|  |  |  |
| --- | --- | --- |
| Name: Adithya M | SRN:PES1UG20CS621 | Section:K |
| Date:19/1/2022 | Week Number:2 |

**Computer Networks**

|  |  |
| --- | --- |
| **Week 1** | Study and understand the basic networking tools - Wireshark, Tcpdump, Ping, Traceroute and Netcat. |

|  |
| --- |
| **Learn and understand Network Tools**   1. Wireshark  * Perform and analyze Ping PDU capture * Examine HTTP packet capture * Analyze HTTP packet capture using filter  1. Netcat  * Establish communication between client and server * Transfer files  1. Tcpdump  * Capture packets  1. Ping  * Test the connectivity between two systems  1. Traceroute  * Perform tranceroute checks  1. Nmap  * Explore an entire network |

**Task 1: Linux interface Configuration (ifconfig / IP command )**

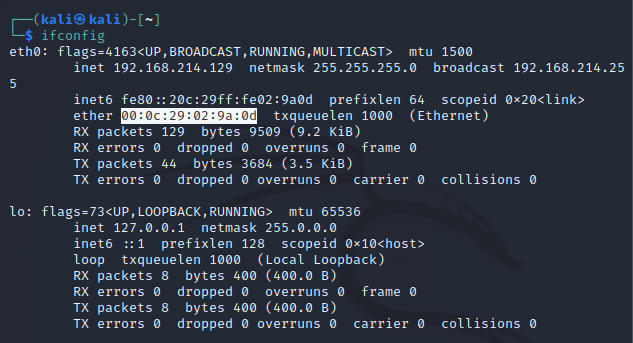
**Step 1:** To display status of all active network interfaces.

**Ifconfig (or) ip addr show**

Analyze and fill the following table:

**Ip address table:**

|  |  |  |
| --- | --- | --- |
| **Interface name** | **IP address (IPv4 / IPv6)** | **MAC address** |
| **eth0** | **192.168.214.129** | **00:0c:29:02:9a:0d** |
| **lo** | **127.0.0.1** | **NA** |

****

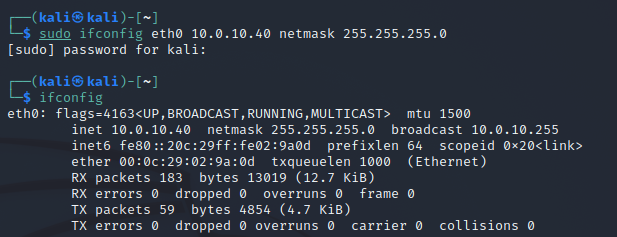
The hardware address and the IP address is mentioned, when **ifconfig** is typed in the

terminal.

**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**

****

**10.0.10.40** is assigned as the **IP address** to the interface.

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

**sudo ifconfig interface\_name** **down**

**sudo ifconfig interface\_name up**

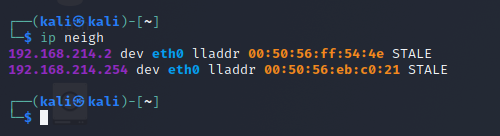
**ip**

The configured interface is set to up and

running if it isn’t.

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

**ip neigh**

****

The neighbour table is shown in the output.

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

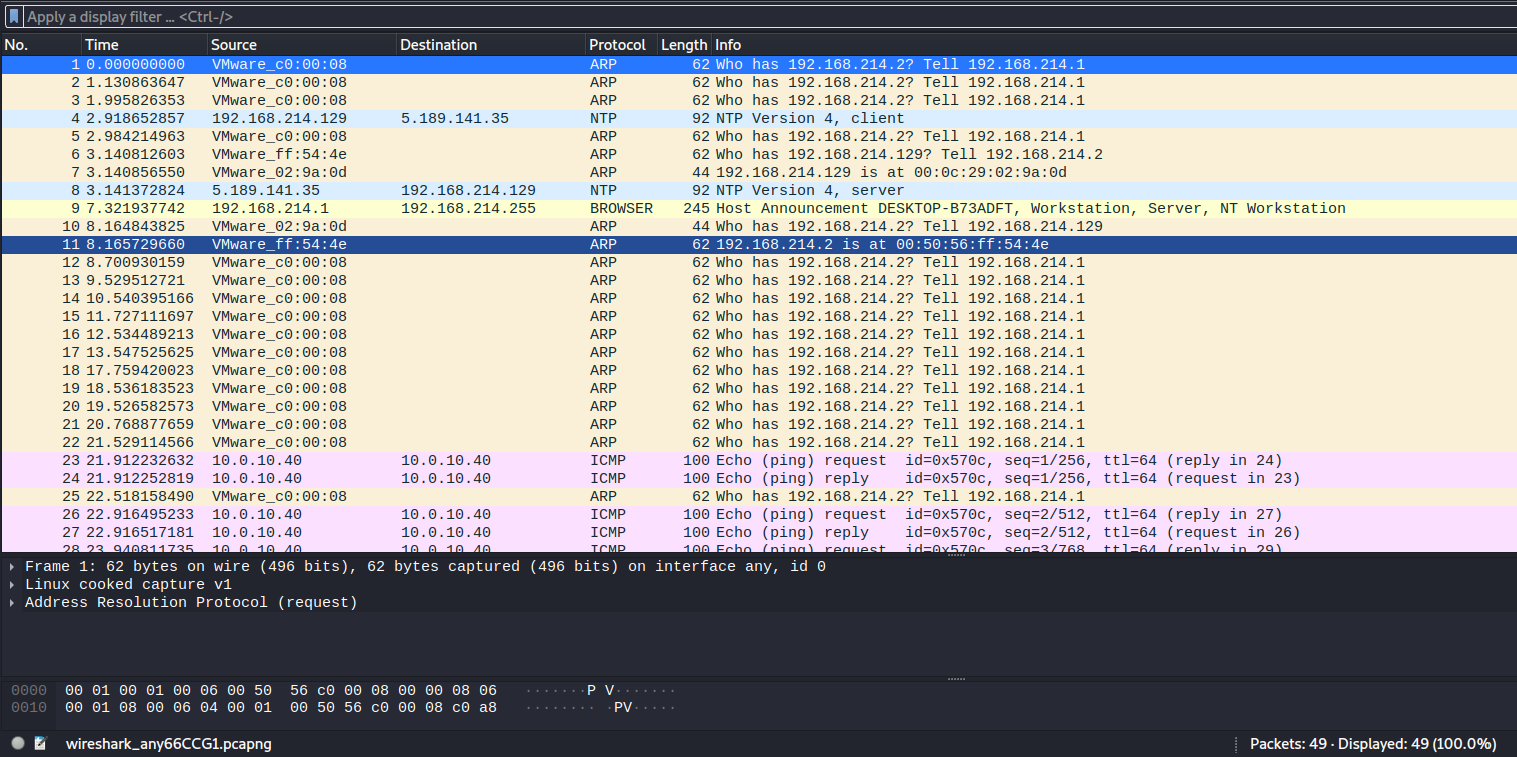
**Step 1:** Assign an IP address to the system (Host).

**Note:** IP address of your system should be **10.0.your\_section.your\_sno.**

****

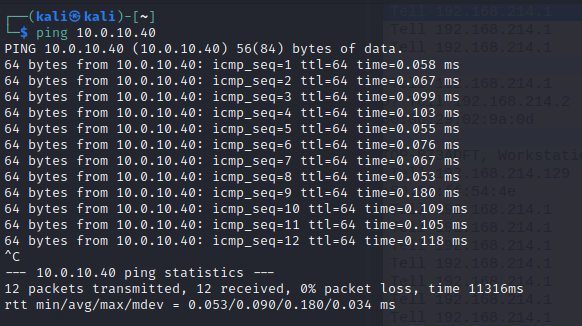
The IP address is set to **10.0.10.40.**

**Step 2:** Launch Wireshark and select ‘**any’** interface

****

Wireshark on launch and opened into “any”.

**Step 3:** In terminal, type ping **10.0.your\_section.your\_sno**

****

Observations to be made

**Step 4:** Analyze the following in Terminal

* TTL
* Protocol used by ping
* Time

The TTL is **64.**

The protocol used by ping is **ICMP.**

The time taken is **0.090 ms** on average.

**Step 5:** Analyze the following in Wireshark

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.

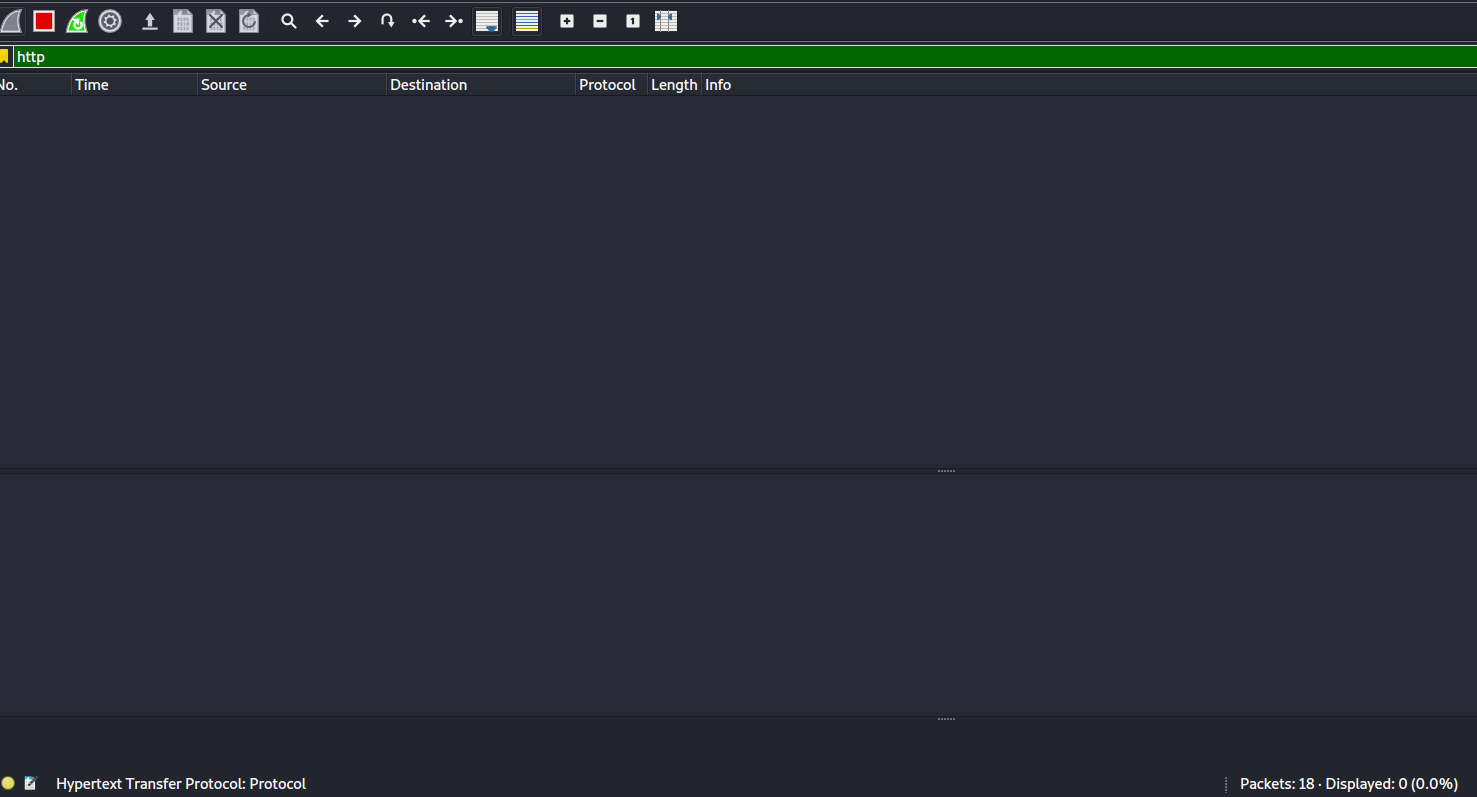
|  |  |  |
| --- | --- | --- |
| **Details** | **First echo request** | **First echo reply** |
| Frame Number | 23 | 24 |
| Source IP address | 10.0.10.40 | 10.0.10.40 |
| Destination IP address | 10.0.10.49 | 10.0.10.40 |
| ICMP Type Value | 8 | 0 |
| ICMP Code Value | 0 | 0 |
| Source Ethernet Address | 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 | 4 | 4 |
| Time To Live (TTL) Value | 64 | 64 |

Task 3: HTTP PDU Capture

Using Wireshark’s Filter feature

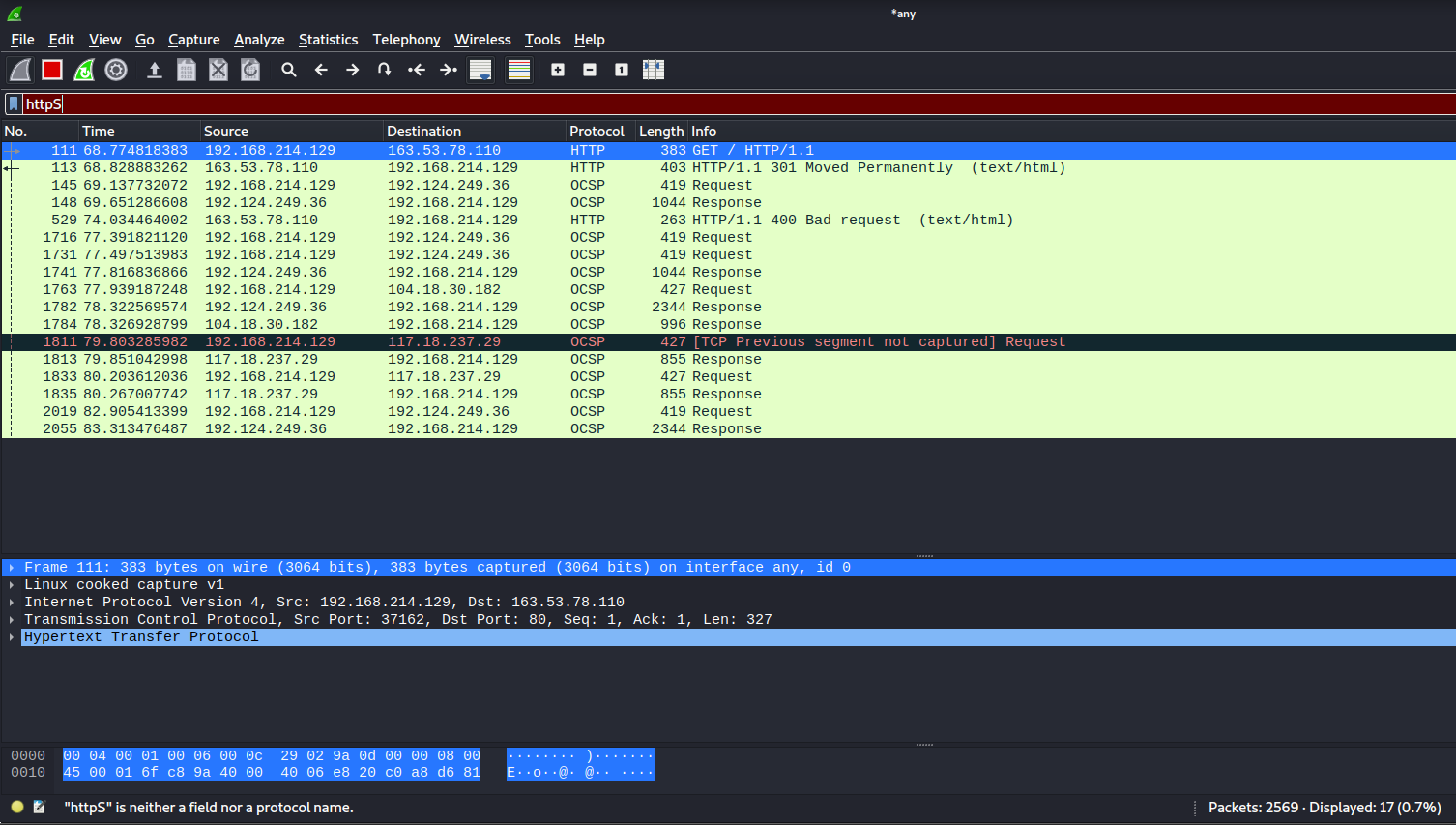
**Step 1:** Launch Wireshark and select ‘any’ interface. On the Filter toolbar, type-in ‘http’ and

press enter



**Step 2:** Open Firefox browser, and browse [www.flipkart.com](http://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 | 111 | 113 |
| Source port | 37162 | 80 |
| Destination port | 80 | 37162 |
| Source IP address | 192.168.214.129 | 163.53.78.110 |
| Destination IP address | 163.53.78.110 | 192.168.214.129 |
| 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 |

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

|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP request** |  | **HTTP response** |  |
| **GET** | / HTTP/1.1 | **Server** | nginx |
| **Host** | www.flipkart.com | **Content-type** | text/html |
| **User-agent** | Mozilla/5.0 | **Date** | Mon 24,Jan 2022 13:47:06 GMT |
| **Accept Language** | en-us,en | **Location** | https://www.flipkart.com |
| **Accept-Encoding** | gzip,deflate | **Content-Length** | 178 |
| **connection** | keep-alive | **connection** | keep-alive |

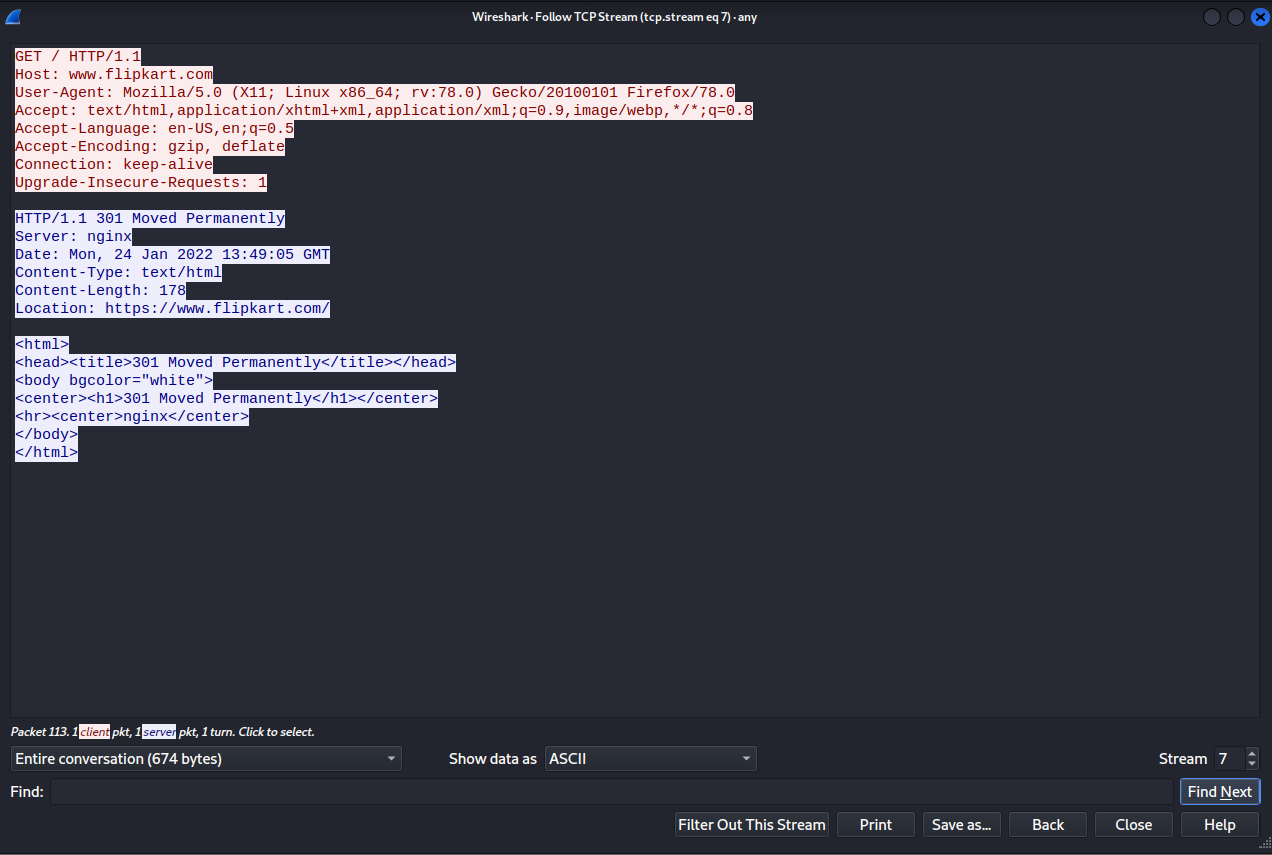
Using Wireshark’s Follow TCP Stream

**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.

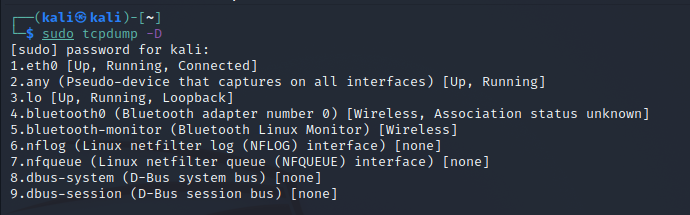
**Step 2:** Upon following a TCP stream, screenshot the whole window.

****

Task 4: Capturing packets with tcpdump

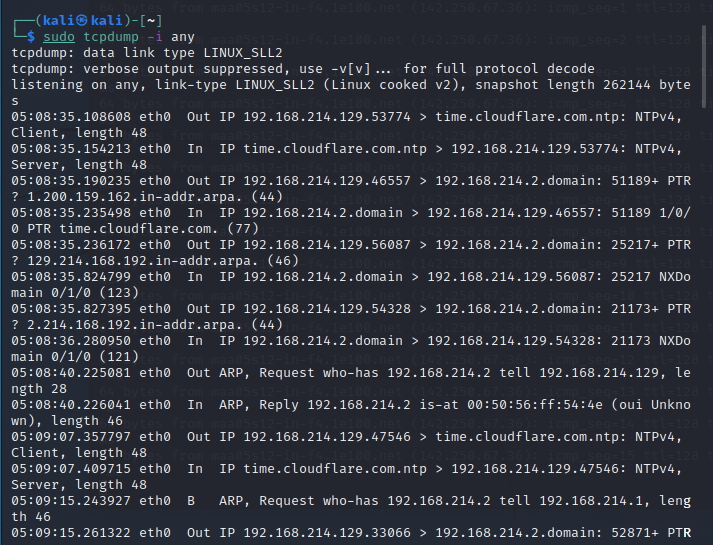
**Step 1:** Use the command **tcpdump -D** to see which interfaces are available for capture.

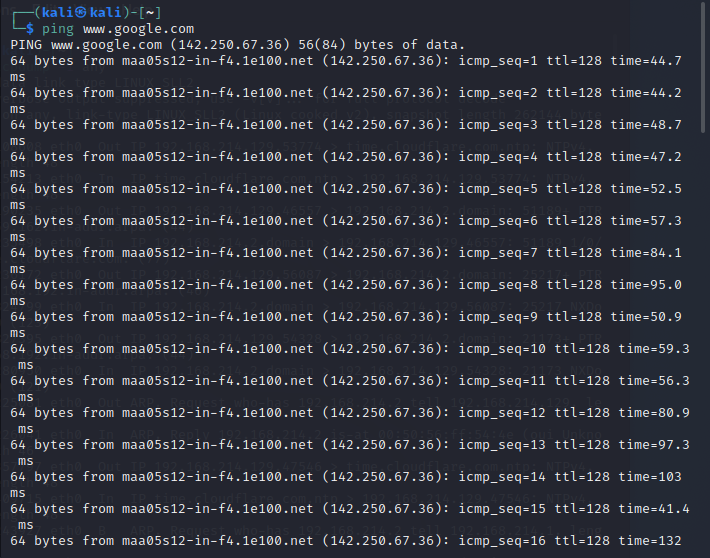
**sudo tcpdump –D**

****

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

**sudo tcpdump -i any**

****

****

**Note:** Perform some pinging operation while giving above command. Also type

www.google.com in browser.

Observation

**Step 3:** Understand the output format.

The above command is used to capture all the packets from all the interfaces.

ICMP, UDP and TCP are the main packets that are visible in the above screenshot.

The timestamp followed by the link level headers, then by ARP/RARP packets if any,

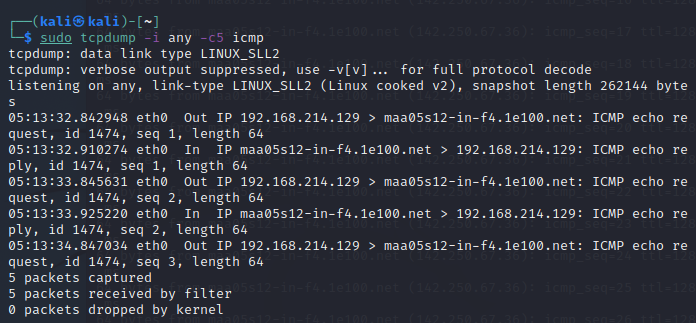
Then by IPv4 packets if any, followed by TCP packets. The sequence numbers and the

length finish defining the outputs.

**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**

****

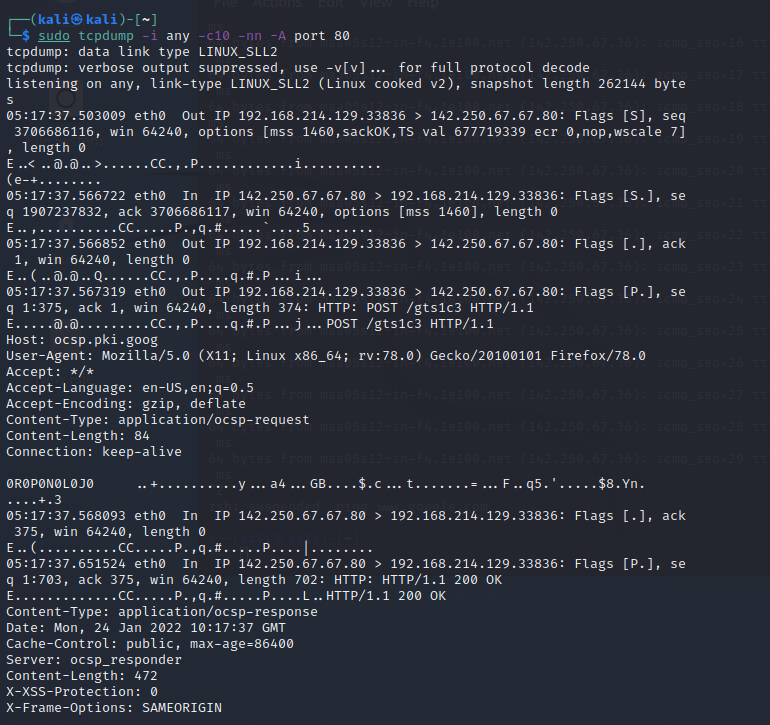
****

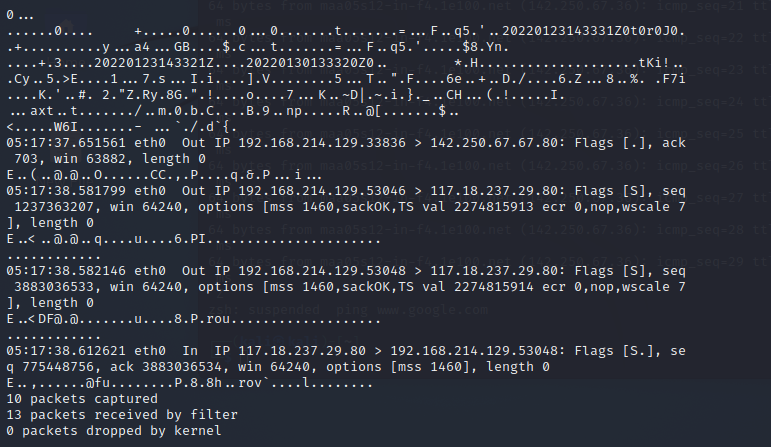
**Step 5:** Check the packet content. For example, inspect the HTTP content of a web request

like this:

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

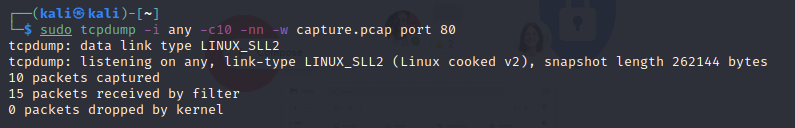
On trying to access the Gmail account sign-in website.

****

****

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

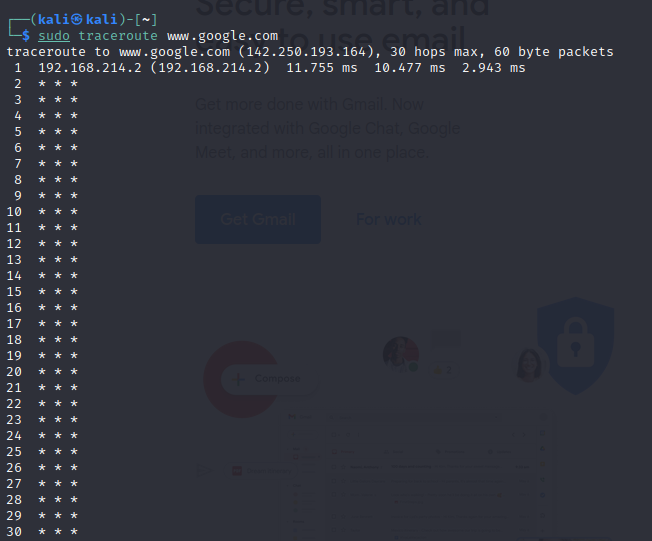
**sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80**

****

Task 5: Perform Traceroute checks

**Step 1:** Run the traceroute using the following command.

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

****

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

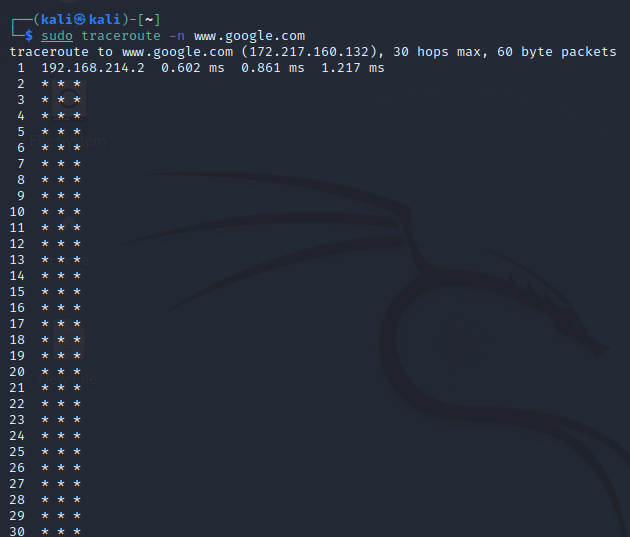
The destination address is **142.250.196.164** [FOUND OUT BY PINGING IN WINDOWS]

The total number of hops is **30**, and most of pings have been timed out.

**Step 3:** To speed up the process, you can disable the mapping of IP addresses with hostnames

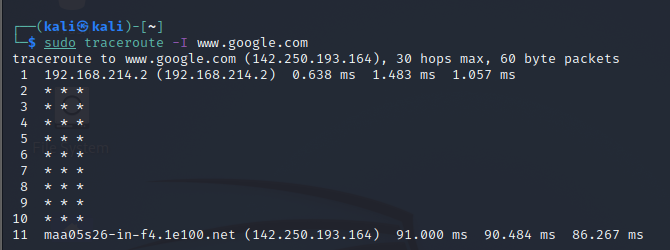
by using the -n option

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

****

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

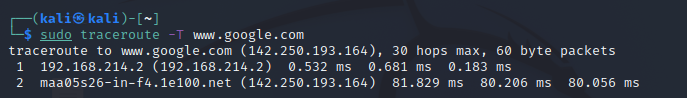
**sudo traceroute -I www.google.com**

****

**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.

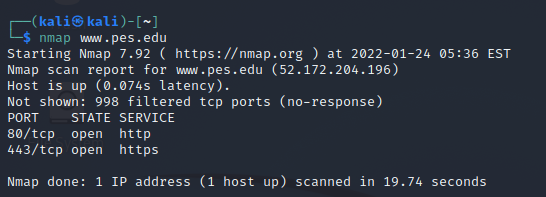
**sudo traceroute -T www.google.com**

****

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

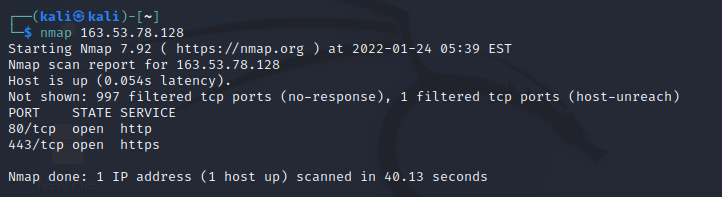
**Step 1:** You can scan a host using its host name or IP address, for instance.

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

****

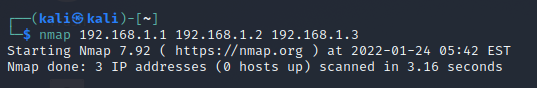
**Step 2:** Alternatively, use an IP address to scan.

**nmap 163.53.78.128**

****

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

**nmap 192.168.1.1 192.168.1.2 192.168.1.3**

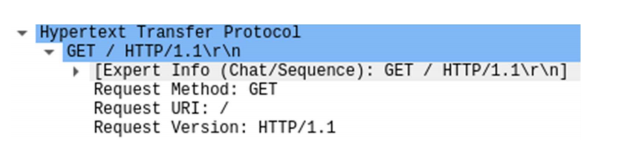
****

**Questions on above observations:**

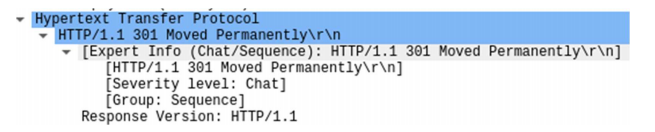
1) Is your browser running HTTP version 1.0 or 1.1? What version of HTTP is the server?

**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:

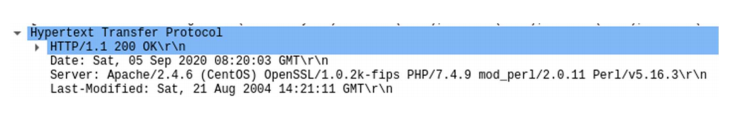


Response:



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:



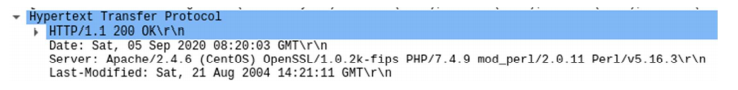
3) How to tell ping to exit after a specified number of ECHO\_REQUEST packets?

**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**

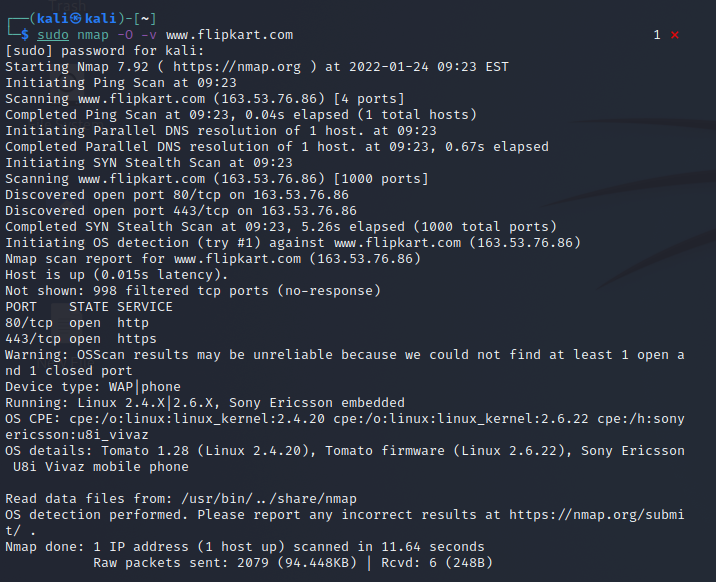
4) How will you identify remote host apps and OS?

**Answer:** 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:



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

**sudo nmap -O -v** [**www.flipkart.co**](http://www.flipkart.co)

****