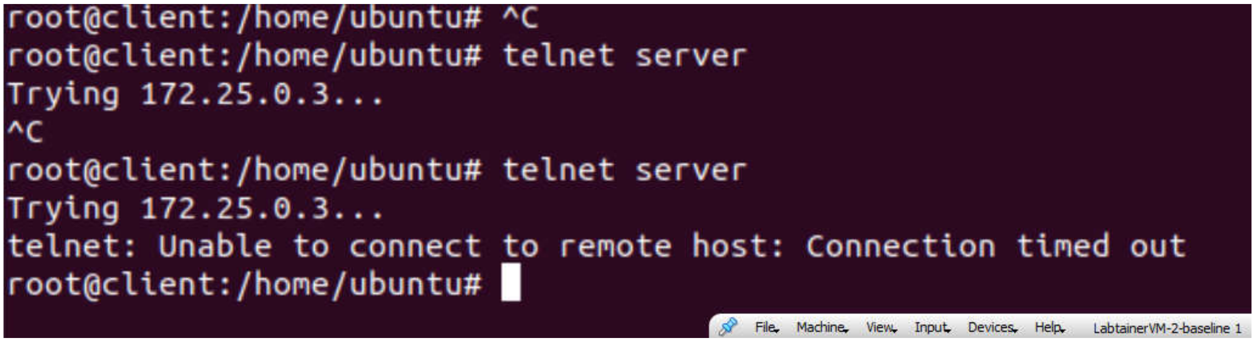
```
root@server's password:

root@client:/home/ubuntu# wget server &
[1] 386
root@client:/home/ubuntu#
Redirecting output to 'wget-log.1'.
```



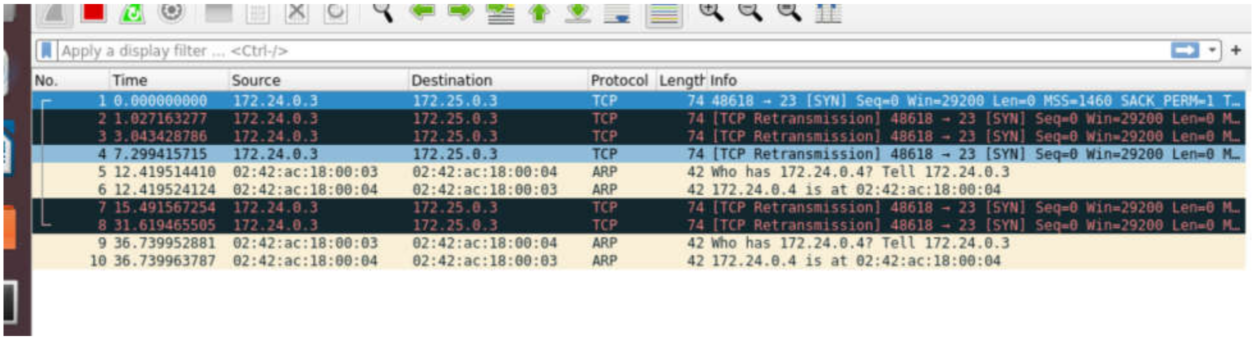
```
No.  Time          Source             Destination        Protocol Length Info
   1 0.000000000   172.24.0.3         172.25.0.3         TCP        74 35814 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
   2 0.000087417   172.25.0.3         172.24.0.3         TCP        74 80 → 35814 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SA…
   3 0.000120305   172.24.0.3         172.25.0.3         TCP        66 35814 → 80 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3358058612…
   4 0.000358631   172.24.0.3         172.25.0.3         HTTP      199 GET / HTTP/1.1
   5 0.000374138   172.25.0.3         172.24.0.3         TCP        66 80 → 35814 [ACK] Seq=1 Ack=134 Win=30080 Len=0 TSval=14754379…
   6 0.000960180   172.25.0.3         172.24.0.3         TCP        83 80 → 35814 [PSH, ACK] Seq=1 Ack=134 Win=30080 Len=17 TSval=14…
   7 0.001085346   172.25.0.3         172.24.0.3         HTTP     1078 HTTP/1.0 200 OK  (text/html)
   8 0.001131726   172.24.0.3         172.25.0.3         TCP        66 35814 → 80 [ACK] Seq=134 Ack=18 Win=29312 Len=0 TSval=3358058…
   9 0.001144968   172.24.0.3         172.25.0.3         TCP        66 35814 → 80 [ACK] Seq=134 Ack=1031 Win=31232 Len=0 TSval=33580…
  10 0.001464427   172.24.0.3         172.25.0.3         TCP        66 35814 → 80 [FIN, ACK] Seq=134 Ack=1031 Win=31232 Len=0 TSval=…
  11 0.001481256   172.25.0.3         172.24.0.3         TCP        66 80 → 35814 [ACK] Seq=1031 Ack=135 Win=30080 Len=0 TSval=14754…
  12 0.240571413   172.24.0.3         172.25.0.3         TCP        66 37932 → 22 [FIN, ACK] Seq=1 Ack=1 Win=262 Len=0 TSval=3358058…
  13 5.104406067   02:42:ac:18:00:04  02:42:ac:18:00:03  ARP        42 Who has 172.24.0.3? Tell 172.24.0.4
  14 5.104459855   02:42:ac:18:00:03  02:42:ac:18:00:04  ARP        42 172.24.0.3 is at 02:42:ac:18:00:03

▸ Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
```

I know http traffic is allowed because wget to the server receives packets in response per the above wireshark screenshot. The iptables rule specifically allows traffic through port 80.

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



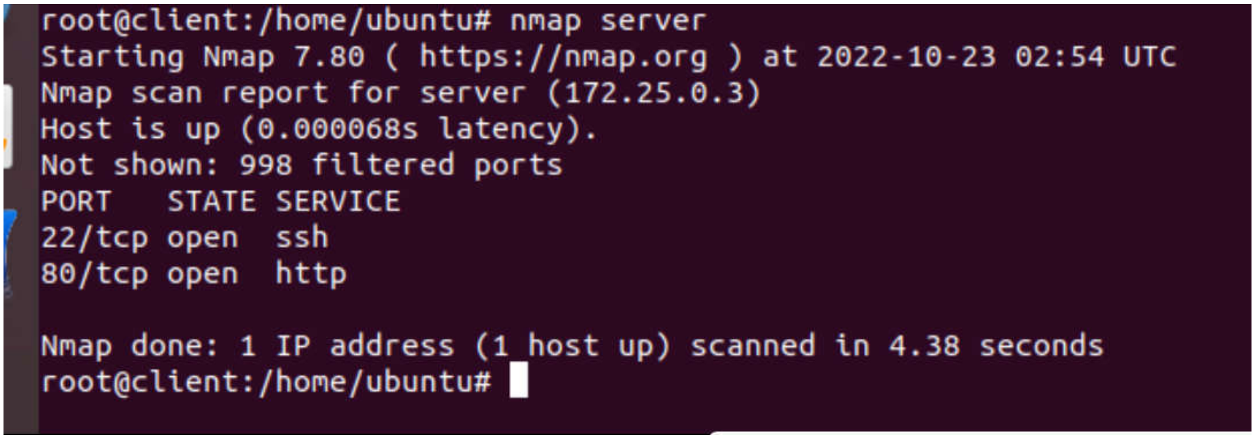
```
root@client:/home/ubuntu# ^C
root@client:/home/ubuntu# telnet server
Trying 172.25.0.3...
^C
root@client:/home/ubuntu# telnet server
Trying 172.25.0.3...
telnet: Unable to connect to remote host: Connection timed out
root@client:/home/ubuntu# 
```



```
No.  Time          Source             Destination        Protocol Length Info
   1 0.000000000   172.24.0.3         172.25.0.3         TCP        74 48618 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 T…
   2 1.027163277   172.24.0.3         172.25.0.3         TCP        74 [TCP Retransmission] 48618 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
   3 3.043428786   172.24.0.3         172.25.0.3         TCP        74 [TCP Retransmission] 48618 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
   4 7.299415715   172.24.0.3         172.25.0.3         TCP        74 [TCP Retransmission] 48618 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
   5 12.419514410  02:42:ac:18:00:03  02:42:ac:18:00:04  ARP        42 Who has 172.24.0.4? Tell 172.24.0.3
   6 12.419524124  02:42:ac:18:00:04  02:42:ac:18:00:03  ARP        42 172.24.0.4 is at 02:42:ac:18:00:04
   7 15.491567254  172.24.0.3         172.25.0.3         TCP        74 [TCP Retransmission] 48618 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
   8 31.619465505  172.24.0.3         172.25.0.3         TCP        74 [TCP Retransmission] 48618 → 23 [SYN] Seq=0 Win=29200 Len=0 M…
   9 36.739952881  02:42:ac:18:00:03  02:42:ac:18:00:04  ARP        42 Who has 172.24.0.4? Tell 172.24.0.3
  10 36.739963787  02:42:ac:18:00:04  02:42:ac:18:00:03  ARP        42 172.24.0.4 is at 02:42:ac:18:00:04
```

I know telnet traffic is blocked because the client never receives a response from the server with a telnet request. The client just continually tries to connect. Per the wiresharkscreenshot, the client sends a packet and a packet is never received back. The iptables rule specifically rejects traffic to the server's port 23.

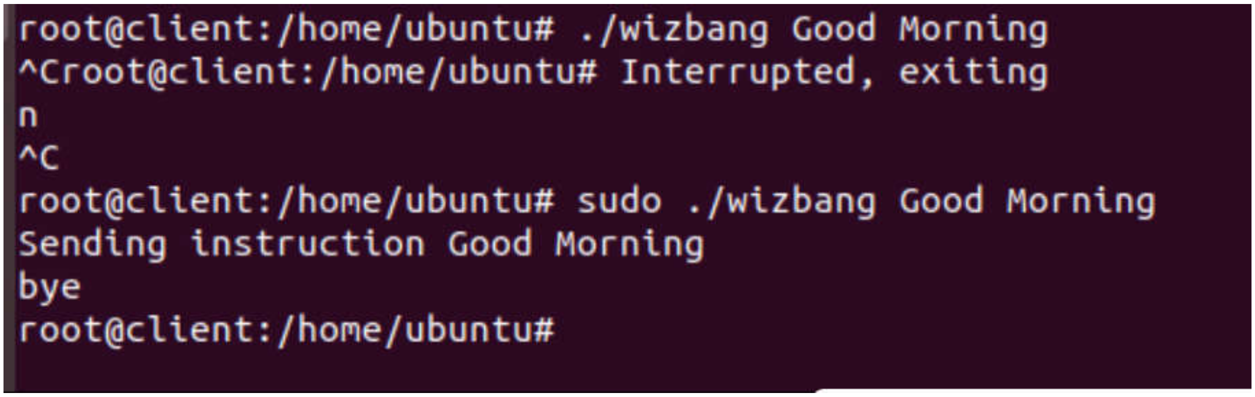
d) At the end, perform a nmap scan on the client for open ports on the server. Show the output in a screenshot.



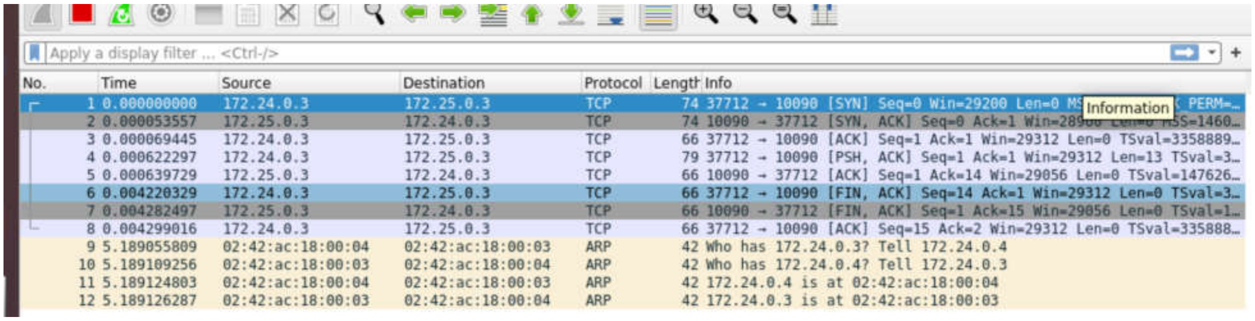
```
root@client:/home/ubuntu# nmap server
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-23 02:54 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.000068s latency).
Not shown: 998 filtered ports
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 4.38 seconds
root@client:/home/ubuntu# 
```

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.



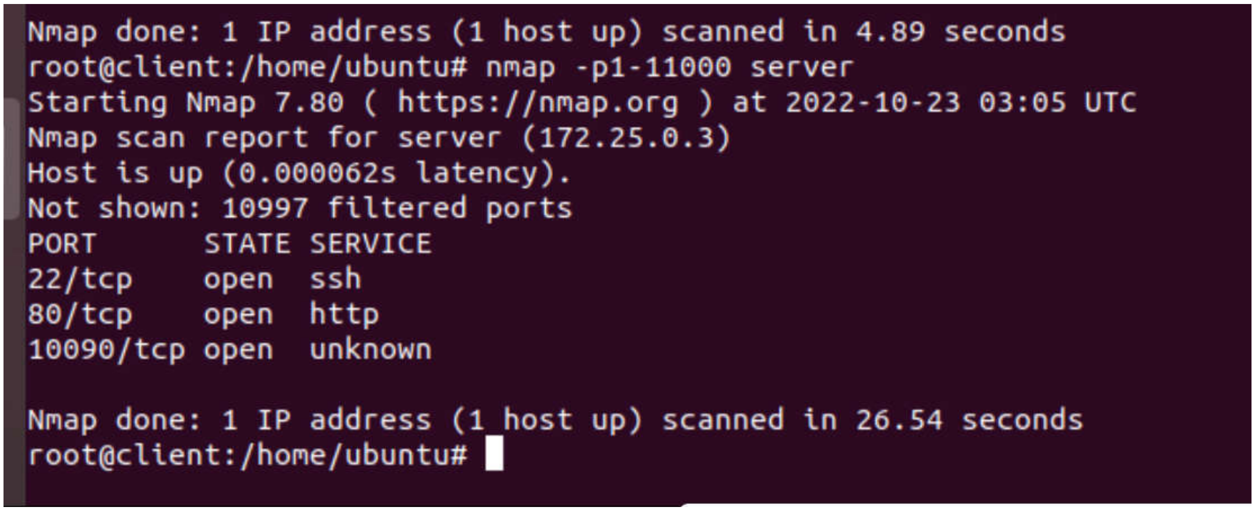
```
root@client:/home/ubuntu# ./wizbang Good Morning
^Croot@client:/home/ubuntu# Interrupted, exiting
n
^C
root@client:/home/ubuntu# sudo ./wizbang Good Morning
Sending instruction Good Morning
bye
root@client:/home/ubuntu#
```



```
No.  Time         Source             Destination        Protocol Length Info
   1 0.000000000  172.24.0.3         172.25.0.3         TCP      74 37712 → 10090 [SYN] Seq=0 Win=29200 Len=0 MS… PERM=…
   2 0.000053557  172.25.0.3         172.24.0.3         TCP      74 10090 → 37712 [SYN, ACK] Seq=0 Ack=1 Win=289… MSS=1460…
   3 0.000069445  172.24.0.3         172.25.0.3         TCP      66 37712 → 10090 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=3358889…
   4 0.000622297  172.24.0.3         172.25.0.3         TCP      79 37712 → 10090 [PSH, ACK] Seq=1 Ack=1 Win=29312 Len=13 TSval=3…
   5 0.000639729  172.25.0.3         172.24.0.3         TCP      66 10090 → 37712 [ACK] Seq=1 Ack=14 Win=29056 Len=0 TSval=147626…
   6 0.004220329  172.24.0.3         172.25.0.3         TCP      66 37712 → 10090 [FIN, ACK] Seq=14 Ack=1 Win=29312 Len=0 TSval=3…
   7 0.004282497  172.25.0.3         172.24.0.3         TCP      66 10090 → 37712 [FIN, ACK] Seq=1 Ack=15 Win=29056 Len=0 TSval=1…
   8 0.004299016  172.24.0.3         172.25.0.3         TCP      66 37712 → 10090 [ACK] Seq=15 Ack=2 Win=29312 Len=0 TSval=335888…
   9 5.189055809  02:42:ac:18:00:04  02:42:ac:18:00:03  ARP      42 Who has 172.24.0.3? Tell 172.24.0.4
  10 5.189109256  02:42:ac:18:00:03  02:42:ac:18:00:04  ARP      42 Who has 172.24.0.4? Tell 172.24.0.3
  11 5.189124803  02:42:ac:18:00:04  02:42:ac:18:00:03  ARP      42 172.24.0.4 is at 02:42:ac:18:00:04
  12 5.189126287  02:42:ac:18:00:03  02:42:ac:18:00:04  ARP      42 172.24.0.3 is at 02:42:ac:18:00:03
```

I know the wizbang program is allowed through the firewall because the client receives a response from the server. Per the wireshark screenshot, packets are sent to and received by the server. The iptables rule allows traffic to port 10090 on the server, which is the port wizbang uses.

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 4.89 seconds
root@client:/home/ubuntu# nmap -p1-11000 server
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-23 03:05 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.000062s latency).
Not shown: 10997 filtered ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
10090/tcp open  unknown

Nmap done: 1 IP address (1 host up) scanned in 26.54 seconds
root@client:/home/ubuntu# 
```