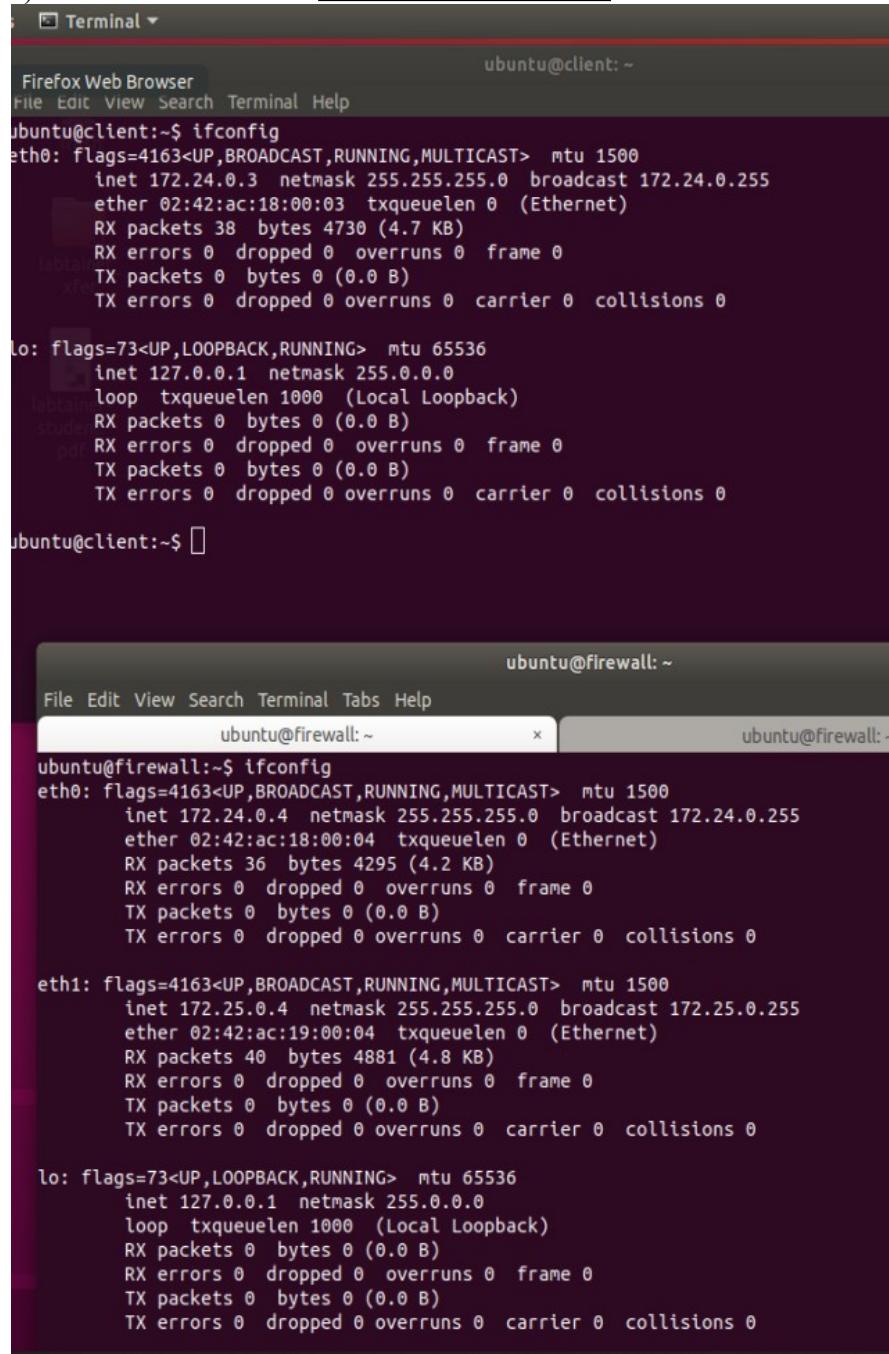


Homework 4 – Linux Firewall

Task 1. Find IP addresses

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



The image shows two terminal windows side-by-side. Both windows have a dark background and white text. The top window is titled "Terminal" and shows the command "ifconfig" being run on a host named "ubuntu@client:~". The bottom window is titled "ubuntu@firewall:~" and also shows the "ifconfig" command being run on a host named "ubuntu@firewall:~". Both outputs show network interface details like eth0 and lo, including their flags, MTU, and IP configurations.

```
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 38 bytes 4730 (4.7 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:~$ 

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 36 bytes 4295 (4.2 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 40 bytes 4881 (4.8 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

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
```

Task 2. Nmap scan

- a) Perform a nmap scan on the client for open ports on the server. [Show the output in a screenshot](#).

```

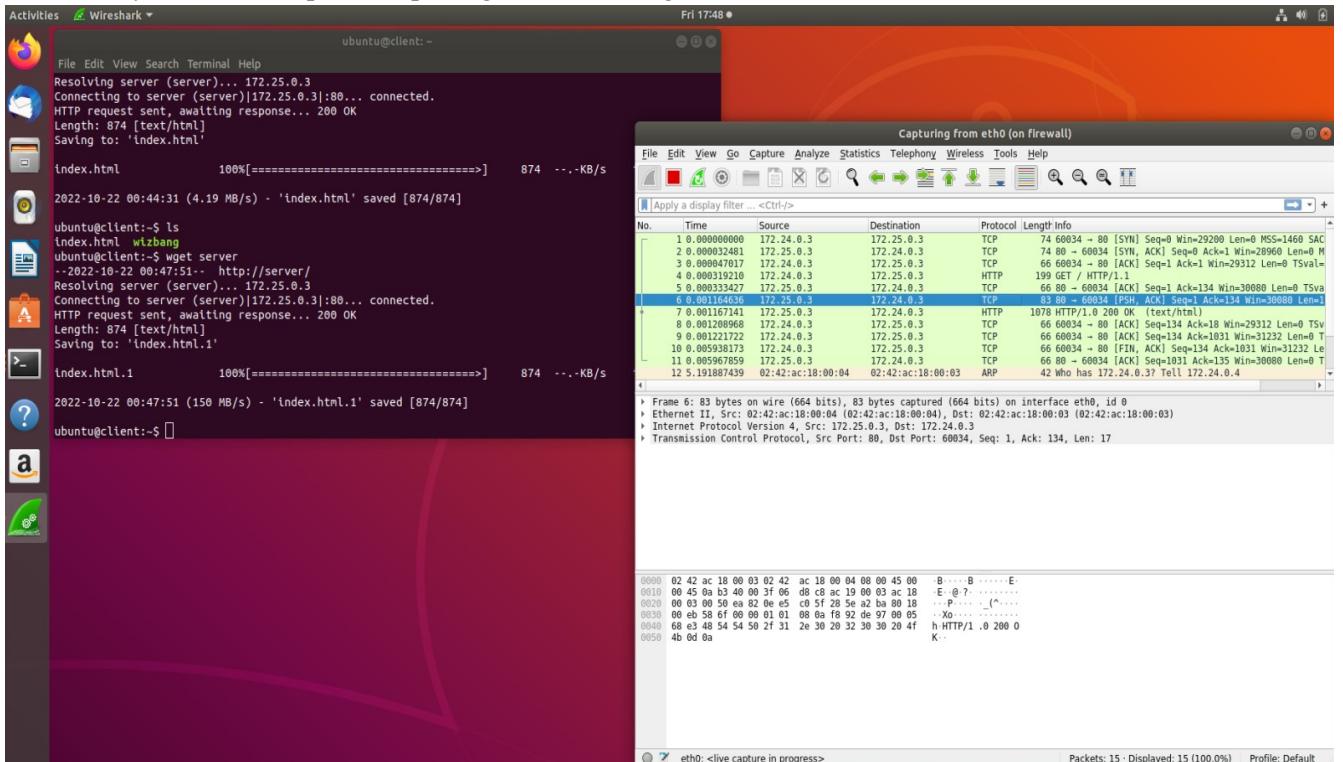
ubuntu@client: ~
File Edit View Search Terminal Help
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

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

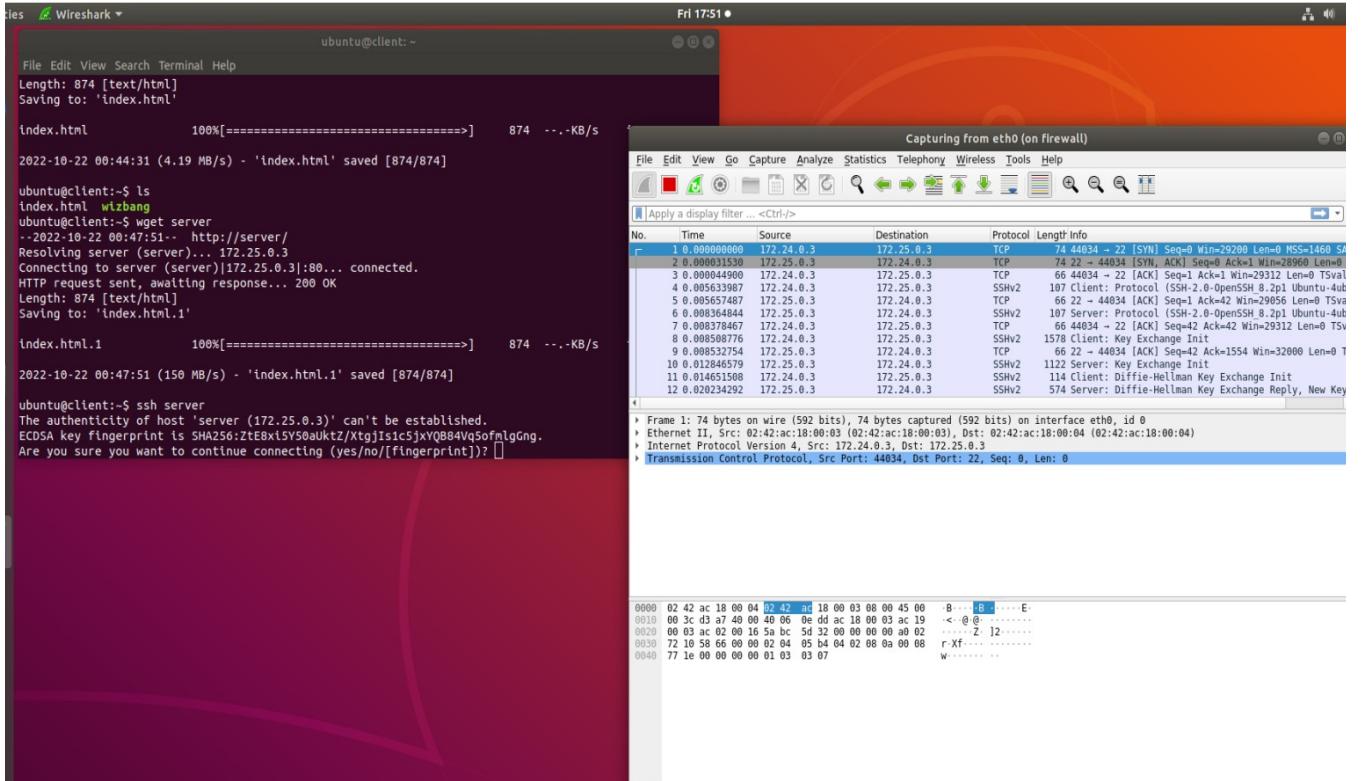
ubuntu@client:~$ nmaop server
-bash: nmaop: command not found
ubuntu@client:~$ nmap server
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-22 00:44 UTC
Nmap scan report for server (172.25.0.3)
Host is up (0.00025s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http

```

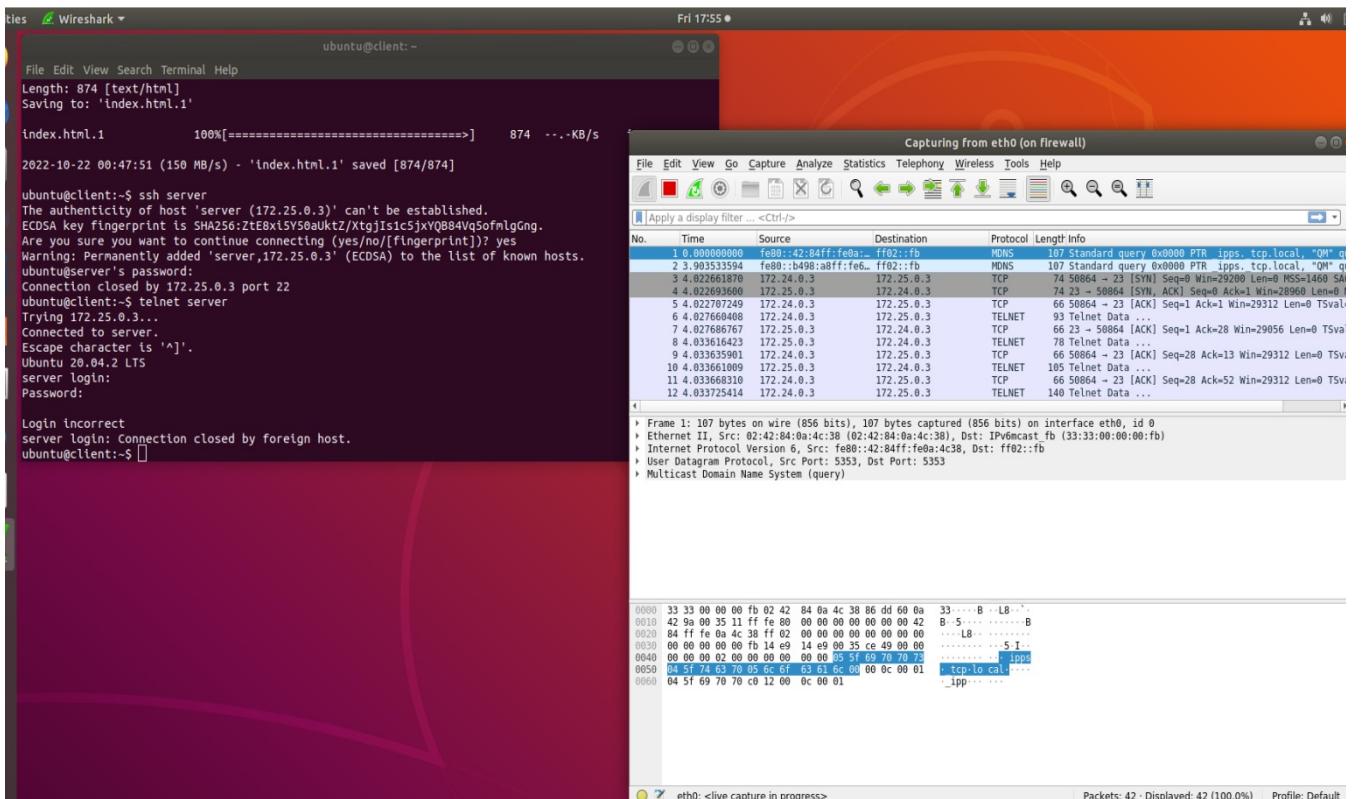
- b) Run `wget` and **report captured packets on wireshark in a screenshot**. To capture packets for a new command, you need to stop/start capturing without exiting wireshark.



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



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



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.

The screenshot displays two terminal windows and a NetworkMiner-like tool interface.

- Terminal 1 (ubuntu@client):**

```
File Edit View Search Terminal Help
Saving to: 'index.html.1'
index.html.1      100%[=====]     874 ...-KB/s  in 0s
2022-10-22 00:47:51 (150 MB/s) - 'index.html.1' saved [874/874]
ubuntu@client:~$ ssh server
The authenticity of host 'server (172.25.0.3)' can't be established.
ECDSA key fingerprint is SHA256:tE8xL5Y50auKtz/XtgJlsic5jxvQ884VgSofmlgGng.
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 hosts.
ubuntu@server's password:
Connection closed by 172.25.0.3 port 22
ubuntu@client:~$ telnet server
Trying 172.25.0.3...
Connected to server.
Escape character is '^].
Ubuntu 20.04.2 LTS
server login:
Password:
Login incorrect
server login: Connection closed by foreign host.
ubuntu@client:~$ ssh server
ubuntu@server's password: []
ubuntu@firewall:~ x      ubuntu@firewall:~
```
- Terminal 2 (ubuntu@firewall):**

```
GNU nano 4.8      cis-im.sh
#
# 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 -j NFLOG -n limit --limit 2/min --nflog-prefix "IPTABLES DROPPED"

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

# log IPTABLES filtering actions
iptables -A FORWARD -j NFLOG -n limit --limit 2/min --nflog-prefix "IPTABLES DROPPED"
```
- NetworkMiner (Capturing from eth0 on firewall):**

Shows captured traffic for the SSH connection. Key entries include:

 - Frame 1: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 0
 - Ethernet II, Src: 02:42:ac:18:00:00 (02:42:ac:18:00:00), Dst: 172.24.0.3 (172.24.0.3)
 - Internet Protocol Version 4, Src: 172.24.0.3, Dst: 172.25.0.3
 - Transmission Control Protocol, Src Port: 44048, Dst Port: 22, Seq: 0, Len: 0
 - Frame 2: 199 bytes on wire (1592 bits), 199 bytes captured (1592 bits) on interface eth0, id 0
 - Ethernet II, Src: 02:42:ac:18:00:00 (02:42:ac:18:00:00), Dst: 172.24.0.3 (172.24.0.3)
 - Internet Protocol Version 4, Src: 172.24.0.3, Dst: 172.25.0.3
 - Transmission Control Protocol, Src Port: 22, Dst Port: 44048, Seq: 1, Ack: 1, Len: 133

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

The screenshot displays two terminal windows and a NetworkMiner-like tool interface.

- Terminal 1 (ubuntu@client):**

```
File Edit View Search Terminal Help
ubuntu@client:~$ wget server
--2022-10-22 01:14:50-- 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.2'

index.html.2      100%[=====]     874 ...-KB/s  in 0s
2022-10-22 01:14:50 (100 MB/s) - 'index.html.2' saved [874/874]
```
- Terminal 2 (ubuntu@firewall):**

```
GNU nano 4.8      cis-im.sh
#
# 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 443 -m conntrack --ctstate NEW,ESTABLISHED -j ACCEPT
$IPTABLES -A FORWARD -p tcp --sport 443 -m conntrack --ctstate ESTABLISHED -j ACCEPT

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

# log IPTABLES filtering actions
iptables -A FORWARD -j NFLOG -n limit --limit 2/min --nflog-prefix "IPTABLES DROPPED"
```
- NetworkMiner (Capturing from eth0 on firewall):**

Shows captured traffic for the HTTP connection. Key entries include:

 - Frame 1: 199 bytes on wire (1592 bits), 199 bytes captured (1592 bits) on interface eth0, id 0
 - Ethernet II, Src: 02:42:ac:18:00:00 (02:42:ac:18:00:00), Dst: 172.24.0.3 (172.24.0.3)
 - Internet Protocol Version 4, Src: 172.24.0.3, Dst: 172.25.0.3
 - Transmission Control Protocol, Src Port: 80, Dst Port: 44048, Seq: 0, Len: 0
 - Frame 2: 199 bytes on wire (1592 bits), 199 bytes captured (1592 bits) on interface eth0, id 0
 - Ethernet II, Src: 02:42:ac:18:00:00 (02:42:ac:18:00:00), Dst: 172.24.0.3 (172.24.0.3)
 - Internet Protocol Version 4, Src: 172.24.0.3, Dst: 172.25.0.3
 - Transmission Control Protocol, Src Port: 44048, Dst Port: 80, Seq: 1, Ack: 1, Len: 133

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

- d) At the end, perform a nmap scan on the client for open ports on the server. Show the output in a screenshot. Said the server was down??? (Not sure if that's correct or not from my last rules created for blocking telnet)

```

Activities Terminal
ubuntu@client: ~
File Edit View Search Terminal Help
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.3'

index.html.3          100%[=====]     874  --.KB/s   in 0s

2022-10-22 01:14:56 (168 MB/s) - 'index.html.3' saved [874/874]

ubuntu@client:~$ telnet server
Trying 172.25.0.3...
Connected to server.
Escape character is '^}'.
Ubuntu 20.04.2 LTS
server login:
Login timed out after 60 seconds.
Connection closed by foreign host.
ubuntu@client:~$ telnet server
Trying 172.25.0.3...
^C
ubuntu@client:~$ nmap server
Starting Nmap 7.80 ( https://nmap.org ) at 2022-10-22 01:41 UTC
Note: Host seems down. If it is really up, but blocking our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 3.05 seconds

```

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. *I tried to make a rule that opened the port 10064 that it showed on wireshark for the TCP protocol. It wasn't working though. I tried a couple other times to get the port to open but I was still unsuccessful. Not sure what else to do to open the port but I at least wanted to show my attempted work here).*

The left screenshot shows a terminal session on an Ubuntu client. The user runs 'wizbang Hello World' which fails due to a connection timeout. They then run 'netstat -lntu' to check for active connections, which shows a listening socket on port 10064. They grep for port 10064 in netstat output and run 'sudo ./wizbang Hello World' again, which succeeds.

```

File Edit View Search Terminal Help
^[[A^[[A^[[B
^C
ubuntu@client:~$ sudo ./wizbang Hello World
ERROR: [Errno 110] Connection timed out
ubuntu@client:~$ sudo ./wizbang Hello World
^C
ubuntu@client:~$ interrupted, exiting
^C
ubuntu@client:~$ netstat -lntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 127.0.0.1:39253          0.0.0.0:*               LISTEN
udp       0      0 127.0.0.1:42341          0.0.0.0:*
ubuntu@client:~$ netstat -na | grep :10064
ubuntu@client:~$ sudo ./wizbang Hello World
^C
ubuntu@client:~$ interrupted, exiting
^C
ubuntu@client:~$ sudo ./wizbang Hello World
ERROR: [Errno 110] Connection timed out
ubuntu@client:~$ sudo ./wizbang Hello World
^C
ubuntu@client:~$ []

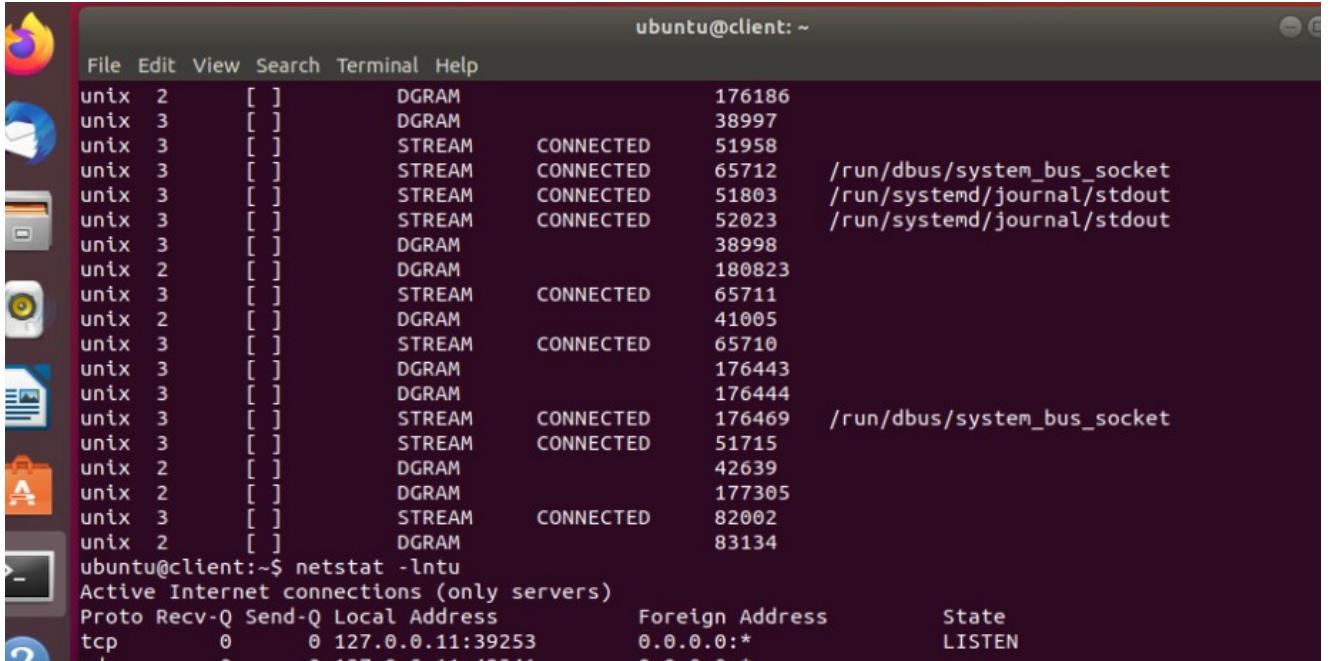
```

The right screenshot shows Wireshark capturing traffic from interface eth0. A single TCP packet is selected, showing details like source (172.24.0.3), destination (172.25.0.3), and sequence numbers. The bottom pane shows the raw hex and ASCII data of the selected frame.

No.	Time	Source	Destination	Protocol	Length Info
1	0.000000000	172.24.0.3	172.25.0.3	TCP	74 41022 - 10064 [SYN] Seq=0 Win=20280 Len=0 MSS=0
2	1.011823822	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
3	3.077842042	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
4	5.107822735	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
5	7.159432466	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
6	9.159432466	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
7	15.347740310	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
8	31.475800309	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
9	64.759547088	172.24.0.3	172.25.0.3	TCP	74 [TCP Retransmission] 41022 - 10064 [SYN] Seq=0
10	69.875951170	0.0.0.0	172.24.0.4	ARP	42 44 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0 Time to live: 64 Type: IPv4 (0800) --> 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) --> Destination: 02:42:ac:18:00:04 (02:42:ac:18:00:04) Source: 02:42:ac:18:00:03 (02:42:ac:18:00:03) Type: IPv4 (0800) Internet Protocol Version 4, Src: 172.24.0.3, Dst: 172.25.0.3 0100 = Version: 4 . .0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSFC: CS0, ECN: Not-ECT) Total Length: 60 Identification: 0x23b3 (9139) Flags: 0x4000, Don't fragment Fragment offset: 0 Time to live: 64 Protocol: TCP (6) Header checksum: 0xeded [validation disabled] Header checksum status: Unverified!
11	69.875951170	0.0.0.0	172.24.0.4	ARP	42 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0 Time to live: 64 Type: ARP (0806) --> Frame 2: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface eth0, id 1 Ethernet II, Src: 02:42:ac:18:00:04 (02:42:ac:18:00:04), Dst: 02:42:ac:18:00:03 (02:42:ac:18:00:03) --> Destination: 02:42:ac:18:00:03 (02:42:ac:18:00:03) Source: 02:42:ac:18:00:04 (02:42:ac:18:00:04) Type: IPv4 (0800) Internet Protocol Version 4, Src: 172.24.0.4, Dst: 172.24.0.3 0100 = Version: 4 . .0101 = Header Length: 20 bytes (5) > Differentiated Services Field: 0x00 (DSFC: CS0, ECN: Not-ECT) Total Length: 60 Identification: 0x23b3 (9139) Flags: 0x4000, Don't fragment Fragment offset: 0 Time to live: 64 Protocol: TCP (6) Header checksum: 0xeded [validation disabled] Header checksum status: Unverified!

- b) At the end, perform a nmap scan on the client for open ports on the server.

- I ran the nmap scan but it came back as “server being down” in the output. I instead ran a netstat scan to show what ports were open, as stated before, I was unsuccessful in opening port 10064 for the wizbang portion of question 4 but I still wanted to shoe my work.*



```
ubuntu@client:~$ netstat -lntu
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State
tcp      0      0 127.0.0.11:39253        0.0.0.0:*              LISTEN
```

The screenshot shows a terminal window titled "ubuntu@client: ~". The window contains two sets of netstat output. The first set, under "Active Internet connections (only servers)", shows a single listening TCP connection on port 39253. The second set, under "unix", lists numerous local unix socket connections, mostly of type DGRAM, with various file descriptors (2, 3, 51958, 65712, etc.) and local addresses (176186, 38997, etc.).