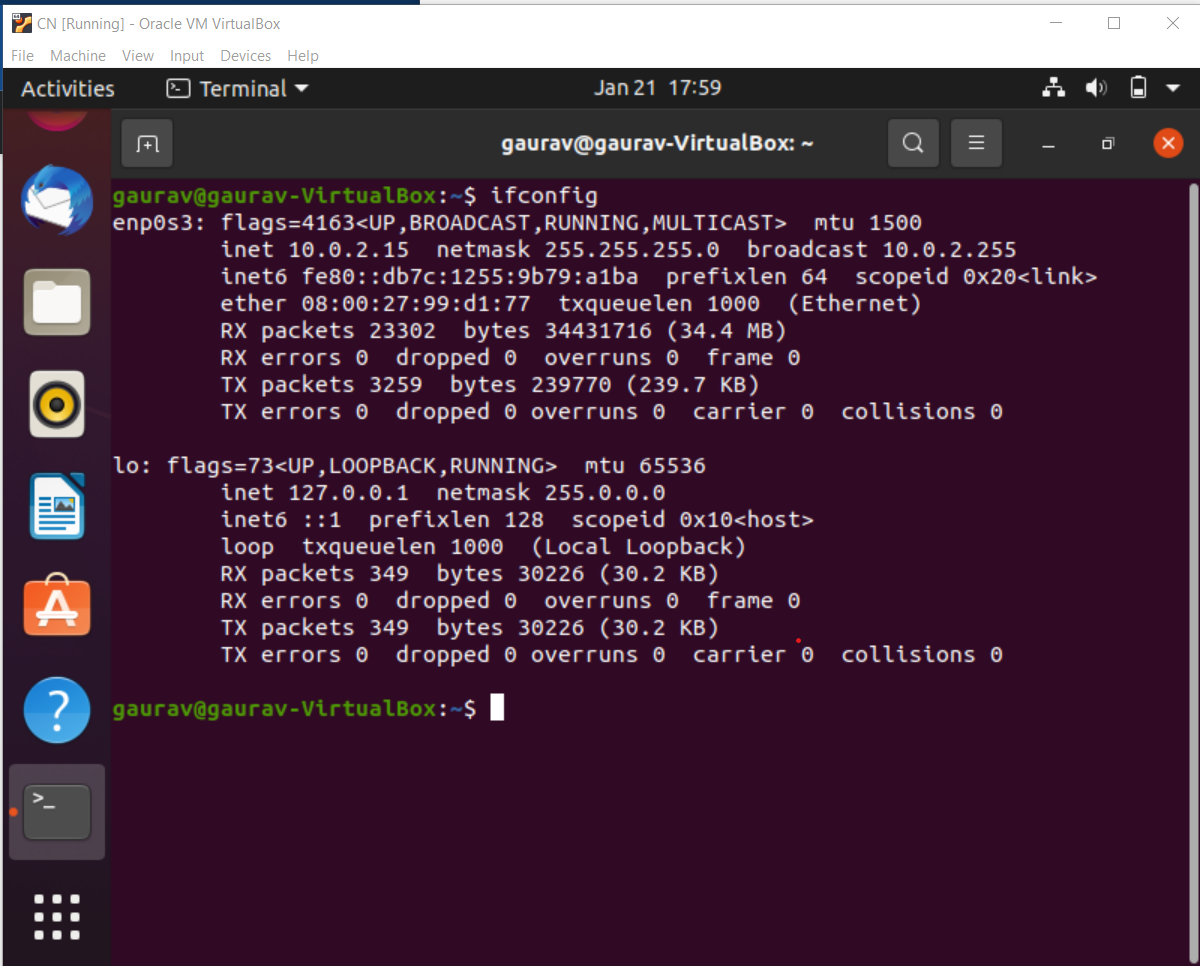
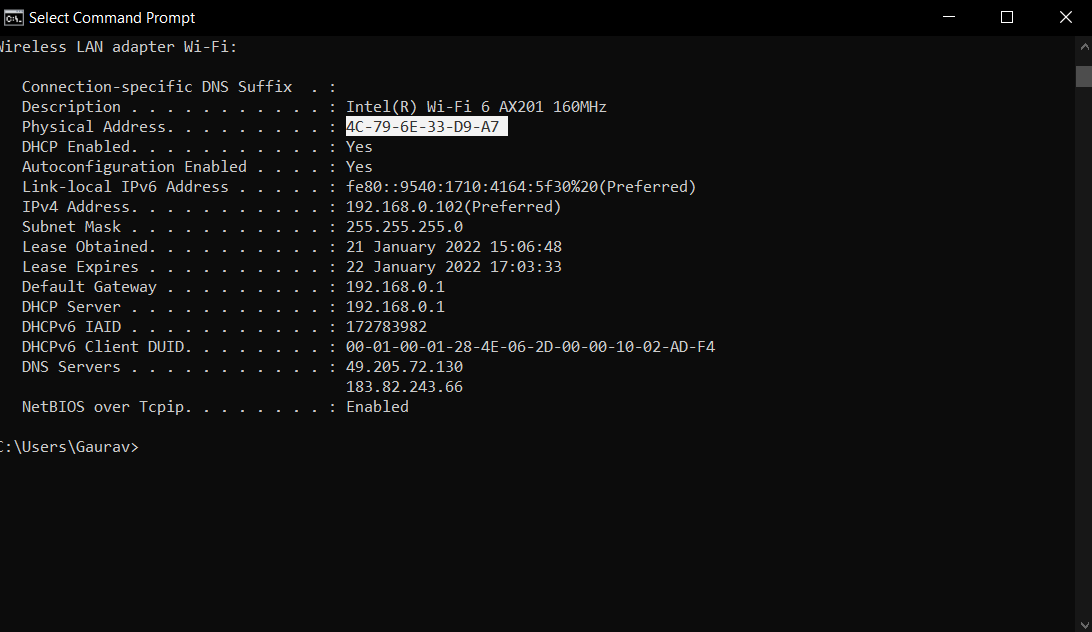
|  |  |  |
| --- | --- | --- |
| **Name: GAURAV MAHAJAN** | **SRN: PES1UG20CS150** | **Section: C** |
| **Date: 19-01-2022** | **WEEK 1** |

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

****

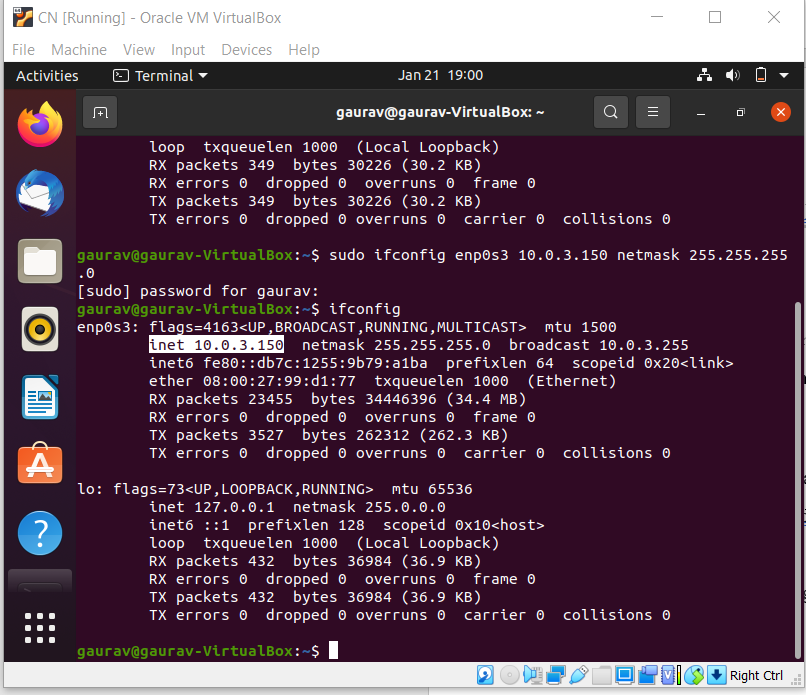
****

|  |  |  |
| --- | --- | --- |
| **INTERFACE** | **IPV4** | **MAC ADDRESS** |
| **enp0s3** | **10.0.2.15** | **08:00:27:99:d1:77** |
| **Wi-fi** | **192.168.0.10** | **4C:79:6E:33:D9:A7** |
|  |  |  |

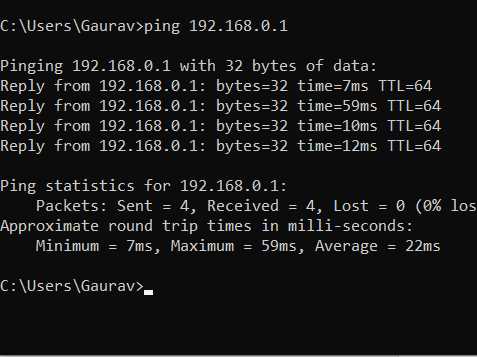
**Step 2:** To assign an IP address to an interface, use the following command. **sudo ifconfig**

**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**



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





* TTL =64
* Protocol used by ping =ICMP
* Time =22ms(AVERAGE)

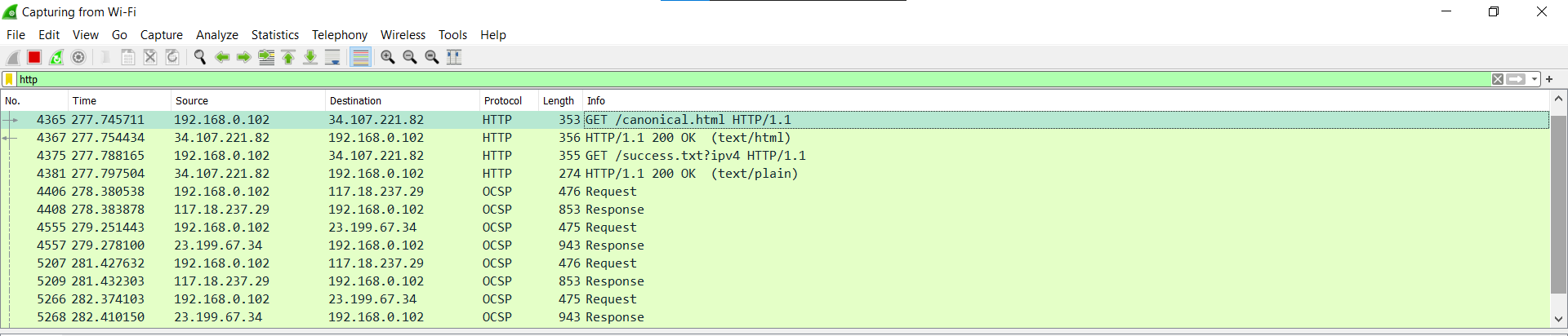
|  |  |  |
| --- | --- | --- |
| **Details** | **First Echo Request** | **First Echo Reply** |
| Frame Number | 21 | 22 |
| Source IP address | 192.168.0.102 | 192.168.0.1 |
| Destination IP address | 192.168.0.1 | 192.168.0.102 |
| ICMP Type Value | 8 | 0 |
| ICMP Code Value | 0 | 0 |
| Source Ethernet Address | 4c:79:6e:33:d9:a7 | 6c:5a:b0:a7:63:fe |
| Destination Ethernet Address | 6c:5a:b0:a7:63:fe | 4c:79:6e:33:d9:a7 |
| 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 | 4365 | 4367 |
| Source Port | 52541 | 80 |
| Destination Port | 80 | 64020 |
| Source IP address | 192.168.0.102 | 34.107.221.82 |
| Destination IP address | 34.107.221.82 | 192.168.0.102 |
| Source Ethernet Address | 4c:79:6e:33:d9:a7 | 6c:5a:b0:a7:63:fe |
| Destination Ethernet Address | 6c:5a:b0:a7:63:fe | 4c:79:6e:33:d9:a7 |

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

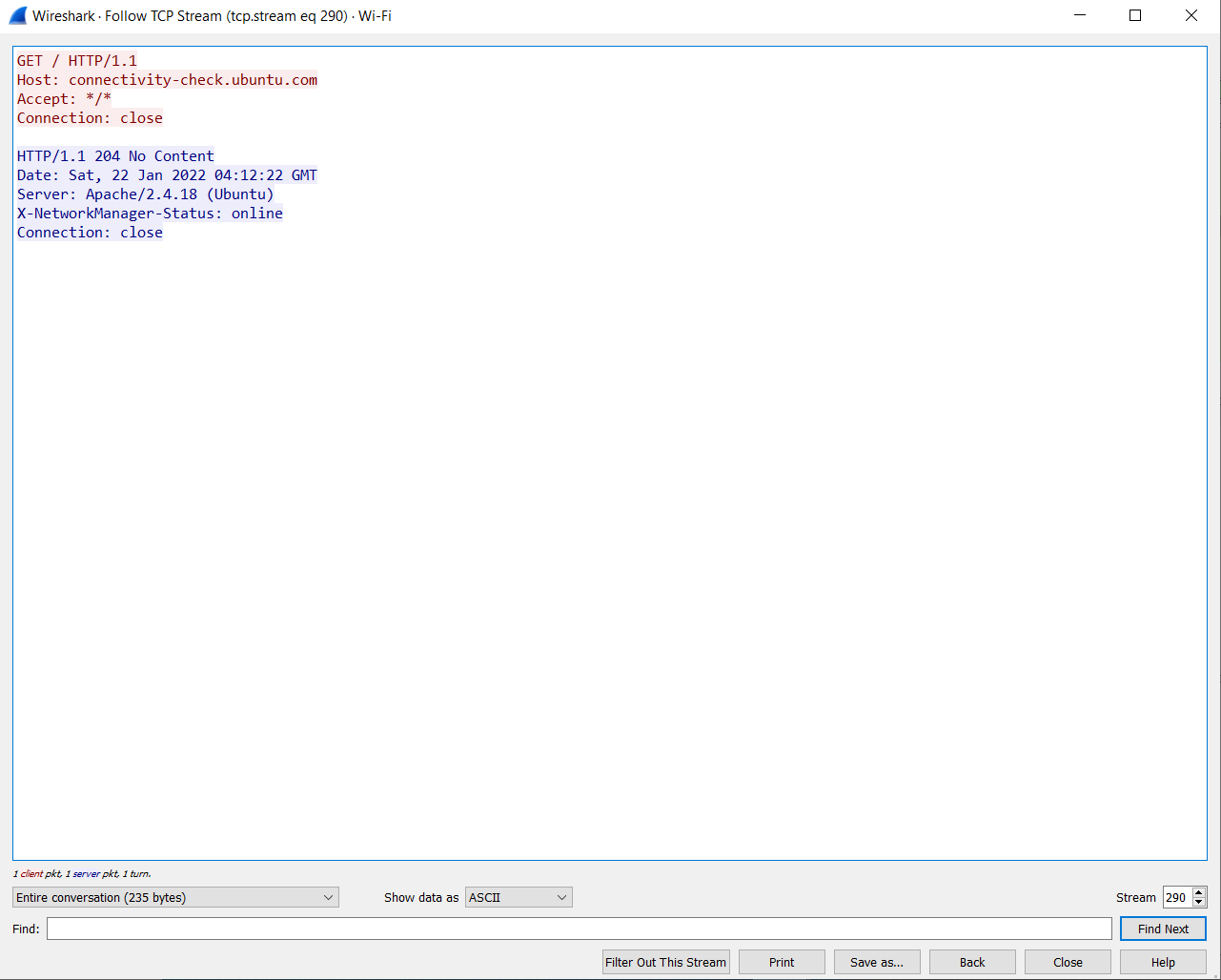
|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP Request** |  | **HTTP Response** |  |
| Get | /canonical.html HTTP/1.1 | Server | nginx |
| Host | detectportal.firefox.com | Content-Type | text/html |
| User-Agent | Mozilla/5.0 | Date | Fri,21 Jan 2022 03:34:40 GMT |
| Accept-Language | en-US | Location | - |
| Accept-Encoding | gzip,deflate | Content-Length | 356 |
| Connection | keep-alive | Connection | close |

**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

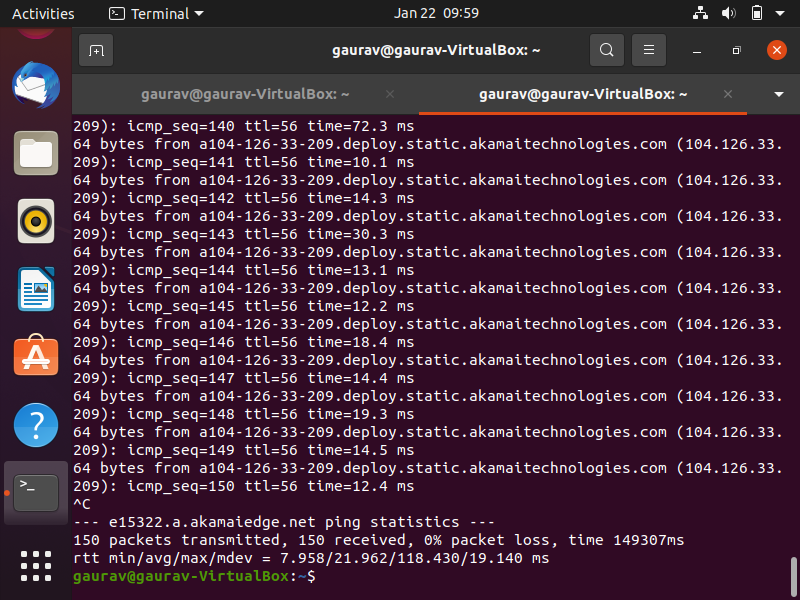
**Step 1:** Use the command **tcpdump -D** to see which interfaces are available for capture.

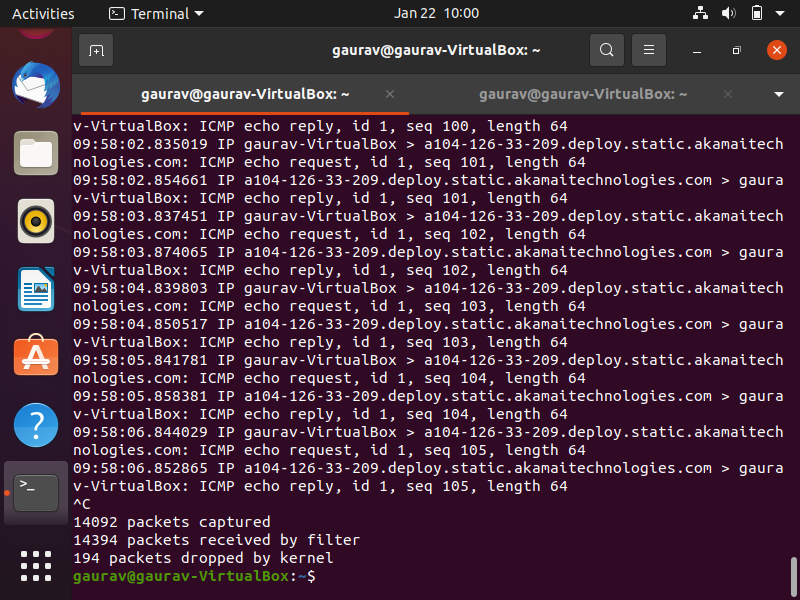
**sudo tcpdump -D**

**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](http://www.google.com/) in browser.



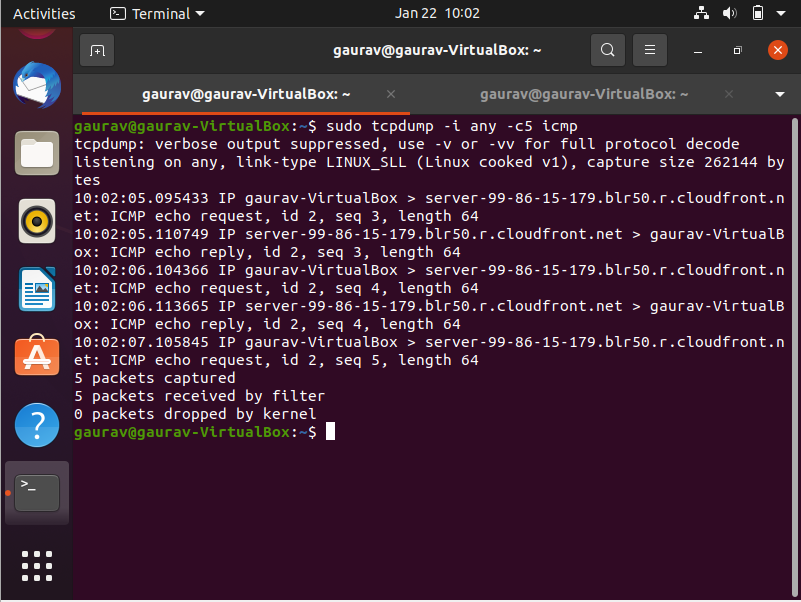


## Observation

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

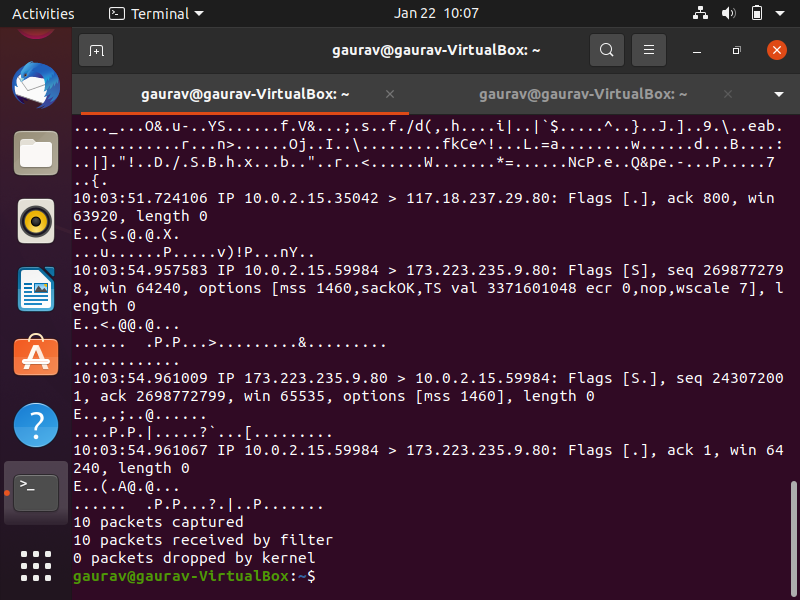
**Step 4:** To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using this command:

## sudo tcpdump -i any -c5 icmp

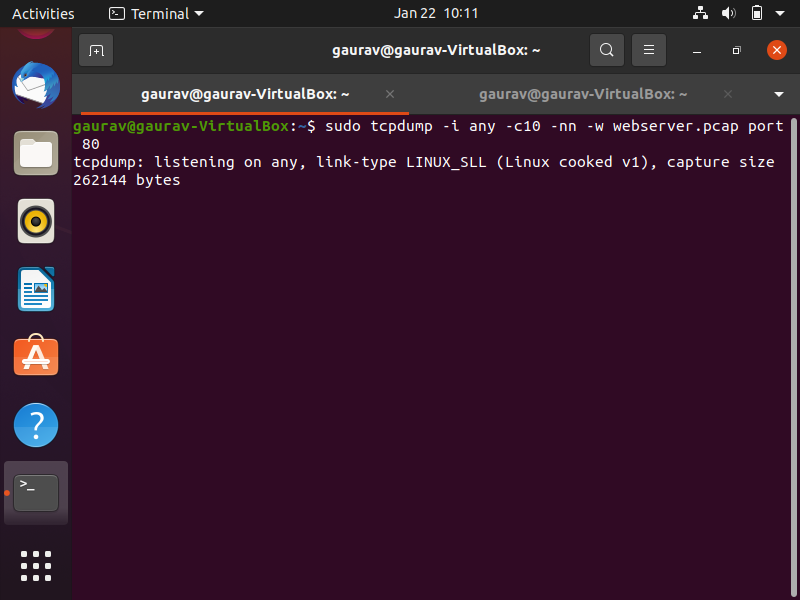


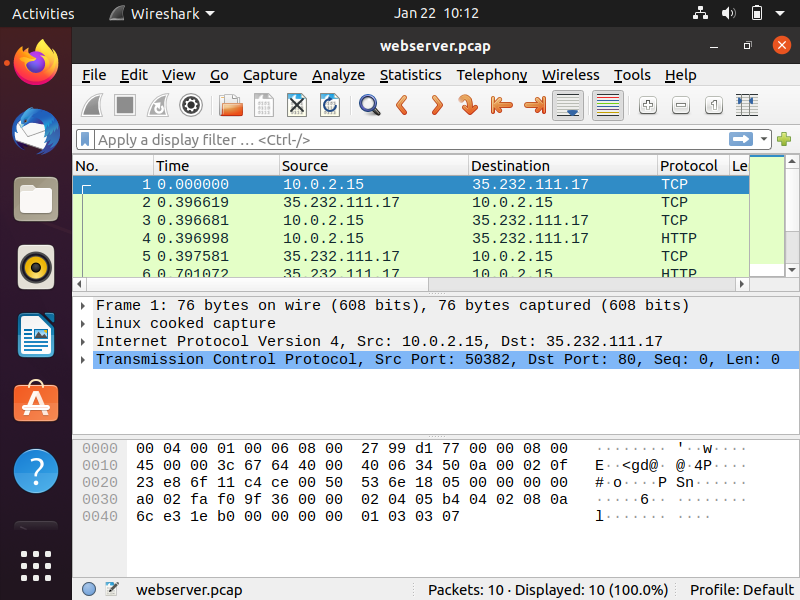
**Step 5:** Check the packet content. For example, inspect the HTTP content of a web request like this:

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



**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**



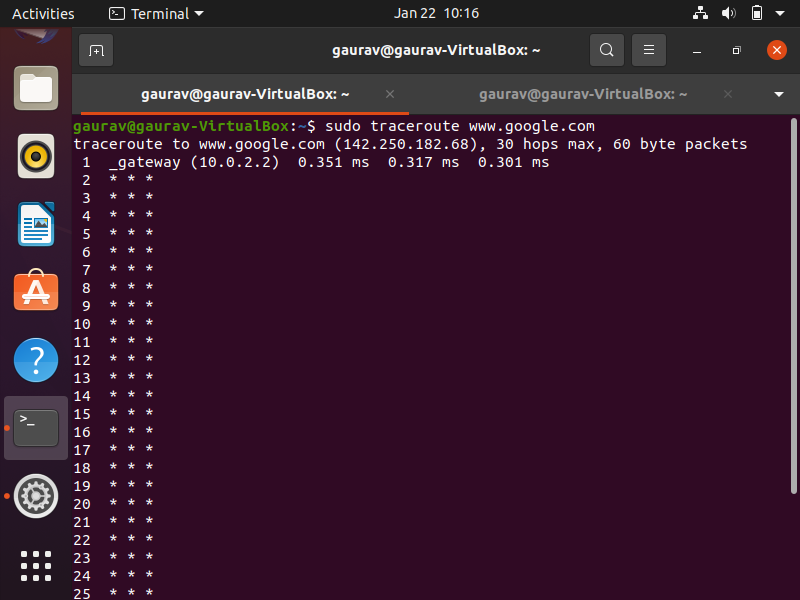


# Task 5: Perform Traceroute checks

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

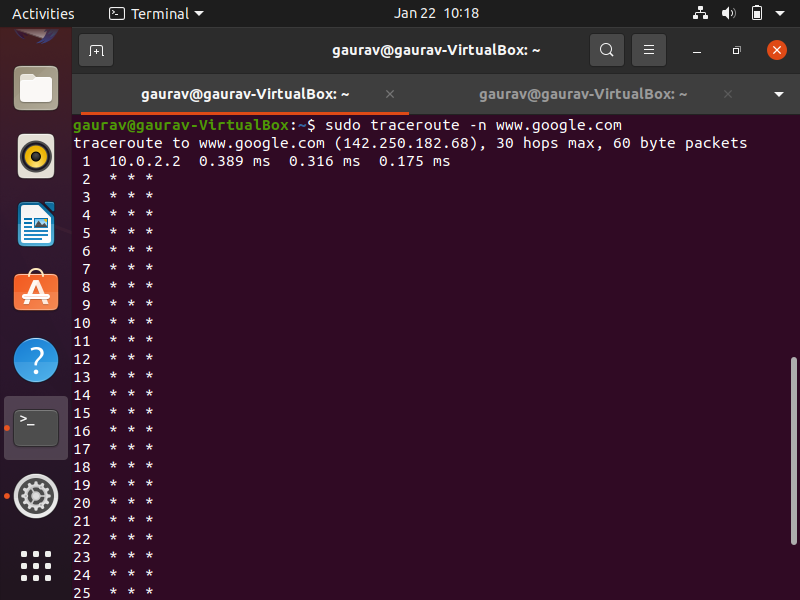
## sudo traceroute [www.google.com](http://www.google.com/)

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



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

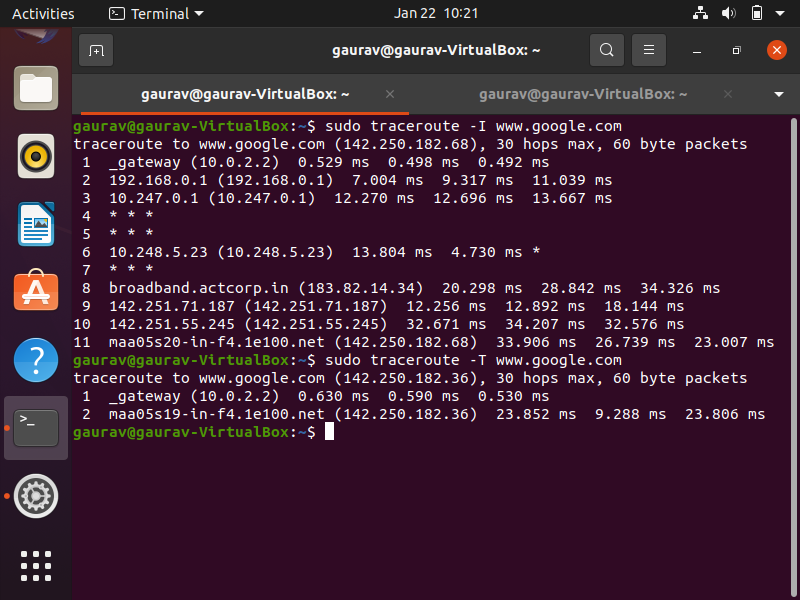
## sudo traceroute -n [www.google.com](http://www.google.com/)



**Step 4:** The -I option is necessary so that the traceroute uses [ICMP.](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) **sudo traceroute -I** [**www.google.com**](http://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**](http://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.

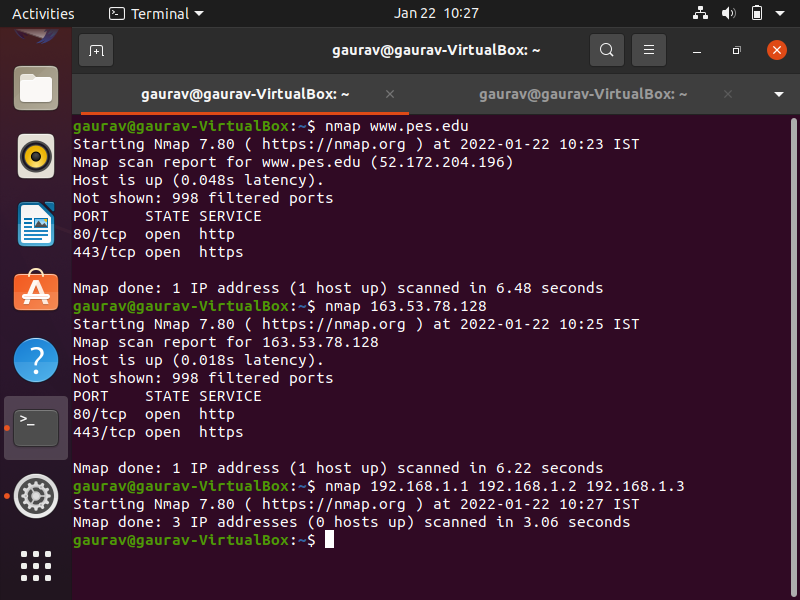
**nmap** [**www.pes.edu**](http://www.pes.edu/)

**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?
2. When was the HTML file that you are retrieving last modified at the server?
3. How to tell ping to exit after a specified number of ECHO\_REQUEST packets?
4. How will you identify remote host apps and OS?

Answers on above observations:

1)Browser is running a HTTP version 1.1 and server is also of the same version

2)After much observation,it has been observed that the Last modified date has been not specified.

3)By using the command ping -c <number > <website>

4)By using nmap