

Computer Networks Lab

UE19CS256

Week 1

Name: Sreenath Saikumar

Semester: 4 Section: G

SRN: PES2UG19CS406

Date: 22/01/2021

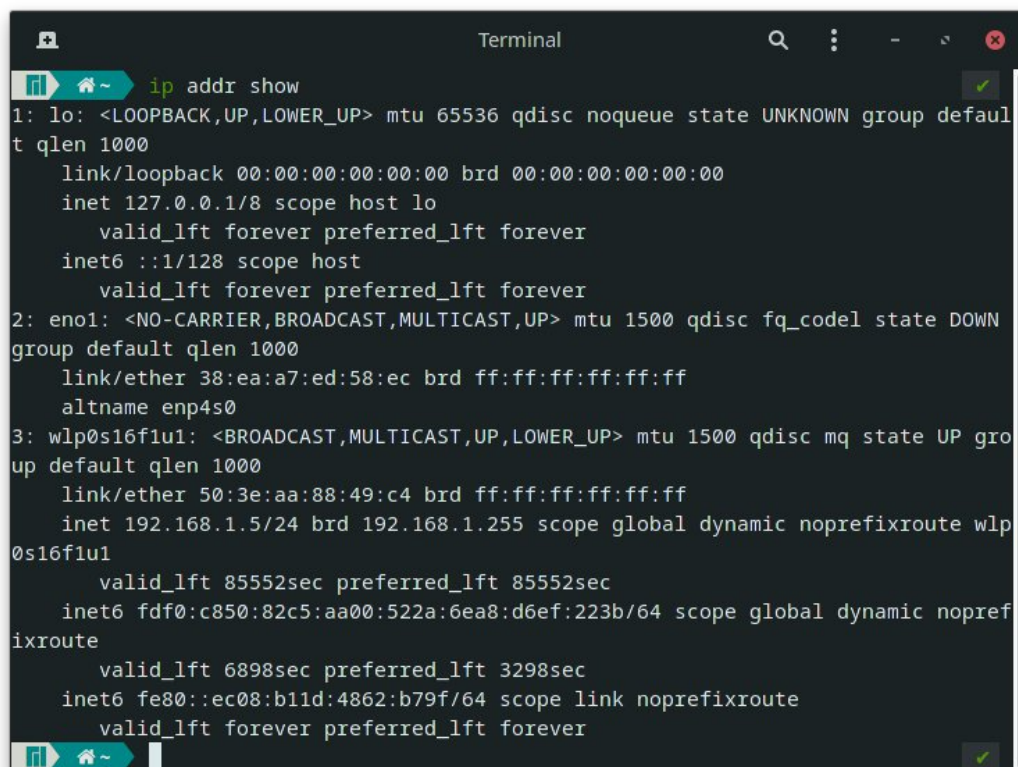
Objectives:

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



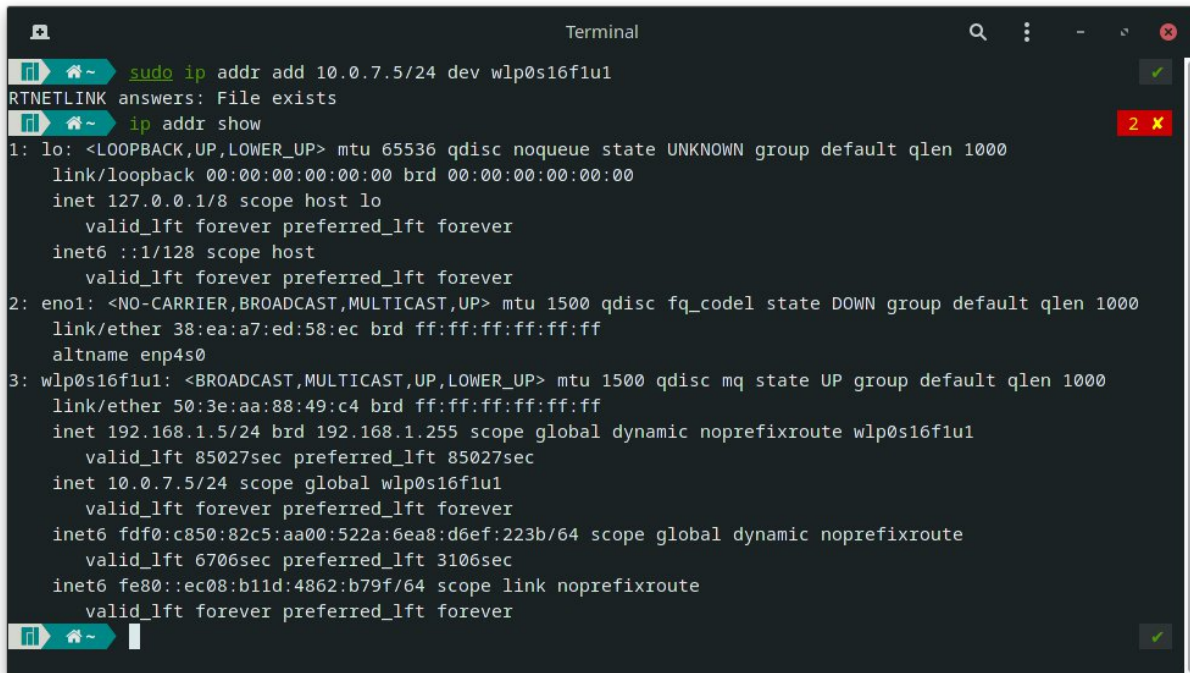
```
Terminal
ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft 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 group
   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
```



A terminal window titled "Terminal" showing the execution of two commands. The first command, `sudo ip addr add 10.0.7.5/24 dev wlp0s16f1u1`, is followed by the message "RTNETLINK answers: File exists". The second command, `ip addr show`, displays the configuration for three network interfaces: `lo`, `eno1`, and `wlp0s16f1u1`. The `wlp0s16f1u1` interface is shown with a state of "UP" and has the IP address `10.0.7.5/24` assigned to it. The terminal window has a dark background and standard window controls.

```
Terminal
~$ sudo ip addr add 10.0.7.5/24 dev wlp0s16f1u1
RTNETLINK answers: File exists
~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default 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
    altnam enp4s0
3: wlp0s16f1u1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP group 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 wlp0s16f1u1
        valid_lft 85027sec preferred_lft 85027sec
    inet 10.0.7.5/24 scope global wlp0s16f1u1
        valid_lft forever preferred_lft forever
    inet6 fdf0:c850:82c5:aa00:522a:6ea8:d6ef:223b/64 scope global dynamic noprefixroute
        valid_lft 6706sec preferred_lft 3106sec
    inet6 fe80::ec08:b11d:4862:b79f/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

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

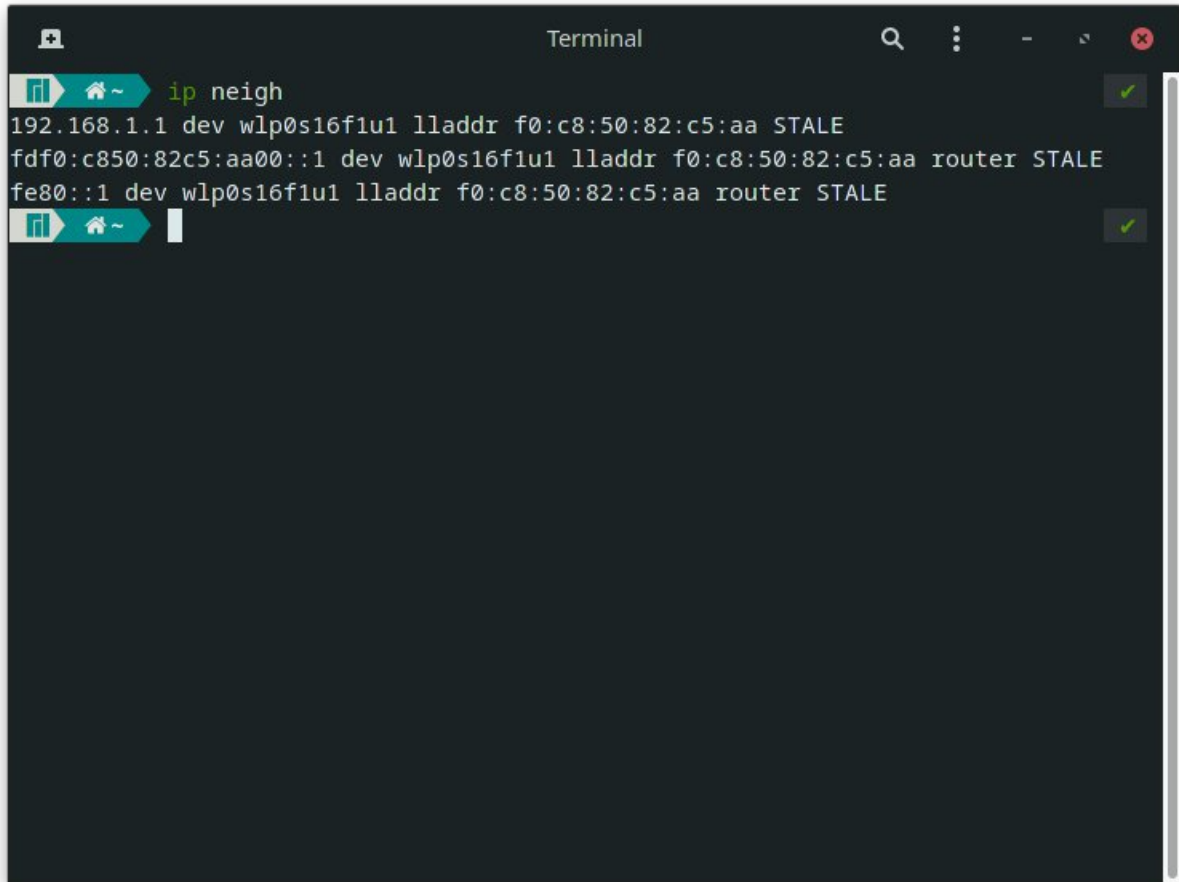
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 28 bytes 1608 (1.5 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 28 bytes 1608 (1.5 KiB)
    TX errors 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 fd0: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`

A terminal window titled "Terminal" with a dark background. The command "ip neigh" has been entered and executed. The output shows three entries in the neighbour table, all marked as "STALE". Each entry includes an IP address, a device name, a link layer address, and a router flag. The first entry is for 192.168.1.1, the second for fdf0:c850:82c5:aa00::1, and the third for fe80::1. All three entries use the same device "wlp0s16f1u1" and link layer address "f0:c8:50:82:c5:aa".

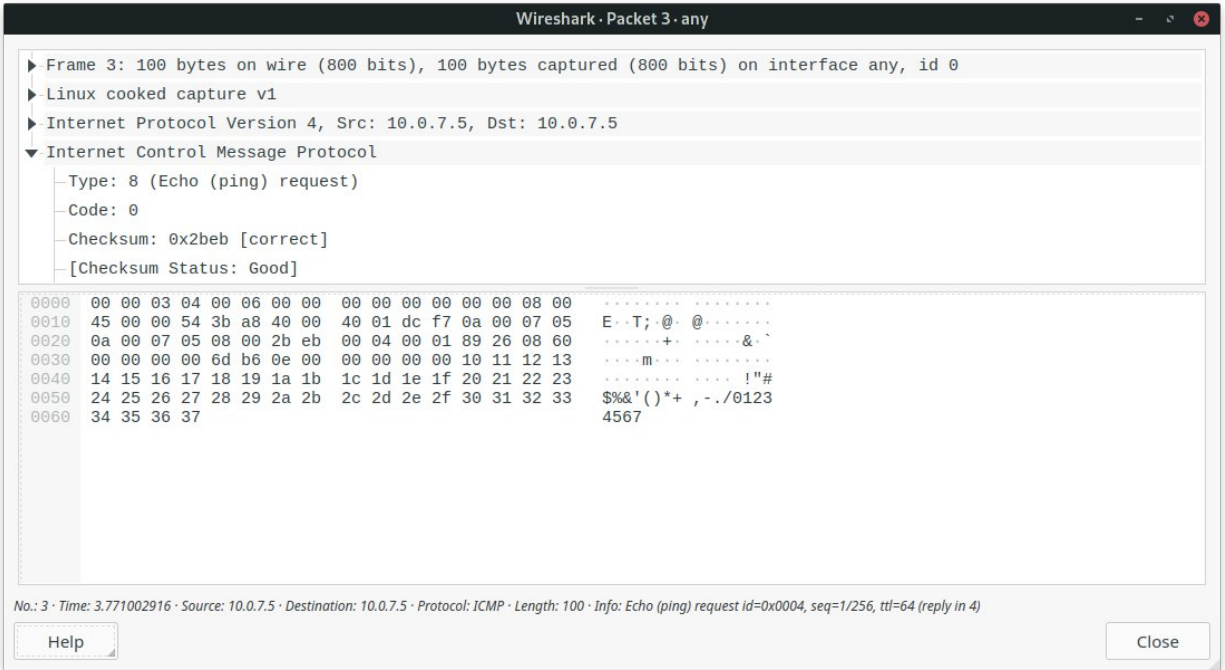
```
ip neigh
192.168.1.1 dev wlp0s16f1u1 lladdr f0:c8:50:82:c5:aa STALE
fdf0:c850:82c5:aa00::1 dev wlp0s16f1u1 lladdr f0:c8:50:82:c5:aa router STALE
fe80::1 dev wlp0s16f1u1 lladdr f0:c8:50:82:c5:aa router STALE
```

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

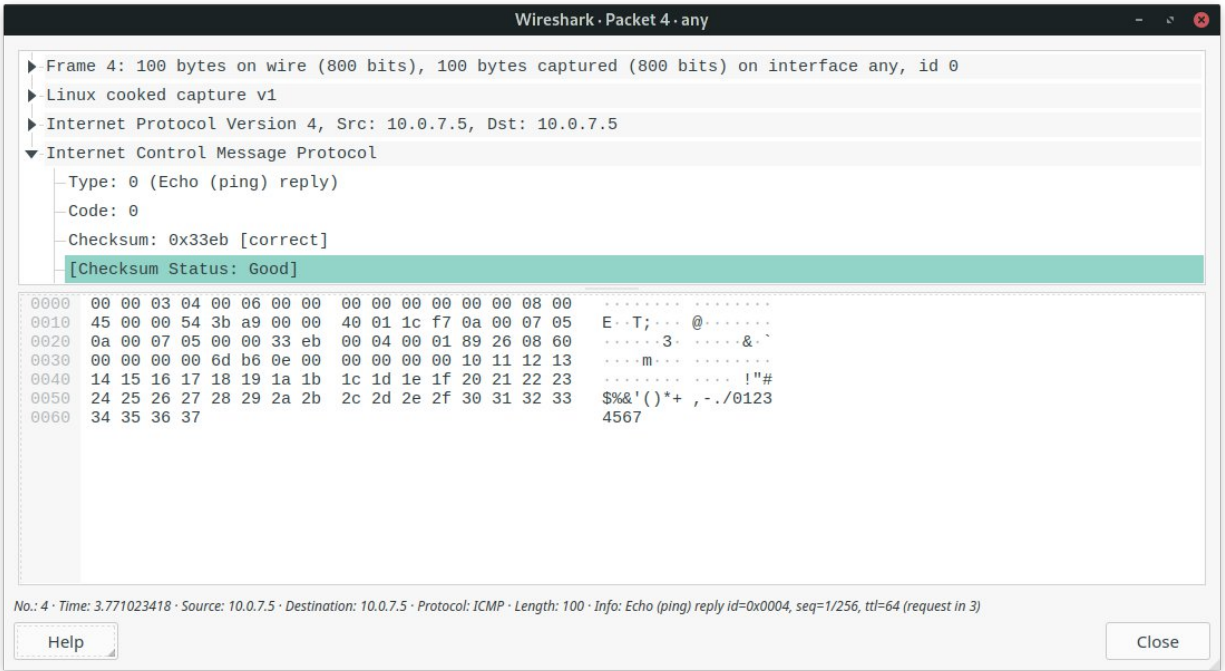
1. Assign an IP address to the system(host).
 - a. Since the host is pinging the host (simple localhost ping), the router doesn't come into play, therefore the 10.0.7.5 IP can be used.
2. Launch Wireshark and select 'any' interface.
3. Use command `ping 10.0.7.5`

```
Terminal
64 bytes from 10.0.7.5: icmp_seq=58 ttl=64 time=0.069 ms
64 bytes from 10.0.7.5: icmp_seq=59 ttl=64 time=0.064 ms
64 bytes from 10.0.7.5: icmp_seq=60 ttl=64 time=0.069 ms
64 bytes from 10.0.7.5: icmp_seq=61 ttl=64 time=0.068 ms
64 bytes from 10.0.7.5: icmp_seq=62 ttl=64 time=0.069 ms
64 bytes from 10.0.7.5: icmp_seq=63 ttl=64 time=0.073 ms
64 bytes from 10.0.7.5: icmp_seq=64 ttl=64 time=0.064 ms
64 bytes from 10.0.7.5: icmp_seq=65 ttl=64 time=0.065 ms
64 bytes from 10.0.7.5: icmp_seq=66 ttl=64 time=0.061 ms
64 bytes from 10.0.7.5: icmp_seq=67 ttl=64 time=0.074 ms
64 bytes from 10.0.7.5: icmp_seq=68 ttl=64 time=0.081 ms
64 bytes from 10.0.7.5: icmp_seq=69 ttl=64 time=0.071 ms
64 bytes from 10.0.7.5: icmp_seq=70 ttl=64 time=0.087 ms
64 bytes from 10.0.7.5: icmp_seq=71 ttl=64 time=0.058 ms
64 bytes from 10.0.7.5: icmp_seq=72 ttl=64 time=0.059 ms
64 bytes from 10.0.7.5: icmp_seq=73 ttl=64 time=0.064 ms
64 bytes from 10.0.7.5: icmp_seq=74 ttl=64 time=0.064 ms
64 bytes from 10.0.7.5: icmp_seq=75 ttl=64 time=0.083 ms
64 bytes from 10.0.7.5: icmp_seq=76 ttl=64 time=0.069 ms
64 bytes from 10.0.7.5: icmp_seq=77 ttl=64 time=0.066 ms
64 bytes from 10.0.7.5: icmp_seq=78 ttl=64 time=0.062 ms
64 bytes from 10.0.7.5: icmp_seq=79 ttl=64 time=0.064 ms
64 bytes from 10.0.7.5: icmp_seq=80 ttl=64 time=0.064 ms
```

TTL	64
Protocol used	ICMP
Time	~0.6-0.8 ms



Request packet



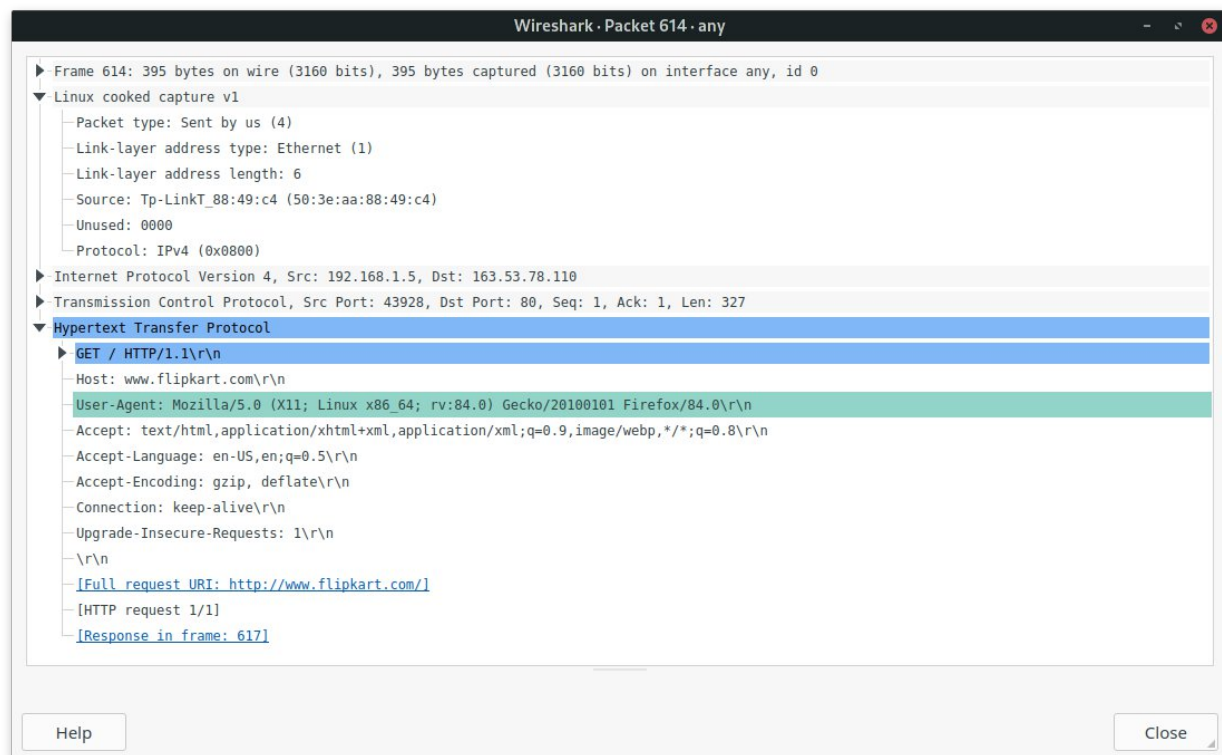
Response Packet

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

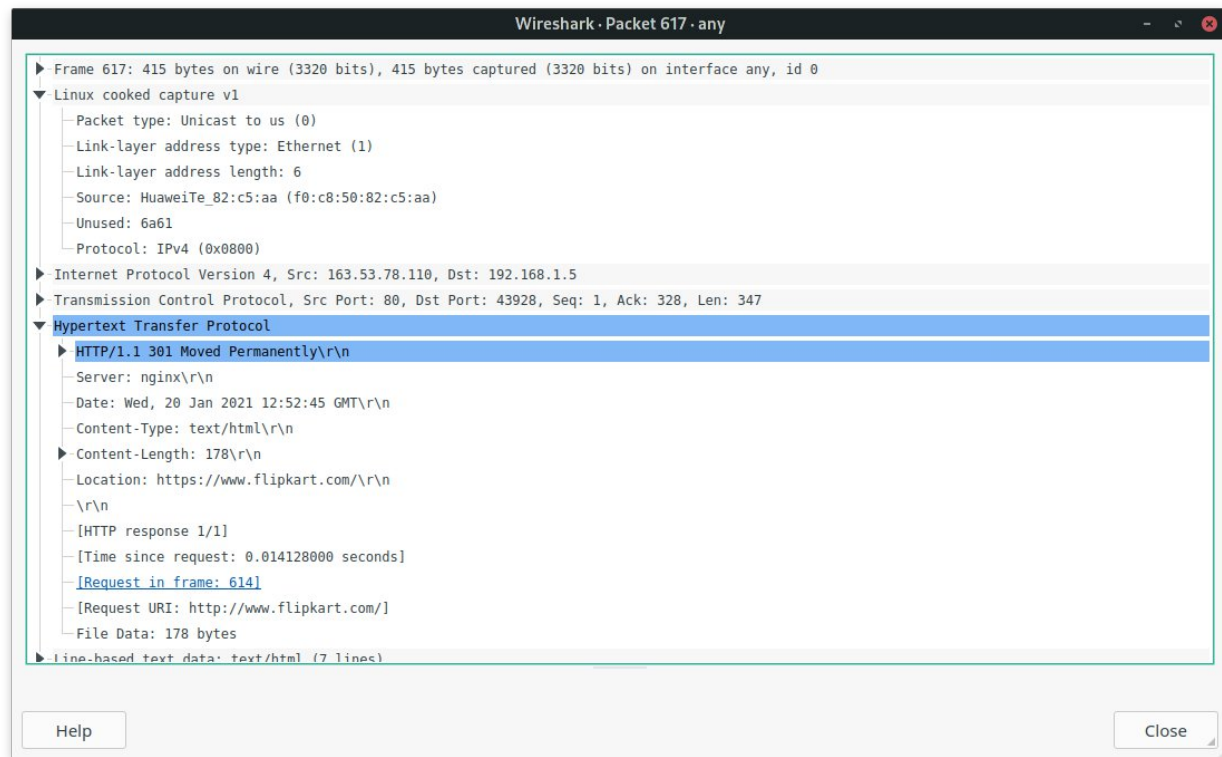
Task 3: HTTP PDU Capture

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

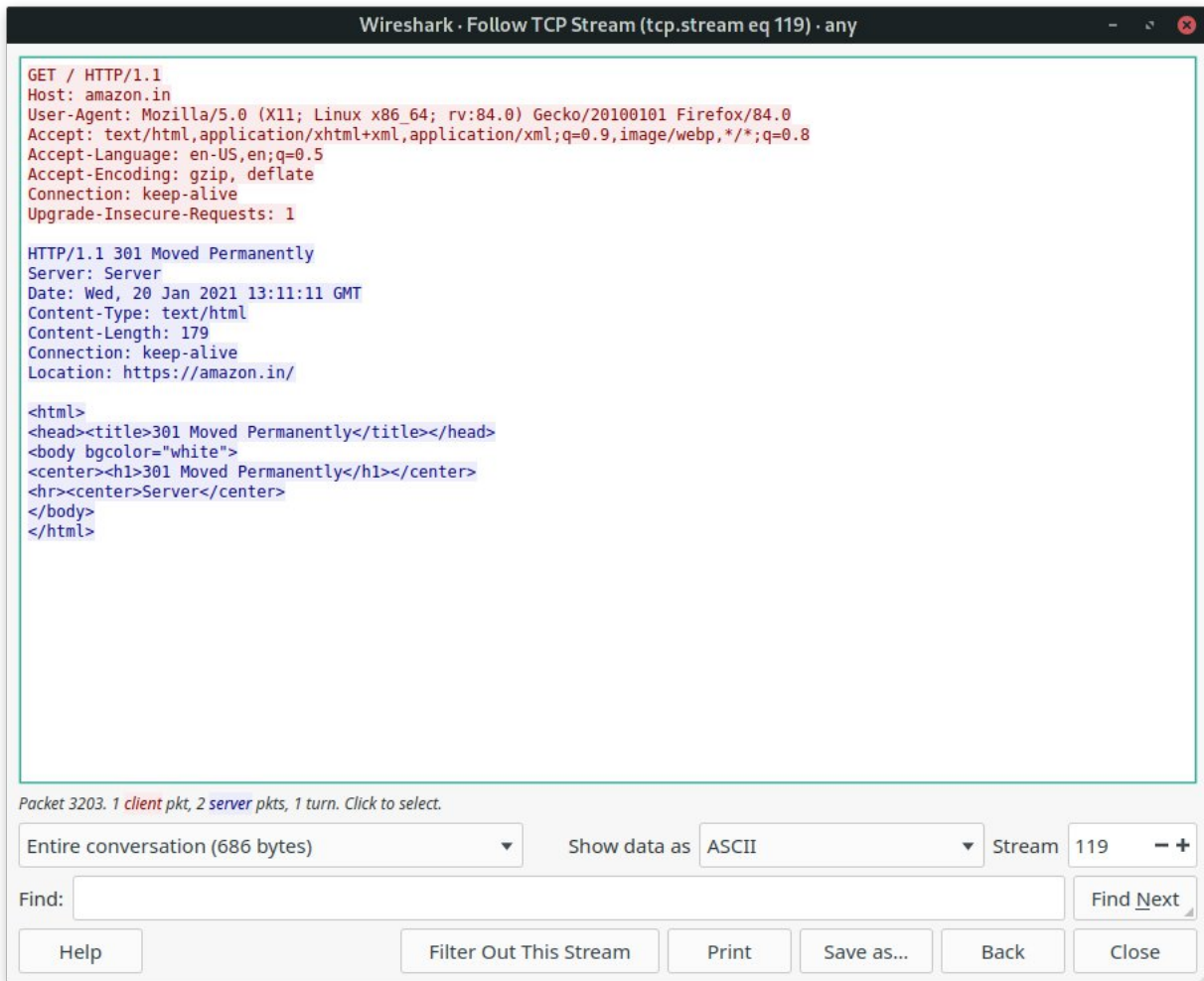
4. HTTP request and response analysis

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

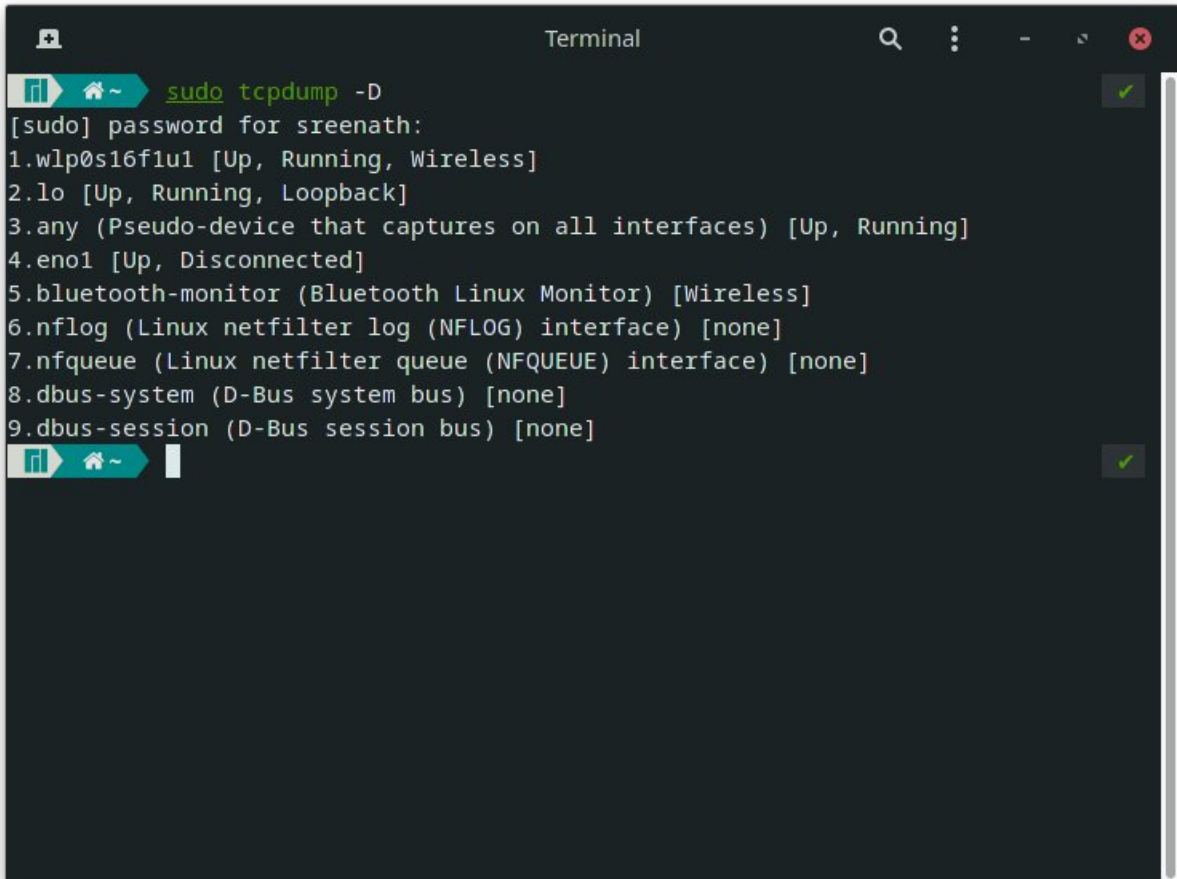
1. Right click a packet and select 'Follow TCP Stream'



Task 4: Capturing Packets with tcpdump

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

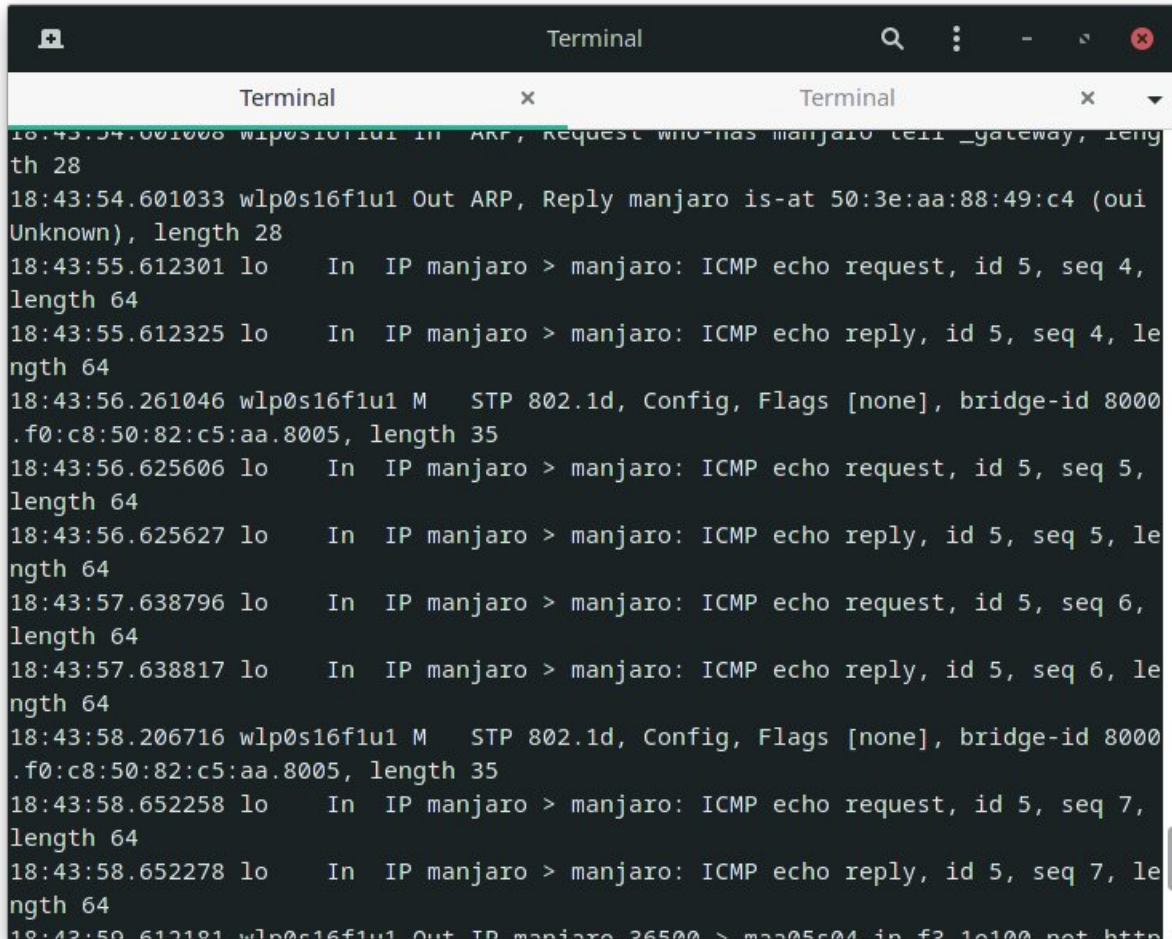
```
sudo tcpdump -D
```

A terminal window titled "Terminal" with a dark background. The command `sudo tcpdump -D` has been entered and executed. The output lists nine available interfaces for capture, each with its status in brackets. The prompt `[sudo]` is visible before the password prompt. The terminal window has standard Linux window controls (minimize, maximize, close) and a search icon in the title bar. A green checkmark icon is visible in the top right corner of the terminal area.

```
Terminal
[~] sudo tcpdump -D
[sudo] password for sreenath:
1.wlp0s16f1u1 [Up, Running, Wireless]
2.lo [Up, Running, Loopback]
3.any (Pseudo-device that captures on all interfaces) [Up, Running]
4.eno1 [Up, Disconnected]
5.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
6.nflog (Linux netfilter log (NFLOG) interface) [none]
7.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
8.dbus-system (D-Bus system bus) [none]
9.dbus-session (D-Bus session bus) [none]
[~]
```

2. Capture all packets in any interface by running:

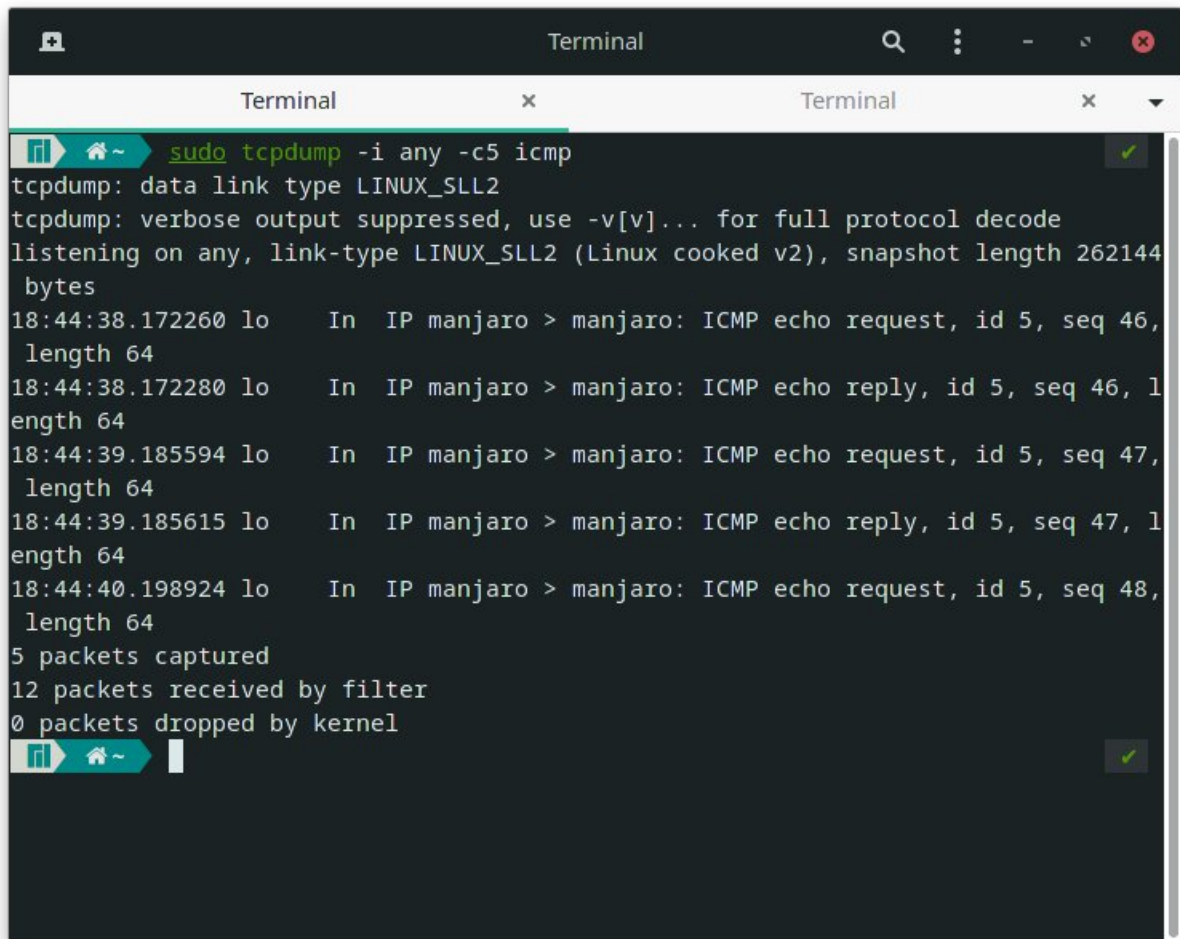
```
sudo tcpdump -i any
```



```
18:43:54.001000 wlan0s16f1u1 In ARP, Request who-has manjaro tell _gateway, length 28
18:43:54.601033 wlan0s16f1u1 Out ARP, Reply manjaro is-at 50:3e:aa:88:49:c4 (oui Unknown), length 28
18:43:55.612301 lo In IP manjaro > manjaro: ICMP echo request, id 5, seq 4, length 64
18:43:55.612325 lo In IP manjaro > manjaro: ICMP echo reply, id 5, seq 4, length 64
18:43:56.261046 wlan0s16f1u1 M STP 802.1d, Config, Flags [none], bridge-id 8000.f0:c8:50:82:c5:aa.8005, length 35
18:43:56.625606 lo In IP manjaro > manjaro: ICMP echo request, id 5, seq 5, length 64
18:43:56.625627 lo In IP manjaro > manjaro: ICMP echo reply, id 5, seq 5, length 64
18:43:57.638796 lo In IP manjaro > manjaro: ICMP echo request, id 5, seq 6, length 64
18:43:57.638817 lo In IP manjaro > manjaro: ICMP echo reply, id 5, seq 6, length 64
18:43:58.206716 wlan0s16f1u1 M STP 802.1d, Config, Flags [none], bridge-id 8000.f0:c8:50:82:c5:aa.8005, length 35
18:43:58.652258 lo In IP manjaro > manjaro: ICMP echo request, id 5, seq 7, length 64
18:43:58.652278 lo In IP manjaro > manjaro: ICMP echo reply, id 5, seq 7, length 64
18:43:59.612181 wlan0s16f1u1 Out IP manjaro 26500 > man05c04 in f2 1a100 not http
```

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

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



```
Terminal
Terminal x Terminal x
~ sudo tcpdump -i any -c5 icmp
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
18:44:38.172260 lo In IP manjaro > manjaro: ICMP echo request, id 5, seq 46,
length 64
18:44:38.172280 lo In IP manjaro > manjaro: ICMP echo reply, id 5, seq 46, l
length 64
18:44:39.185594 lo In IP manjaro > manjaro: ICMP echo request, id 5, seq 47,
length 64
18:44:39.185615 lo In IP manjaro > manjaro: ICMP echo reply, id 5, seq 47, l
length 64
18:44:40.198924 lo In IP manjaro > manjaro: ICMP echo request, id 5, seq 48,
length 64
5 packets captured
12 packets received by filter
0 packets dropped by kernel
~
```

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

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

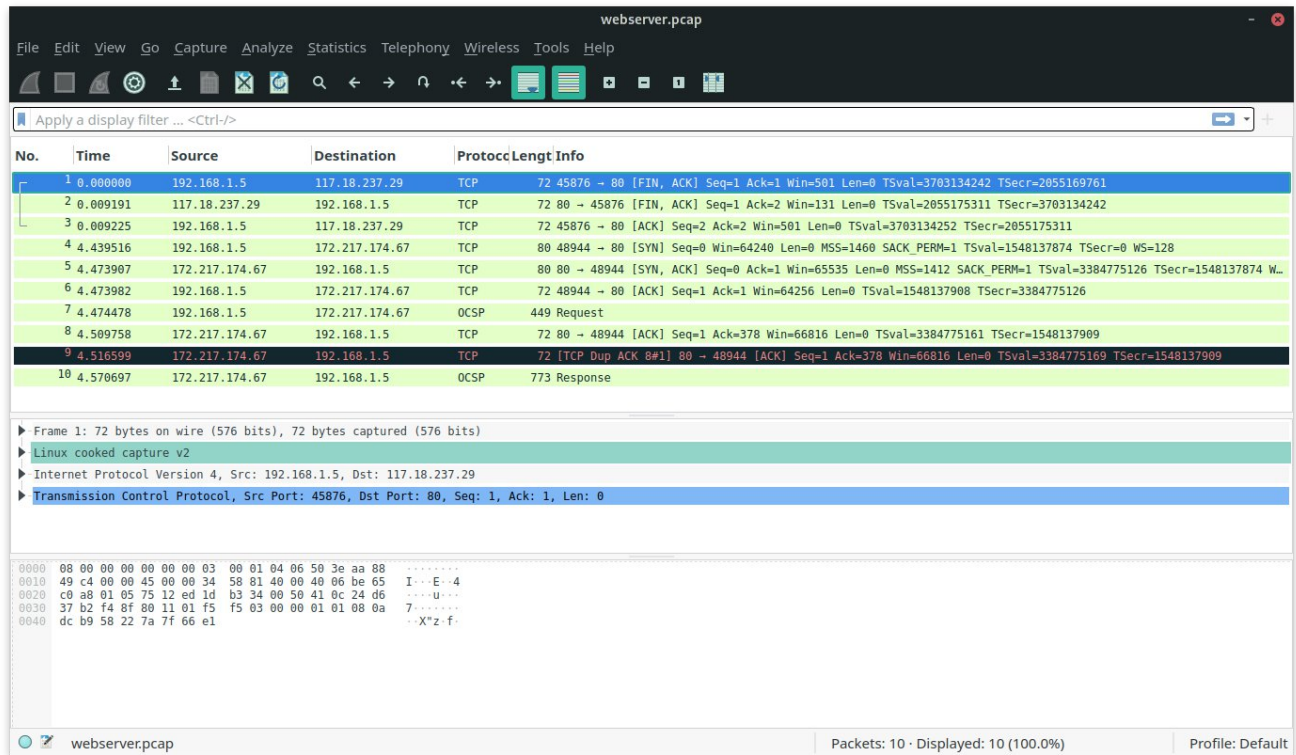


```
Terminal
Terminal x Terminal x
~ sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
18:46:35.342380 wlp0s16f1u1 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
```

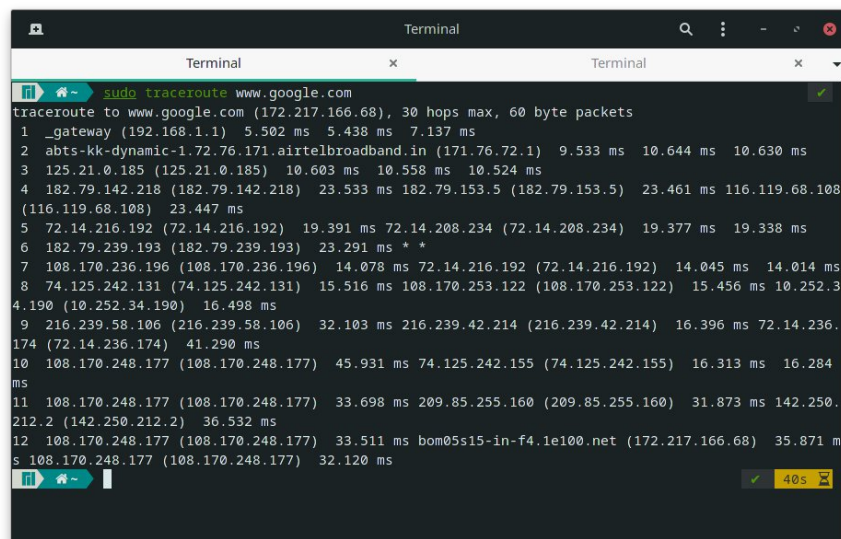
```
Terminal
Terminal x Terminal x
~ sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80
tcpdump: data link type LINUX_SLL2
tcpdump: listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
10 packets captured
11 packets received by filter
0 packets dropped by kernel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

webserver.pcap viewed in Wireshark

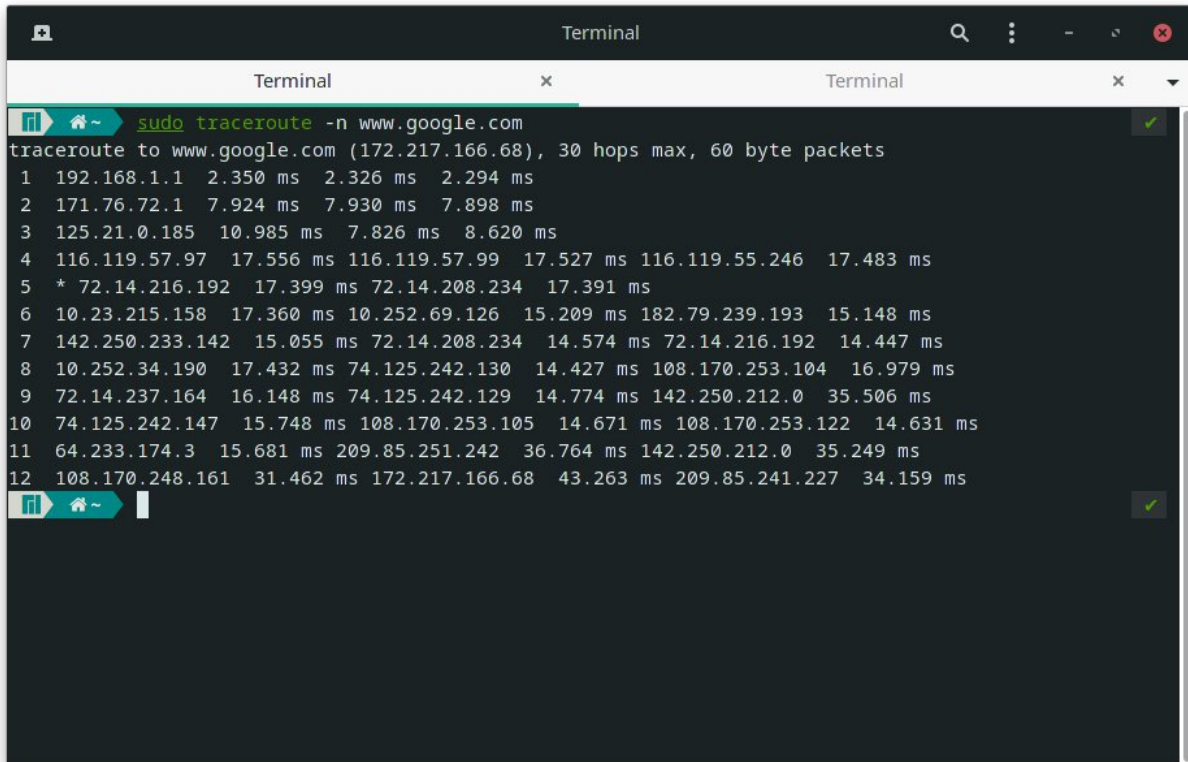
Task 5: Perform Traceroute Checks

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
```

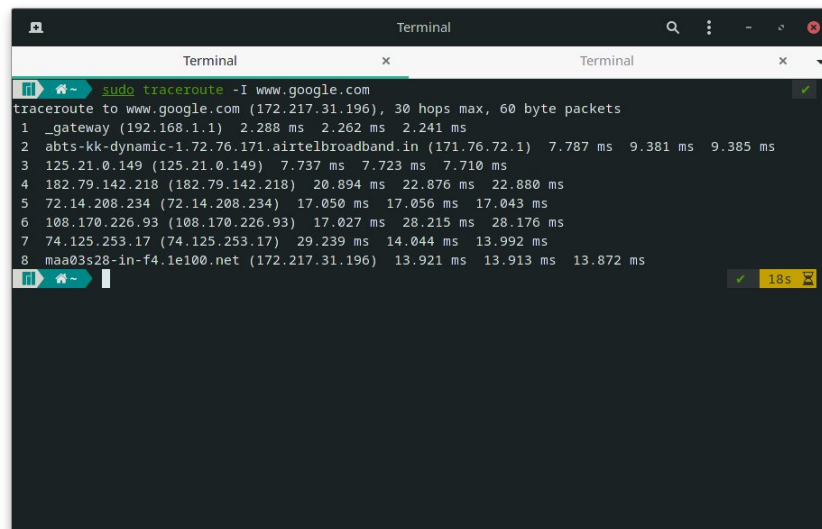


A terminal window titled "Terminal" showing the execution of the command `sudo traceroute -n www.google.com`. The output displays the network path to `www.google.com` (172.217.166.68) with 30 hops max and 60 byte packets. The path consists of 12 hops, each showing three round-trip times in milliseconds. The output is as follows:

```
traceroute to www.google.com (172.217.166.68), 30 hops max, 60 byte packets
 1 192.168.1.1 2.350 ms 2.326 ms 2.294 ms
 2 171.76.72.1 7.924 ms 7.930 ms 7.898 ms
 3 125.21.0.185 10.985 ms 7.826 ms 8.620 ms
 4 116.119.57.97 17.556 ms 116.119.57.99 17.527 ms 116.119.55.246 17.483 ms
 5 * 72.14.216.192 17.399 ms 72.14.208.234 17.391 ms
 6 10.23.215.158 17.360 ms 10.252.69.126 15.209 ms 182.79.239.193 15.148 ms
 7 142.250.233.142 15.055 ms 72.14.208.234 14.574 ms 72.14.216.192 14.447 ms
 8 10.252.34.190 17.432 ms 74.125.242.130 14.427 ms 108.170.253.104 16.979 ms
 9 72.14.237.164 16.148 ms 74.125.242.129 14.774 ms 142.250.212.0 35.506 ms
10 74.125.242.147 15.748 ms 108.170.253.105 14.671 ms 108.170.253.122 14.631 ms
11 64.233.174.3 15.681 ms 209.85.251.242 36.764 ms 142.250.212.0 35.249 ms
12 108.170.248.161 31.462 ms 172.217.166.68 43.263 ms 209.85.241.227 34.159 ms
```

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

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

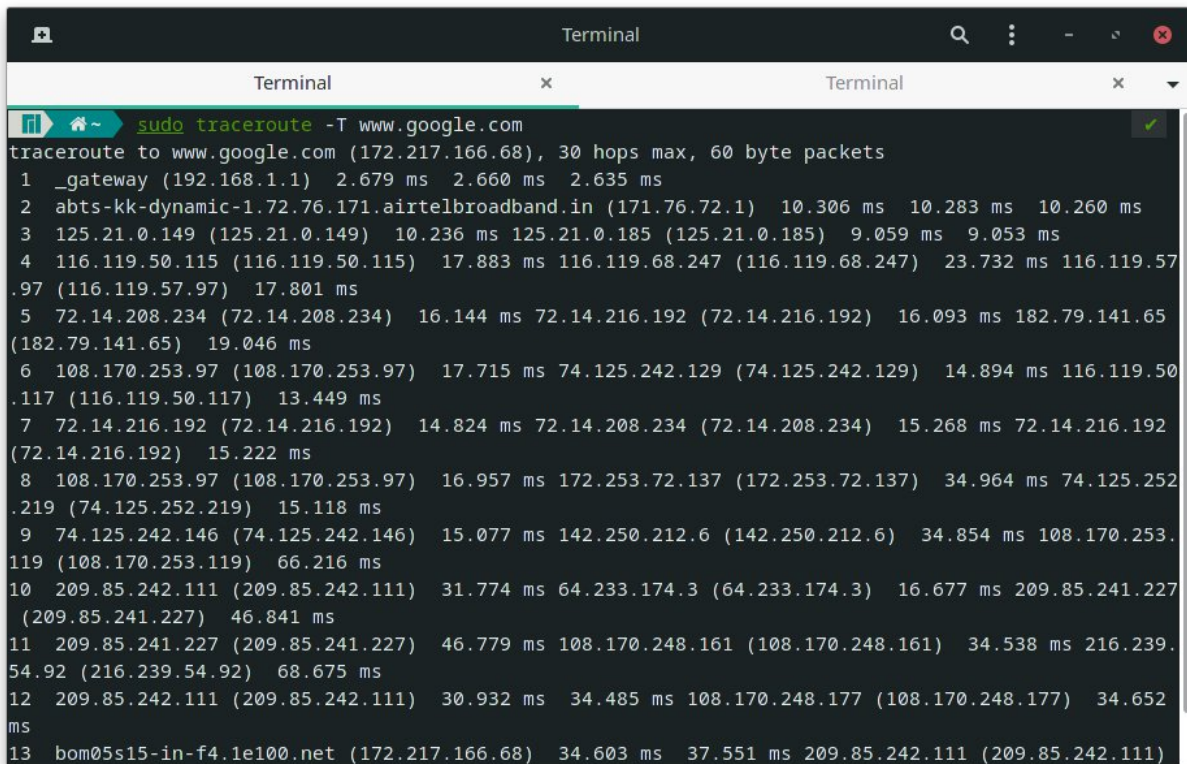


A terminal window titled "Terminal" showing the execution of the command `sudo traceroute -I www.google.com`. The output displays the network path to `www.google.com` (172.217.31.196) with 30 hops max and 60 byte packets. The path consists of 8 hops, each showing three round-trip times in milliseconds. The output is as follows:

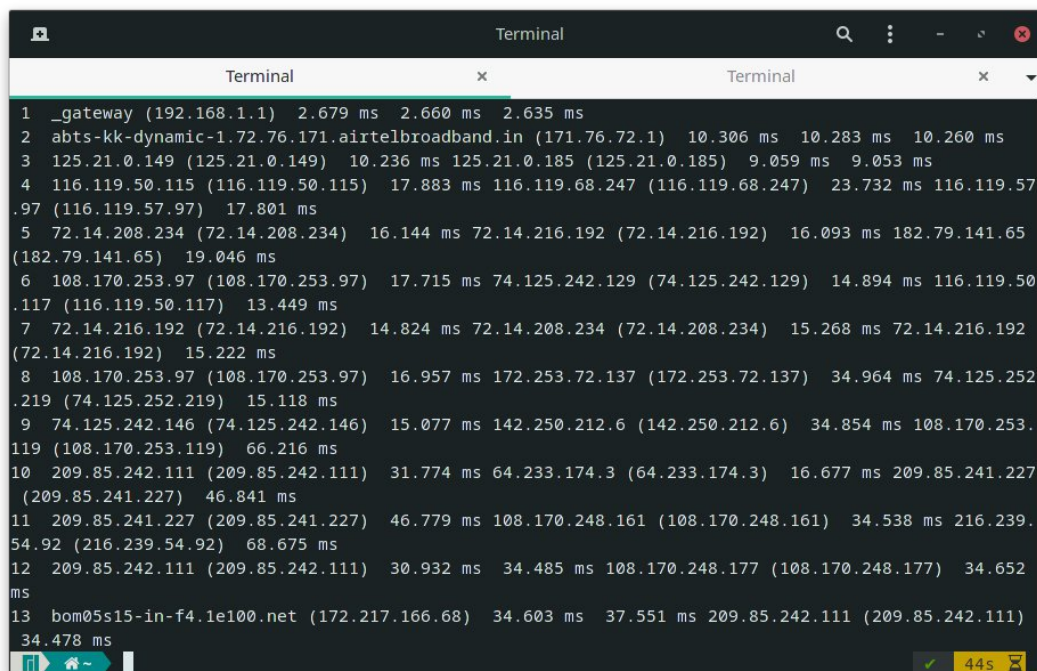
```
traceroute to www.google.com (172.217.31.196), 30 hops max, 60 byte packets
 1 _gateway (192.168.1.1) 2.288 ms 2.262 ms 2.241 ms
 2 abts-kk-dynamic-1.72.76.171.airtelbroadband.in (171.76.72.1) 7.787 ms 9.381 ms 9.385 ms
 3 125.21.0.149 (125.21.0.149) 7.737 ms 7.723 ms 7.710 ms
 4 182.79.142.218 (182.79.142.218) 20.894 ms 22.876 ms 22.880 ms
 5 72.14.208.234 (72.14.208.234) 17.050 ms 17.056 ms 17.043 ms
 6 108.170.226.93 (108.170.226.93) 17.027 ms 28.215 ms 28.176 ms
 7 74.125.253.17 (74.125.253.17) 29.239 ms 14.044 ms 13.992 ms
 8 maa03s28-in-f4.1e100.net (172.217.31.196) 13.921 ms 13.913 ms 13.872 ms
```

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

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



```
Terminal
Terminal x Terminal
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)
```

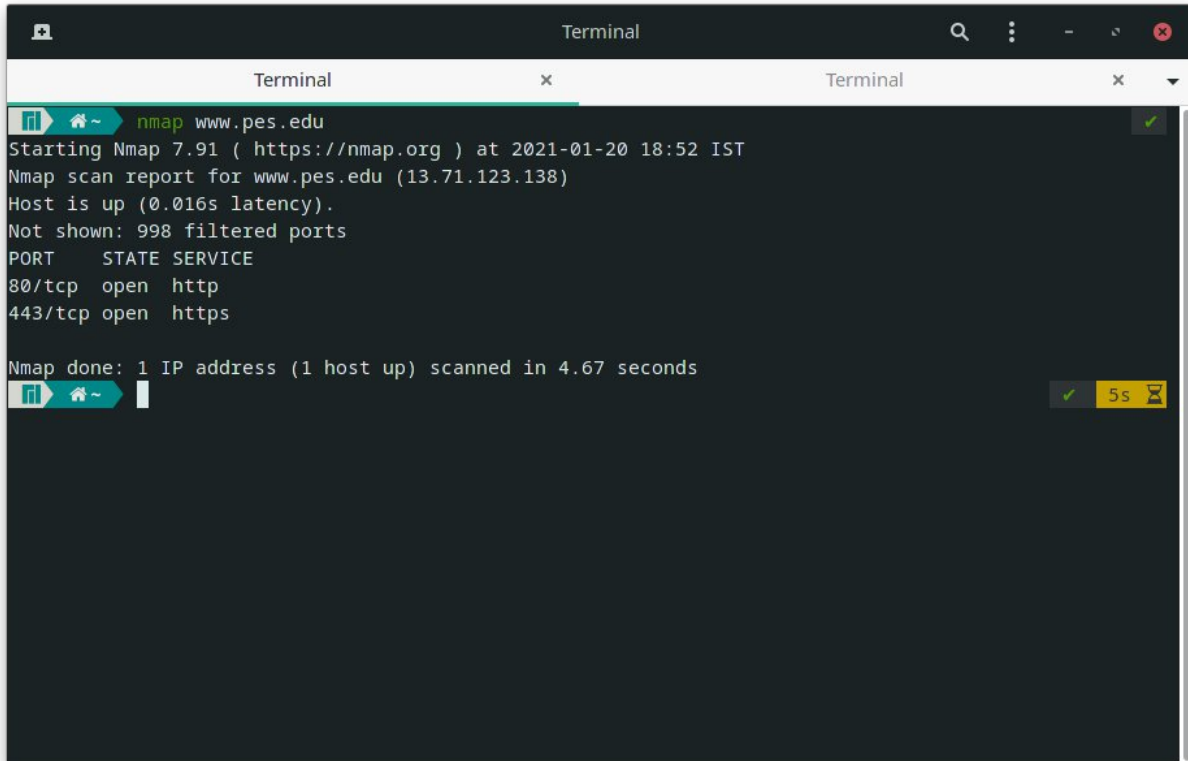


```
Terminal
Terminal x Terminal
1 _gateway (192.168.1.1)  2.679 ms  2.660 ms  2.635 ms
2 abts-kk-dynamic-1.72.76.171.airtelbroadband.in (171.76.72.1)  10.306 ms  10.283 ms  10.260 ms
3 125.21.0.149 (125.21.0.149)  10.236 ms  125.21.0.185 (125.21.0.185)  9.059 ms  9.053 ms
4 116.119.50.115 (116.119.50.115)  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
```

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`

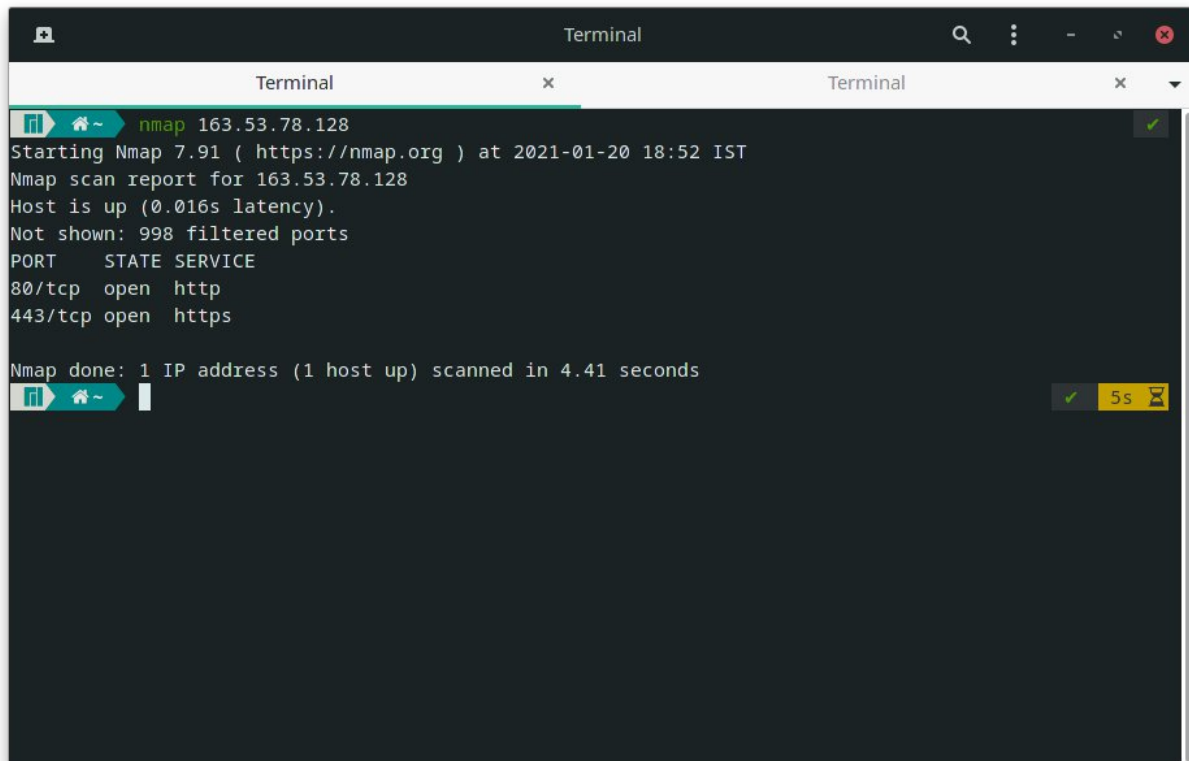
A terminal window titled "Terminal" with a dark background. The prompt shows the user is in a directory with icons for a file manager, home, and terminal. The command entered is `nmap www.pes.edu`. The output shows the Nmap version (7.91), the scan time (2021-01-20 18:52 IST), the host IP (13.71.123.138), and that the host is up. It lists two open ports: 80/tcp for http and 443/tcp for https. The scan took 4.67 seconds. A status bar at the bottom right shows a green checkmark, "5s", and a clock icon.

```
Terminal
Terminal x Terminal x
nmap www.pes.edu
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-20 18:52 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.016s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 4.67 seconds
nmap www.pes.edu
```

2. Alternatively, using an IP address.

`nmap 163.53.78.128`



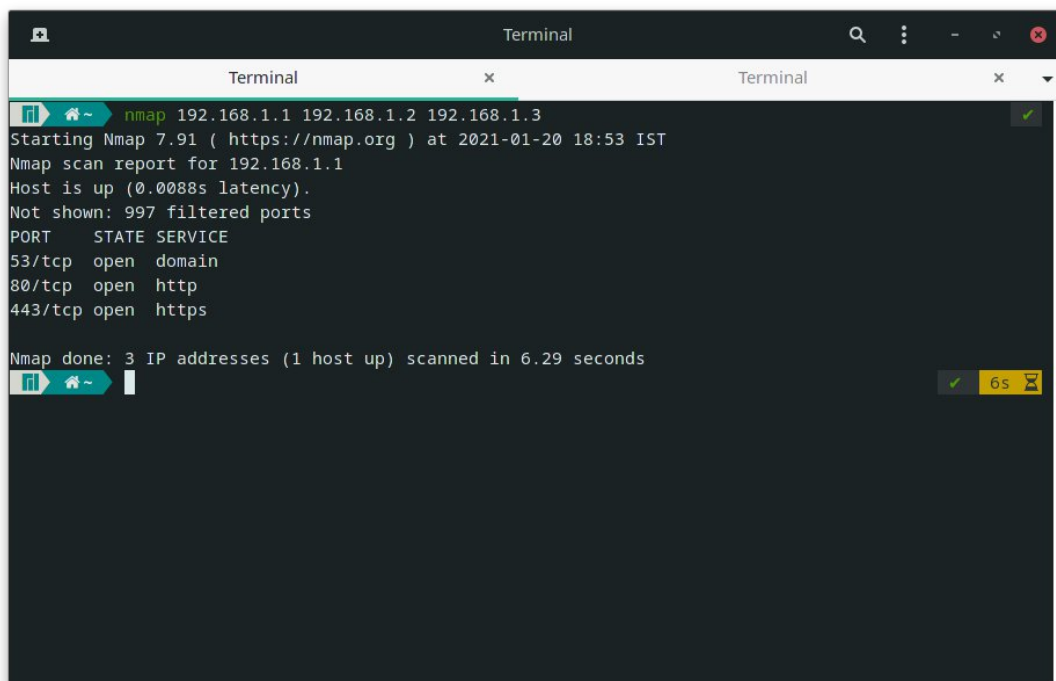
A terminal window titled "Terminal" with a search icon and window controls. It shows the execution of the command `nmap 163.53.78.128`. The output includes the Nmap version (7.91), the target IP, a confirmation that the host is up, and a list of open ports (80/tcp for http and 443/tcp for https). The scan completed in 4.41 seconds. A status bar at the bottom right shows a green checkmark, a 5-second timer, and a pause icon.

```
nmap 163.53.78.128
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-20 18:52 IST
Nmap scan report for 163.53.78.128
Host is up (0.016s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https

Nmap done: 1 IP address (1 host up) scanned in 4.41 seconds
```

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

```
nmap 192.168.1.1 192.168.1.2 192.168.1.3
```



A terminal window titled "Terminal" showing the execution of the command `nmap 192.168.1.1 192.168.1.2 192.168.1.3`. The output shows the scan of the first IP (192.168.1.1) with open ports 53/tcp (domain), 80/tcp (http), and 443/tcp (https). The scan completed in 6.29 seconds. A status bar at the bottom right shows a green checkmark, a 6-second timer, and a pause icon.

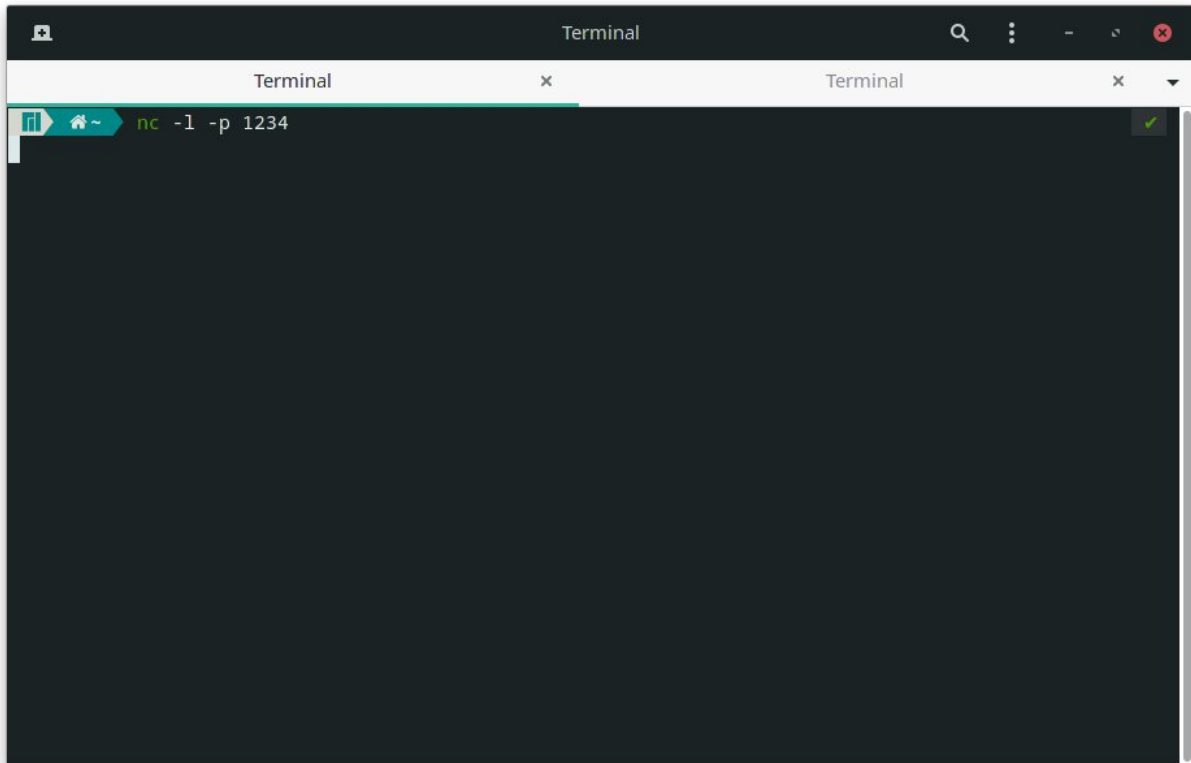
```
nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.91 ( https://nmap.org ) at 2021-01-20 18:53 IST
Nmap scan report for 192.168.1.1
Host is up (0.0088s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https

Nmap done: 3 IP addresses (1 host up) scanned in 6.29 seconds
```

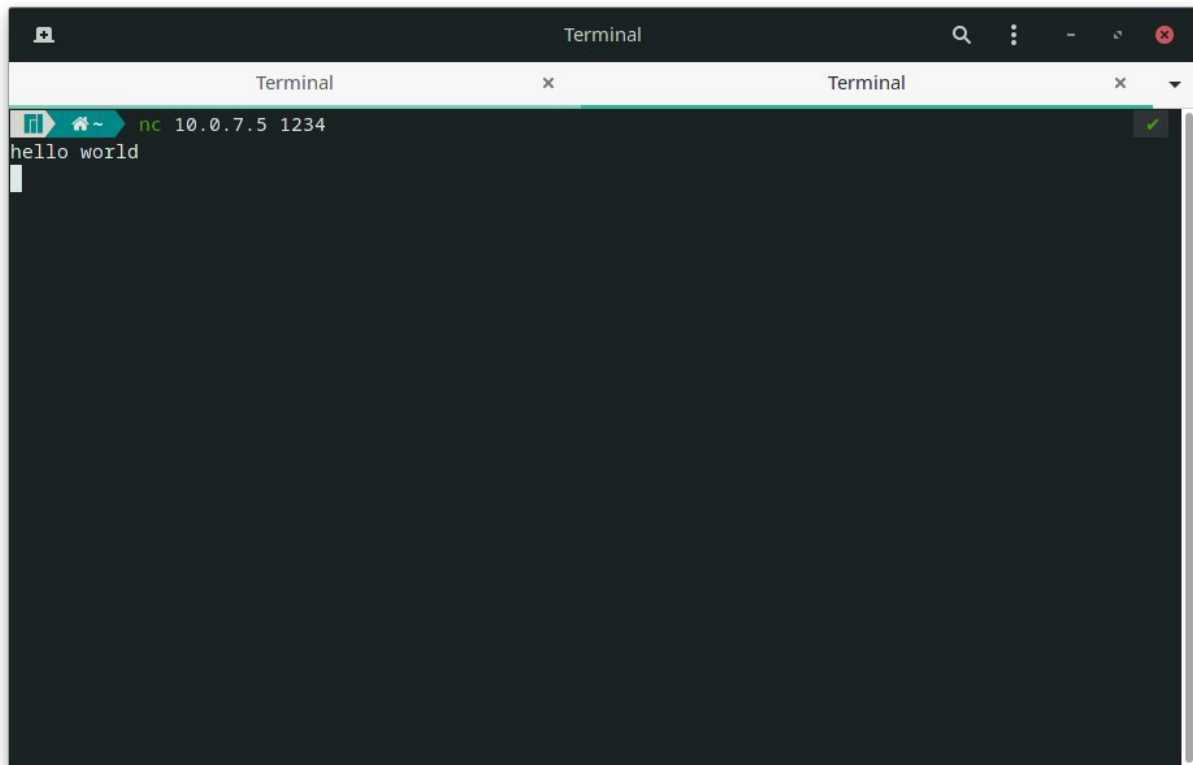
Task 7 a): Netcat as a chat tool

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

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



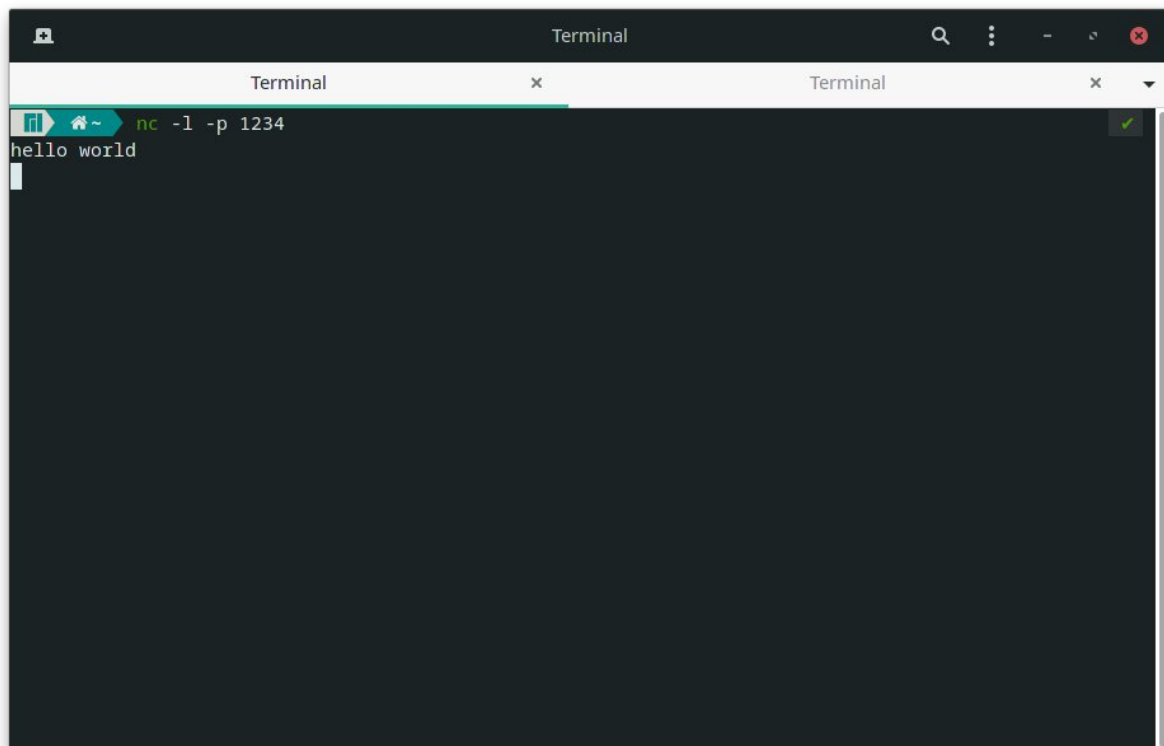
```
nc 10.0.7.5 1234 (Client)
```

```
Terminal
nc 10.0.7.5 1234
hello world
```

A terminal window titled "Terminal" with a search icon and window controls. It shows a netcat listener on IP 10.0.7.5 port 1234. The prompt is "nc 10.0.7.5 1234" and the received message is "hello world". A green checkmark icon is in the top right corner.

Now, we check the server for any messages.

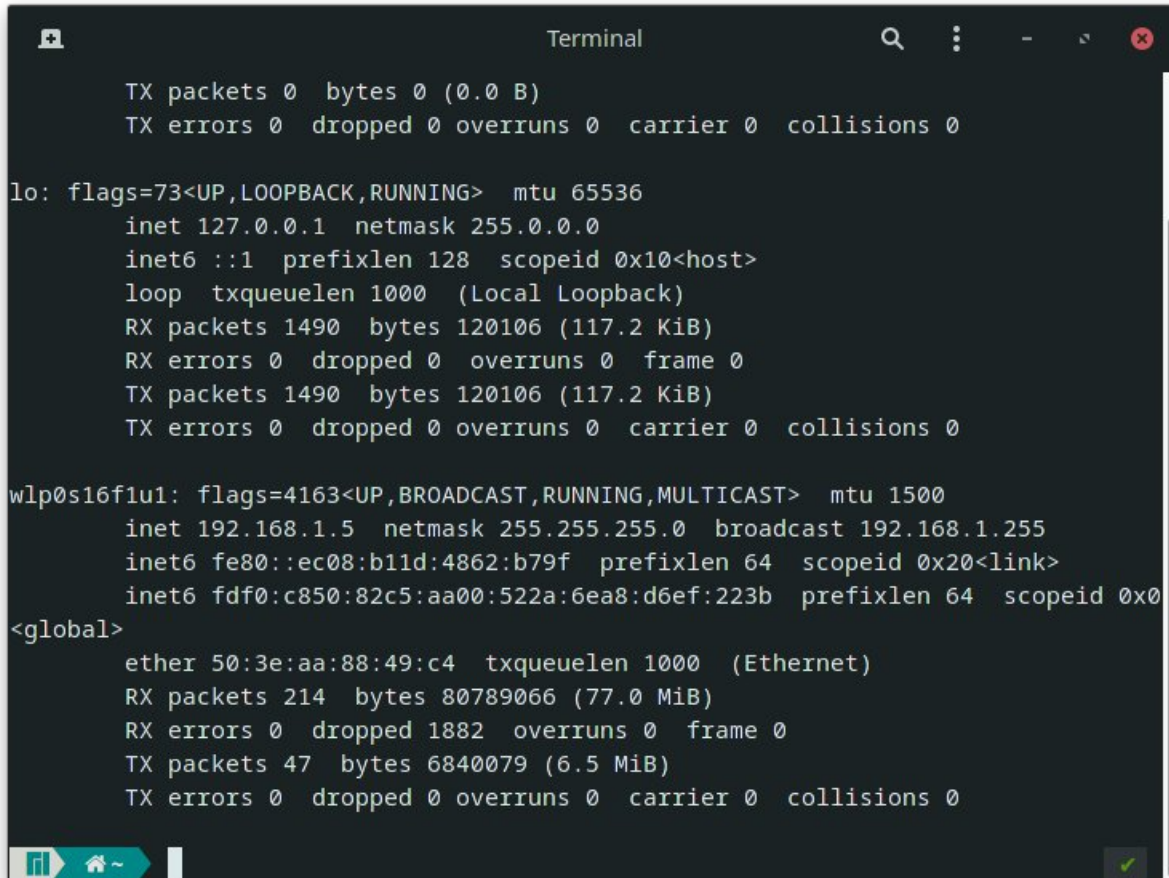


```
Terminal
nc -l -p 1234
hello world
```

A terminal window titled "Terminal" with a search icon and window controls. It shows a netcat listener on port 1234. The prompt is "nc -l -p 1234" and the received message is "hello world". A green checkmark icon is in the top right corner.

b) Inter System Communication

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

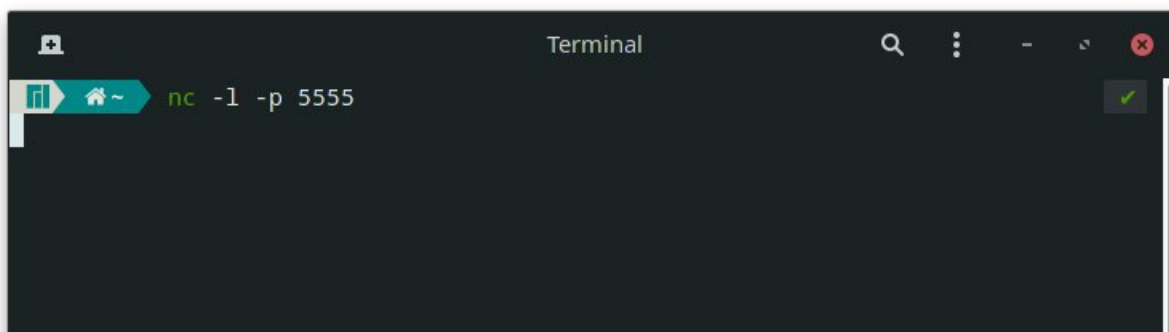


```
TX packets 0  bytes 0 (0.0 B)
TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

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 1490  bytes 120106 (117.2 KiB)
    RX errors 0  dropped 0  overruns 0  frame 0
    TX packets 1490  bytes 120106 (117.2 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 214  bytes 80789066 (77.0 MiB)
    RX errors 0  dropped 1882  overruns 0  frame 0
    TX packets 47  bytes 6840079 (6.5 MiB)
    TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

`nc -l -p 5555`



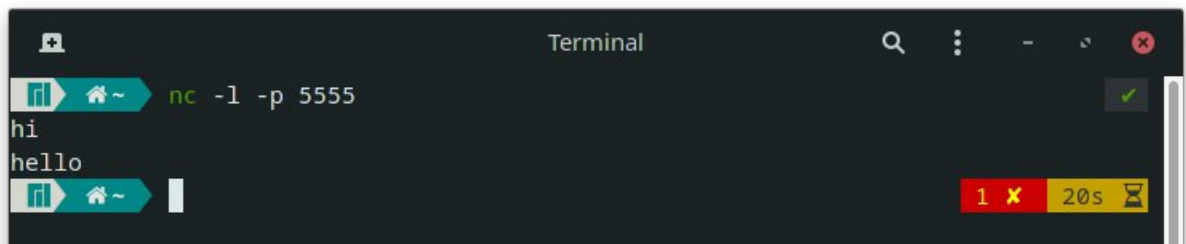
```
nc -l -p 5555
```

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

```
nc 192.168.1.5 5555
```

```
1  []= st
[sreenath@archvm ~]$ nc 192.168.1.5 5555
hi
hello
^C
[sreenath@archvm ~]$ |
```

On checking the host for any messages,



Task 7 b): Use Netcat to transfer files

1. Create a listening server using

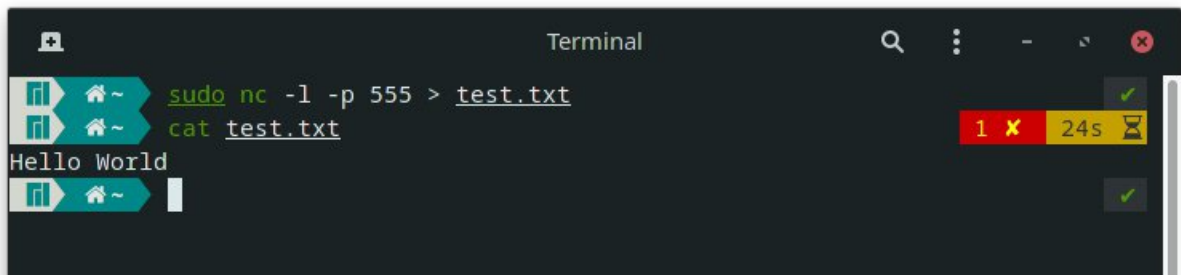
```
sudo nc -l -p 555 > test.txt
```

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

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

```
1  []= st
[sreenath@archvm ~]$ cat testfile.txt
Hello World
[sreenath@archvm ~]$ sudo nc 192.168.1.5 555 < testfile.txt
^C
[sreenath@archvm ~]$ |
```

Now, checking the client side again

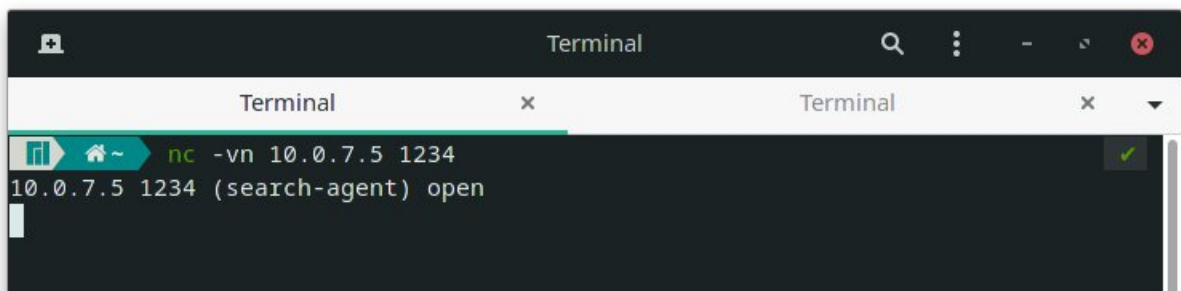


```
Terminal
sudo nc -l -p 555 > test.txt
cat test.txt
Hello World
```

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
```



```
Terminal
nc -vn 10.0.7.5 1234
10.0.7.5 1234 (search-agent) open
```

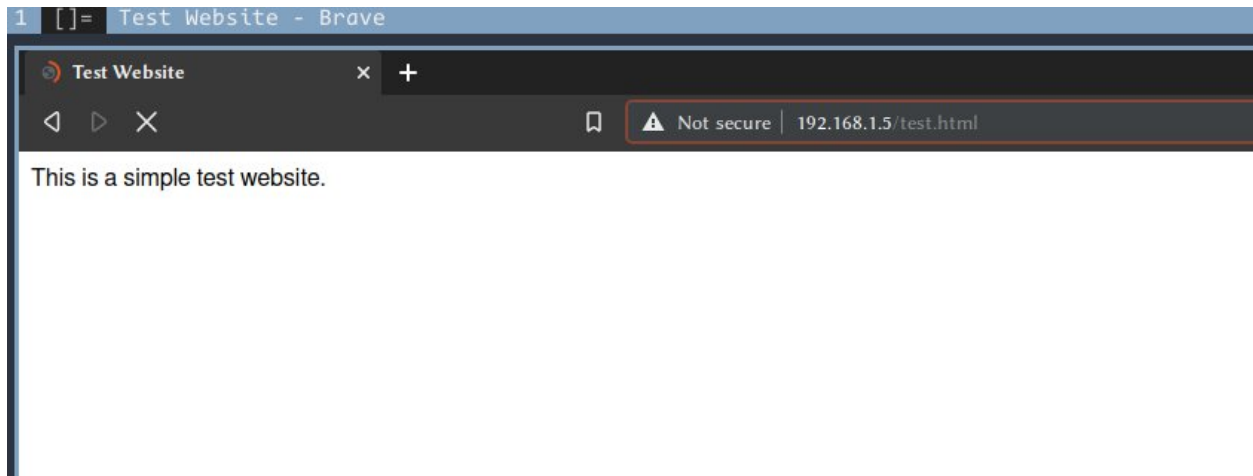
2. To run a web server with a static web page through nc, use

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



```
Terminal
cd Downloads
vim test.html
cat test.html
<html>
  <head>
    <title>Test Website</title>
  </head>
  <body>
    This is a simple test website.
  </body>
</html>
```

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



Questions on above observations:

- 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.

