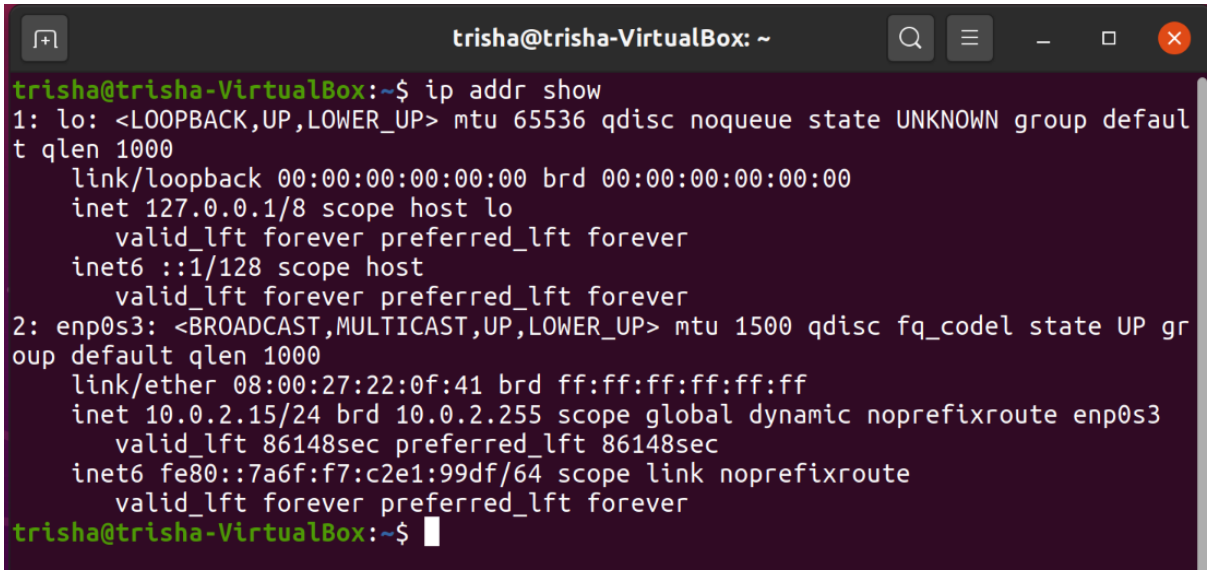
# Computer Networks Lab - Week 1
# SRN : PES1UG19CS542
# Name : Trisha Jain

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

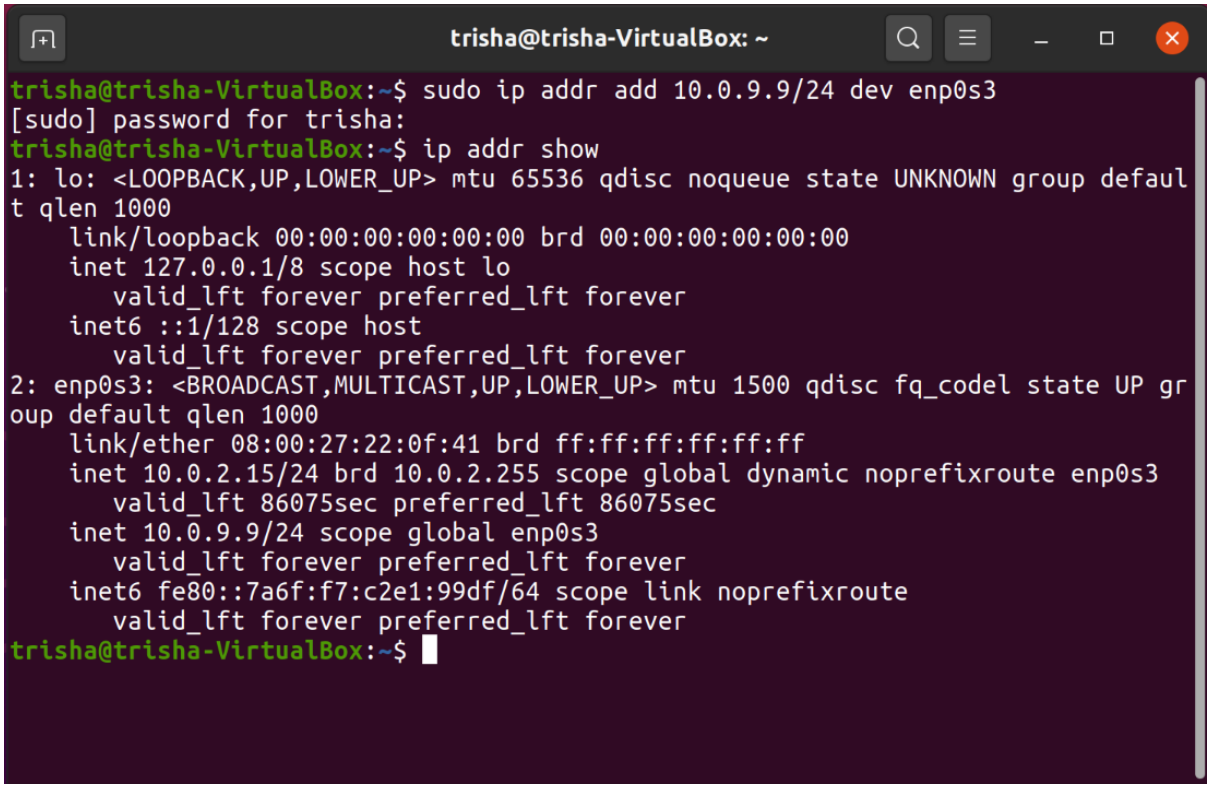
Step 1: To display status of all active network interfaces.

ip addr show




| **Interface Name** | **IP Address(IPv4 / IPv6)** | **MAC Address** |
|---|---|---|
| lo | 127.0.0.1 / ::1 | 00:00:00:00:00:00 |
| enp0s3 | 10.0.2.15 / fe80::7a6f:f7:c2e1:99df | 08:00:27:22:0f:41 |

Step 2 : To assign an IP address to an interface, used the following command:
sudo ip addr add 10.0.9.9 /24 dev enp0s3



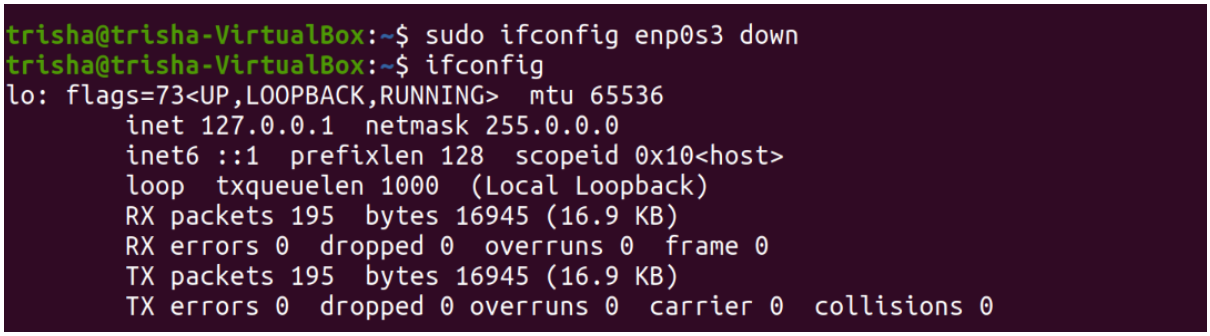

inet 10.0.9.9/24 scope global enp0s3

Step 3 : To activate and deactivate a network interface :

*To Deactivate* :

Command used is:-

sudo ifconfig enp0s3 down




Only lo is displayed above

*To Activate* :

Command used is:-

sudo ifconfig enp0s3 up

```
trisha@trisha-VirtualBox:~$ sudo ifconfig enp0s3 up
trisha@trisha-VirtualBox:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.0.2.15  netmask 255.255.255.0  broadcast 10.0.2.255
        inet6 fe80::7a6f:f7:c2e1:99df  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:22:0f:41  txqueuelen 1000  (Ethernet)
        RX packets 4329  bytes 5679155 (5.6 MB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 2463  bytes 169851 (169.8 KB)
        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 217  bytes 18635 (18.6 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 217  bytes 18635 (18.6 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

trisha@trisha-VirtualBox:~$
```

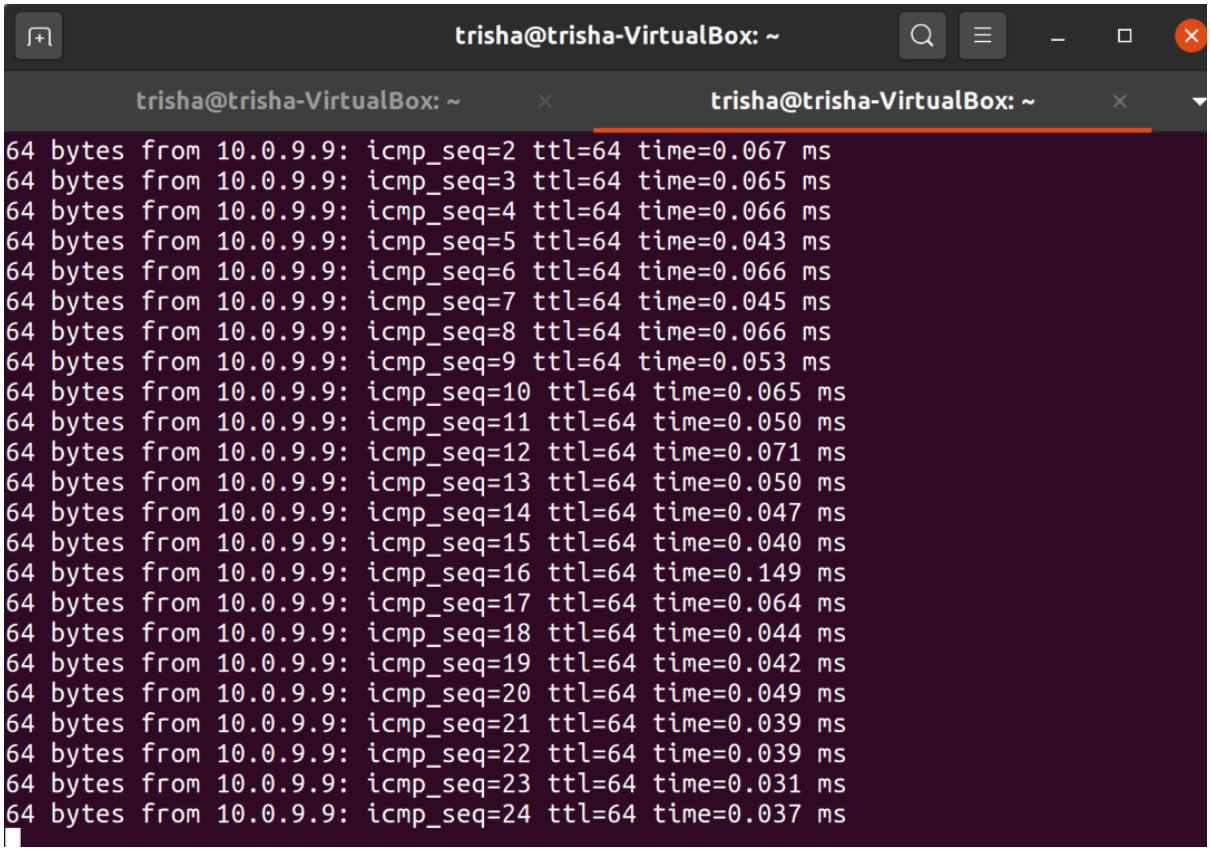
Now enp0s3 is also shown since its reactivated

Step 4 : To show the current neighbour table in kernel, ip neigh is used.

```
trisha@trisha-VirtualBox:~$ ip neigh
10.0.2.2 dev enp0s3 lladdr 52:54:00:12:35:02 REACHABLE
```

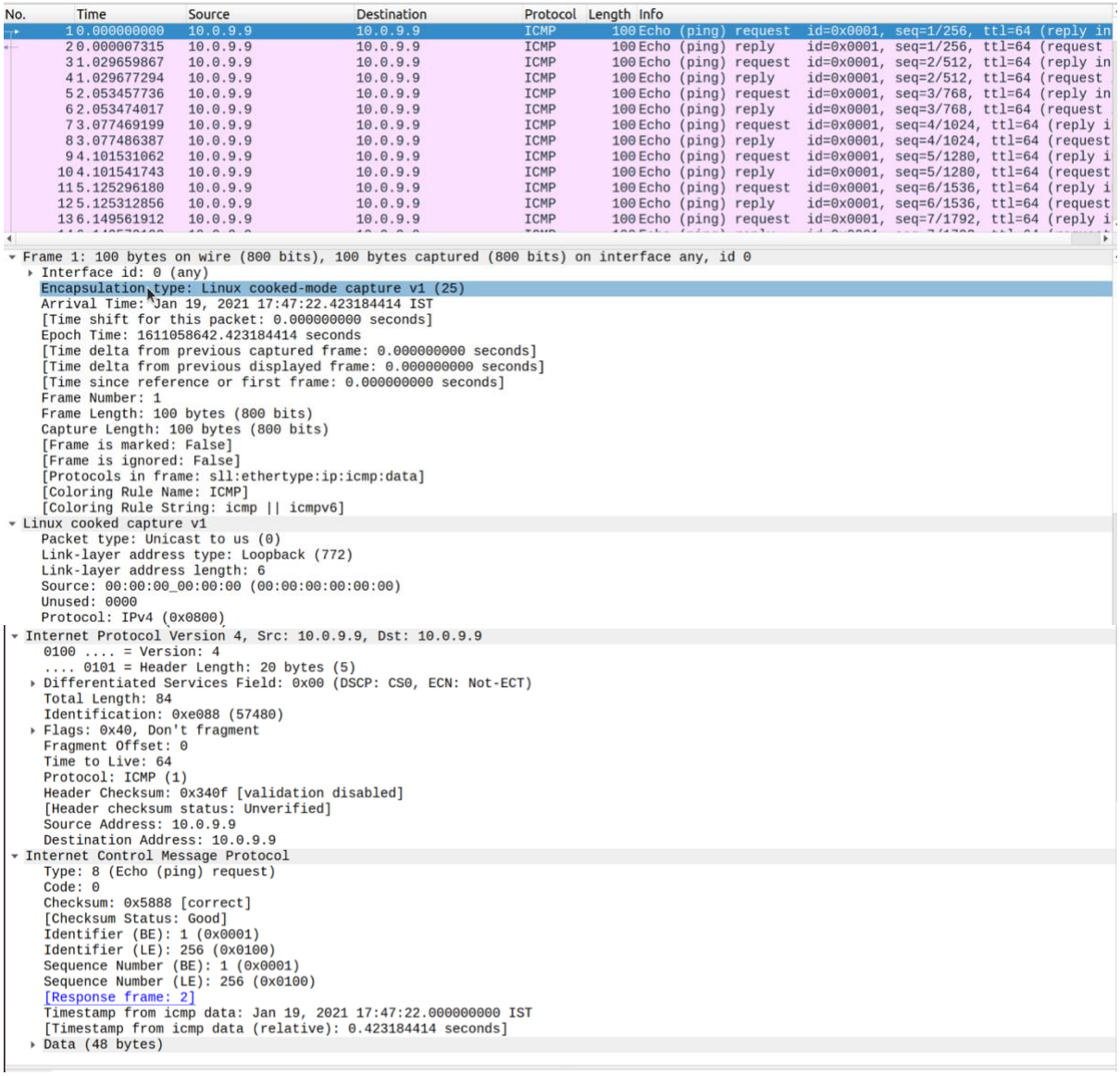
ip neigh

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

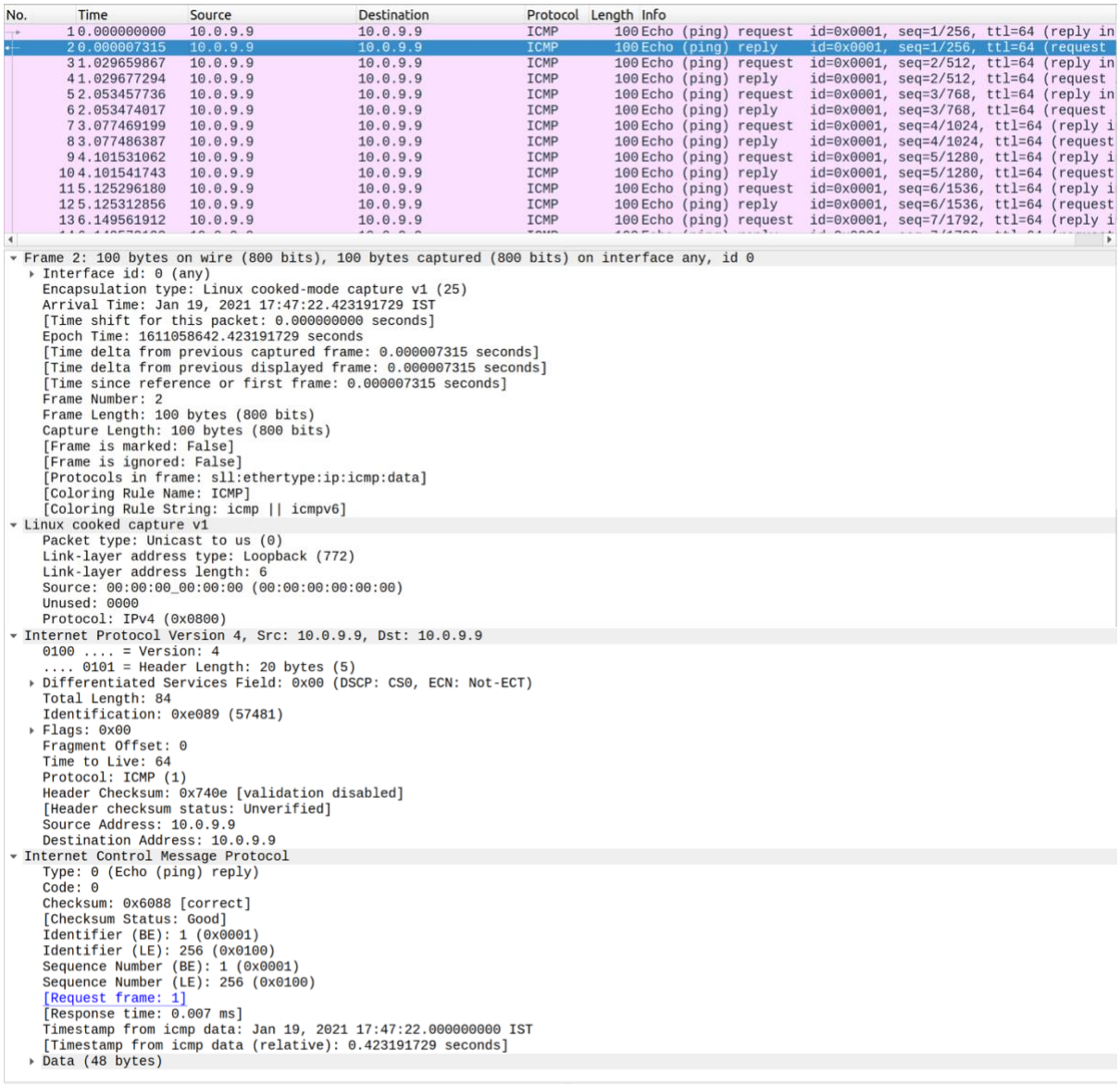



ping 10.0.9.9

| | |
|---|---|
| **TTL** | 64 |
| **Protocol used by ping** | ICMP |
| **Time** | Order of $10^{-2}$ ms |



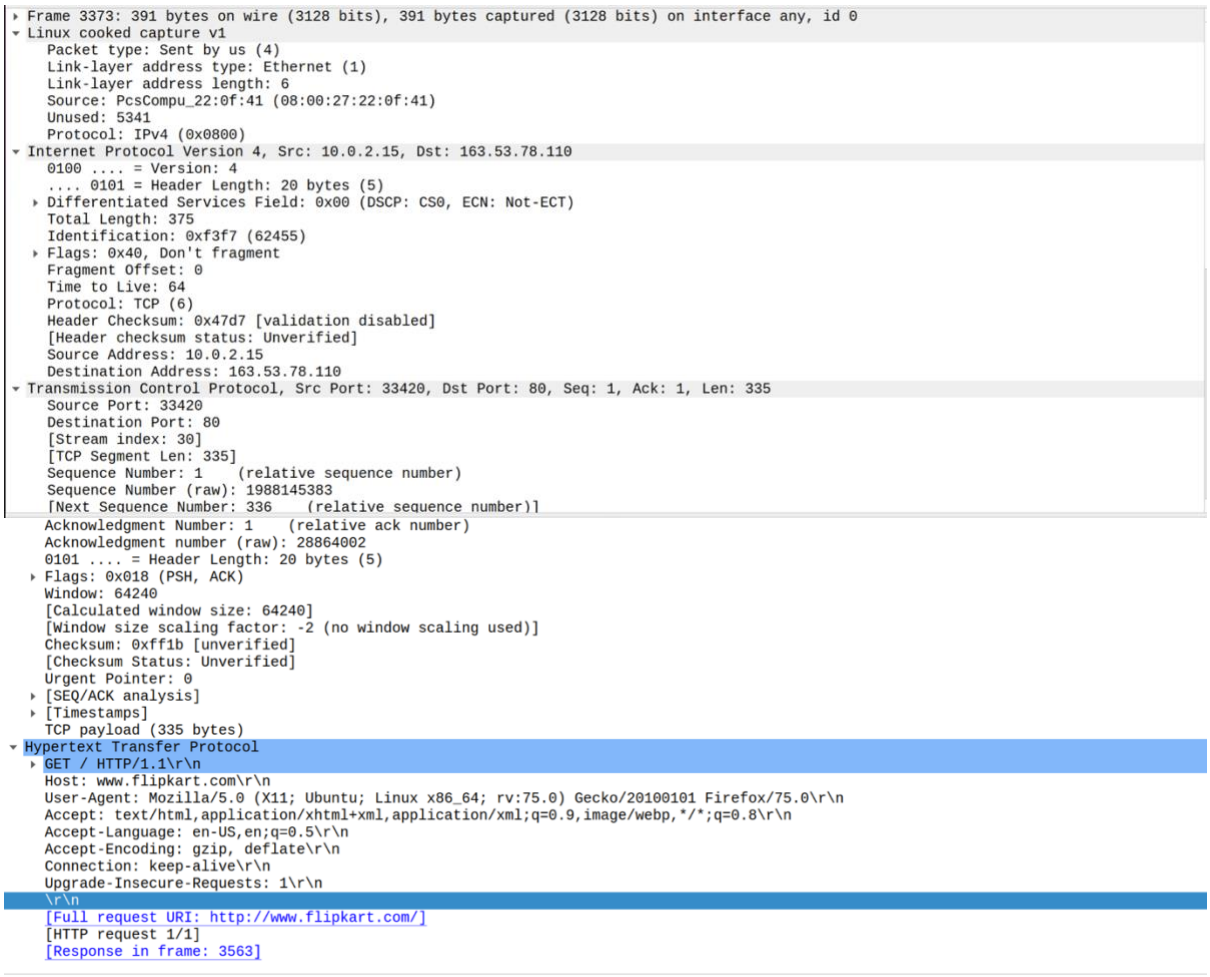

Request Packet



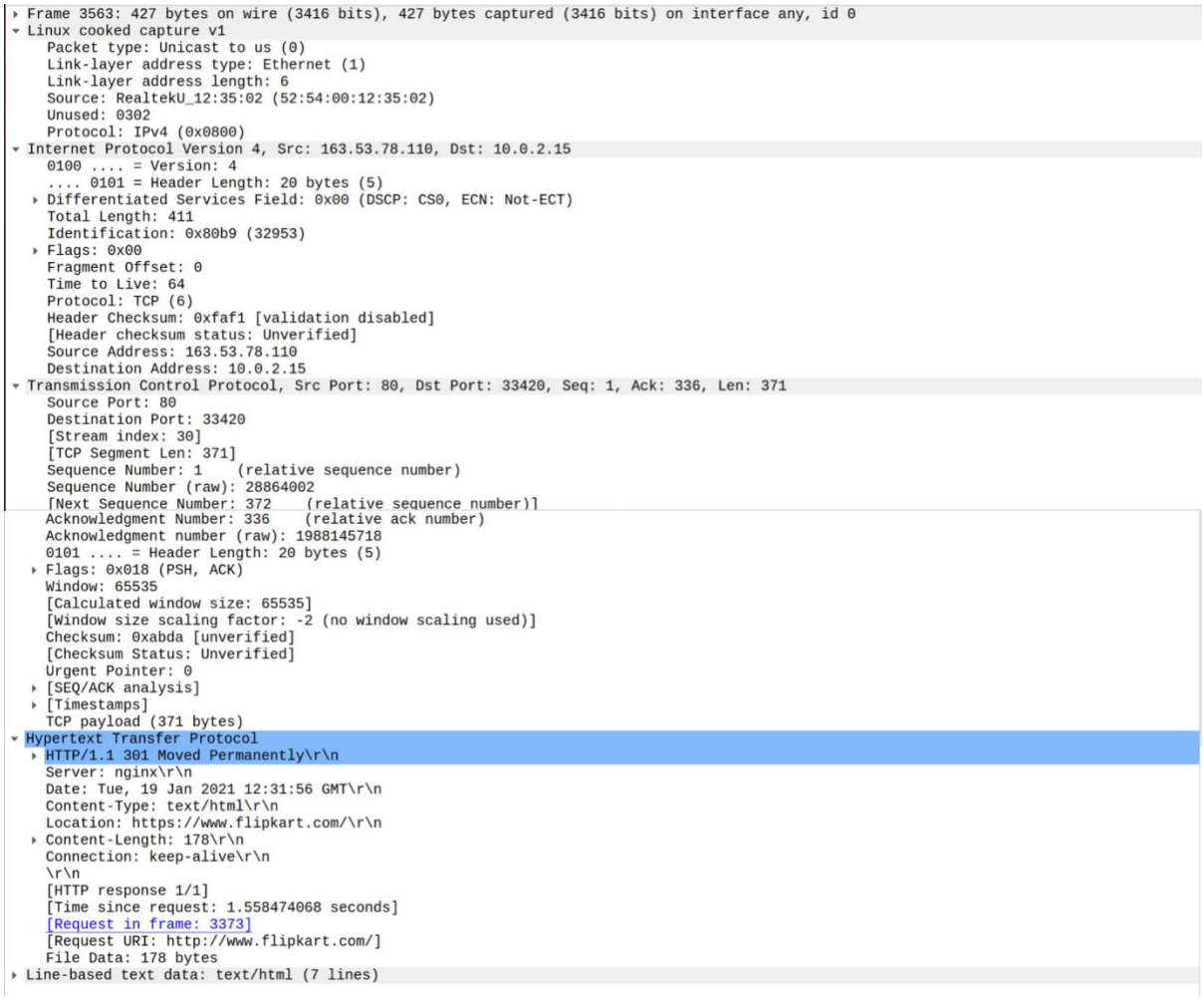

Response Packet

| Details | First Echo Request | First Echo Reply |
|---|---|---|
| **Frame Number** | 1 | 2 |
| **Source IP Address** | 10.0.9.9 | 10.0.9.9 |
| **Destination IP Address** | 10.0.9.9 | 10.0.9.9 |
| **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 |
| **TTL Value** | 64 | 64 |

# Task 3: HTTP PDU (Packet Data Units or Packets) Capture




Request Packet



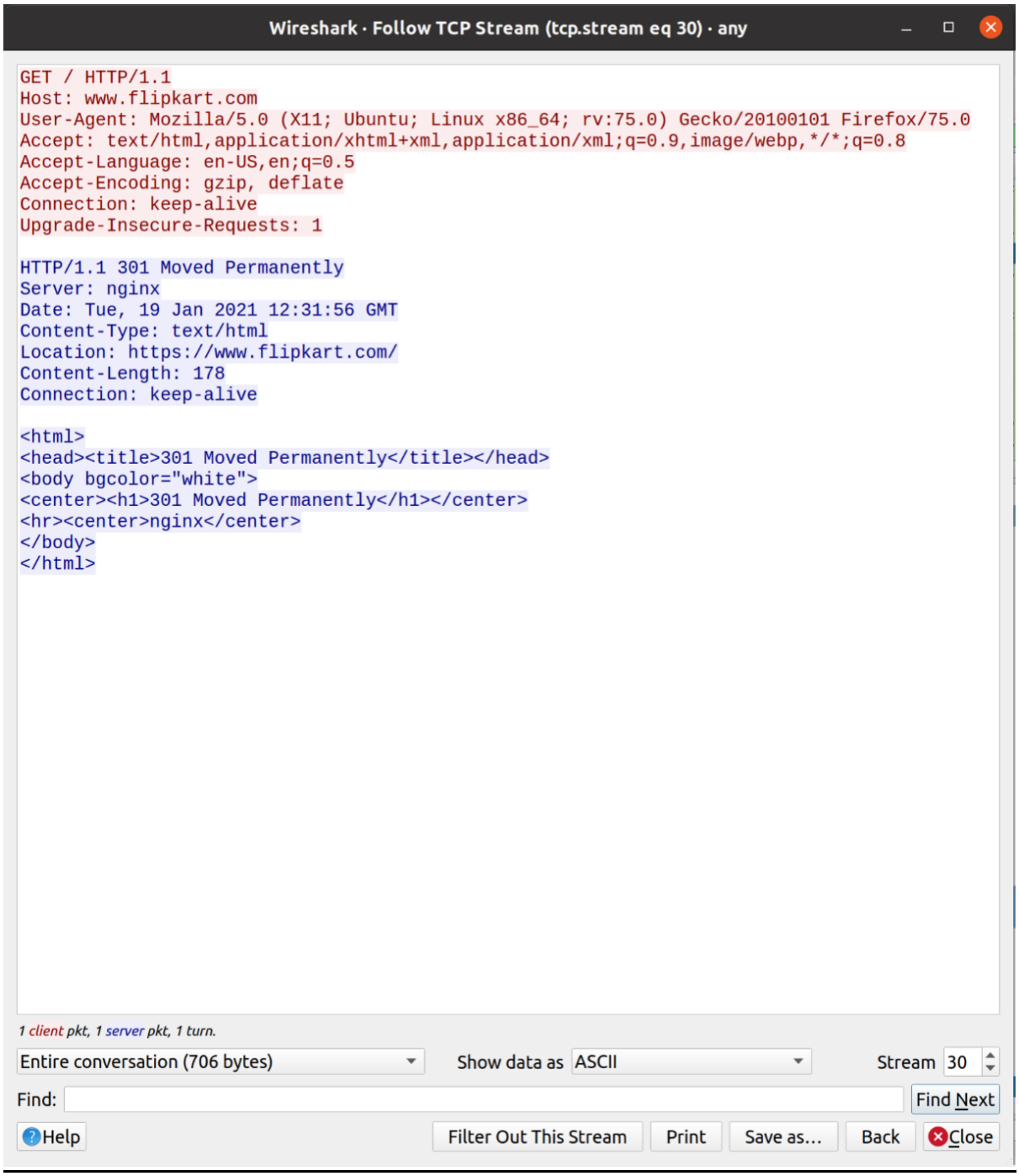

Response Packet

| Details | First Echo Request | First Echo Reply |
|---|---|---|
| **Frame Number** | 3373 | 3563 |
| **Source Port** | 33420 | 80 |
| **Destination Port** | 80 | 33420 |
| **Source IP Address** | 10.0.2.15 | 163.53.78.110 |
| **Destination IP Address** | 163.53.78.110 | 10.0.2.15 |
| **Source Ethernet Address** | 08:00:27:22:0f:41 | 52:54:00:12:35:02 |
| **Destination Ethernet Address** | 52:54:00:12:35:02 | 08:00:27:22:0f:41 |

## HTTP Request and Response

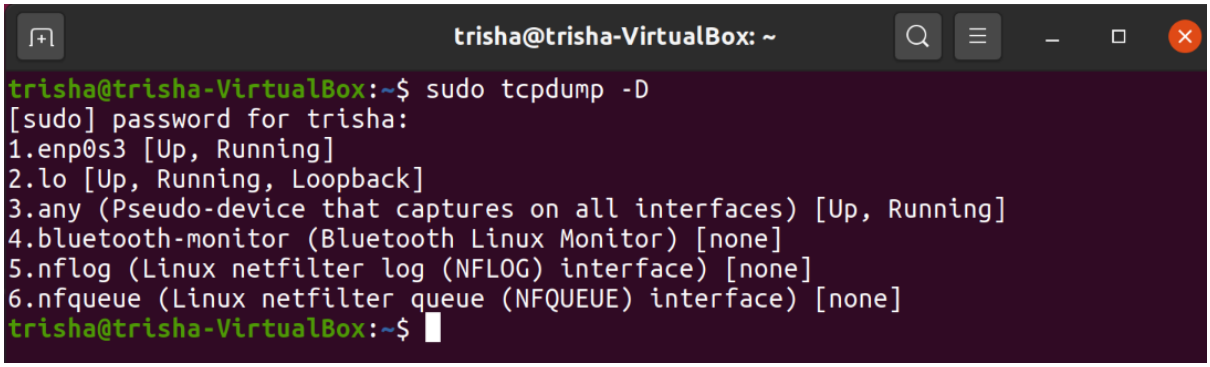
| HTTP Request | | HTTP Response | |
|---|---|---|---|
| **Get** | GET/HTTP/1.1\r\n | **Server** | nginx |
| **Host** | www.flipkart.com | **Content-Type** | text/html |
| **User-Agent** | Mozilla/5.0 (X11; Ubuntu; Linux x86__64; rv:75.0) Gecko/20100101 Firefox/80.0 | **Date** | Tue, 19 Jan 2021 12:31:56 GMT |
| **Accept-Language** | en-US,en;q=0.5 | **Location** | https://www.flipkart.com/ |
| **Accept-Encoding** | gzip, deflate | **Content-Length** | 178 |
| **Connection** | keep-alive | **Connection** | keep-alive |

# Using Wireshark to follow TCP Stream



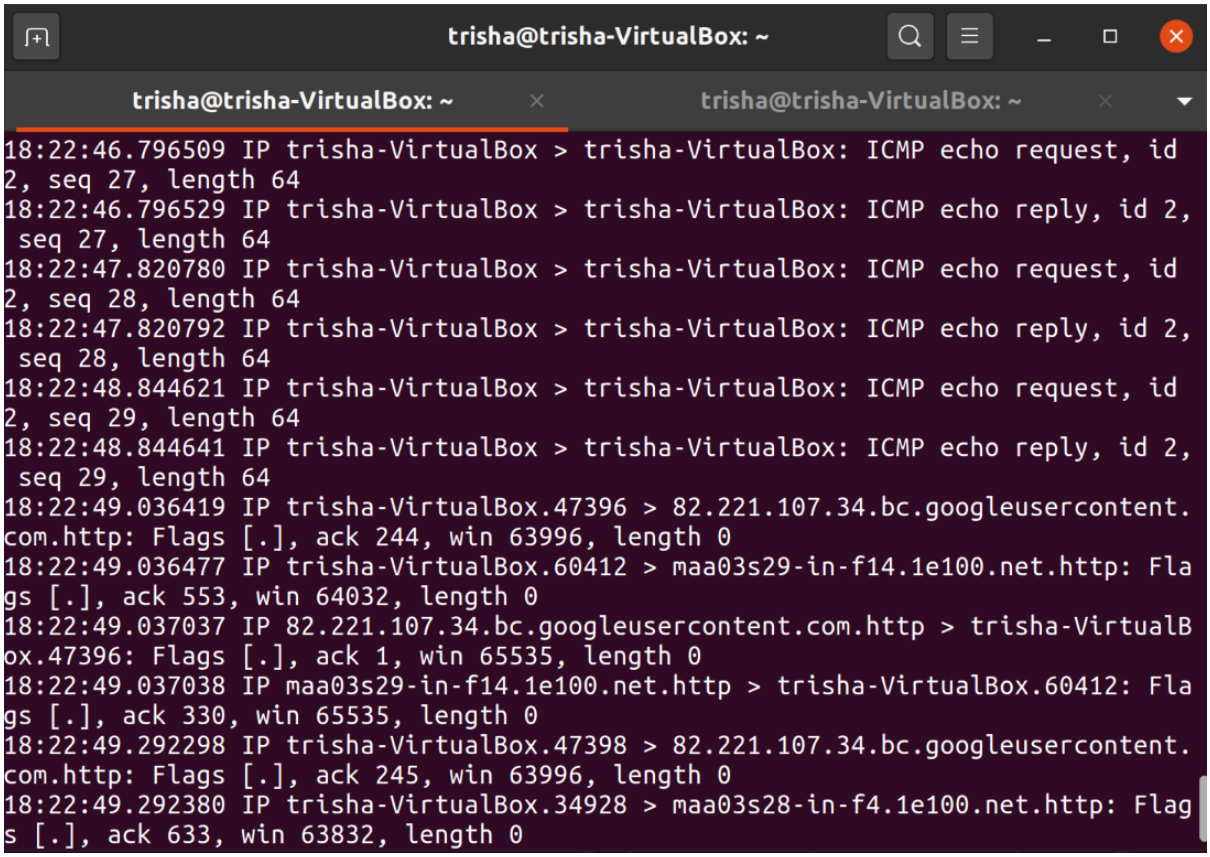
## Task 4: Capturing Packets with tcpdump

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




sudo tcpdump -D

Step 2: Using the command sudo tcpdump -i any to capture all packets in any interface.




sudo tcpdump -i any

Step 3: Understanding the output format of tcpdump :-

*Using the following line as an example :-*
18:22:49.292298 IP trisha-VirtualBox.47398 > 82.221.107.34.bc.googleusercontent.com.http: Flags [.], ack 245, win 63996, length 0

The first field, 18:22:49.292298 represents the timestamp of the received packet as per the local clock.

The next field, IP represents the network layer protocol, in this case IPv4.(For IPv6 it is specified as IP6)

The next field, trisha-VirtualBox.47398, is the source IP address and port
and 82.221.107.34.bc.googleusercontent.com.http is the destination ip address and port.

After the destination, the TCP flags are specified.

The next field is the Ack Number which in this case is 245 and it specifies the next expected byte (data) on this flow. (Ack number 1 is for the side sending the data)

The next field is the window size which represents the number of bytes available in the receiving buffer which in this case is 63996.

And lastly we have the packet length which in this example is 0.

Step 4 : Capturing Packets based on Protocol

Capturing ICMP packets using the command :
sudo tcpdump -i any -c5 icmp

```
trisha@trisha-VirtualBox:~$ sudo tcpdump -i any -c5 icmp
tcpdump: verbose output suppressed, use -v or -vv for full protocol dec
ode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 2
62144 bytes
18:30:05.678551 IP trisha-VirtualBox > trisha-VirtualBox: ICMP echo req
uest, id 2, seq 395, length 64
18:30:05.678563 IP trisha-VirtualBox > trisha-VirtualBox: ICMP echo rep
ly, id 2, seq 395, length 64
18:30:06.702998 IP trisha-VirtualBox > trisha-VirtualBox: ICMP echo req
uest, id 2, seq 396, length 64
18:30:06.703016 IP trisha-VirtualBox > trisha-VirtualBox: ICMP echo rep
ly, id 2, seq 396, length 64
18:30:07.727085 IP trisha-VirtualBox > trisha-VirtualBox: ICMP echo req
uest, id 2, seq 397, length 64
5 packets captured
12 packets received by filter
0 packets dropped by kernel
trisha@trisha-VirtualBox:~$
```

sudo tcpdump -i any -c5 icmp

## Step 5 : Checking the packet content

Inspect the HTTP content of a web request using the command :
sudo tcpdump -i any -c10 -nn -A port 80

```
trisha@trisha-VirtualBox:~$ sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on any, link-type LINUX_SLL (Linux cooked v1), capture size 262144
bytes
18:33:52.133064 IP 10.0.2.15.47464 > 34.107.221.82.80: Flags [S], seq 3812885
594, win 64240, options [mss 1460,sackOK,TS val 3036175840 ecr 0,nop,wscale 7
], length 0
E..</@.@.A.
..."k.R.h.P.D.Z..................
..].........
18:33:52.171933 IP 34.107.221.82.80 > 10.0.2.15.47464: Flags [S.], seq 142336
001, ack 3812885595, win 65535, options [mss 1460], length 0
E..,....@..8"k.R
....P.h.{...D.[`....u........
18:33:52.171977 IP 10.0.2.15.47464 > 34.107.221.82.80: Flags [.], ack 1, win
64240, length 0
E..(.0@.@.A.
..."k.R.h.P.D.[.{..P.......
18:33:52.172428 IP 10.0.2.15.47464 > 34.107.221.82.80: Flags [P.], seq 1:297,
 ack 1, win 64240, length 296: HTTP: GET /success.txt HTTP/1.1
E..P.1@.@.@.
..."k.R.h.P.D.[.{..P.......GET /success.txt HTTP/1.1
```

```
Host: detectportal.firefox.com
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 F
irefox/75.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cache-Control: no-cache
Pragma: no-cache
Connection: keep-alive


18:33:52.172653 IP 34.107.221.82.80 > 10.0.2.15.47464: Flags [.], ack 297, wi
n 65535, length 0
E..(....@..;"k.R
....P.h.{...D..P....
........
18:33:52.238320 IP 34.107.221.82.80 > 10.0.2.15.47464: Flags [P.], seq 1:245,
 ack 297, win 65535, length 244: HTTP: HTTP/1.1 200 OK
E.......@..F"k.R
....P.h.{...D..P.......HTTP/1.1 200 OK
Server: nginx
Date: Mon, 18 Jan 2021 16:52:22 GMT
Content-Type: text/plain
Via: 1.1 google
Age: 72690
Cache-Control: public, must-revalidate, max-age=0, s-maxage=86400
Content-Length: 8
Connection: keep-alive

success

18:33:52.238722 IP 10.0.2.15.47464 > 34.107.221.82.80: Flags [.], ack 245, wi
n 63996, length 0
E..(.2@.@.A.
..."k.R.h.P.D...{..P.......
18:33:52.255769 IP 10.0.2.15.47466 > 34.107.221.82.80: Flags [S], seq 3038851
270, win 64240, options [mss 1460,sackOK,TS val 3036175962 ecr 0,nop,wscale 7
], length 0
E..<<3@.@...
..."k.R.j.P.!0..................
..^Z........
18:33:52.314356 IP 34.107.221.82.80 > 10.0.2.15.47466: Flags [S.], seq 142400
001, ack 3038851271, win 65535, options [mss 1460], length 0
E..,....@..1"k.R
....P.j.|...!0.`...
)........
18:33:52.314387 IP 10.0.2.15.47466 > 34.107.221.82.80: Flags [.], ack 1, win
64240, length 0
```

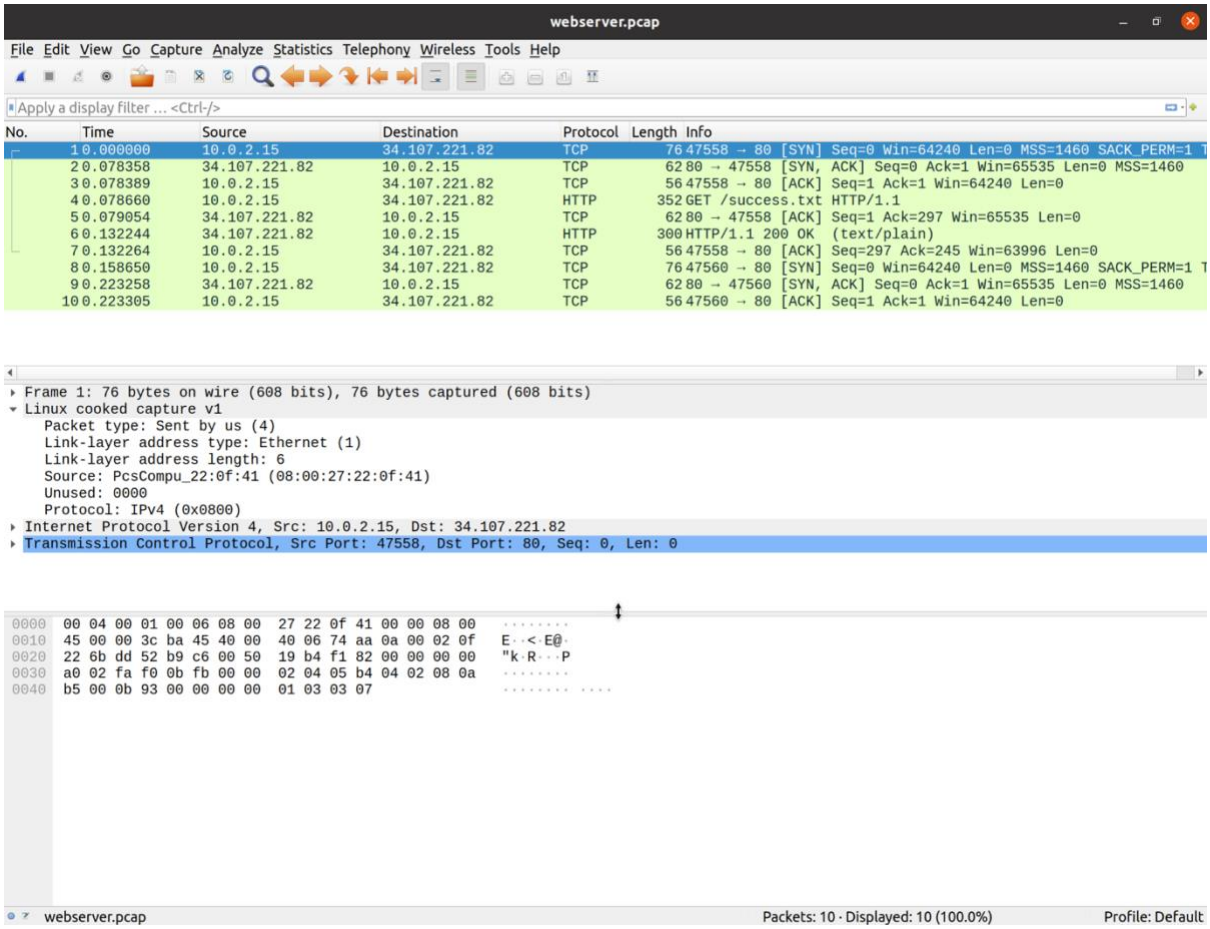
```
E..(.2@.@.A.
..."k.R.h.P.D...{..P.......
18:33:52.255769 IP 10.0.2.15.47466 > 34.107.221.82.80: Flags [S], seq 3038851
270, win 64240, options [mss 1460,sackOK,TS val 3036175962 ecr 0,nop,wscale 7
], length 0
E..<<3@.@...
..."k.R.j.P.!0...................
..^Z........
18:33:52.314356 IP 34.107.221.82.80 > 10.0.2.15.47466: Flags [S.], seq 142400
001, ack 3038851271, win 65535, options [mss 1460], length 0
E..,....@..1"k.R
....P.j.|...!0.`...
).........
18:33:52.314387 IP 10.0.2.15.47466 > 34.107.221.82.80: Flags [.], ack 1, win
64240, length 0
E..(<4@.@...
..."k.R.j.P.!0..|..P.......
10 packets captured
10 packets received by filter
0 packets dropped by kernel
```

## Step 6 : Saving packets to a file

Using the -w option to save the packets to a file instead of displaying them on screen,
sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80

```
trisha@trisha-VirtualBox:~$ sudo tcpdump -i any -c10 -nn -w webserver.pcap po
rt 80
[sudo] password for trisha:
tcpdump: listening on any, link-type LINUX_SLL (Linux cooked v1), capture siz
e 262144 bytes
10 packets captured
14 packets received by filter
0 packets dropped by kernel
trisha@trisha-VirtualBox:~$
```

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




webserver.pcap

## Task 5: Perform Traceroute Checks

Step 1: Running the traceroute using the command :
sudo traceroute www.google.com

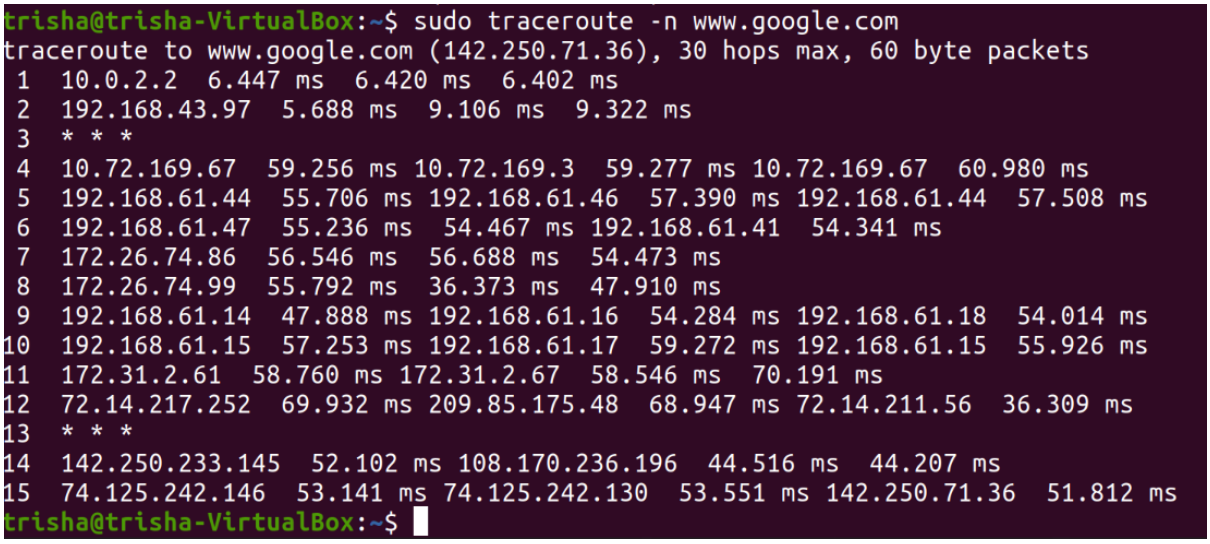
```
trisha@trisha-VirtualBox:~$ sudo traceroute www.google.com
traceroute to www.google.com (172.217.31.196), 30 hops max, 60 byte packets
 1  _gateway (10.0.2.2)  0.286 ms  0.226 ms  0.210 ms
 2  192.168.43.97 (192.168.43.97)  13.241 ms  14.444 ms  14.429 ms
 3  * * *
 4  10.72.169.3 (10.72.169.3)  69.286 ms  69.415 ms  72.745 ms
 5  192.168.61.44 (192.168.61.44)  69.145 ms 192.168.61.46 (192.168.61.46)  69.269 ms  69.359 ms
 6  192.168.61.41 (192.168.61.41)  69.194 ms  44.983 ms 192.168.61.45 (192.168.61.45)  45.242 ms
 7  172.26.74.86 (172.26.74.86)  44.932 ms  48.900 ms  49.090 ms
 8  172.26.74.99 (172.26.74.99)  49.075 ms  29.782 ms  29.652 ms
 9  192.168.61.14 (192.168.61.14)  41.920 ms  41.644 ms 192.168.61.18 (192.168.61.18)  47.763 ms
10  192.168.61.19 (192.168.61.19)  47.998 ms 192.168.61.17 (192.168.61.17)  49.087 ms  52.215 ms
11  172.26.29.42 (172.26.29.42)  58.638 ms  56.257 ms  50.197 ms
12  172.26.29.42 (172.26.29.42)  49.665 ms  49.371 ms 172.26.29.107 (172.26.29.107)  35.062 ms
13  10.70.80.197 (10.70.80.197)  44.897 ms  32.652 ms  42.655 ms
14  10.70.80.225 (10.70.80.225)  42.746 ms  42.549 ms  47.062 ms
15  74.125.48.26 (74.125.48.26)  38.375 ms  38.001 ms  37.943 ms
16  108.170.253.97 (108.170.253.97)  55.644 ms  55.898 ms 108.170.253.113 (108.170.253.113)  49.664 ms
17  74.125.253.13 (74.125.253.13)  51.666 ms  41.615 ms 74.125.253.17 (74.125.253.17)  41.563 ms
18  maa03s28-in-f4.1e100.net (172.217.31.196)  31.787 ms  44.309 ms  56.523 ms
trisha@trisha-VirtualBox:~$
```

sudo traceroute www.google.com

Step 2:Ananlyzing destination address of google.com and number of hops

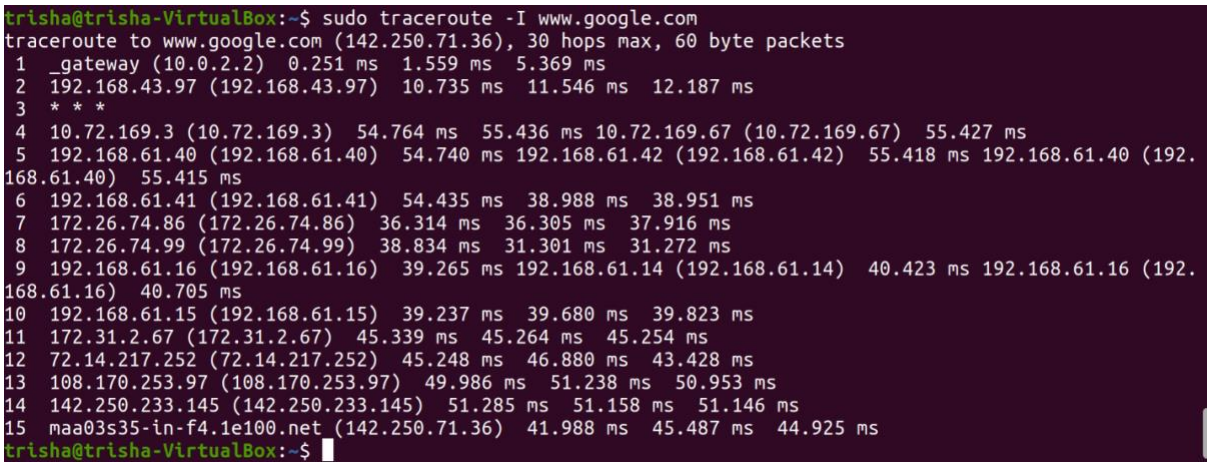
*Number of hops* : 18 (Max number of hops allowed = 30)
*Destination Address of google.com* : 172.217.31.196 (Destination server : maa03s28-in-f4.1e100.net)

Step 3: Speeding up the process by disabling the mapping of IP address with hostnames by using the -n option



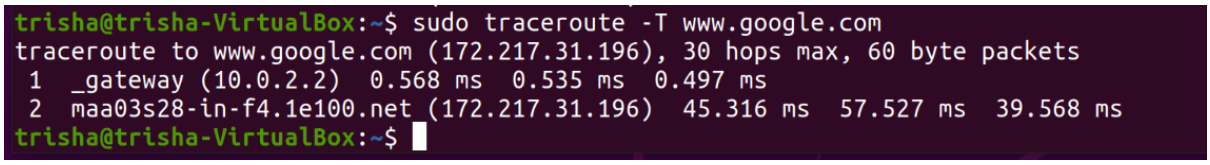

sudo traceroute -n www.google.com

Step 4: Using the -I option so that the traceroute uses ICMP




sudo traceroute -I www.google.com

## Step 5: Using the -T flag to test a TCP connection to gather data more relevant to web server




sudo traceroute -T www.google.com

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

Step 1: Scanning a host using host name (Command: nmap www.pes.edu)

```
trisha@trisha-VirtualBox:~$ nmap www.pes.edu
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-25 14:57 IST
Nmap scan report for www.pes.edu (13.71.123.138)
Host is up (0.016s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
443/tcp  open  https
554/tcp  open  rtsp
1723/tcp open  pptp

Nmap done: 1 IP address (1 host up) scanned in 4.79 seconds
```

Step 2: Scanning a host using IP Address
Command used : nmap 163.53.78.128

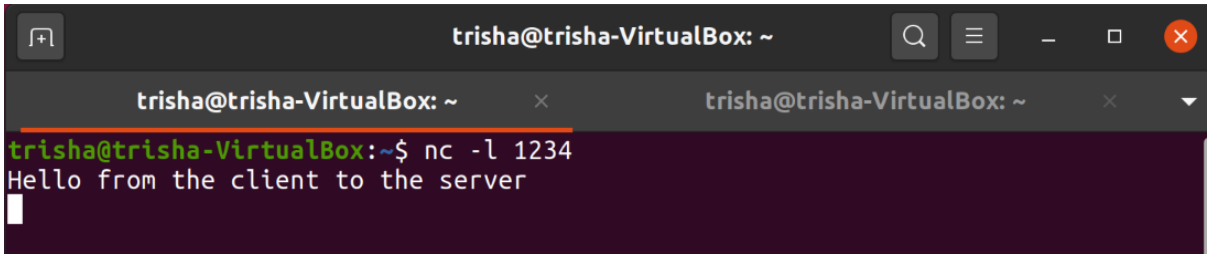
```
trisha@trisha-VirtualBox:~$ nmap 163.53.78.128
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-25 14:57 IST
Nmap scan report for 163.53.78.128
Host is up (0.17s latency).
Not shown: 995 filtered ports
PORT     STATE SERVICE
21/tcp   open  ftp
80/tcp   open  http
443/tcp  open  https
554/tcp  open  rtsp
1723/tcp open  pptp

Nmap done: 1 IP address (1 host up) scanned in 42.35 seconds
```

Step 3: Scanning multiple IP Addresses
Command used : nmap 192.168.1.1 192.168.1.2 192.168.1.3

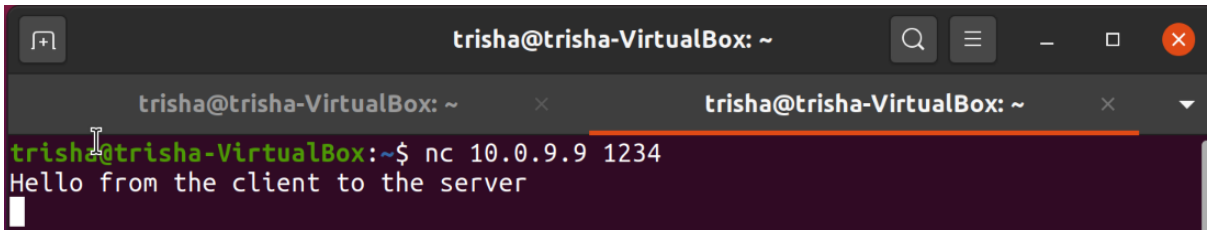
```
trisha@trisha-VirtualBox:~$ nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.80 ( https://nmap.org ) at 2021-01-25 14:58 IST
Nmap done: 3 IP addresses (0 hosts up) scanned in 3.09 seconds
```

## Task 7 a): Netcat as Chat tool

a) Intra system communication (Using 2 terminals in the same system)

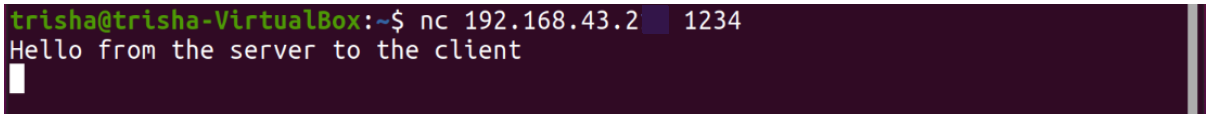



nc -l 1234

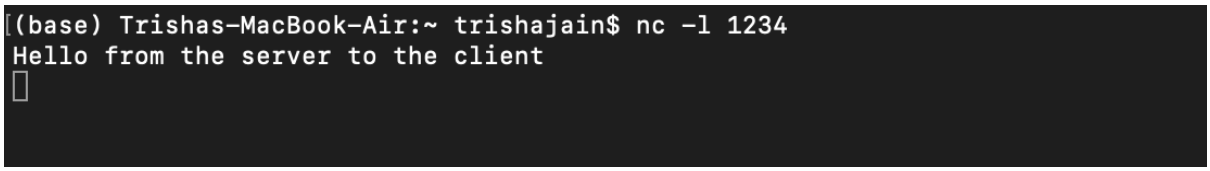



nc 10.0.9.9 1234

b) Inter system communication



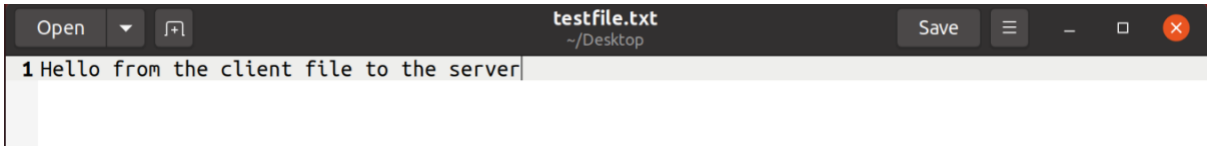

nc 192.168.43.2?? 1234




nc -l 1234

Note: The last two digits of the IP address have been hidden.

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




testfile.txt

```
trisha@trisha-VirtualBox:~/Desktop$ sudo nc 192.168.43.2  555 < testfile.txt
trisha@trisha-VirtualBox:~/Desktop$
```

sudo nc 192.168.43.2?? 555 < testfile.txt

```
(base) Trishas-MacBook-Air:Desktop trishajain$ sudo nc -l 555 > test.txt
Password:
```

```
(base) Trishas-MacBook-Air:Desktop trishajain$ cat test.txt
Hello from the client file to the server
```

sudo nc -l 555 > test.txt

Task 7 c): Other commands

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

```
trisha@trisha-VirtualBox: ~/Desktop
trisha@trisha-VirtualBox:~/Desktop$ nc -l 1234

trisha@trisha-VirtualBox: ~/Desktop
trisha@trisha-VirtualBox:~/Desktop$ nc -vn 10.0.2.15 555
nc: connect to 10.0.2.15 port 555 (tcp) failed: Connection refused
trisha@trisha-VirtualBox:~/Desktop$ nc -vn 10.0.2.15 1234
Connection to 10.0.2.15 1234 port [tcp/*] succeeded!
```

nc -vn 10.0.2.15 555

2) Run a web server with a static web page

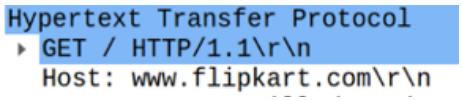
```
trisha@trisha-VirtualBox:~/Desktop$ while true; do sudo nc -lp 80 < test.html; d
one
GET /test.html HTTP/1.1
Host: 10.0.2.15
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:75.0) Gecko/20100101 Fire
fox/75.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0
.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
```

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

Questions on above observations:-

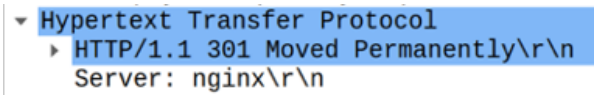
Q1. Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?

A1. The browser is running HTTP version 1.1.




HTTP 1.1 for request

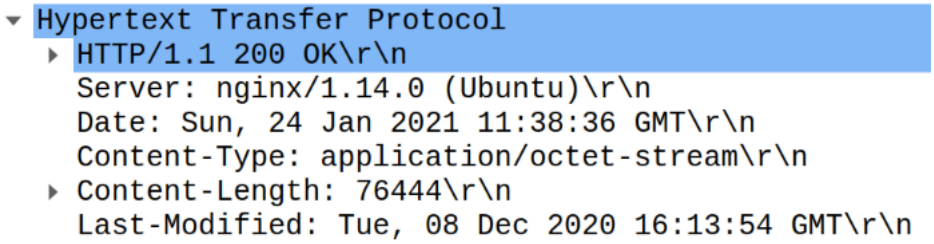
The server is also running HTTP version 1.1.




HTTP 1.1 for response

Q2. When was the HTML file that you are retrieving last modified at the server?

A2. The date the HTML file was last modified is specified in the Last-Modified field.



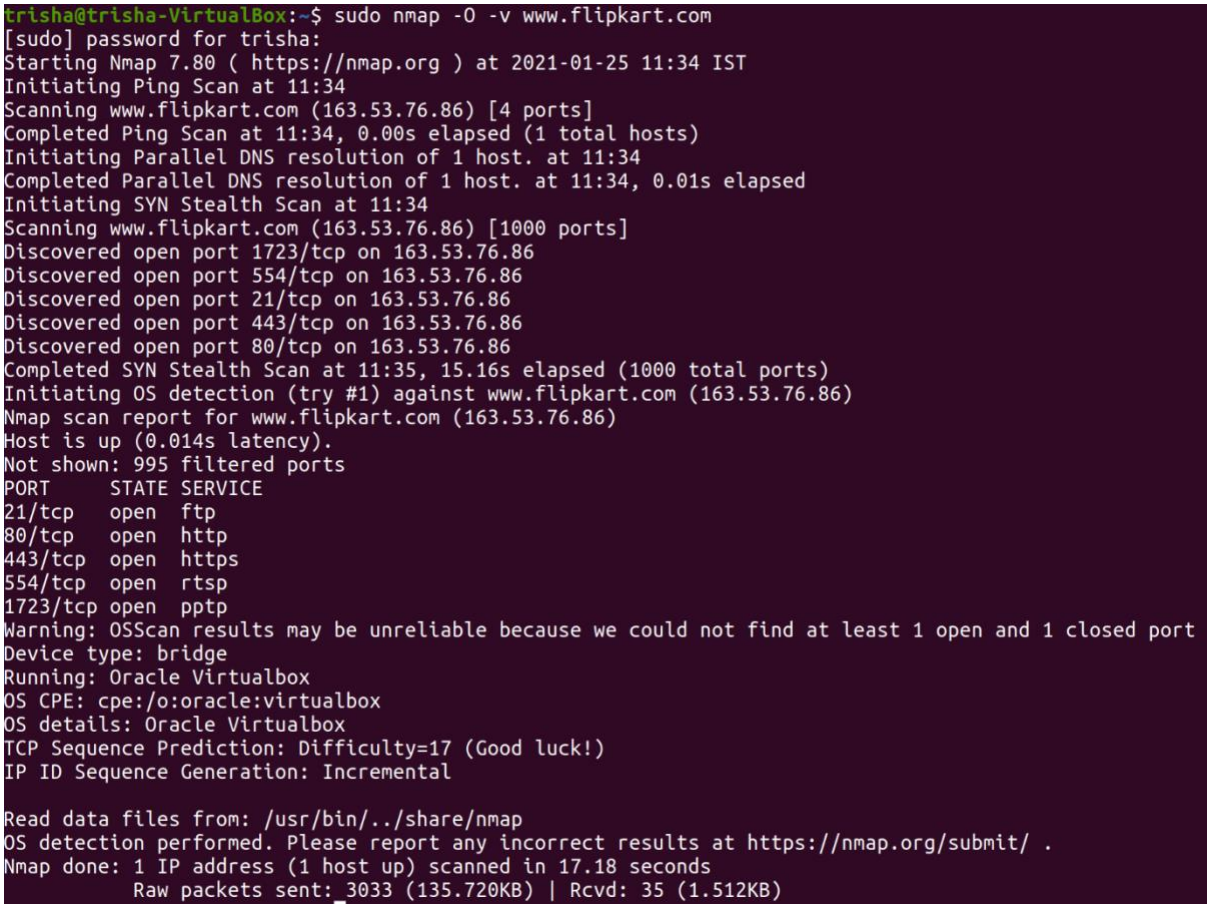

Specified as Tue, 08 Dec 2020 16:13:54 GMT\r\n

Q3. How to tell ping to exit after a specified number of ECHO_REQUEST packets?

A3. The ICMP packages continue to be sent until interrupted. Therefore to exit after a specified number of ECHO_REQUEST packets, we can use the -c option followed by the number of packets.

Q4. How will you identify remote host apps and OS?

A4. The remote host apps and OS can be identified using the command as shown below:




sudo nmap -O -v www.flipkart.com