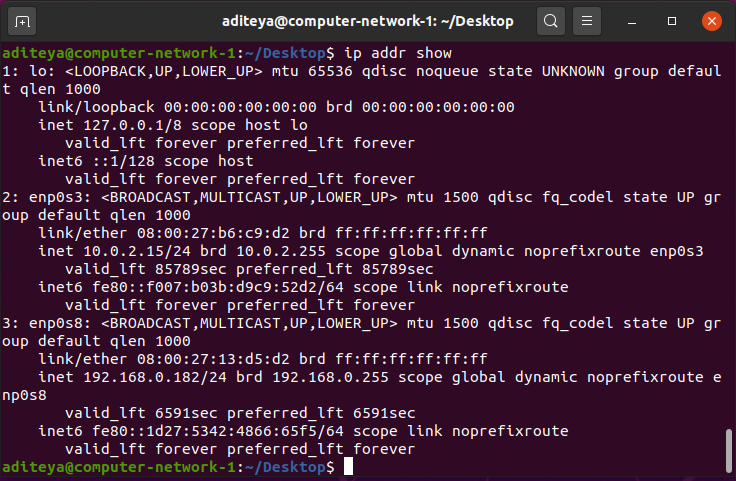
Computer Networks Lab - Week 1

PES1201800366

Aditeya Baral

1. Linux Interface Configuration

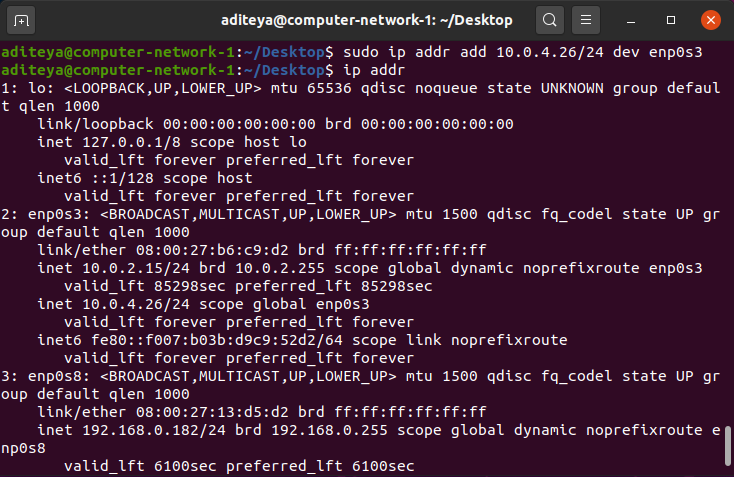
1.1 ip addr show



|  |  |  |
| --- | --- | --- |
| **Interface Name** | **IPv4/IPv6** | **MAC Address** |
| lo | 127.0.0.1/::1 | 00:00:00:00:00:00 |
| enp0s3 | 10.0.2.15/fe80::f007:b03b:d9c9:52d2 | 08:00:27:b6:c9:d2 |
| enp0s8 | 192.168.0.182/fe80::1d27:5342:4866:65f5 | 08:00:27:13:d5:d2 |

1.2 Assigning an IP

Command used: sudo ip addr add 10.0.4.26/24 dev enp0s3

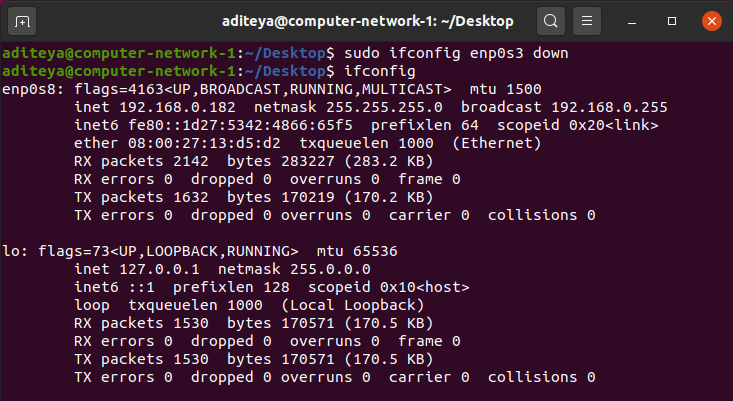


inet 10.0.4.26/24 scope global enp0s3

1.3 Activating and Deactivating Network Interfaces

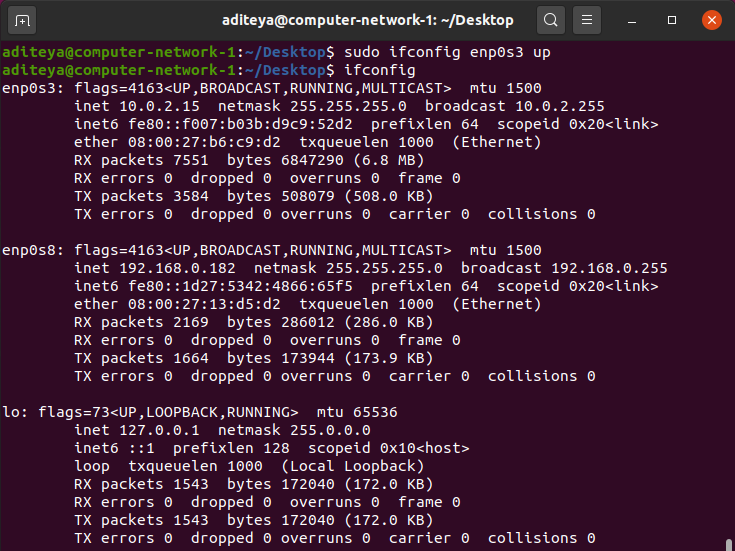
1.3.1 Deactivating enp0s3

Command used: sudo ifconfig enp0s3 down



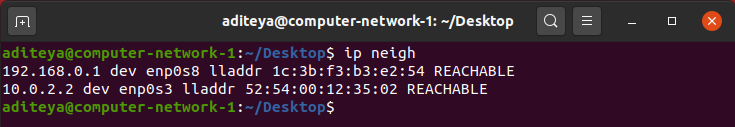
Only enp0s8 and lo are displayed above

1.3.2 Activating enp0s3

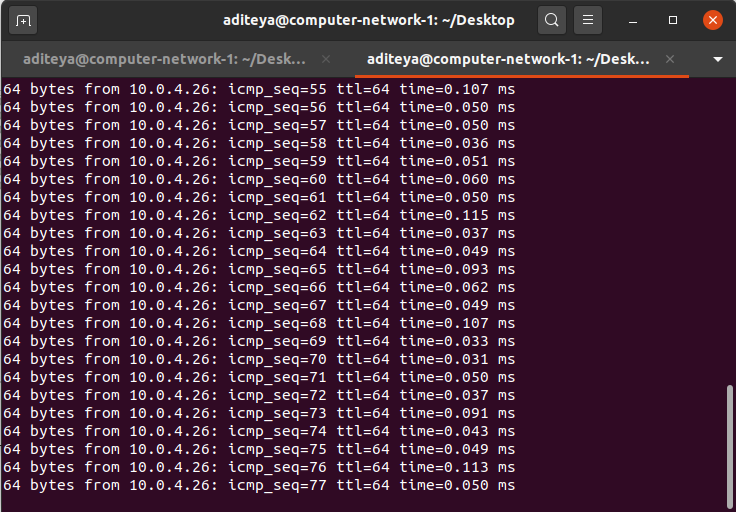


enp0s3 has been reactivated

1.4 Step 4 – ip neigh

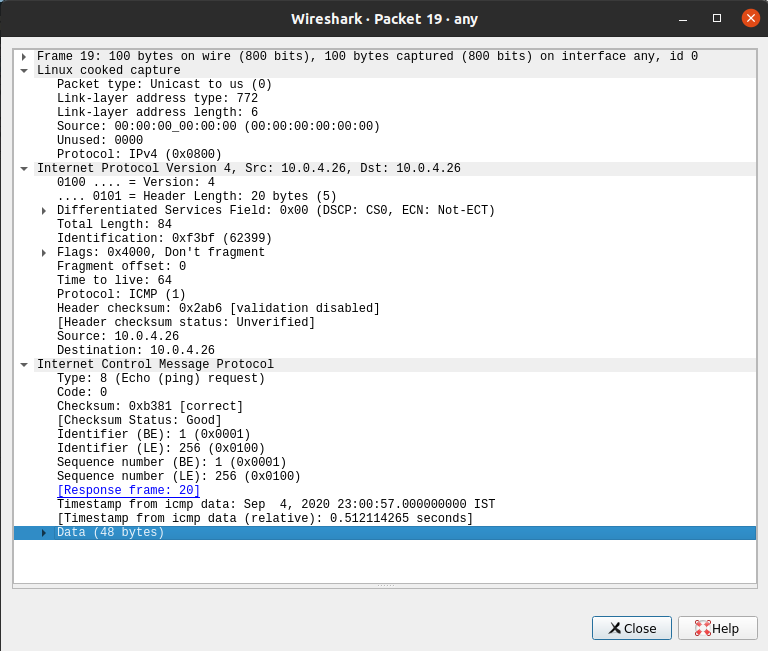


2. Ping PDU (Packet Data Units) Capture

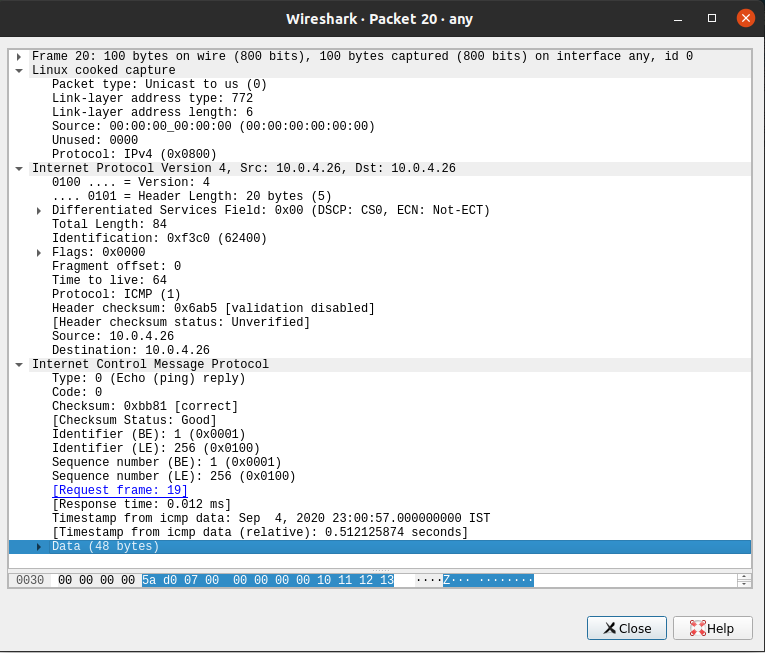


ping 10.0.4.26

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



Request Packet

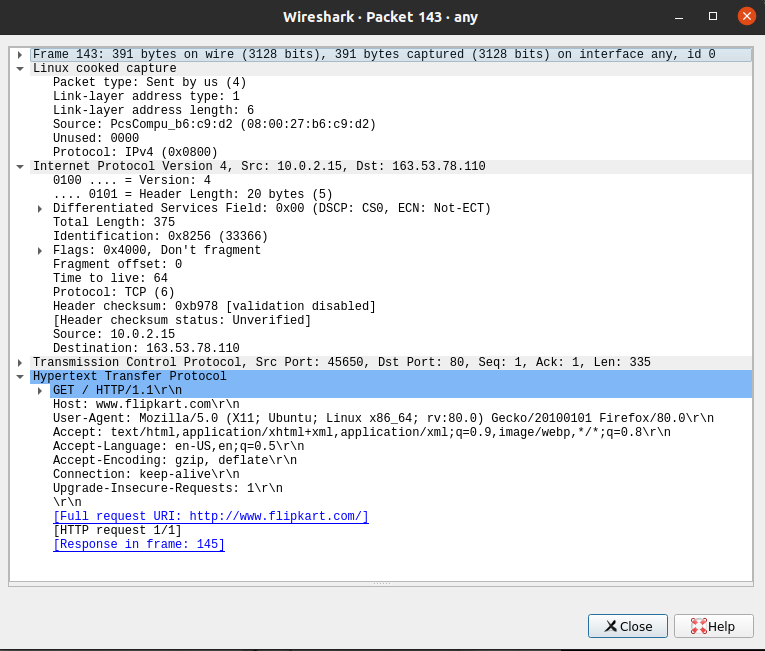


Response Packet

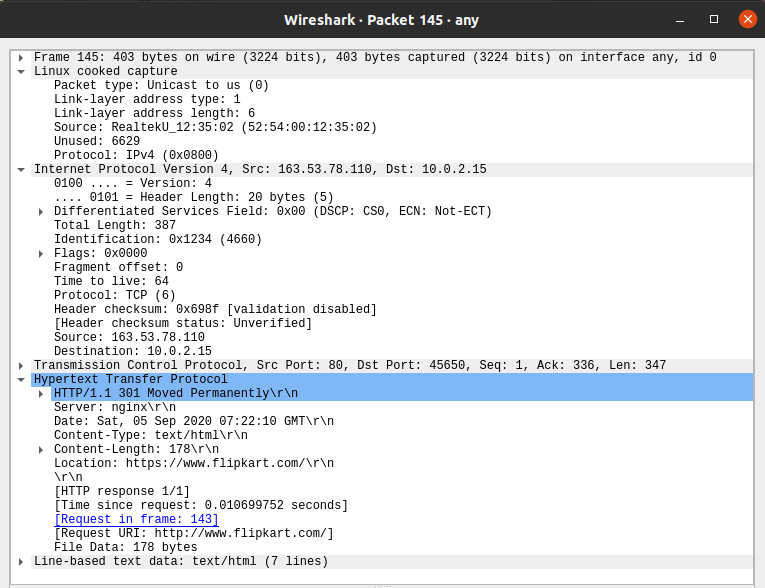
|  |  |  |
| --- | --- | --- |
| **Details** | **First Echo Request** | **First Echo Reply** |
| Frame Number | 19 | 20 |
| Source IP address | 10.0.4.26 | 10.0.4.26 |
| Destination IP address | 10.0.4.26 | 10.0.4.26 |
| ICMP Type Value | 8 | 0 |
| ICMP Code Value | 0 | 0 |
| Source Ethernet Address | 00:00:00:00:00:00 | 00:00:00:00:00:00 |
| Destination Ethernet | 00:00:00:00:00:00 | 00:00:00:00:00:00 |
| Internet Protocol Version | IPv4 | IPv4 |
| Time To Live (TTL) | 64 | 64 |

3. HTTP PDU Capture

3.1 Echo Request and Reply



Request Packet



Response Packet

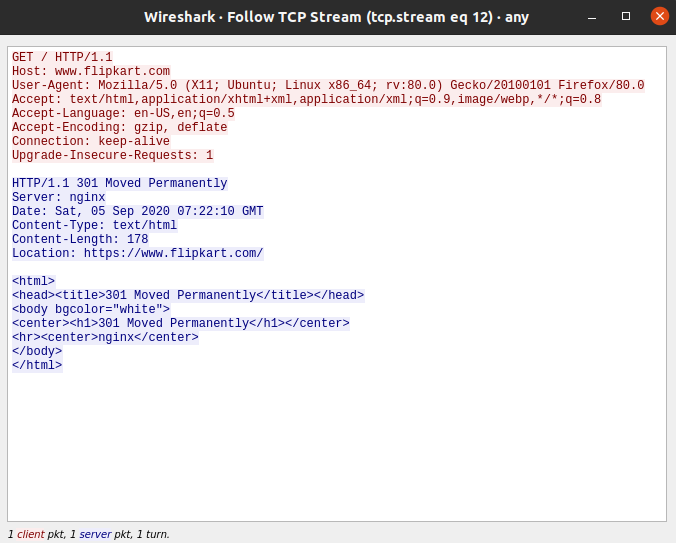
|  |  |  |
| --- | --- | --- |
| **Details** | **First Echo Request** | **First Echo Reply** |
| Frame Number | 143 | 145 |
| Source Port | 45650 | 80 |
| Destination Port | 80 | 45650 |
| Source IP Address | 10.0.2.15 | 163.53.78.110 |
| Destination IP Address | 163.53.78.110 | 10.0.2.15 |
| Source Ethernet Address | 08:00:27:b6:c9:d2 | 52:54:00:12:35:02 |
| Destination Ethernet Address | 52:54:00:12:35:02 | 08:00:27:b6:c9:d2 |

Connection Details

3.2 HTTP Request and Response

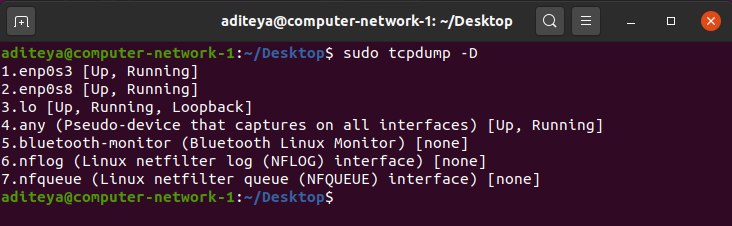
|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP Request** |  | **HTTP Response** |  |
| **Get** | GET / HTTP/1.1\r\n | **Server** | nginx |
| **Host** | www.flipkart.com | **Content-Type** | text/html |
| **User-Agent** | Mozilla/5.0 (X11; Ubuntu; Linux x86\_64; rv:80.0) Gecko/20100101 Firefox/80.0 | **Date** | Sat, 05 Sep 2020 07:22:10 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 |

3.3 Following TCP Stream



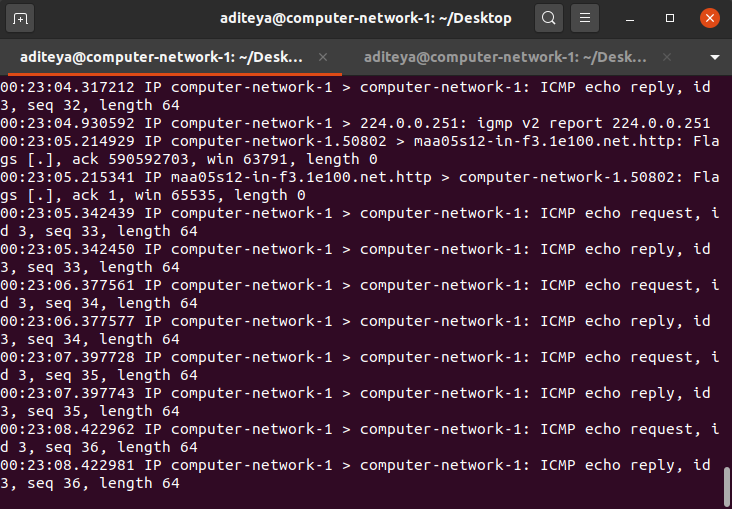
4. Capturing Packets with tcpdump

4.1 Viewing Interfaces available for Capture



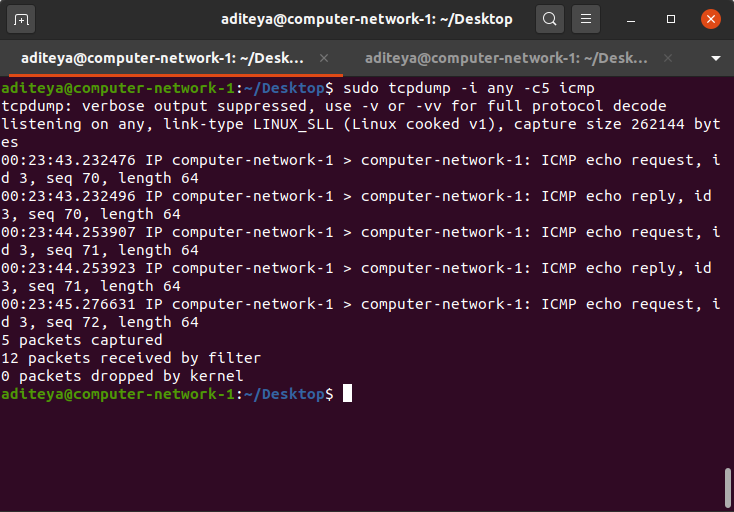
tcpdump -D

4.2 Capturing all Packets in any Interface



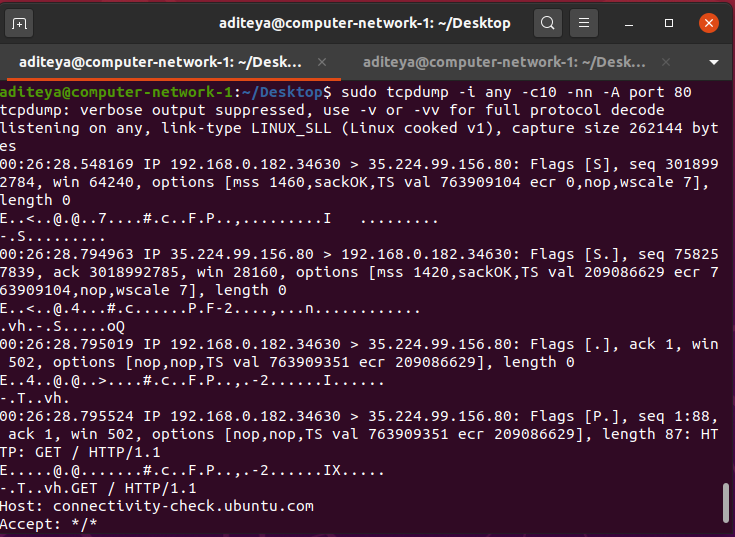
tcpdump -i any

4.3 Filtering Packets based on Protocol

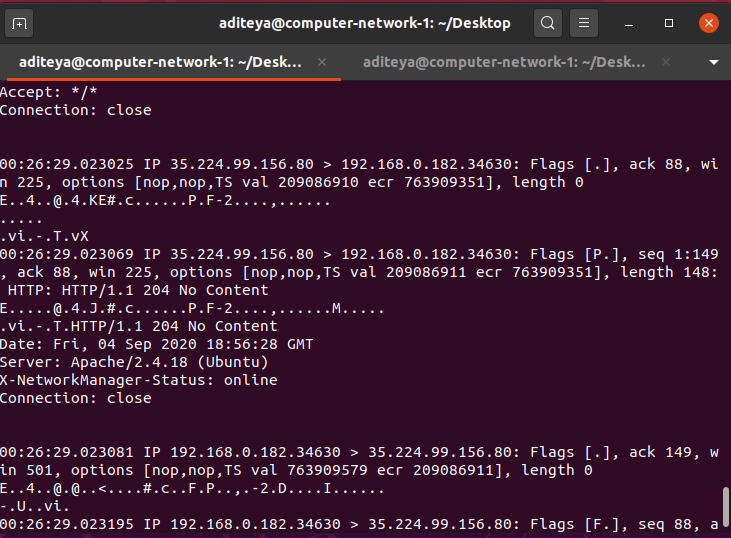


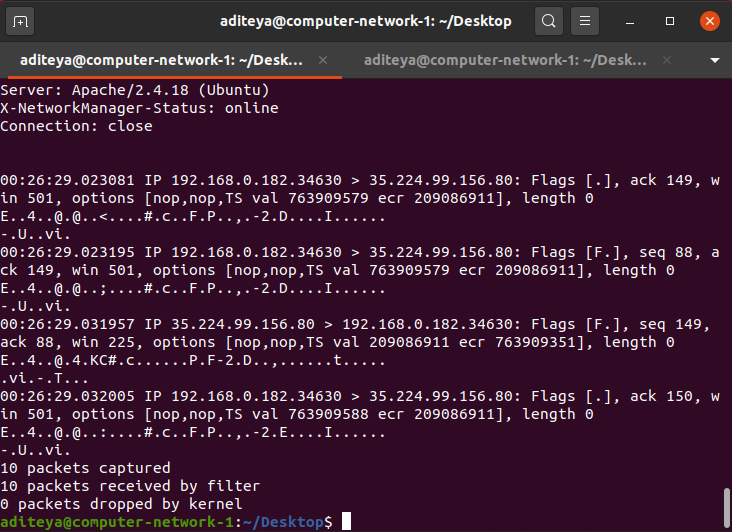
sudo tcpdump -i any -c5 icmp

4.4 Checking Packet Content

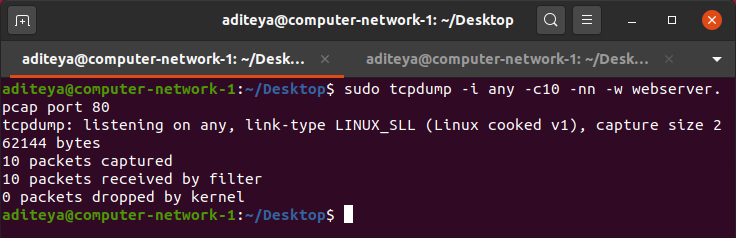


sudo tcpdump -i any -c10 -nn -A port 80

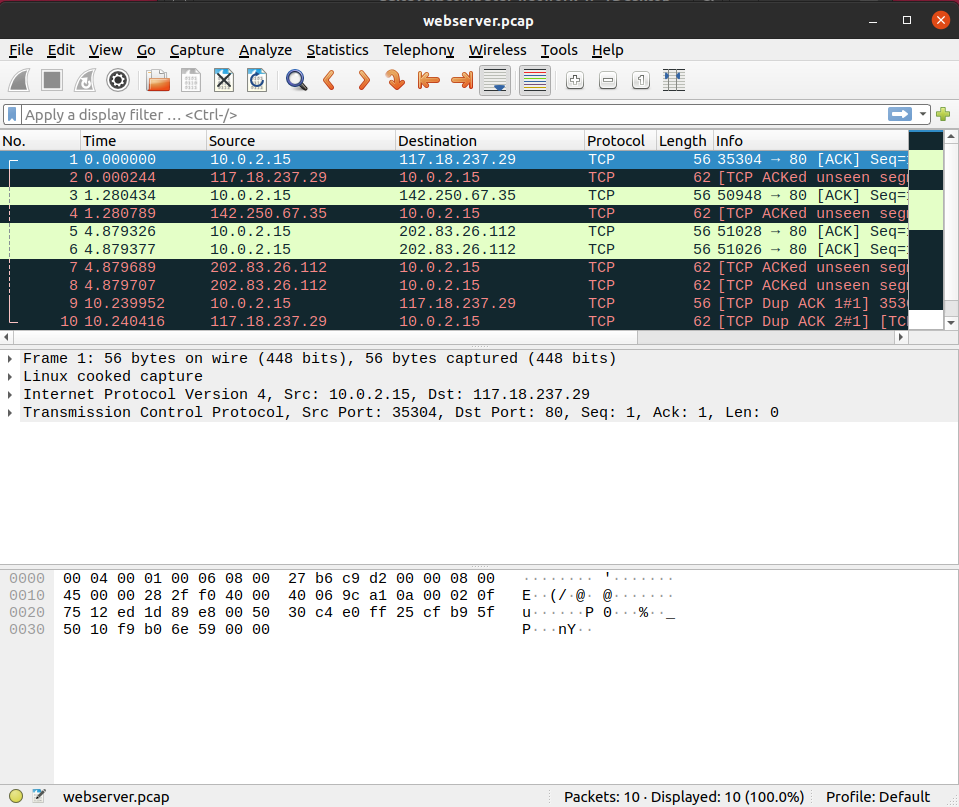




4.5 Saving Packets to a File



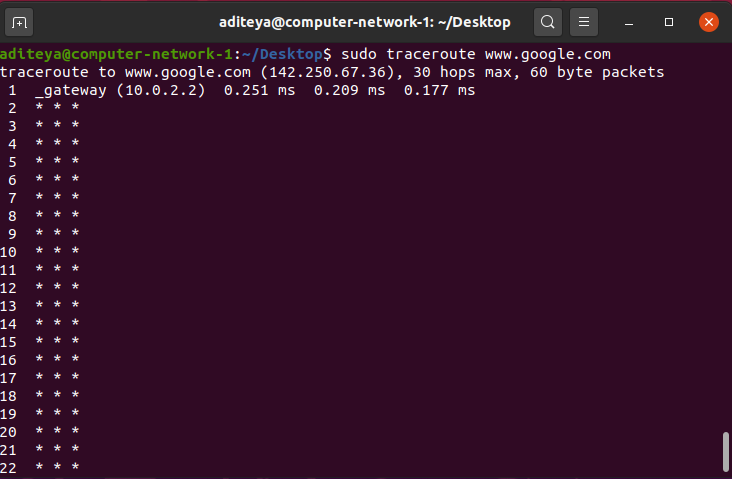
sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80



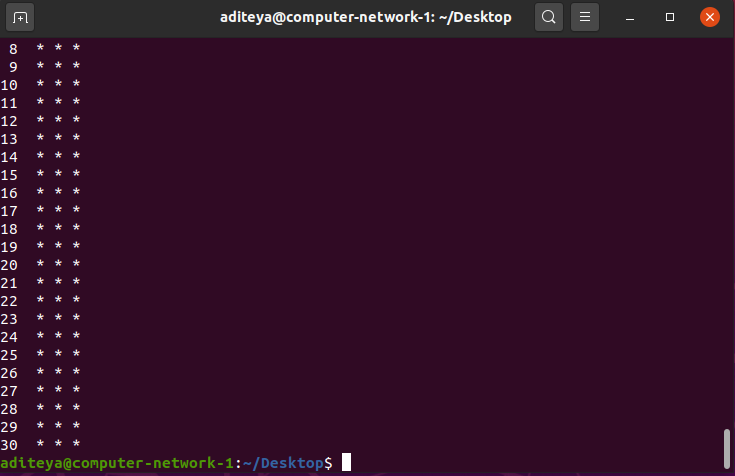
webserver.pcap

5. Perform traceroute Checks

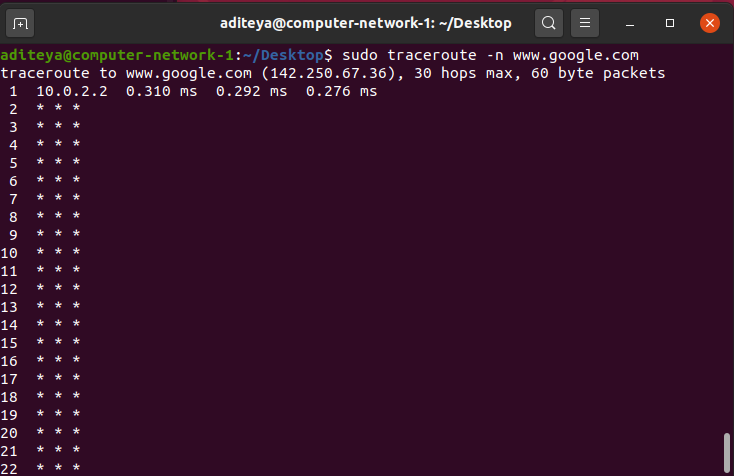
5.1 Running traceroute



sudo traceroute [www.google.com](http://www.google.com)

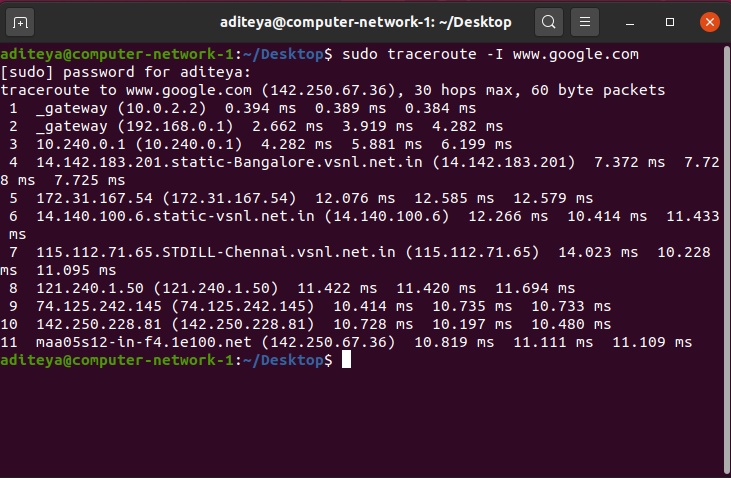


5.2 Disabling mapping of IP addresses with hostnames



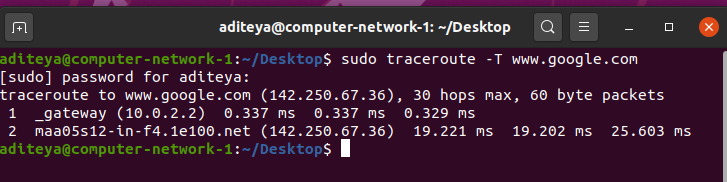
sudo traceroute -n [www.google.com](http://www.google.com)

5.3 traceroute with ICMP Protocol



sudo traceroute -I [www.google.com](http://www.google.com)

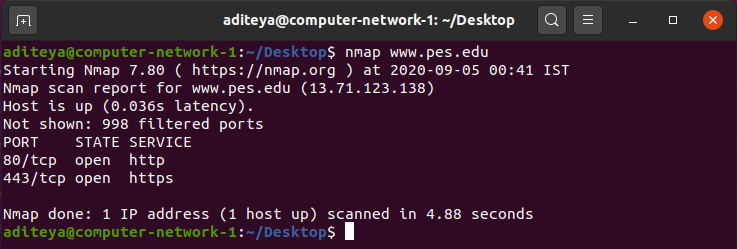
5.4 Testing TCP Connection with traceroute



sudo traceroute -T [www.google.com](http://www.google.com)

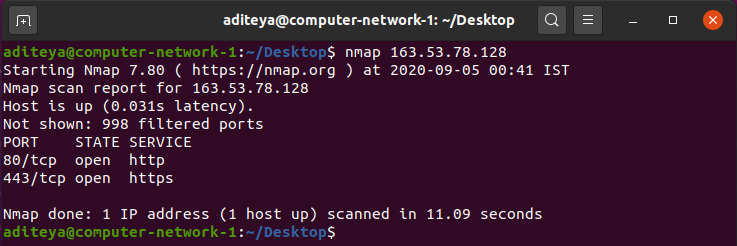
6. Exploring a network with nmap

6.1 Scanning Host with Hostname



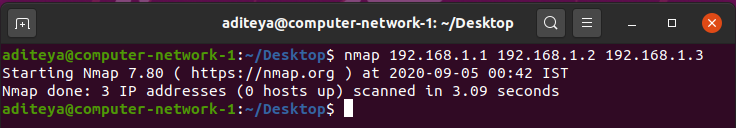
nmap [www.pes.edu](http://www.pes.edu)

6.2 Scanning Host with IP Address



nmap 163.53.78.128

6.3 Scanning Multiple IP Addresses or Subnet (IPv4)

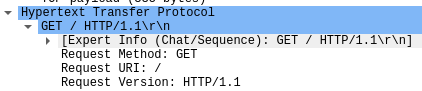


nmap 192.168.1.1 192.168.1.2 192.168.1.3

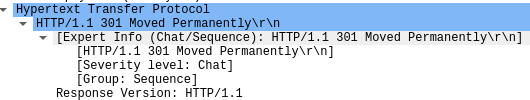
7. Questions

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

**Answer –** The Firefox browser used is running HTTP v1.1, and this can be seen in the request header which contains the method (GET) followed by the HTTP version. Similarly, the HTTP version of the web server is v1.1 and can be seen in the header of the HTTP response sent back to the browser.



Request



Response

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

**Answer** **–** We can find the last modified time of the HTML file at the server by observing the **Last-Modified** field of the HTTP response object. The Last-Modified field stores a timestamp of the last modification time. Example:



**3. How to tell ping to exit after a specified number of ECHO\_REQUEST packets?**

**Answer –** Ping continues to send ICMP packages until it receives an interrupt signal. To specify the number of ECHO\_REQUEST packages after which ping will exit, we can use the -c option followed by the number of packages.

ping -c 10 [www.pes.edu](http://www.pes.edu)

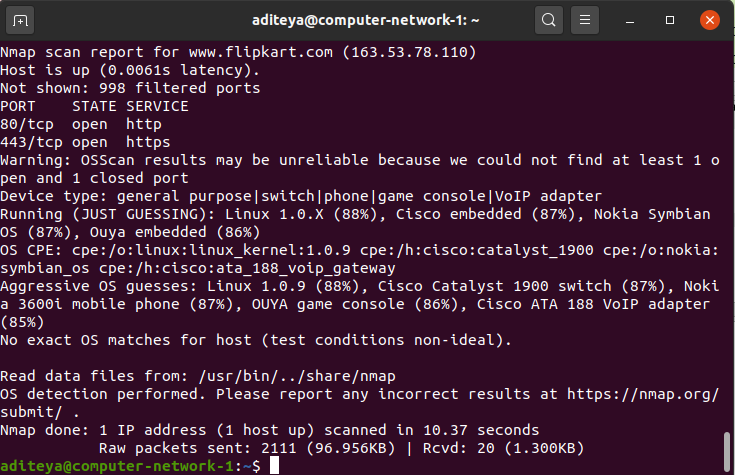
**4. How will you identify remote host apps and OS?**

**Answer –**

1. We can obtain the remote host app and OS of the server by observing the **Server** files of the HTTP response object. The Server field stores the remote host app or server on which it is hosted and the OS too. Example:



1. We can use nmap to find the OS too. It will scan the network to find information about the remote host apps and OS.

  
nmap -O -v www.flipkart.com