# Computer Networks Lab

# UE19CS256

# Week 1

Name: Sreenath Saikumar

Semester: 4  Section: G

**SRN:** PES2UG19CS406

Date: 22/01/2021

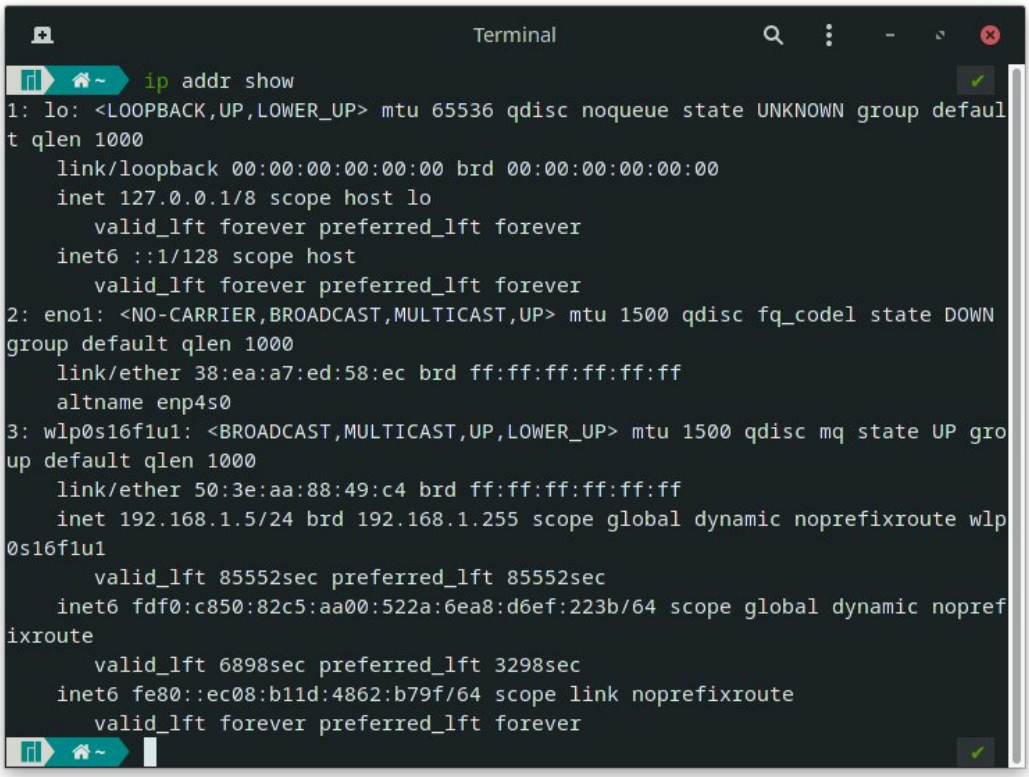
## Objectives:

1. Wireshark: Perform and Analyse Ping PDU capture, examine HTTP packet capture, analyse HTTP packet capture using filter.
2. Netcat: Establish communication between client and server, transfer files.
3. Tcpdump: Capture packets.
4. Ping: Test the connectivity between 2 systems.
5. Traceroute: Perform traceroute checks.
6. Nmap: Explore an entire network.

### Task 1: Linux Interface Configuration (ifconfig/ ip command)

1. To display the status of all active network interfaces.

```
ifconfig (or) ip addr show
```



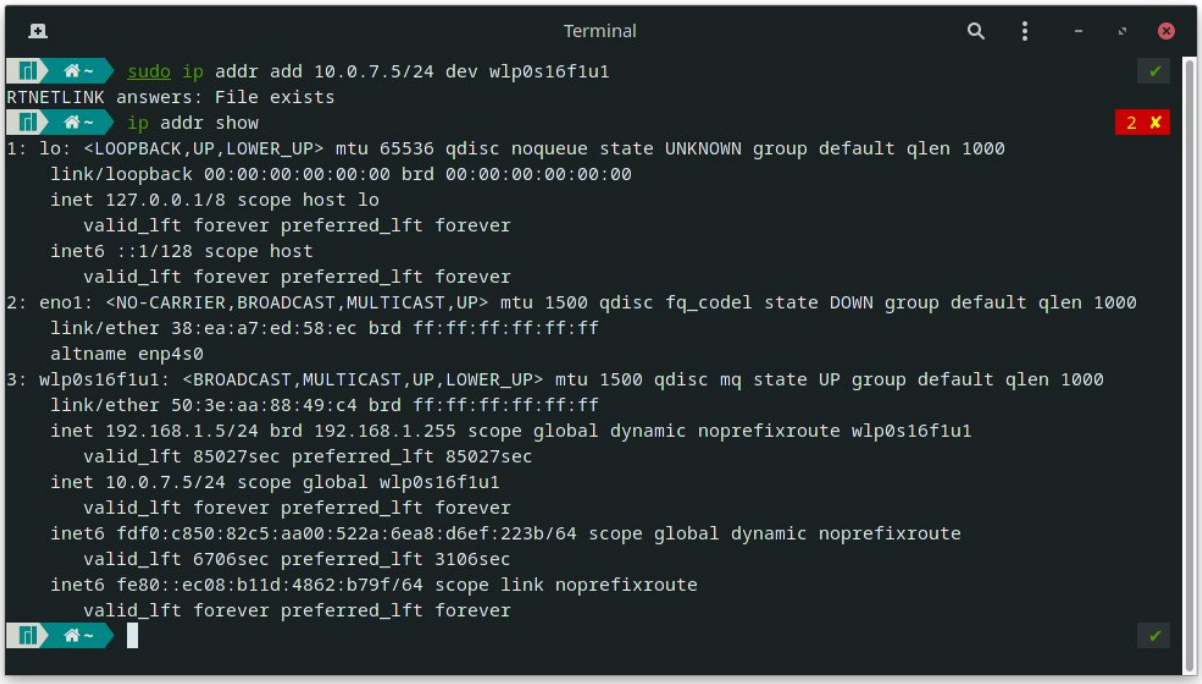
```
Terminal
~ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eno1: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state DOWN
group default qlen 1000
    link/ether 38:ea:a7:ed:58:ec brd ff:ff:ff:ff:ff:ff
    altname enp4s0
3: wlp0s16f1u1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP gro
up default qlen 1000
    link/ether 50:3e:aa:88:49:c4 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.5/24 brd 192.168.1.255 scope global dynamic noprefixroute wlp
0s16f1u1
       valid_lft 85552sec preferred_lft 85552sec
    inet6 fdf0:c850:82c5:aa00:522a:6ea8:d6ef:223b/64 scope global dynamic nopref
ixroute
       valid_lft 6898sec preferred_lft 3298sec
    inet6 fe80::ec08:b11d:4862:b79f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
~
```

IP address table:

| Interface Name | IP Address (IPv4/IPv6) | MAC Address |
|---|---|---|
| lo | 127.0.0.1/::1 | 00:00:00:00:00:00 |
| eno1 | None/None | 38:ea:a7:ed:58:ec |
| wlp0s16f1u1 | 192.168.1.5/fdf0:c850:82c5:aa00:522a:6ea8:d6ef:223b | 50:3e:aa:88:49:c4 |

2. To assign an IP address to an interface.

```
sudo ip addr add 10.0.7.5/24 dev wlp0s16f1u1
```




The new IP allocated **is invalid** and may not be used for procedures later on **due to the DHCP server running on the home network router** that allocates IP's in the range 192.168.1.2 to 192.168.1.255. **Therefore, for most procedures that involve connecting to the internet or using multiple devices, the IP 192.168.1.5 will be used. SRN or USN number cannot be used as it is outside the IP address range of the router.**

3. To active/ deactivate a network interface.

```
sudo ifconfig wlp0s16f1u1 down

sudo ifconfig wlp0s16f1u1 up
```

```
Terminal
~ sudo ifconfig wlp0s16f1u1 down
~ ifconfig
eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 38:ea:a7:ed:58:ec  txqueuelen 1000  (Ethernet)
        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

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 28  bytes 1608 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 1608 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

~
```

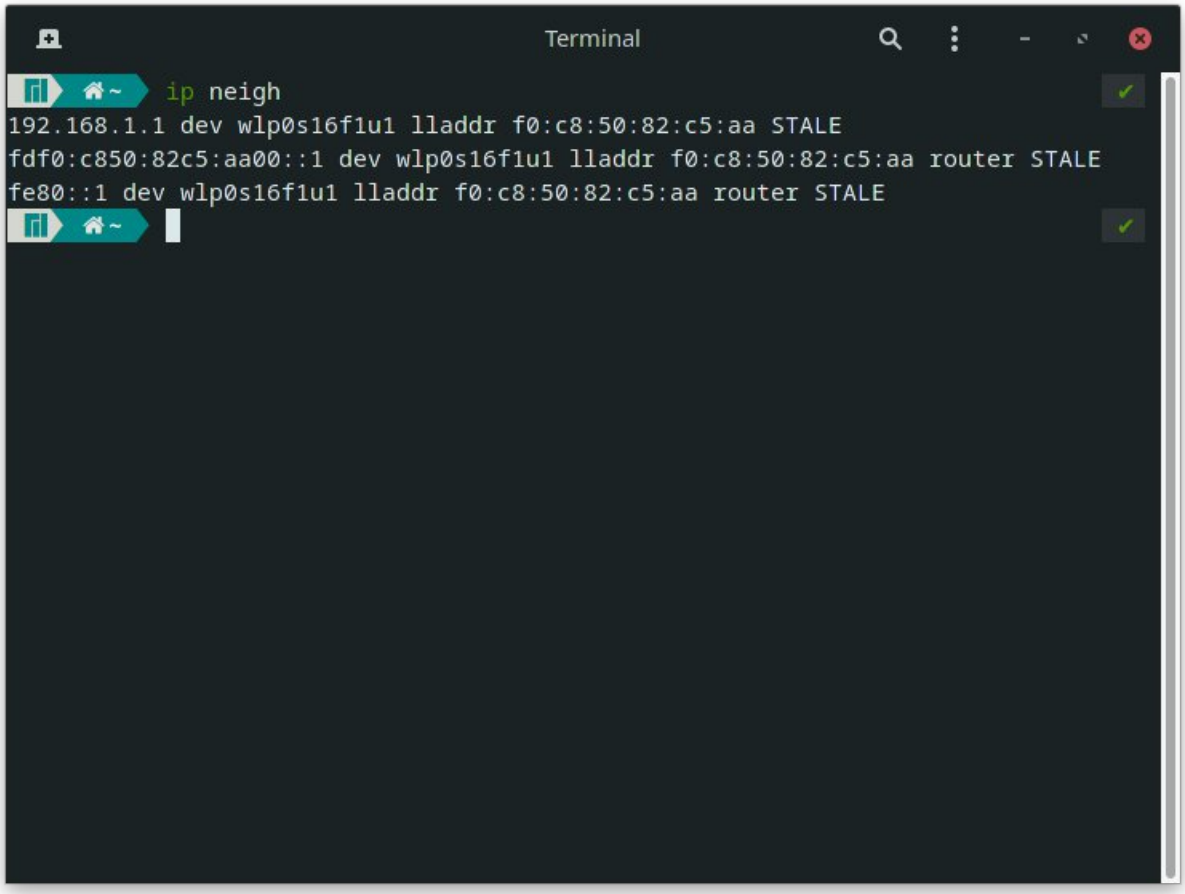
`wlp0s16f1u1` has been deactivated.

```
Terminal
~ sudo ifconfig wlp0s16f1u1 up
~ ifconfig
eno1: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        ether 38:ea:a7:ed:58:ec  txqueuelen 1000  (Ethernet)
        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

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 28  bytes 1608 (1.5 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 28  bytes 1608 (1.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlp0s16f1u1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.1.5  netmask 255.255.255.0  broadcast 192.168.1.255
        inet6 fe80::ec08:b11d:4862:b79f  prefixlen 64  scopeid 0x20<link>
        inet6 fdf0:c850:82c5:aa00:522a:6ea8:d6ef:223b  prefixlen 64  scopeid 0x0<global>
        ether 50:3e:aa:88:49:c4  txqueuelen 1000  (Ethernet)
        RX packets 25  bytes 57107287 (54.4 MiB)
        RX errors 0  dropped 629  overruns 0  frame 0
        TX packets 18  bytes 2806491 (2.6 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

~
```

`wlp0s16f1u1` has been reactivated.

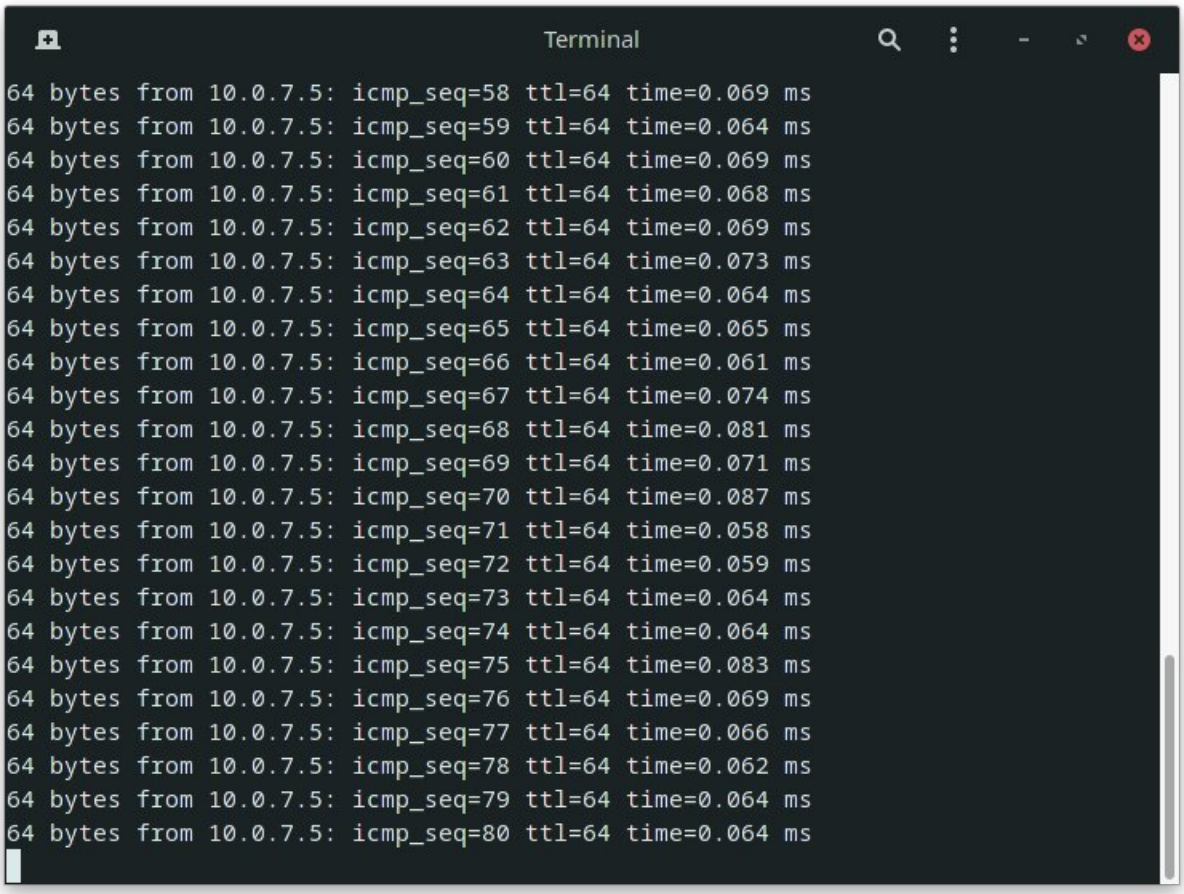
4. To show the current neighbour table in kernel.

```
ip neigh
```

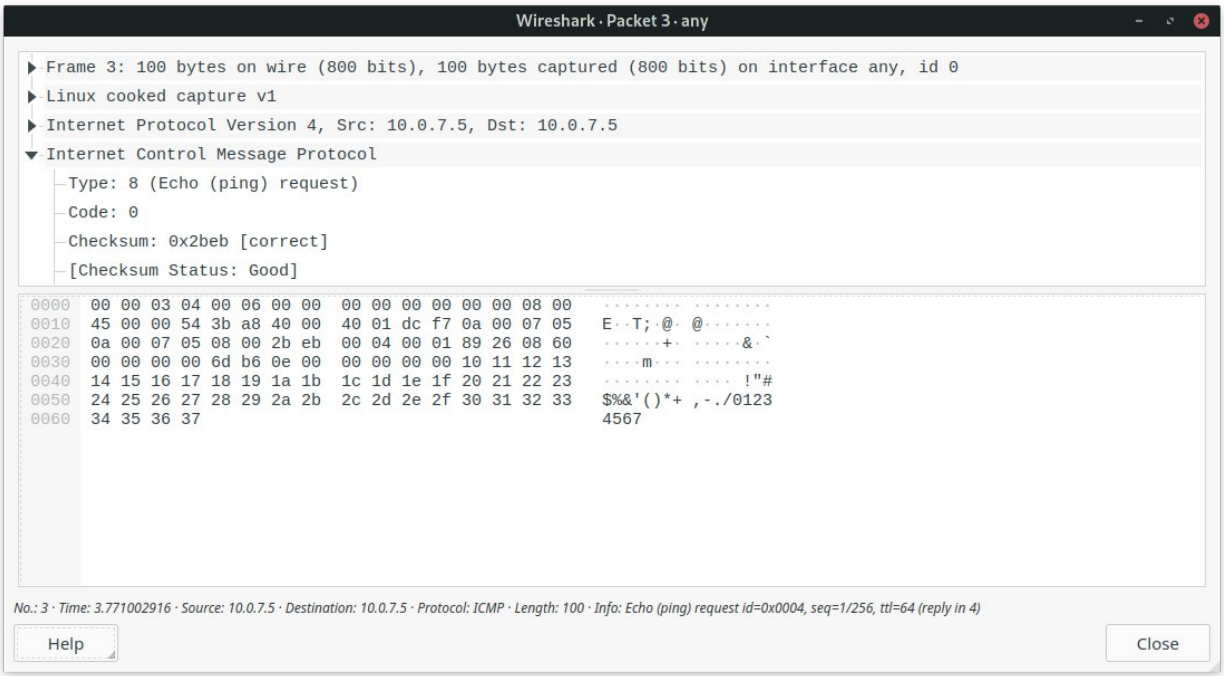



## Task 2: Ping PDU ( Packet Data Units or Packets) Capture

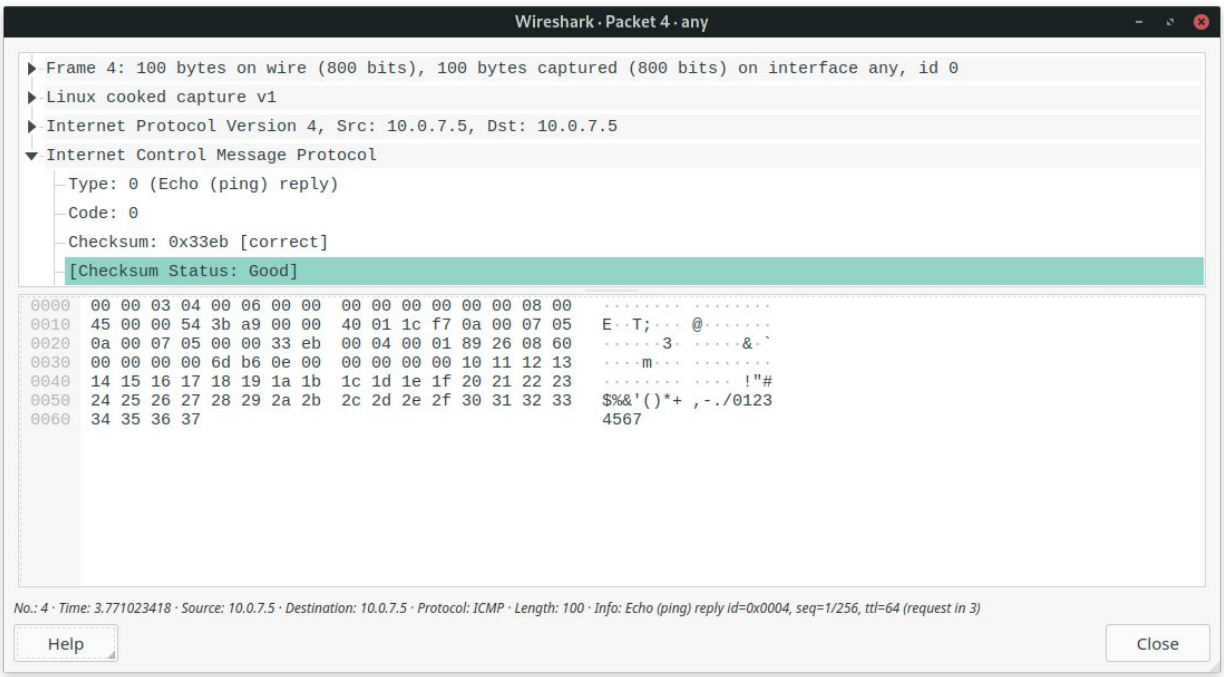
1. Assign an IP address to the system(host).
   a. Since the host is pinging the host (simple localhost ping), the router doesn't come into play, therefore the 10.0.7.5 IP can be used.
2. Launch Wireshark and select 'any' interface.
3. Use command `ping 10.0.7.5`




| | |
|---|---|
| TTL | 64 |
| Protocol used | ICMP |
| Time | ~0.6–0.8 ms |




Request packet



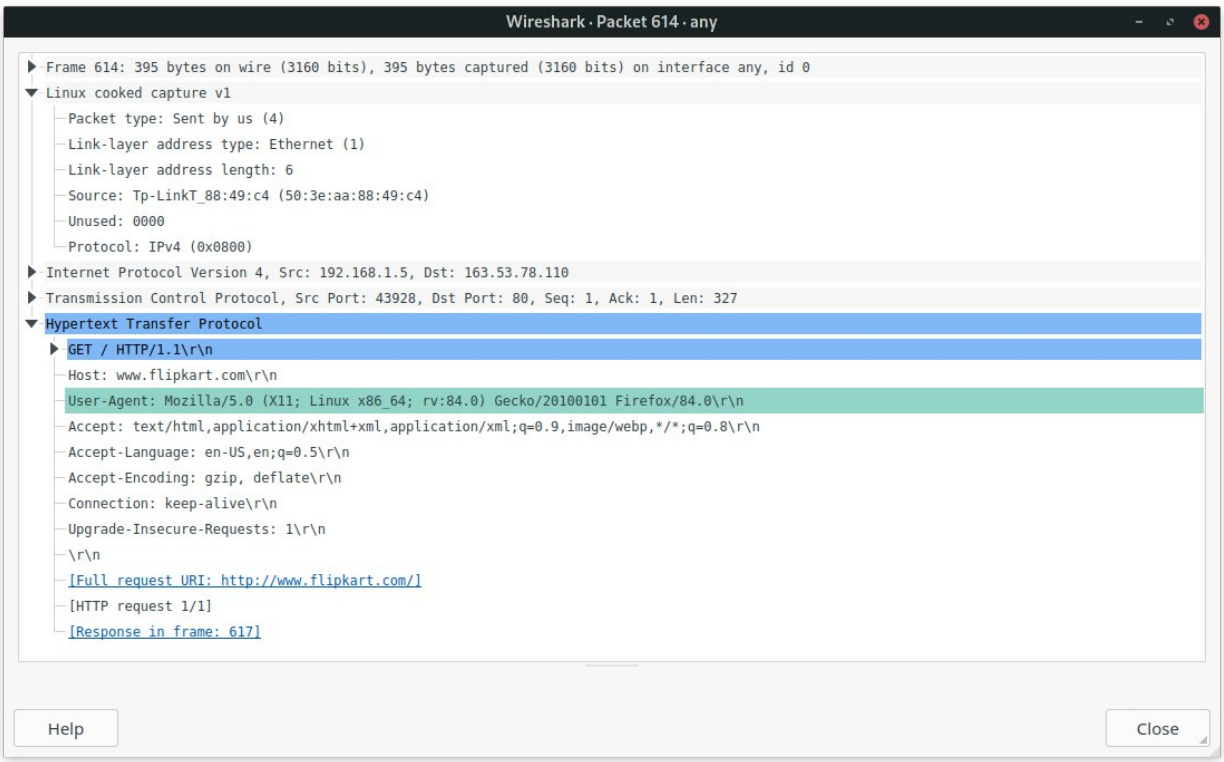

Response Packet

| Details | First Echo Request | First Echo Reply |
|---|---|---|
| Frame Number | 3 | 4 |
| Source IP address | 10.0.7.5 | 10.0.7.5 |
| Destination IP address | 10.0.7.5 | 10.0.7.5 |
| ICMP Type value | 8 | 0 |
| ICMP Code value | 0 | 0 |
| Source Ethernet address | 00:00:00:00:00:00 | 00:00:00:00:00:00 |
| Destination Ethernet address | 00:00:00:00:00:00 | 00:00:00:00:00:00 |
| Internet Protocol Version | IPv4 | IPv4 |
| Time to Live (TTL) value | 64 | 64 |

## Task 3: HTTP PDU Capture

Using Wireshark's Filter feature

1. Open Wireshark and choose the 'any' interface. Type in 'http' on the filter toolbar.
2. Open Firefox and browse www.flipkart.com
3.




Request packet


Response packet

| Details | First Echo Request | First Echo Reply |
|---|---|---|
| Frame Number | 614 | 617 |
| Source Port | 80 | 43928 |
| Destination Port | 43928 | 80 |
| Source IP address | 192.168.1.5 | 163.53.78.110 |
| Destination IP address | 163.53.78.110 | 192.168.1.5 |
| Source Ethernet address | 50:3e:aa:88:49:c4 | f0:c8:50:82:c5:aa |
| Destination Ethernet address | f0:c8:50:82:c5:aa | 50:3e:aa:88:49:c4 |

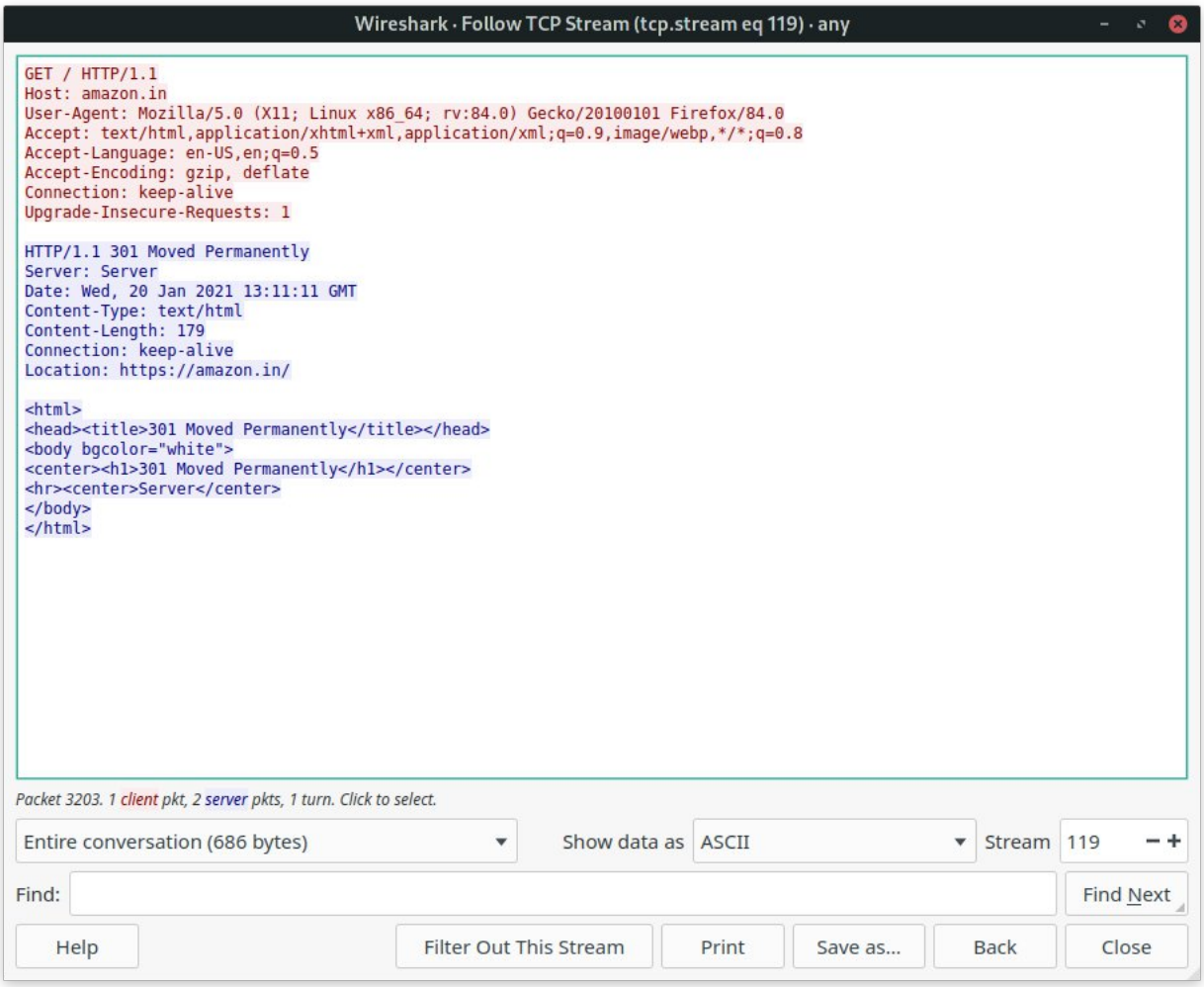
4. HTTP request and response analysis

| HTTP Request | | HTTP Response | |
|---|---|---|---|
| Get | GET / HTTP/1.1\r\n | Server | nginx\r\n |
| Host | www.flipkart.com\r\n | Content-Type | text/html\r\n |
| User-Agent | Mozilla/5.0 (X11; Linux x86_64; rv: 84.0) Gecko /20100101 Firefox/84.0\r\n | Date | Wed, 20 Jan 2021 12:52:45 GMT\r\n |

| | | | |
|---|---|---|---|
| Accept-Language | en-US,en;q=0.5\r\n | Location | https://www.flipkart.com/\r\n |
| Accept-Encoding | gzip, deflate\r\n | Content-length | 178\r\n |
| Connection | keep-alive\r\n | Connection | keep-alive\r\n |

Using Wireshark's Follow TCP Stream

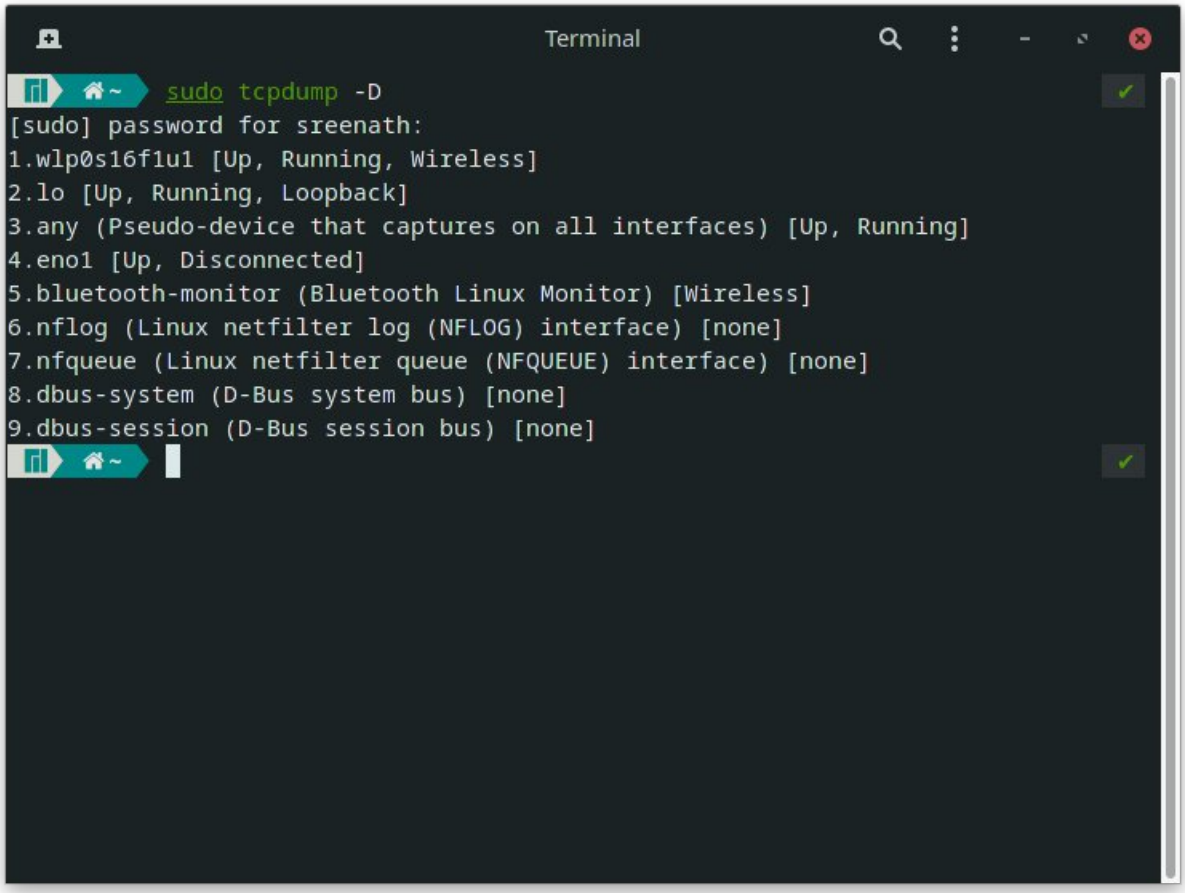
1. Right click a packet and select 'Follow TCP Stream'



## Task 4: Capturing Packets with tcpdump

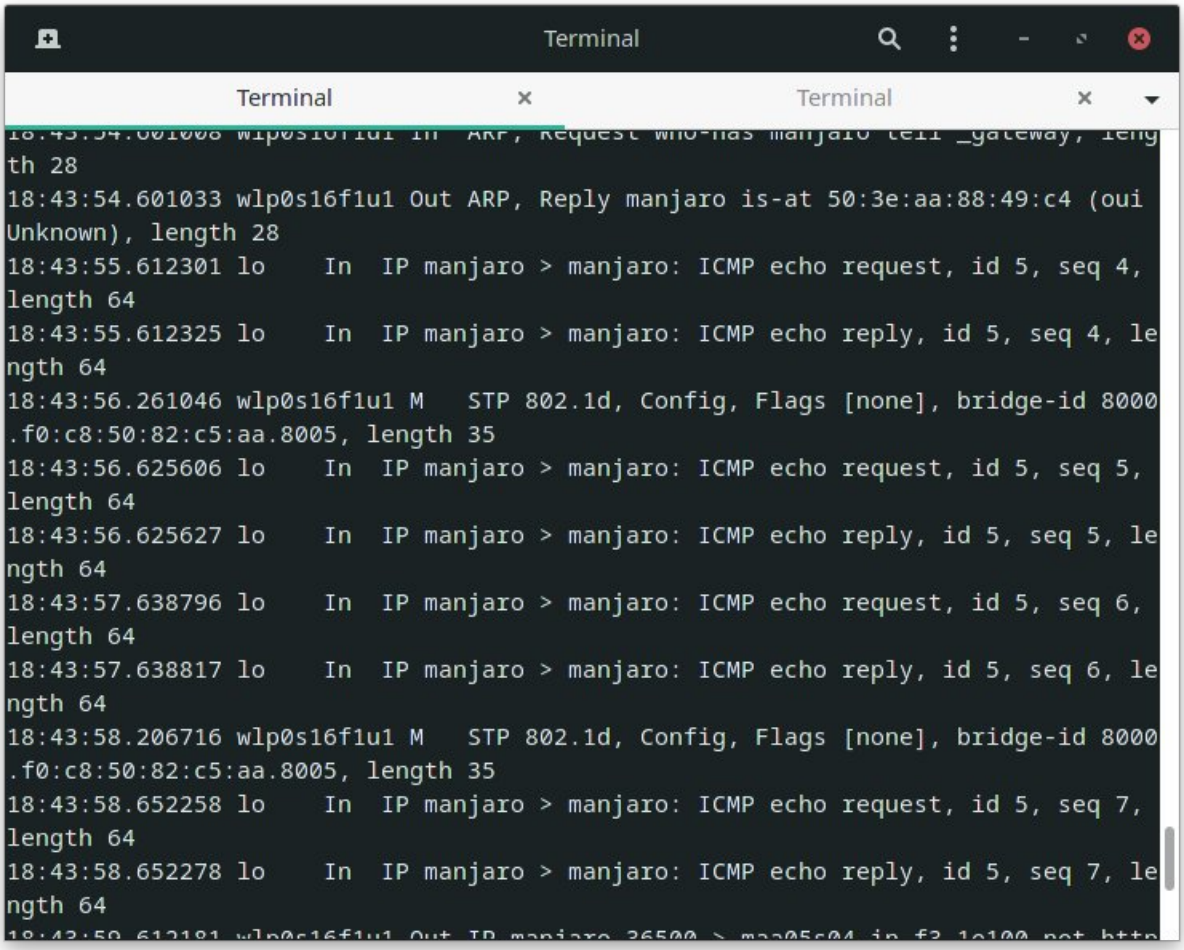
1. To see the interfaces available for capture, use the command :

```
sudo tcpdump -D
```



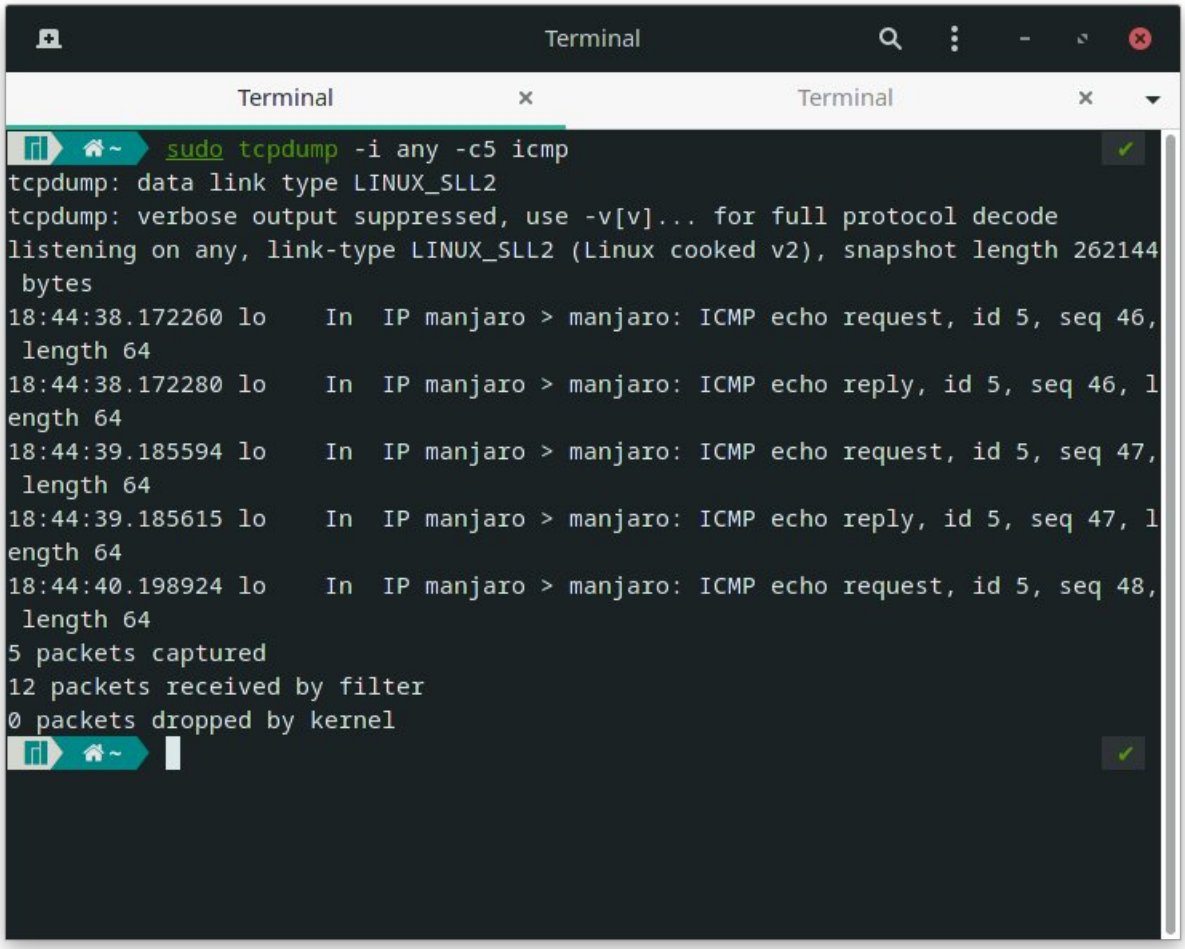
2. Capture all packets in any interface by running:

```
sudo tcpdump -i any
```




3. To filter packets based on protocol, specifying the protocol in the command line, use the following command:

```
sudo tcpdump -i any -c5 icmp
```




4. To inspect the HTTP content of a web request, use

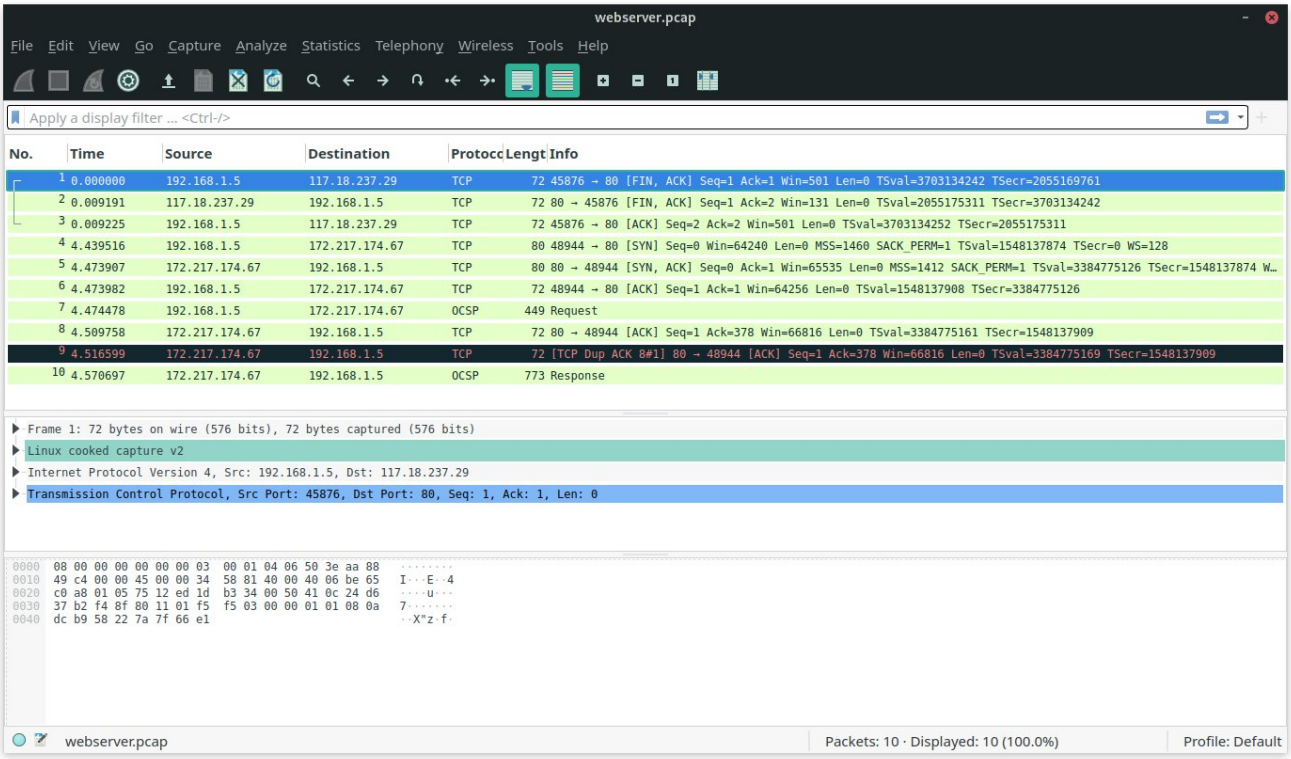
```
sudo tcpdump -i any -c10 -nn -A port 80
```

```
Terminal
Terminal                                   Terminal
~ sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
18:46:35.342380 wlp0s16f1u1 Out IP 192.168.1.5.39864 > 95.217.163.246.80: Flags [S], seq 4064378061,
 win 64240, options [mss 1460,sackOK,TS val 2772271708 ecr 0,nop,wscale 7], length 0
E..<..@.@.\....._......P.A|.........T..........
.=.\........
18:46:35.509721 wlp0s16f1u1 In  IP 95.217.163.246.80 > 192.168.1.5.39864: Flags [S.], seq 1775439521
, ack 4064378062, win 65160, options [mss 1412,sackOK,TS val 846948227 ecr 2772271708,nop,wscale 7],
 length 0
E..<..@.5..?_........P..i....A|.....>..........
2{g..=.\....
18:46:35.509793 wlp0s16f1u1 Out IP 192.168.1.5.39864 > 95.217.163.246.80: Flags [.], ack 1, win 502,
 options [nop,nop,TS val 2772271875 ecr 846948227], length 0
E..4..@.@.\....._......P.A|.i.......i|.....
.=..2{g.
18:46:35.510025 wlp0s16f1u1 Out IP 192.168.1.5.39864 > 95.217.163.246.80: Flags [P.], seq 1:100, ack
 1, win 502, options [nop,nop,TS val 2772271876 ecr 846948227], length 99: HTTP: GET /check_network_
status.txt HTTP/1.1
E.....@.@.\"...._......P.A|.i..............
.=..2{g.GET /check_network_status.txt HTTP/1.1
Host: www.archlinux.org
Accept: */*
```

5. To save packets to a file, use the option -w:

```
sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
```

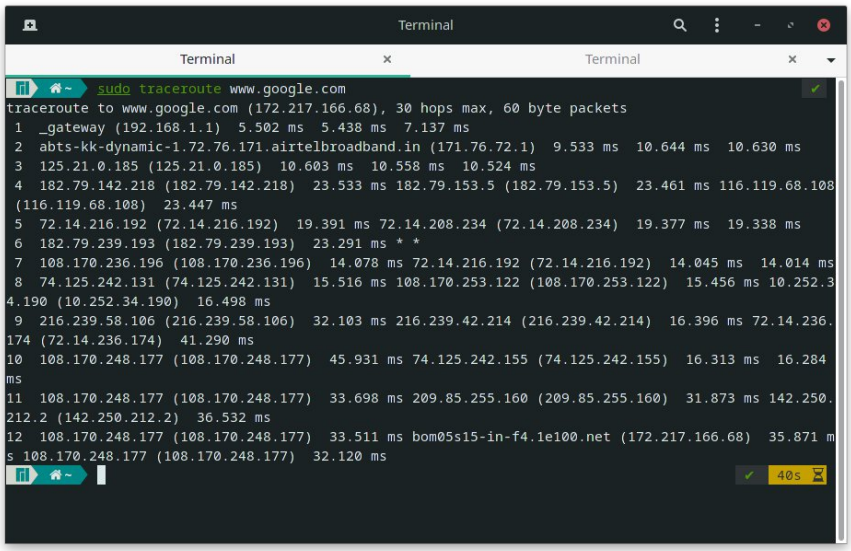
```
Terminal
Terminal                                   Terminal
~ sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
10 packets captured
11 packets received by filter
0 packets dropped by kernel
~ ls                                                                                    9s
Desktop    Downloads  Pictures  Templates  rtl8192eu-linux-driver
Documents  Music      Public    Videos     webserver.pcap
~
```




webserver.pcap viewed in Wireshark

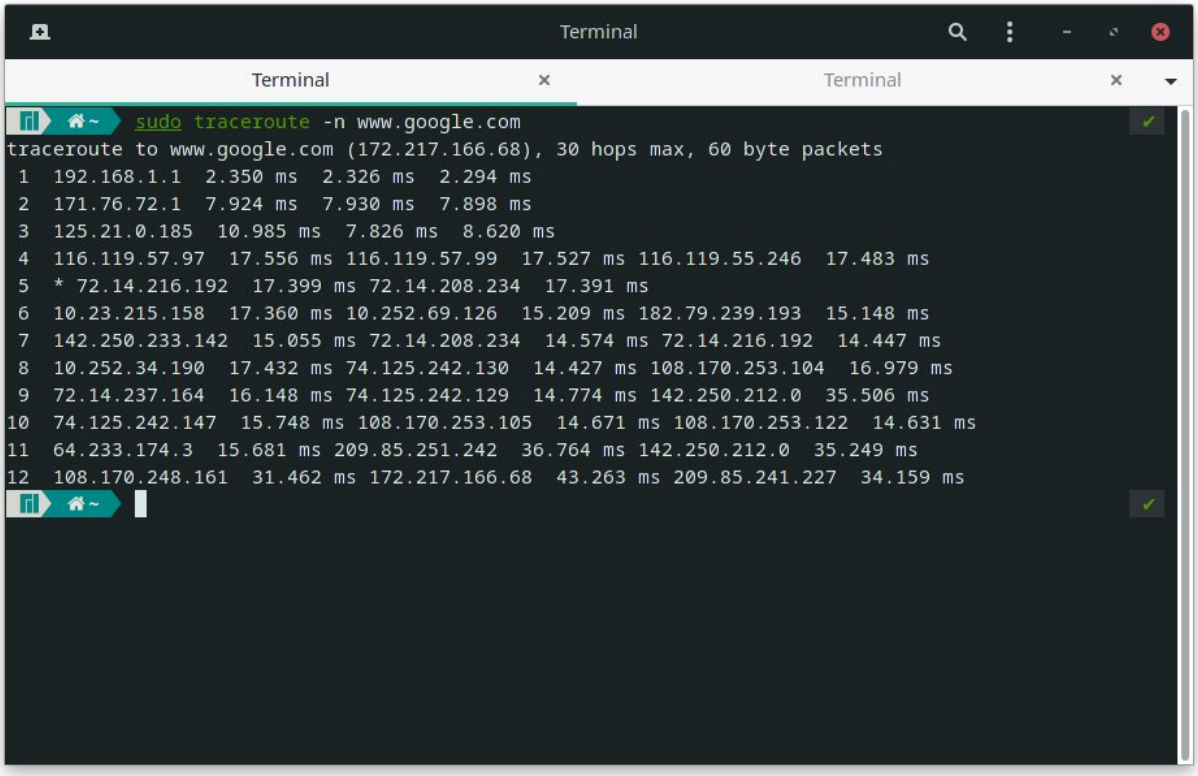
## Task 5: Perform Traceroute Checks

1. Run traceroute using `sudo traceroute www.google.com`



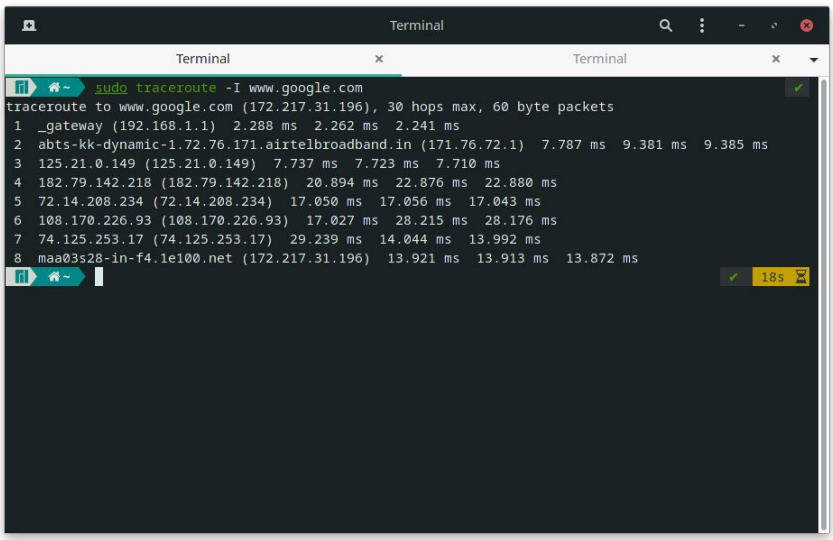
2. To disable IP address mapping with hostnames, use the -n option:

```
sudo traceroute -n www.google.com
```



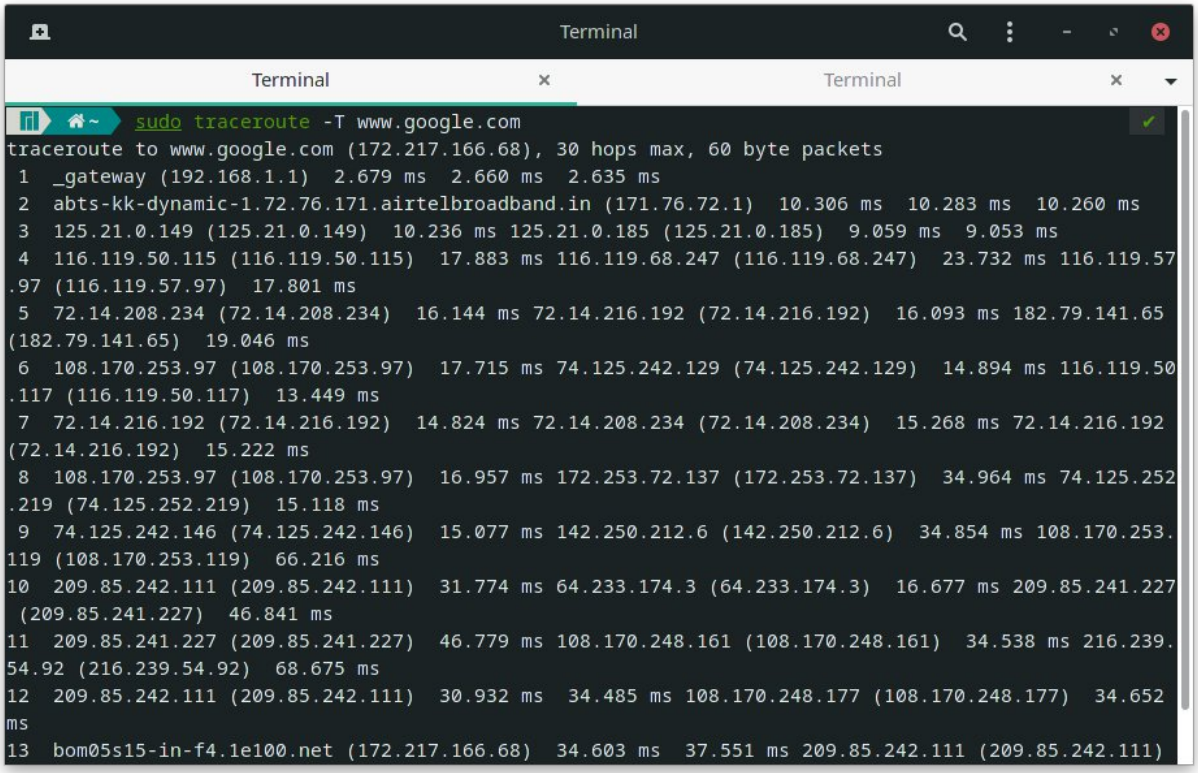

3. Use the -I option to make traceroute use ICMP.

```
sudo traceroute -I www.google.com
```

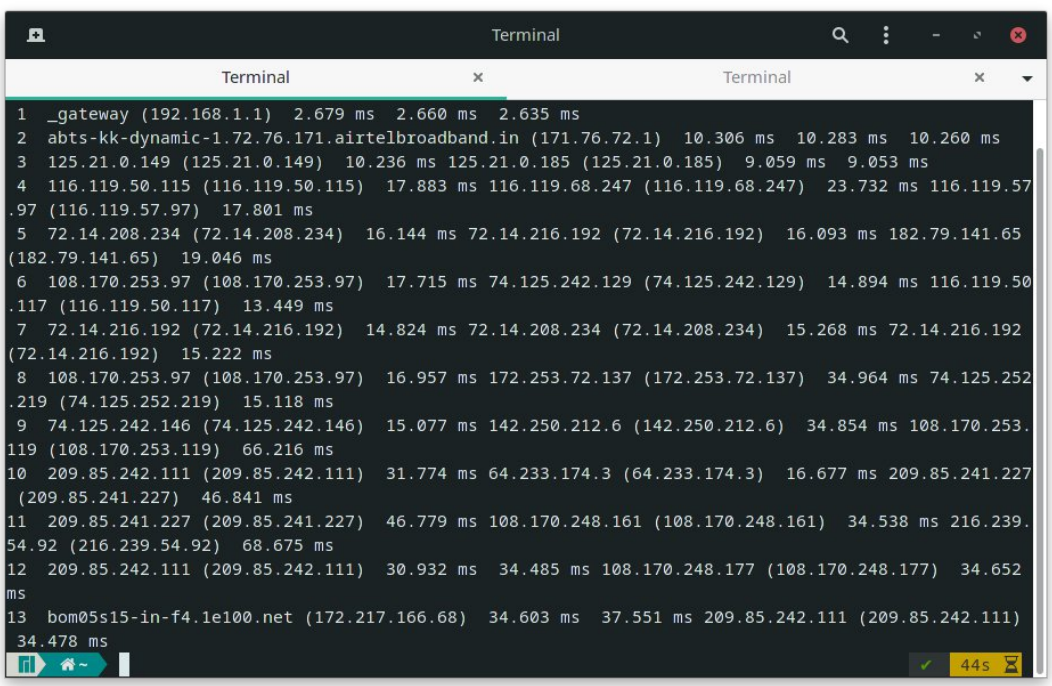


4. To test a TCP connection, use the -T flag.

```
sudo traceroute -T www.google.com
```



```
~ sudo traceroute -T www.google.com
traceroute to www.google.com (172.217.166.68), 30 hops max, 60 byte packets
 1  _gateway (192.168.1.1)  2.679 ms  2.660 ms  2.635 ms
 2  abts-kk-dynamic-1.72.76.171.airtelbroadband.in (171.76.72.1)  10.306 ms  10.283 ms  10.260 ms
 3  125.21.0.149 (125.21.0.149)  10.236 ms 125.21.0.185 (125.21.0.185)  9.059 ms  9.053 ms
 4  116.119.50.115 (116.119.50.115)  17.883 ms 116.119.68.247 (116.119.68.247)  23.732 ms 116.119.57
.97 (116.119.57.97)  17.801 ms
 5  72.14.208.234 (72.14.208.234)  16.144 ms 72.14.216.192 (72.14.216.192)  16.093 ms 182.79.141.65
(182.79.141.65)  19.046 ms
 6  108.170.253.97 (108.170.253.97)  17.715 ms 74.125.242.129 (74.125.242.129)  14.894 ms 116.119.50
.117 (116.119.50.117)  13.449 ms
 7  72.14.216.192 (72.14.216.192)  14.824 ms 72.14.208.234 (72.14.208.234)  15.268 ms 72.14.216.192
(72.14.216.192)  15.222 ms
 8  108.170.253.97 (108.170.253.97)  16.957 ms 172.253.72.137 (172.253.72.137)  34.964 ms 74.125.252
.219 (74.125.252.219)  15.118 ms
 9  74.125.242.146 (74.125.242.146)  15.077 ms 142.250.212.6 (142.250.212.6)  34.854 ms 108.170.253.
119 (108.170.253.119)  66.216 ms
10  209.85.242.111 (209.85.242.111)  31.774 ms 64.233.174.3 (64.233.174.3)  16.677 ms 209.85.241.227
 (209.85.241.227)  46.841 ms
11  209.85.241.227 (209.85.241.227)  46.779 ms 108.170.248.161 (108.170.248.161)  34.538 ms 216.239.
54.92 (216.239.54.92)  68.675 ms
12  209.85.242.111 (209.85.242.111)  30.932 ms  34.485 ms 108.170.248.177 (108.170.248.177)  34.652
ms
13  bom05s15-in-f4.1e100.net (172.217.166.68)  34.603 ms  37.551 ms 209.85.242.111 (209.85.242.111)
```

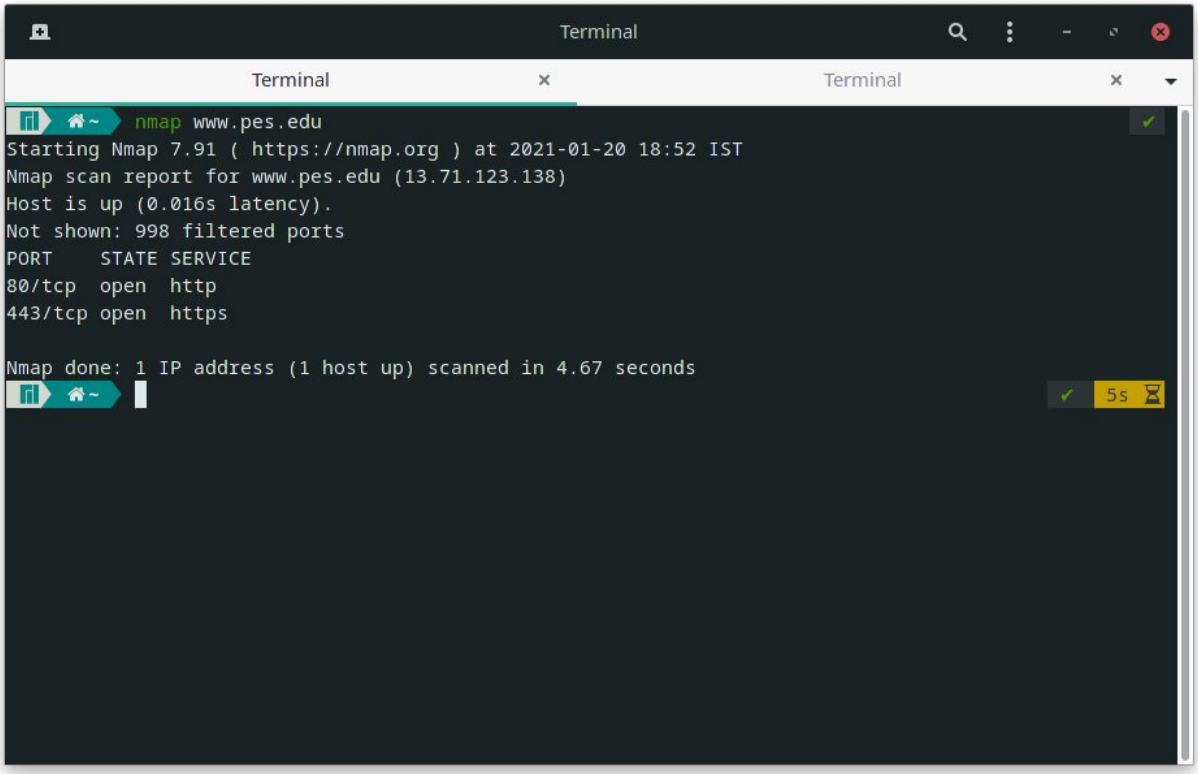


```
 1  _gateway (192.168.1.1)  2.679 ms  2.660 ms  2.635 ms
 2  abts-kk-dynamic-1.72.76.171.airtelbroadband.in (171.76.72.1)  10.306 ms  10.283 ms  10.260 ms
 3  125.21.0.149 (125.21.0.149)  10.236 ms 125.21.0.185 (125.21.0.185)  9.059 ms  9.053 ms
 4  116.119.50.115 (116.119.50.115)  17.883 ms 116.119.68.247 (116.119.68.247)  23.732 ms 116.119.57
.97 (116.119.57.97)  17.801 ms
 5  72.14.208.234 (72.14.208.234)  16.144 ms 72.14.216.192 (72.14.216.192)  16.093 ms 182.79.141.65
(182.79.141.65)  19.046 ms
 6  108.170.253.97 (108.170.253.97)  17.715 ms 74.125.242.129 (74.125.242.129)  14.894 ms 116.119.50
.117 (116.119.50.117)  13.449 ms
 7  72.14.216.192 (72.14.216.192)  14.824 ms 72.14.208.234 (72.14.208.234)  15.268 ms 72.14.216.192
(72.14.216.192)  15.222 ms
 8  108.170.253.97 (108.170.253.97)  16.957 ms 172.253.72.137 (172.253.72.137)  34.964 ms 74.125.252
.219 (74.125.252.219)  15.118 ms
 9  74.125.242.146 (74.125.242.146)  15.077 ms 142.250.212.6 (142.250.212.6)  34.854 ms 108.170.253.
119 (108.170.253.119)  66.216 ms
10  209.85.242.111 (209.85.242.111)  31.774 ms 64.233.174.3 (64.233.174.3)  16.677 ms 209.85.241.227
 (209.85.241.227)  46.841 ms
11  209.85.241.227 (209.85.241.227)  46.779 ms 108.170.248.161 (108.170.248.161)  34.538 ms 216.239.
54.92 (216.239.54.92)  68.675 ms
12  209.85.242.111 (209.85.242.111)  30.932 ms  34.485 ms 108.170.248.177 (108.170.248.177)  34.652
ms
13  bom05s15-in-f4.1e100.net (172.217.166.68)  34.603 ms  37.551 ms 209.85.242.111 (209.85.242.111)
 34.478 ms
~ 44s
```

Task 6: Explore an entire network for information (Nmap)

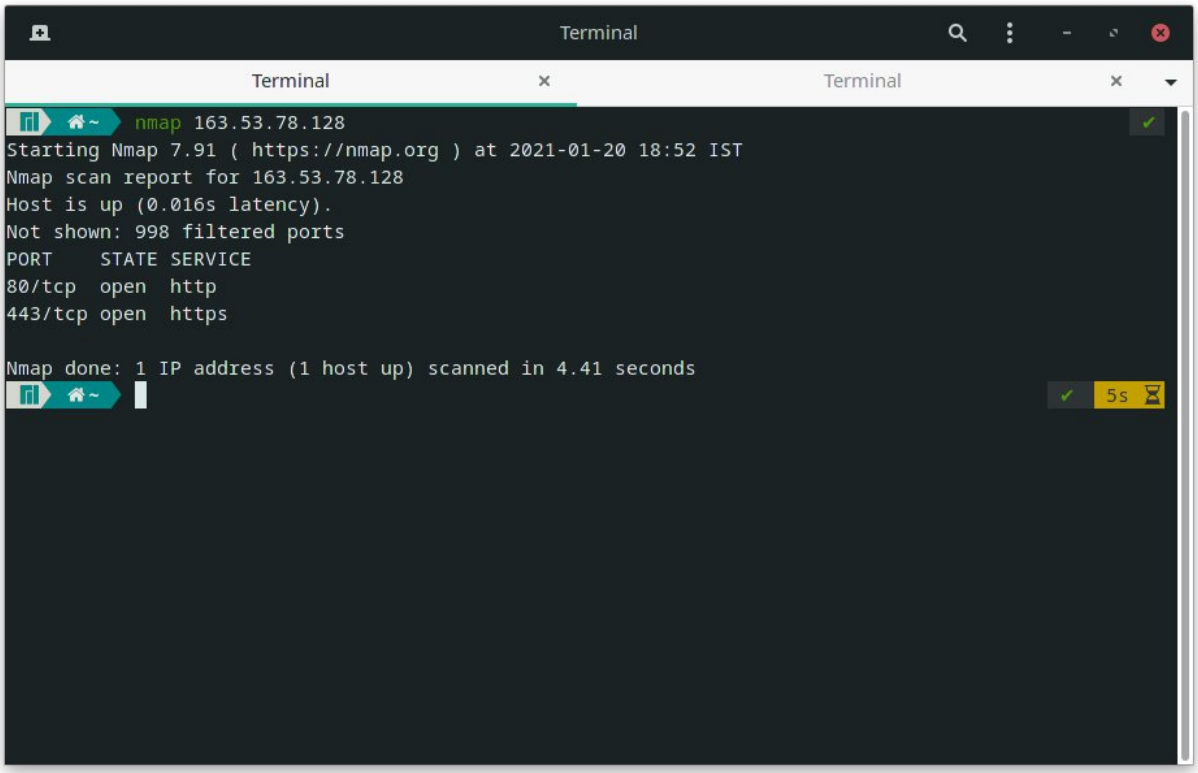
1. To scan a host using it's hostname or IP address, use

```
nmap www.pes.edu
```



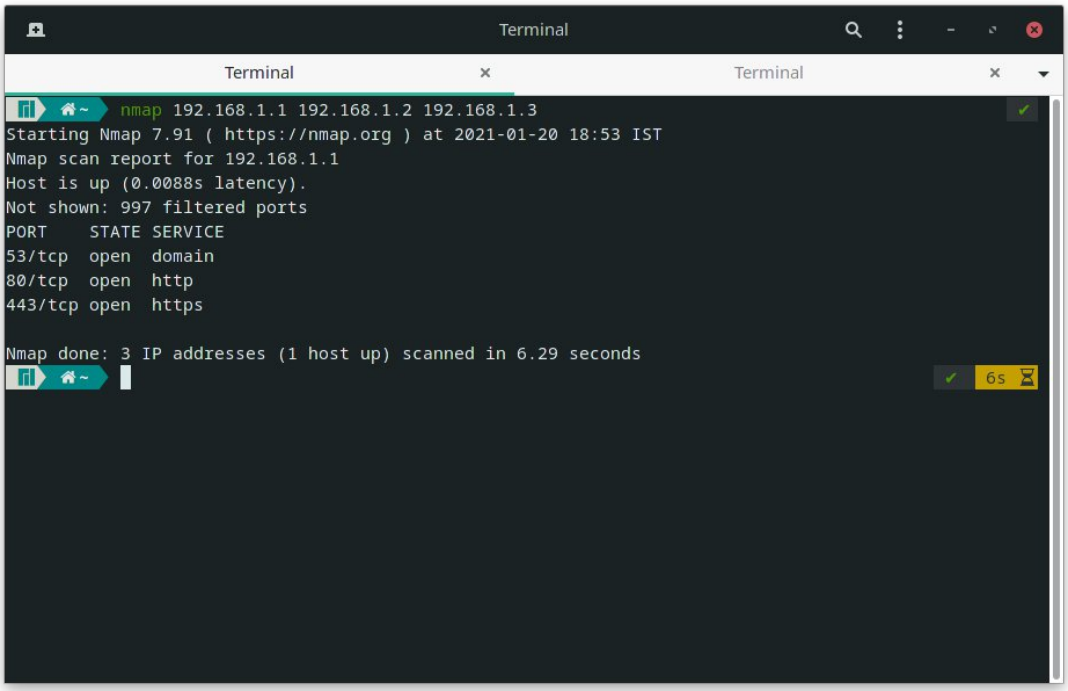

2. Alternatively, using an IP address.

```
nmap 163.53.78.128
```




3. To scan multiple IP addresses or subnets (IPv4)

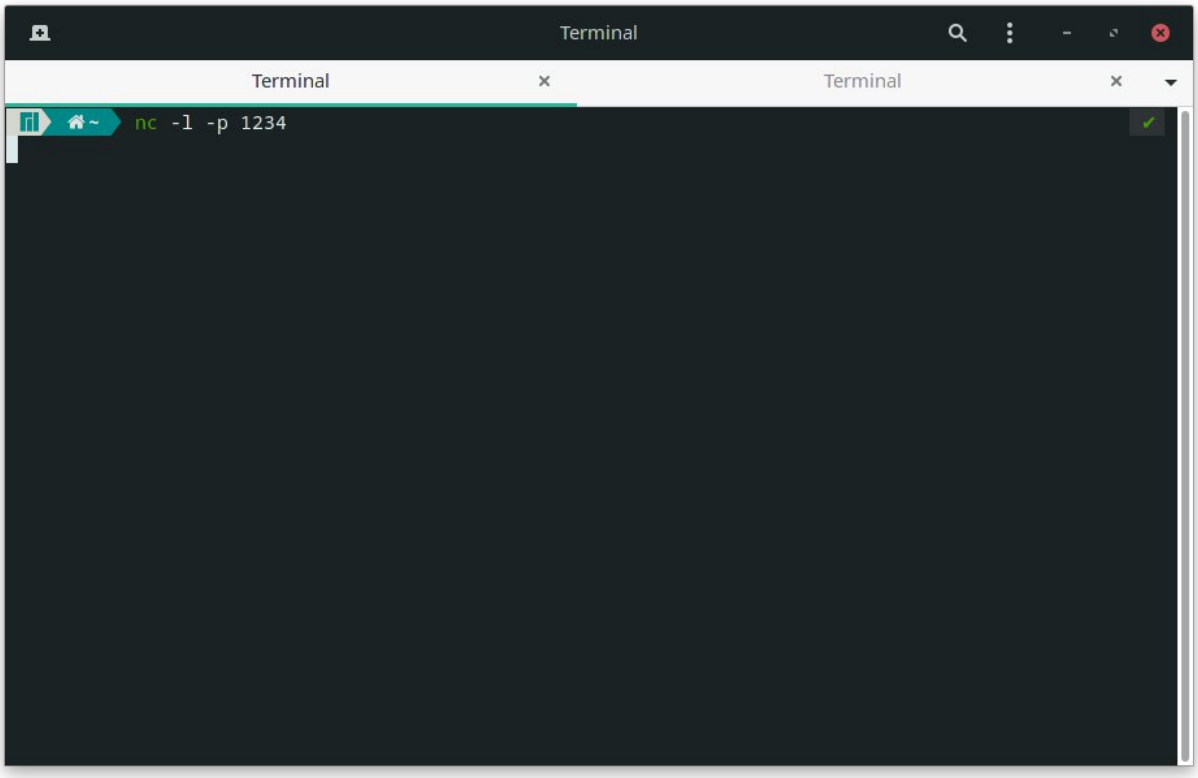
```
nmap 192.168.1.1 192.168.1.2 192.168.1.3
```



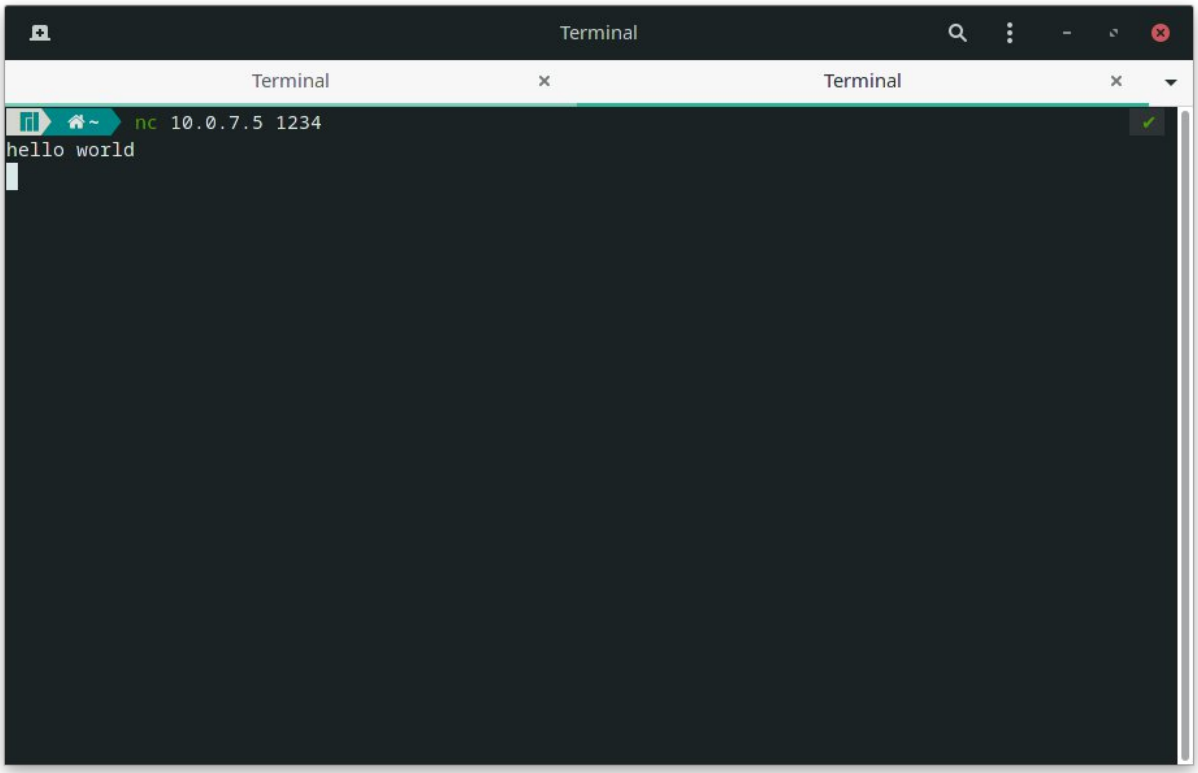
Task 7 a): Netcat as a chat tool

a) Intra System communication (using 2 terminals at once)

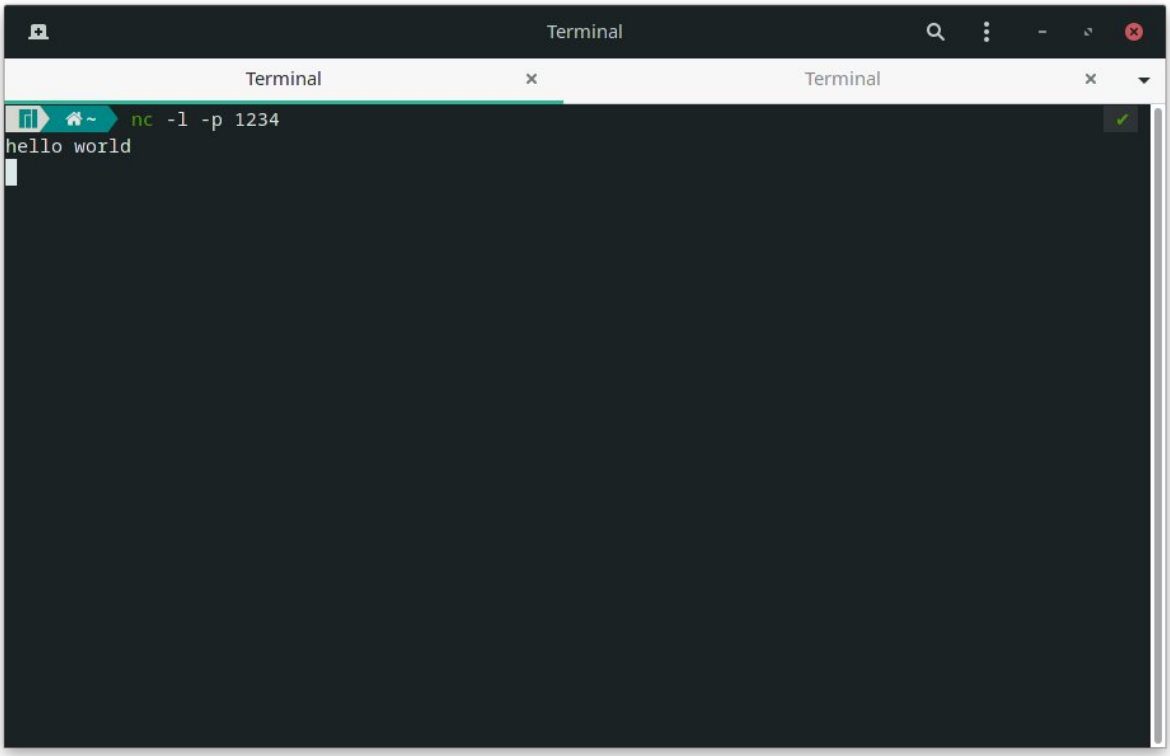
`nc -l -p 1234 (listening mode)`




`nc 10.0.7.5 1234 (Client)`

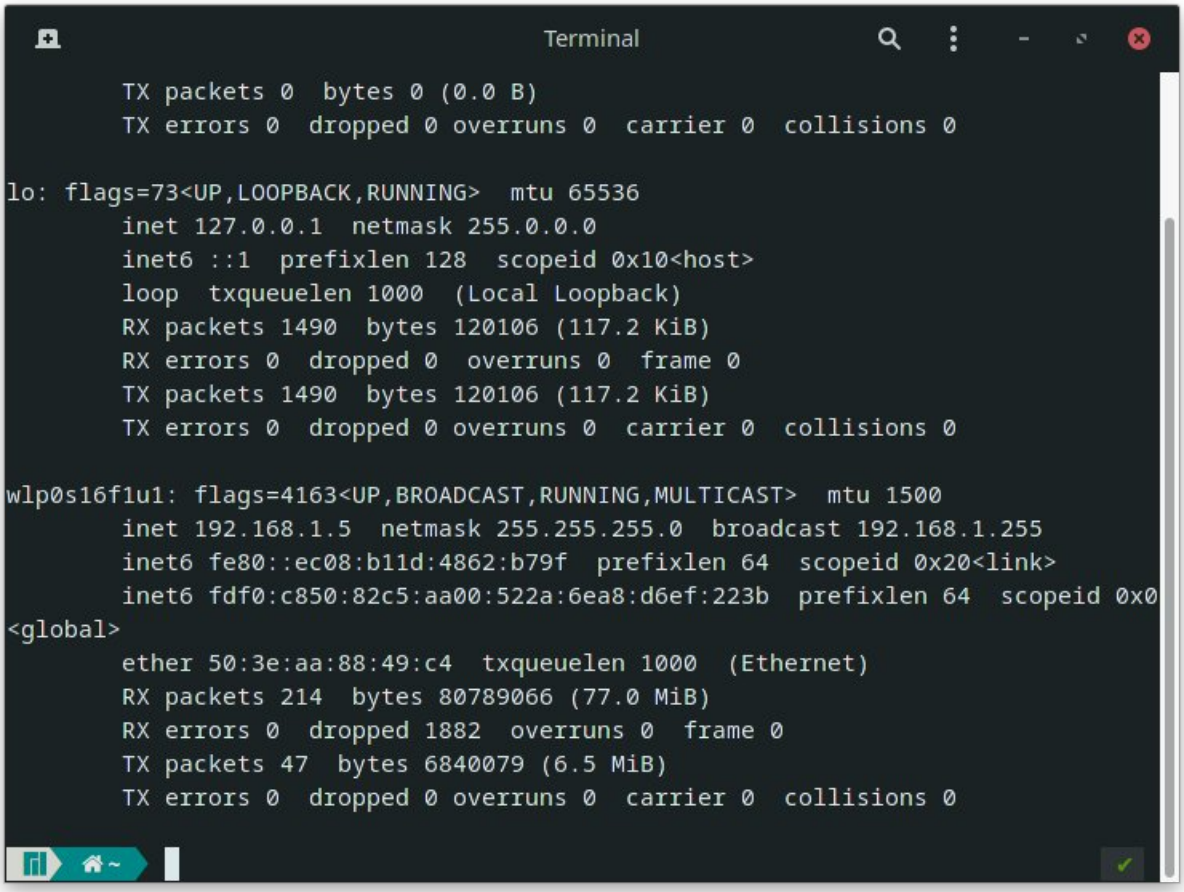



Now, we check the server for any messages.

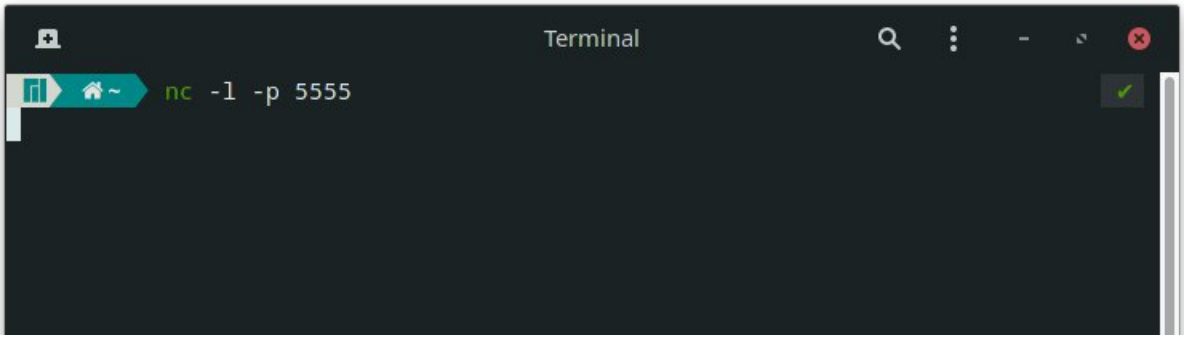


## b) Inter System Communication

Since this is over a home network, we will be using the IP address **192.168.1.5.**

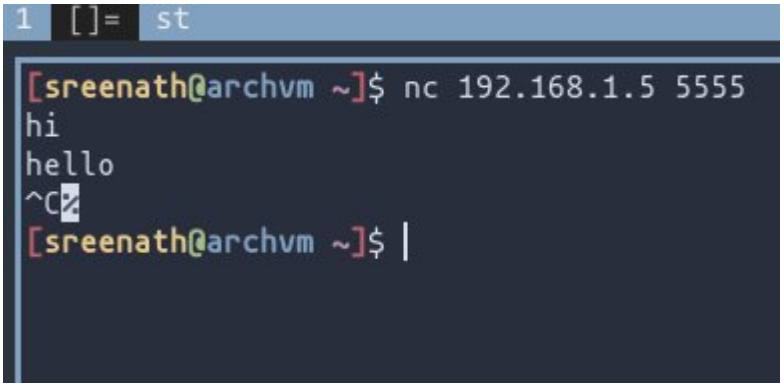



```
nc -l -p 5555
```



Now, on another system on the same network, we use the command

```
nc 192.168.1.5 5555
```




On checking the host for any messages,

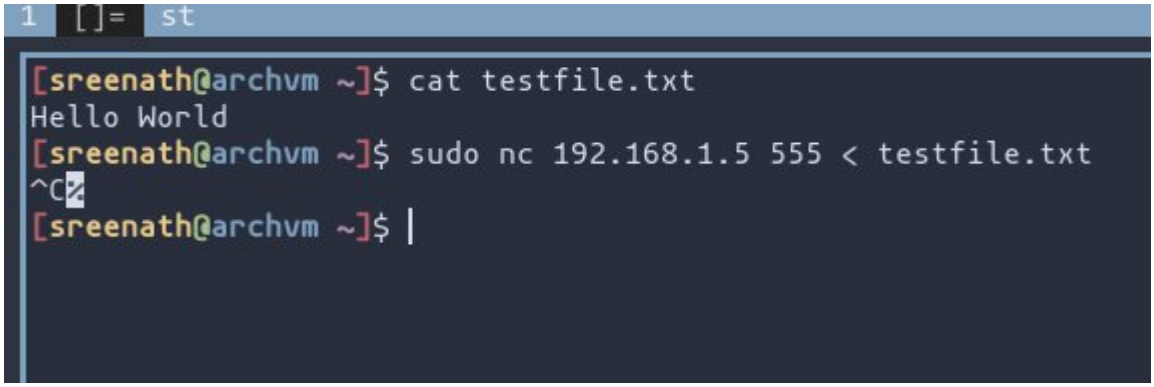

## Task 7 b): Use Netcat to transfer files

1. Create a listening server using

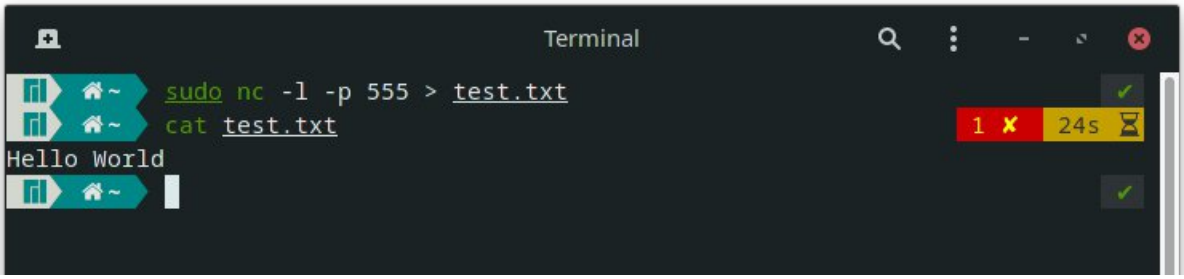
```
sudo nc -l -p 555>test.txt
```

2. On the client side, create a text file and add some contents and run

```
sudo nc 192.168.1.5 555 < testfile.txt
```



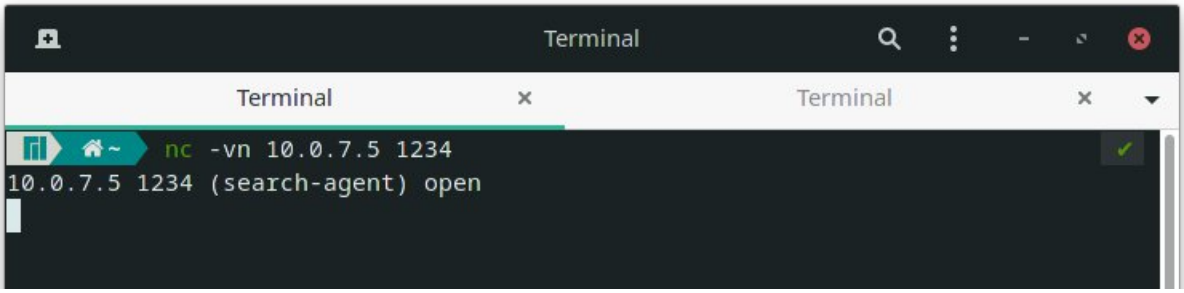
Now, checking the client side again




## Task 7 c): Other commands

1. To test if a particular TCP port of a remote host is open

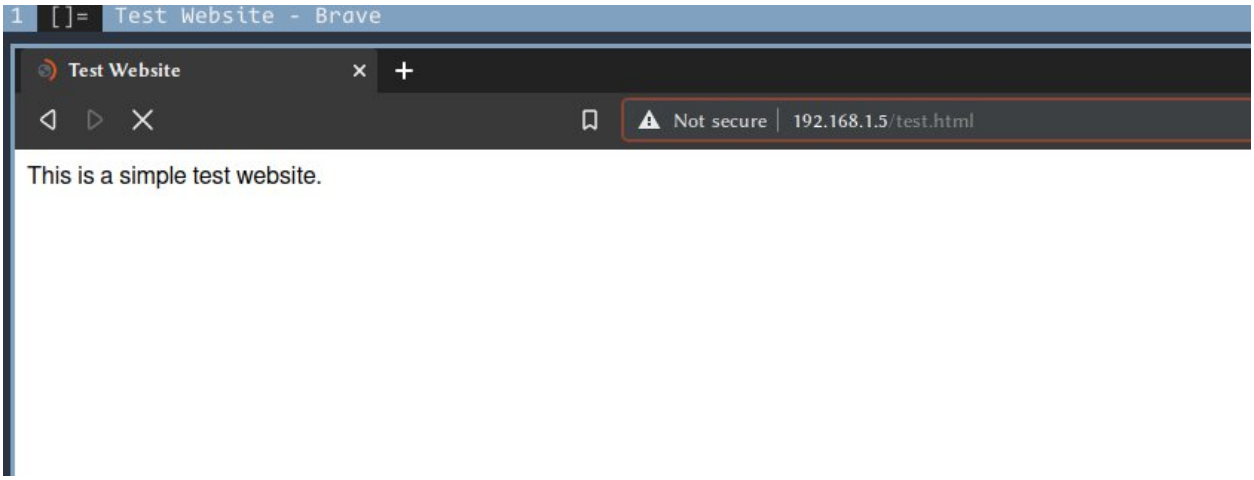
```
nc -vn 10.0.7.5 555
```




2. To run a web server with a static web page through nc, use

```
while true; do sudo nc -lp 80 <test.html;done
```

Now, we access the site from another system on the same network




**<u>Questions on above observations:</u>**

1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?

The browser ( Firefox) is running HTTP v1.1 and the server is running version 1.1 as well which can be seen in the packet captures in the above experiments.

2) When was the HTML file that you are retrieving last modified at the server?

Wireshark allows us to view the last modified field of an HTML file under the Hyper Text Transfer Protocol field of an HTTP response packet.

3) How to tell ping to exit after a specified number of ECHO_REQUEST packets?

We use the `-c` flag followed by the number of packets after which ping will terminate. It can also be terminated with an interrupt signal such as `Ctrl-c` in the terminal.

4) How will you identify remote host apps and OS?

We can use the `nmap` command to get relevant info about the server.