

Week #1
Siri N Shetty
PES2UG22CS556

Study and understand the basic networking tools - Wireshark, Tcpdump, Ping, Traceroute.

Learn and Understand Network Tools

1. Wireshark

- ☐ Perform and analyze Ping PDU capture
- ☐ Examine HTTP packet capture
- ☐ Analyze HTTP packet capture using filter

2. Tcpdump

- Capture packets

3. Ping

- Test the connectivity between 2 systems

4. Traceroute

- Perform traceroute checks

5. Nmap

- Explore an entire network

IMPORTANT INSTRUCTIONS:

- This manual is written for Ubuntu Linux OS only. You can also execute these experiments on VirtualBox or VMWare platform.
- For few tasks, you may need to create 2 VMs for experimental setup.
- Perform **sudo apt-get update** before installing any tool or utility.
- Install any tool or utility using the command **sudo apt-get install name_of_the_tool**
Take screenshots wherever necessary and upload it as a single PDF file. (The PDF must contain: Lab Number and Title, SRN and Name of the student, Section)
- To define an IP address for your machine (e.g., Section – ‘a’ & Serial number is 1, then your IP address should be 10.0.1.1. Section – ‘h’ & Serial number is 23, then your IP address should be 10.0.8.23) – applicable only for relevant tasks (which doesn’t requires internet connectivity to execute the tasks).

Task 1: Linux Interface Configuration (ifconfig / IP command)

Step 1: To display status of all active network interfaces.

ifconfig (or) ip addr show

ip address table:

Interface name	IP address (IPv4 / IPv6)	MAC address
eth0	172.27.97.244	00:15:5d:79:5c:7f
lo	127.0.0.1	

```
(kali@DESKTOP-F9UMPJU)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.27.97.244 netmask 255.255.240.0 broadcast 172.27.111.255
    inet6 fe80::215:5dff:fe79:5c7f prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:79:5c:7f txqueuelen 1000 (Ethernet)
    RX packets 211172 bytes 84474390 (80.5 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 311099 bytes 10612675087 (9.8 GiB)
    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 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
```

Step 2: To assign an IP address to an interface, use the following command.

sudo ifconfig interface_name 10.0.your_section.your_sno netmask 255.255.255.0 (or)

sudo ip addr add 10.0.your_section.your_sno /24 dev interface_name

```
kali@DESKTOP-F9UMPJU: ~
File Actions Edit View Help
(kali@DESKTOP-F9UMPJU)-[~]
$ sudo ifconfig eth0 10.0.10.56 netmask 255.255.255.0
[sudo] password for kali:
(kali@DESKTOP-F9UMPJU)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.10.56 netmask 255.255.255.0 broadcast 10.0.10.255
    inet6 fe80::215:5dff:fe8b:eef1 prefixlen 64 scopeid 0x20<link>
    ether 00:15:5d:8b:ee:f1 txqueuelen 1000 (Ethernet)
    RX packets 208 bytes 110871 (108.2 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 208 bytes 24409 (23.8 KiB)
    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 2901 bytes 6696382 (6.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2901 bytes 6696382 (6.3 MiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Step 3: To activate / deactivate a network interface, type.

sudo ifconfig interface_name down

```
(kali@DESKTOP-F9UMPJU)-[~]
$ sudo ifconfig eth0 down

(kali@DESKTOP-F9UMPJU)-[~]
$ ifconfig
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 8135  bytes 18683516 (17.8 MiB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 8135  bytes 18683516 (17.8 MiB)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

sudo ifconfig interface_name up

```
(kali@DESKTOP-F9UMPJU)-[~]
$ sudo ifconfig eth0 up

(kali@DESKTOP-F9UMPJU)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
    inet 10.0.10.56  netmask 255.255.255.0  broadcast 10.0.10.255
    inet6 fe80::215:5dff:fe8b:eef1  prefixlen 64  scopeid 0x20<link>
    ether 00:15:5d:8b:ee:f1  txqueuelen 1000  (Ethernet)
RX packets 217  bytes 112713 (110.0 KiB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 218  bytes 25271 (24.6 KiB)
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 10537  bytes 24990106 (23.8 MiB)
RX errors 0  dropped 0  overruns 0  frame 0
TX packets 10537  bytes 24990106 (23.8 MiB)
TX errors 0  dropped 0  overruns 0  carrier 0  collisions 0
```

Step 4: To show the current neighbor table in kernel, type

ip neigh

```
(kali@DESKTOP-F9UMPJU)-[~]
$ ip neigh
172.27.96.1 dev eth0 lladdr 00:15:5d:97:ae:1a REACHABLE
```

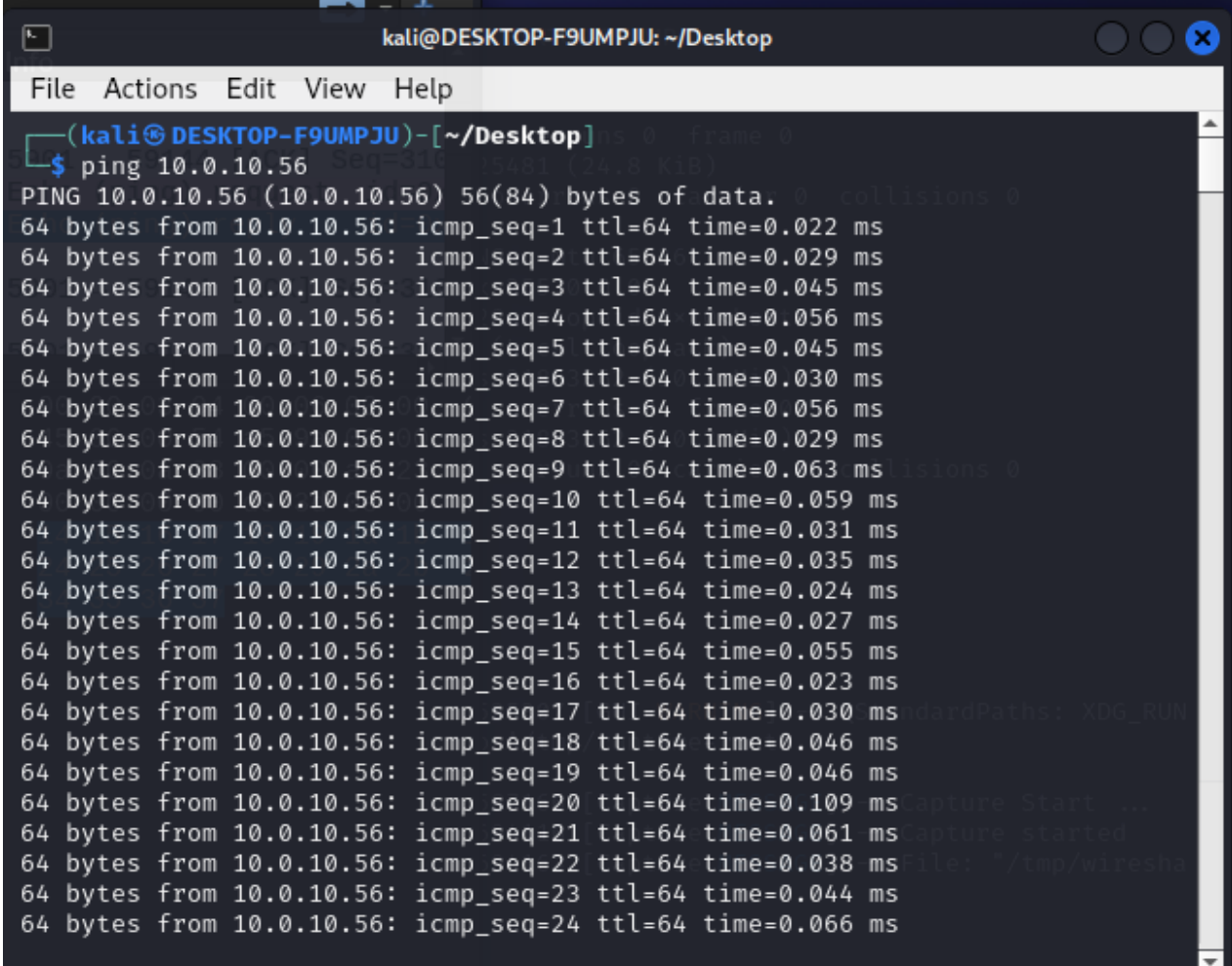
Task 2: Ping PDU (Packet Data Units or Packets) Capture

Step 1: Assign an IP address to the system (Host).

Note: IP address of your system should be 10.0.your_section.your_sno.

Step 2: Launch Wireshark and select 'any' interface

Step 3: In terminal, type **ping 10.0.your_section.your_sno**



```
kali@DESKTOP-F9UMPJU: ~/Desktop
File Actions Edit View Help
(kali@DESKTOP-F9UMPJU)-[~/Desktop]
$ ping 10.0.10.56
PING 10.0.10.56 (10.0.10.56) 56(84) bytes of data. 0 collisions: 0
64 bytes from 10.0.10.56: icmp_seq=1 ttl=64 time=0.022 ms
64 bytes from 10.0.10.56: icmp_seq=2 ttl=64 time=0.029 ms
64 bytes from 10.0.10.56: icmp_seq=3 ttl=64 time=0.045 ms
64 bytes from 10.0.10.56: icmp_seq=4 ttl=64 time=0.056 ms
64 bytes from 10.0.10.56: icmp_seq=5 ttl=64 time=0.045 ms
64 bytes from 10.0.10.56: icmp_seq=6 ttl=64 time=0.030 ms
64 bytes from 10.0.10.56: icmp_seq=7 ttl=64 time=0.056 ms
64 bytes from 10.0.10.56: icmp_seq=8 ttl=64 time=0.029 ms
64 bytes from 10.0.10.56: icmp_seq=9 ttl=64 time=0.063 ms
64 bytes from 10.0.10.56: icmp_seq=10 ttl=64 time=0.059 ms
64 bytes from 10.0.10.56: icmp_seq=11 ttl=64 time=0.031 ms
64 bytes from 10.0.10.56: icmp_seq=12 ttl=64 time=0.035 ms
64 bytes from 10.0.10.56: icmp_seq=13 ttl=64 time=0.024 ms
64 bytes from 10.0.10.56: icmp_seq=14 ttl=64 time=0.027 ms
64 bytes from 10.0.10.56: icmp_seq=15 ttl=64 time=0.055 ms
64 bytes from 10.0.10.56: icmp_seq=16 ttl=64 time=0.023 ms
64 bytes from 10.0.10.56: icmp_seq=17 ttl=64 time=0.030 ms
64 bytes from 10.0.10.56: icmp_seq=18 ttl=64 time=0.046 ms
64 bytes from 10.0.10.56: icmp_seq=19 ttl=64 time=0.046 ms
64 bytes from 10.0.10.56: icmp_seq=20 ttl=64 time=0.109 ms
64 bytes from 10.0.10.56: icmp_seq=21 ttl=64 time=0.061 ms
64 bytes from 10.0.10.56: icmp_seq=22 ttl=64 time=0.038 ms
64 bytes from 10.0.10.56: icmp_seq=23 ttl=64 time=0.044 ms
64 bytes from 10.0.10.56: icmp_seq=24 ttl=64 time=0.066 ms
```

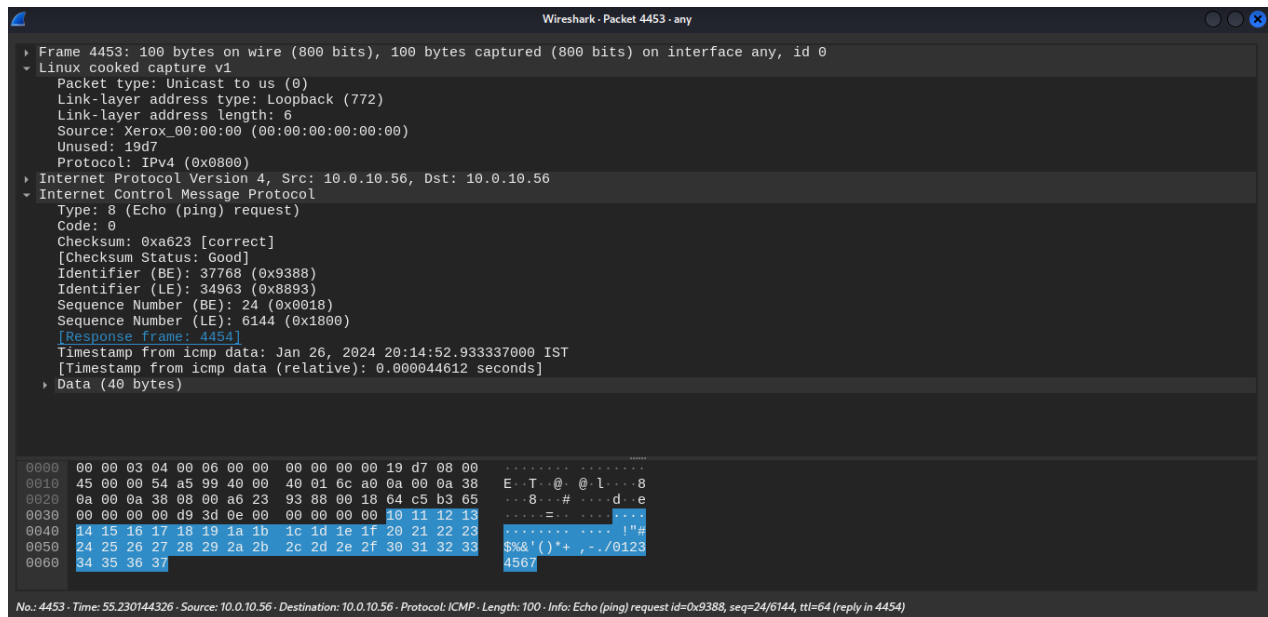
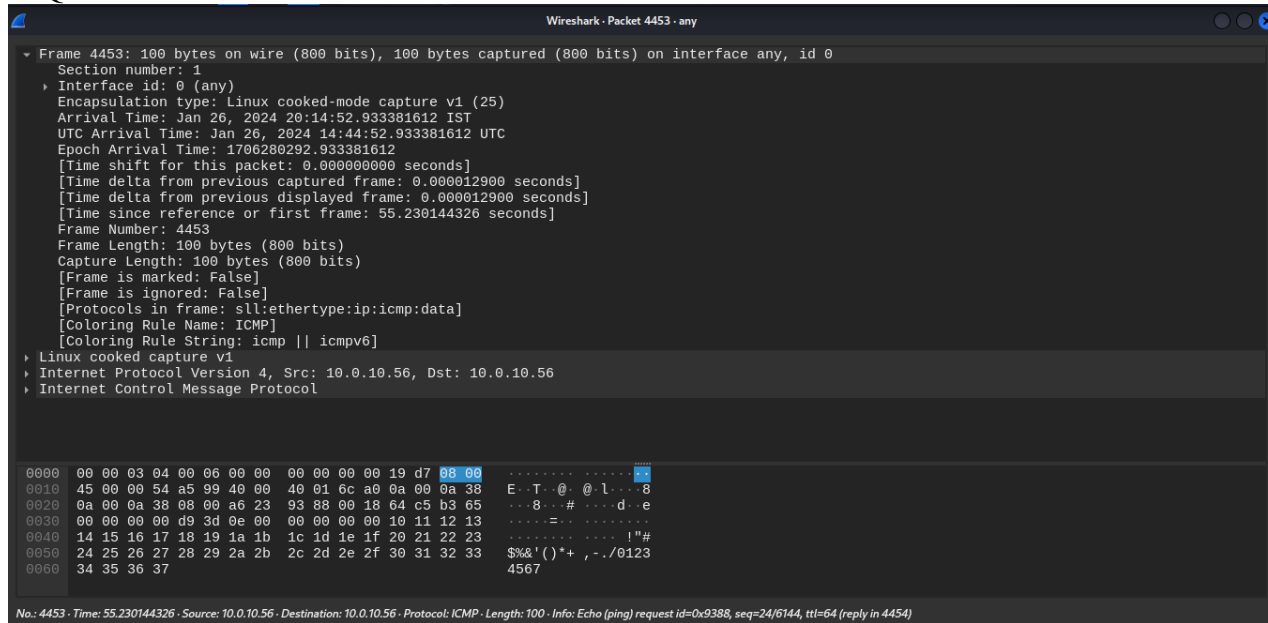
Observations to be made

Step 4: Analyze the following in Terminal

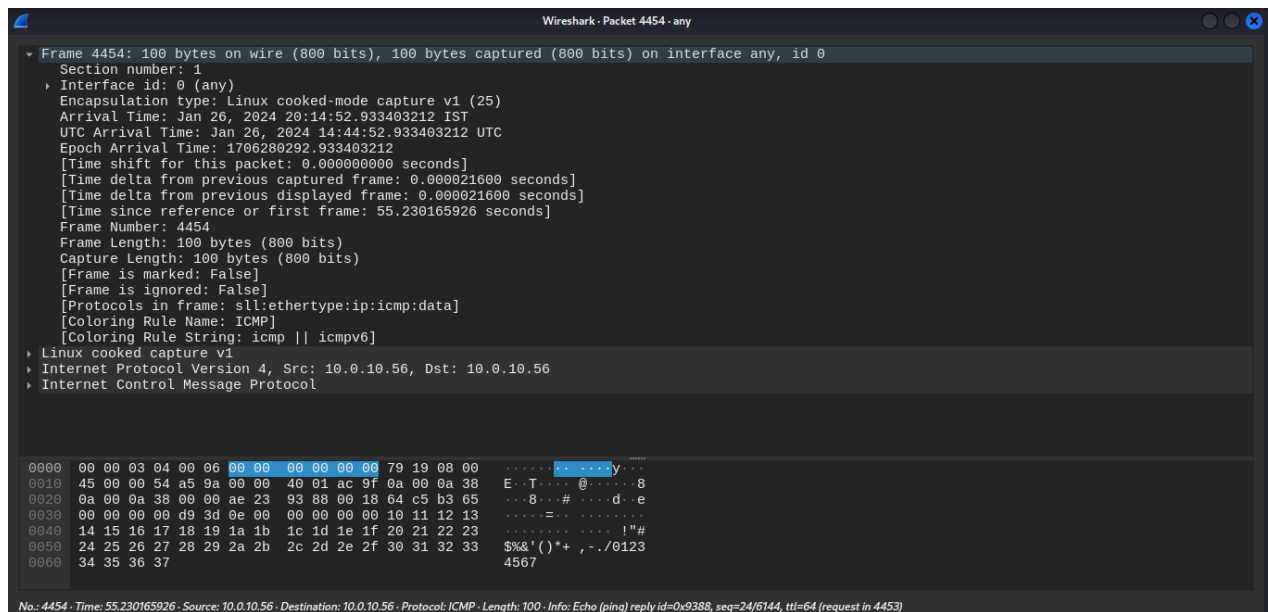
- TTL
- Protocol used by ping
- Time

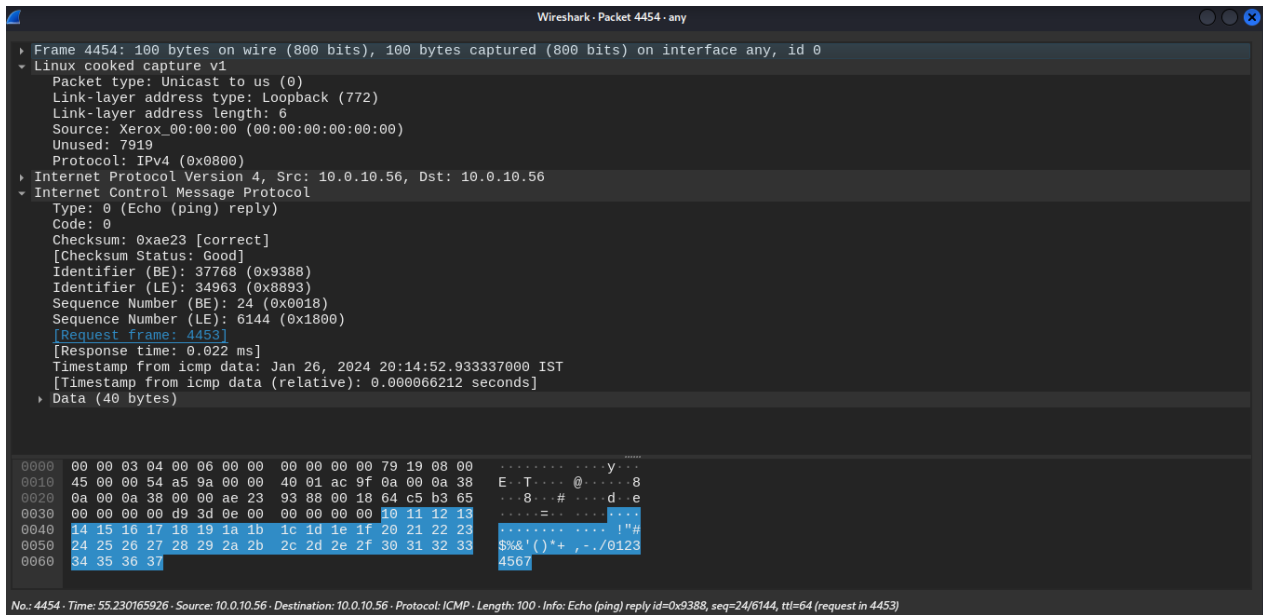
Step 5: Analyze the following in Wireshark

REQUEST



REPLY





On Packet List Pane, select the first echo packet on the list. On Packet Details Pane, click on each of the four “+” to expand the information. Analyze the frames with the first echo request and echo reply and complete the table below.

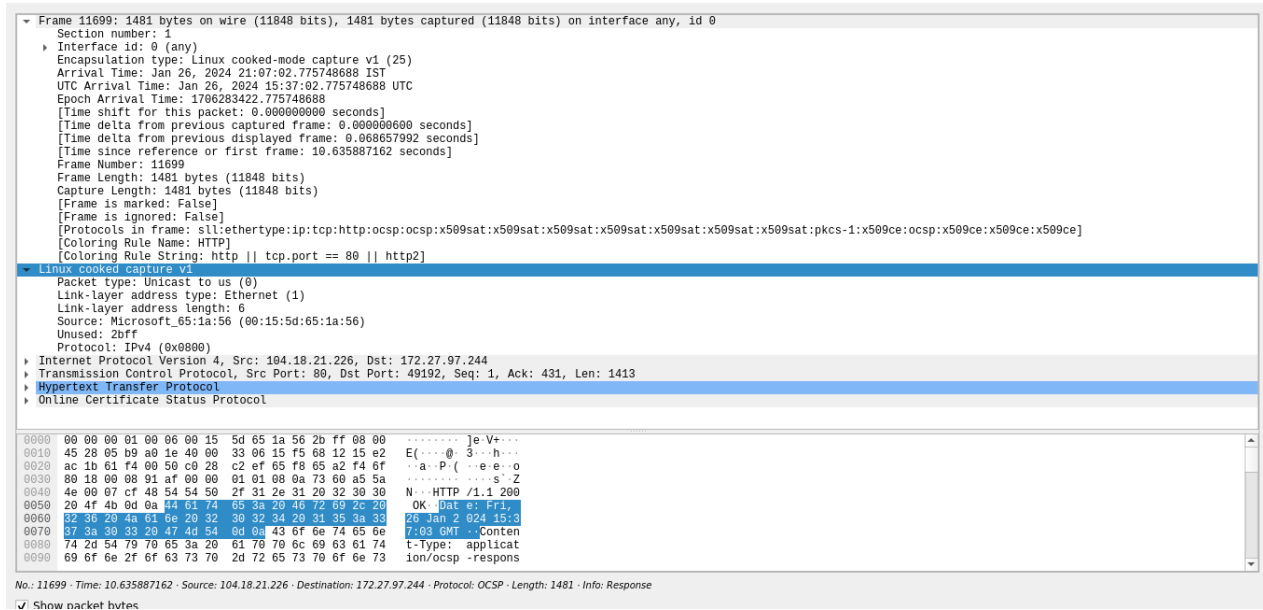
Details	First Echo Request	First Echo Reply
Frame Number	4453	4454
Source IP address	10.0.10.56	10.0.10.56
Destination IP address	10.0.10.56	10.0.10.56
ICMP Type Value	8 (Echo (ping(request)	0(Echo (ping) reply)
ICMP Code Value	0	0
Source Ethernet Address	00:00:00:00:00:00	00:00:00:00:00:00
Destination Ethernet Address	Unused	Unused
Internet Protocol Version	4	4
Time To Live (TTL) Value	64	64

Task 3: HTTP PDU Capture Using

Wireshark's Filter feature

Step 1: Launch Wireshark and select 'any' interface. On the Filter toolbar, type-in 'http' and press enter

Step 2: Open Firefox browser, and browse www.flipkart.com



Observations to be made

Step 3: Analyze the first (interaction of host to the web server) and second frame (response of server to the client). By analyzing the filtered frames, complete the table below:

Details	First Echo Request	First Echo Reply
Frame Number	11692	11699
Source Port	42296	80
Destination Port	80	42296
Source IP address	172.27.97.244	104.18.21.226
Destination IP address	104.18.21.226	172.27.97.244
Source Ethernet Address	00:15:5d:8b:e9:7c	00:15:5d:65:1a:56
Destination Ethernet Address	00:15:5d:65:1a:56	00:15:5d:8b:e9:7c

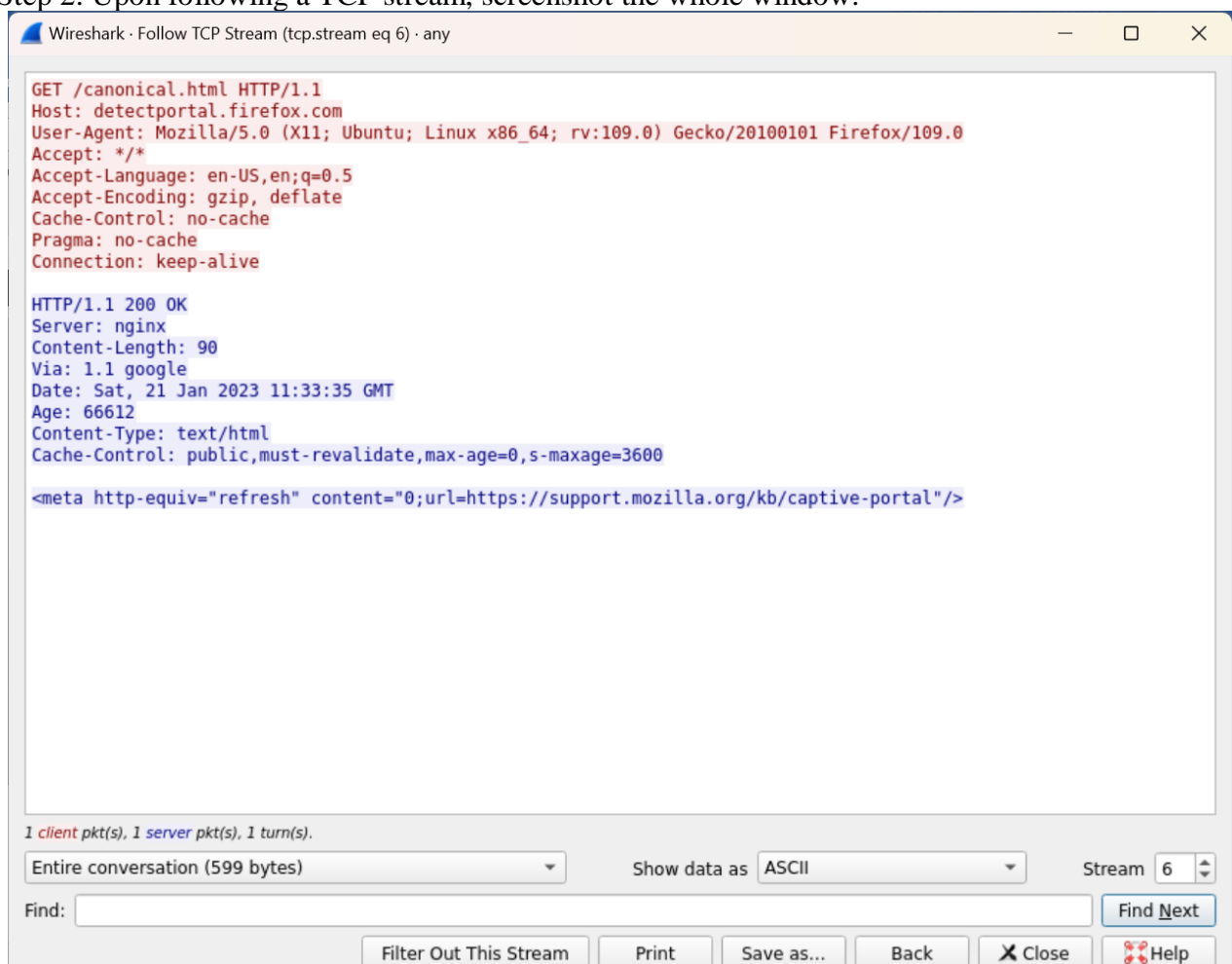
Step 4: Analyze the HTTP request and response and complete the table below.

HTTP Request		HTTP Response	
POST	POST /gseccovsslca2018	Server	nginx\r\n
host	ocsp.globalsign.com\r\n	Content-Type	text/html\r\n
User-Agent	Mozilla/5.0	Date	Fri, 26 Jan 2024 12:59:26 GMT r\n
Accept-Language	en-US,en;q=0.5\r\n	Location	-----
Accept-Encoding	gzip,deflate\r\n	Content-Length	938\r\n
Connection	keep-alive\r\n	Connection	keep-alive\r\n

Using Wireshark's Follow TCP Stream

Step 1: Make sure the filter is blank. Right-click any packet inside the Packet List Pane, then select 'Follow TCP Stream'. For demo purpose, a packet containing the HTTP GET request "GET / HTTP / 1.1" can be selected.

Step 2: Upon following a TCP stream, screenshot the whole window.



Task 4: Capturing packets with tcpdump

Step 1: Use the command **tcpdump -D** to see which interfaces are available for capture.

sudo tcpdump -D

```
(kali@DESKTOP-F9UMPJU)-[~]
$ sudo tcpdump -D
[sudo] password for kali:
1.eth0 [Up, Running, Connected]
2.any (Pseudo-device that captures on all interfaces) [Up, Running]
3.lo [Up, Running, Loopback]
4.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
5.nflog (Linux netfilter log (NFLOG) interface) [none]
6.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
7.dbus-system (D-Bus system bus) [none]
8.dbus-session (D-Bus session bus) [none]
```

Step 2: Capture all packets in any interface by running this command:

sudo tcpdump -i any

```
(kali@DESKTOP-F9UMPJU)-[~]
$ sudo tcpdump -i any
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 262
144 bytes
21:25:44.021172 eth0 Out IP 172.27.97.244.40968 > DESKTOP-F9UMPJU.x11-3: Fla
gs [P.], seq 993379648:993380616, ack 1402418344, win 501, length 968
21:25:44.037234 eth0 Out IP 172.27.97.244.58428 > DESKTOP-F9UMPJU.x11-3: Fla
gs [P.], seq 0:8, ack 1, win 3277, length 8
21:25:44.040703 eth0 Out IP 172.27.97.244.58428 > DESKTOP-F9UMPJU.x11-3: Fla
gs [P.], seq 1:33, ack 8, win 8212, length 32
21:25:44.041601 eth0 In IP DESKTOP-F9UMPJU.x11-3 > 172.27.97.244.58428: Fla
gs [P.], seq 8:33588, ack 33, win 3277, length 33580
21:25:44.042238 eth0 Out IP 172.27.97.244.58428 > DESKTOP-F9UMPJU.x11-3: Fla
gs [P.], seq 33588:67168, ack 33, win 3277, length 33580
21:25:44.042266 eth0 Out IP 172.27.97.244.58428 > DESKTOP-F9UMPJU.x11-3: Fla
gs [P.], seq 67168:94908, ack 33, win 3277, length 27740
21:25:44.042918 eth0 In IP DESKTOP-F9UMPJU.x11-3 > 172.27.97.244.58428: Fla
gs [P.], seq 94908:132868, ack 33, win 3277, length 37960
21:25:44.042949 eth0 Out IP 172.27.97.244.58428 > DESKTOP-F9UMPJU.x11-3: Fla
gs [P.], seq 132868:156228, ack 33, win 3277, length 23360
21:25:44.043516 eth0 Out IP 172.27.97.244.58428 > DESKTOP-F9UMPJU.x11-3: Fla
gs [P.], seq 156228:194188, ack 33, win 3277, length 37960
```

Note: Perform some pinging operation while giving above command. Also type www.google.com in browser.

Observation

Step 3: Understand the output format.

Step 4: To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using this command:

sudo tcpdump -i any -c5 icmp

```
(kali@DESKTOP-F9UMPJU)-[~]
$ 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
22:22:19.784952 eth0 Out IP 172.27.97.244 > DESKTOP-F9UMPJU: ICMP 172.27.97.244 udp port 41435 unreachable, length 126
22:22:21.868345 eth0 Out IP 172.27.97.244 > DESKTOP-F9UMPJU: ICMP 172.27.97.244 udp port 35786 unreachable, length 80
22:22:24.882697 eth0 Out IP 172.27.97.244 > DESKTOP-F9UMPJU: ICMP 172.27.97.244 udp port 35786 unreachable, length 80
22:22:24.882714 eth0 Out IP 172.27.97.244 > DESKTOP-F9UMPJU: ICMP 172.27.97.244 udp port 35786 unreachable, length 80
22:22:24.883671 eth0 Out IP 172.27.97.244 > DESKTOP-F9UMPJU: ICMP 172.27.97.244 udp port 35786 unreachable, length 80
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

Step 5: Check the packet content. For example, inspect the HTTP content of a web request like this:

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
11:38:44.694959 eth0 Out IP 172.17.158.182.37644 > 34.107.221.82.80: Flags [S], seq 3389668248, win 64240, options [mss 1460, sackOK,TS val 2726775024 ecr 0,nop,wscale 7], length 0
E...</y@.@....."k.R...P.
...J.....
...H.....
11:38:44.702701 eth0 In IP 34.107.221.82.80 > 172.17.158.182.37644: Flags [S.], seq 1425979628, ack 3389668249, win 65535, options [mss 1412,sackOK,TS val 725566228 ecr 2726775024,nop,wscale 8], length 0
E...<...@.x... "k.R...P..T....
...J.....
...H.....
11:38:44.702759 eth0 Out IP 172.17.158.182.37644 > 34.107.221.82.80: Flags [.], ack 1, win 502, options [nop,nop,TS val 2726775032 ecr 725566228], length 0
E...4/z@.@....."k.R...P.
...T.....J.....
...H.+?C.
11:38:44.702919 eth0 Out IP 172.17.158.182.37644 > 34.107.221.82.80: Flags [P.], seq 1:302, ack 1, win 502, options [nop,nop,TS val 2726775032 ecr 725566228], length 301: HTTP: GET /canonical.html HTTP/1.1
E..a/|@.@....."k.R...P.
...T.....K.....
...H.+?C.GET /canonical.html HTTP/1.1
Host: detectportal.firefox.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/109.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
```

```
[NEW] |*1*|*2*| 3 |*4*|*5*| 6 |
E`.^..f..x.gN"k.R....P..T....
<.....Lq.....
+?C...H.HTTP/1.1 200 OK
Server: nginx
Content-Length: 90
Via: 1.1 google
Date: Sat, 21 Jan 2023 20:14:31 GMT
Age: 35653
Content-Type: text/html
Cache-Control: public,must-revalidate,max-age=0,s-maxage=3600

<meta http-equiv="refresh" content="0;url=https://support.mozilla.org/kb/captive-portal"/>
11:38:44.712857 eth0 Out IP 172.17.158.182.37644 > 34.107.221.82.80: Flags [.], ack 299, win 501, options [nop,nop,TS val 2726775042 ecr 725566237], length 0
E...4/|@.@....."k.R...P.
<...T.....J.....
...I.+?C.
11:38:44.820890 eth0 Out IP 172.17.158.182.37646 > 34.107.221.82.80: Flags [S], seq 591477368, win 64240, options [mss 1460,sackOK,TS val 2726775150 ecr 0,nop,wscale 7], length 0
E...<L@.@...m... "k.R...P#A:x.....J.....
...In.....
11:38:44.829968 eth0 In IP 34.107.221.82.80 > 172.17.158.182.37646: Flags [S.], seq 23370223, ack 591477369, win 65535, options [mss 1412,sackOK,TS val 2181072818 ecr 2726775150,nop,wscale 8], length 0
E...<...@.v..6"k.R....P...d..#A:y...f.....
...In....
11:38:44.830053 eth0 Out IP 172.17.158.182.37646 > 34.107.221.82.80: Flags [.], ack 1, win 502, options [nop,nop,TS val 2726775159 ecr 2181072818], length 0
E...4L@.@...t... "k.R...P#A:y.d.....J.....
...Iw.....
10 packets captured
14 packets received by filter
0 packets dropped by kernel
```

Step 6: To save packets to a file instead of displaying them on screen, use the option -w:

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
21 packets received by filter
0 packets dropped by kernel
```

Task 5: Perform Traceroute checks

Step 1: Run the traceroute using the following command.

sudo traceroute www.google.com

```
traceroute to www.google.com (142.250.195.132), 30 hops max, 60 byte packets
 1  LAPTOP-FVC6J040.mshome.net (172.17.144.1)  0.370 ms  0.288 ms  0.282 ms
 2  192.168.0.1 (192.168.0.1)  2.391 ms  3.064 ms  3.061 ms
 3  * * *
 4  49.205.72.51.actcorp.in (49.205.72.51)  52.622 ms  52.613 ms  52.607 ms
 5  10.248.5.10 (10.248.5.10)  12.558 ms !X  12.482 ms !X  12.474 ms !X
```

Step 2: Analyze destination address of google.com and no. of hops

Step 3: To speed up the process, you can disable the mapping of IP addresses with hostnames by using the **-n** option

sudo traceroute -n www.google.com

```
traceroute to www.google.com (142.250.205.228), 30 hops max, 60 byte packets
 1  172.17.144.1  0.229 ms  0.217 ms  0.186 ms
 2  192.168.0.1  2.222 ms  2.218 ms  1.421 ms
 3  * * *
 4  49.205.72.51  6.374 ms  6.319 ms  6.310 ms
 5  10.248.5.10  14.773 ms !X  14.768 ms !X  14.714 ms !X
```

Step 4: The **-I** option is necessary so that the traceroute uses ICMP.

sudo traceroute -I www.google.com

```
traceroute to www.google.com (142.250.195.132), 30 hops max, 60 byte packets
 1  LAPTOP-FVC6J040.mshome.net (172.17.144.1)  0.189 ms  0.176 ms  0.174 ms
 2  192.168.0.1 (192.168.0.1)  1.558 ms  1.556 ms  1.555 ms
 3  10.185.0.1 (10.185.0.1)  3.462 ms  3.461 ms  3.460 ms
 4  49.205.72.51.actcorp.in (49.205.72.51)  4.497 ms  4.495 ms  4.625 ms
 5  * * *
 6  10.248.5.23 (10.248.5.23)  3.840 ms  2.692 ms  3.422 ms
 7  49.205.72.39.actcorp.in (49.205.72.39)  7.027 ms  7.682 ms  7.643 ms
 8  72.14.243.242 (72.14.243.242)  10.415 ms  8.238 ms  8.212 ms
 9  216.239.51.91 (216.239.51.91)  8.672 ms  8.876 ms  8.875 ms
10  142.251.55.61 (142.251.55.61)  8.143 ms  8.143 ms  8.142 ms
11  maa03s40-in-f4.1e100.net (142.250.195.132)  8.424 ms  8.423 ms  8.503 ms
```

Step 5: By default, traceroute uses icmp (ping) packets. If you'd rather test a TCP connection to gather data more relevant to web server, you can use the **-T** flag.

sudo traceroute -T www.google.com

```
traceroute to www.google.com (142.250.195.132), 30 hops max, 60 byte packets
 1  LAPTOP-FVC6J040.mshome.net (172.17.144.1)  0.276 ms  0.176 ms  0.168 ms
 2  192.168.0.1 (192.168.0.1)  1.469 ms  1.770 ms  1.738 ms
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  49.205.72.39.actcorp.in (49.205.72.39)  7.710 ms  7.030 ms  6.925 ms
 8  72.14.243.242 (72.14.243.242)  8.271 ms  7.986 ms  7.936 ms
 9  108.170.227.7 (108.170.227.7)  9.102 ms  10.904 ms  216.239.51.91 (216.239.51.91)  8.623 ms
10  142.251.55.61 (142.251.55.61)  8.060 ms  142.251.55.63 (142.251.55.63)  16.684 ms  142.251.55.61 (142.251.55.61)  8.161 ms
11  maa03s40-in-f4.1e100.net (142.250.195.132)  16.573 ms  8.487 ms  8.504 ms
```

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

Step 1: You can scan a host using its host name or IP address, for instance.

nmap www.pes.edu

```
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-21 23:50 IST
Nmap scan report for www.pes.edu (52.172.204.196)
Host is up (0.027s 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 6.27 seconds
```

Step 2: Alternatively, use an IP address to scan.

nmap 163.53.78.128

```
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-21 23:53 IST
Nmap scan report for 163.53.78.128
Host is up (0.0095s 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 5.57 seconds
jawahar@LAPTOP-FVC6J04Q ~/learn/computer_network
```

Step 3: Scan multiple IP address or subnet (IPv4)

nmap 192.168.1.1 192.168.1.2 192.168.1.3

```
Starting Nmap 7.80 ( https://nmap.org ) at 2023-01-21 23:55 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.03 seconds
```

Submission:

Students are expected to take the screenshot of results - after execution of every command in every task.

They are expected to write the Task and 2-3 lines of their observation followed by screenshots. Submissions will be through google forms.

Questions on above observations: (Optional)

- 1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server? Answer: 1.1
- 2) When was the HTML file that you are retrieving last modified at the server?
12:59:26
- 3) How to tell ping to exit after a specified number of ECHO_REQUEST packets?
ping 127.0.0.1 -c 5

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data:
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.121 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.064 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.044 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.110 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.123 ms

--- 127.0.0.1 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 4097ms
rtt min/avg/max/mdev = 0.044/0.092/0.123/0.032 ms
```

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

Based on SSH version