**Atul Anurag**

**PES2UG19CS075**

**COMPUTER NETWORKS LAB**

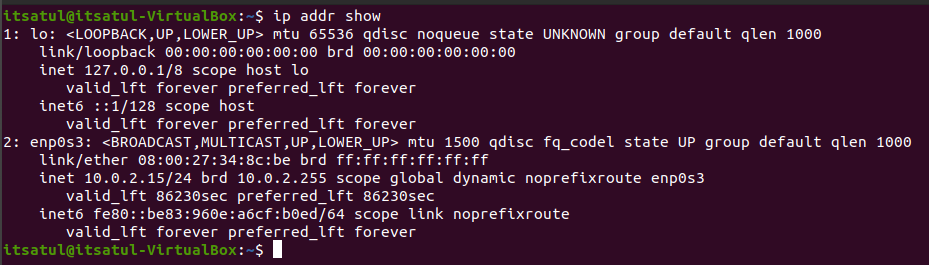
**Week #1**

**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 2. **Netcat**     * Establish communication between client and server    * Transfer files 3. **Tcpdump**    * Capture packets 4. **Ping**    * Test the connectivity between 2 systems 5. **Traceroute**    * Perform traceroute checks 6. **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** |
| lo | IPV4: 127.0.0.1/8  IPV6: 1/128 | 00:00:00:00:00:00 |
| enp0s3 | IPV4: 10.0.2.15/24  IPV6: fe80::be83:960e:a6cf:b0ed/64 | 08:00:27:34:8c:be |

**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**



**Step 3:** To activate / deactivate a network interface, type. **sudo ifconfig interface\_name down sudo ifconfig interface\_name up**

****

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

**ip neigh**

****

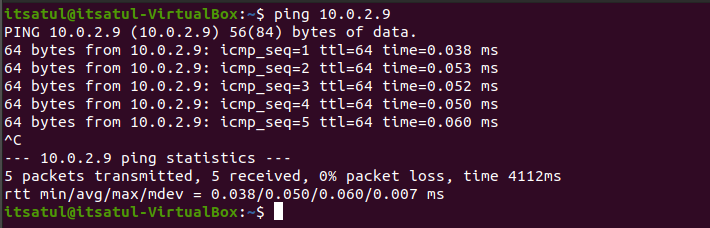
# Task 2: Ping PDU (Packet Data Units or Packets) Capture

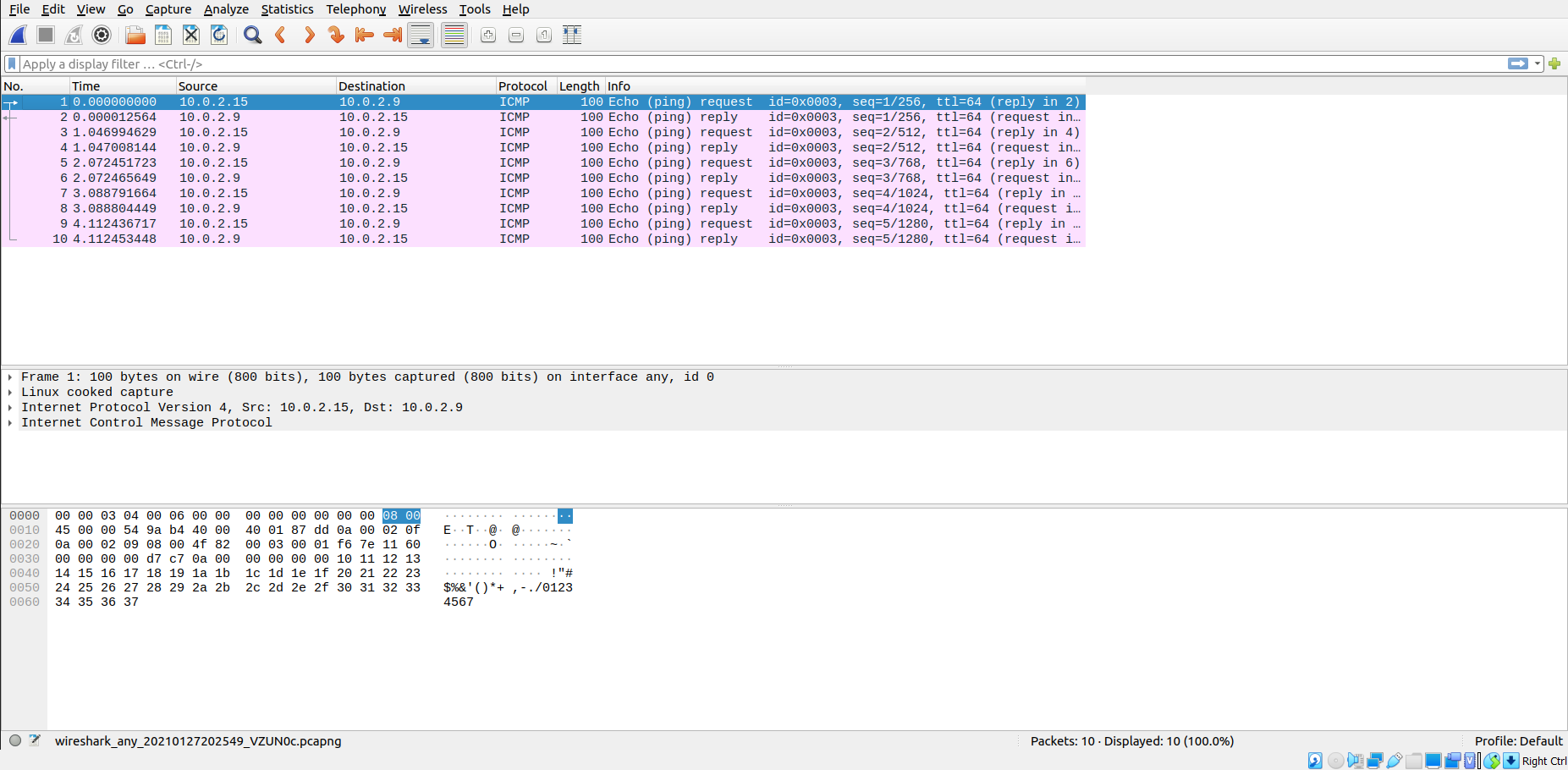
**Step 1:** Assign an IP address to the system (Host).

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**



****

## Observations to be made

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

* TTL - 64
* Protocol used by ping - ICMP
* Time – 4112 ms

**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 | 1 | 2 |
| Source IP address | 10.0.2.15 | 10.0.2.9 |
| Destination IP address | 10.0.2.9 | 10.0.2.15 |
| 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**

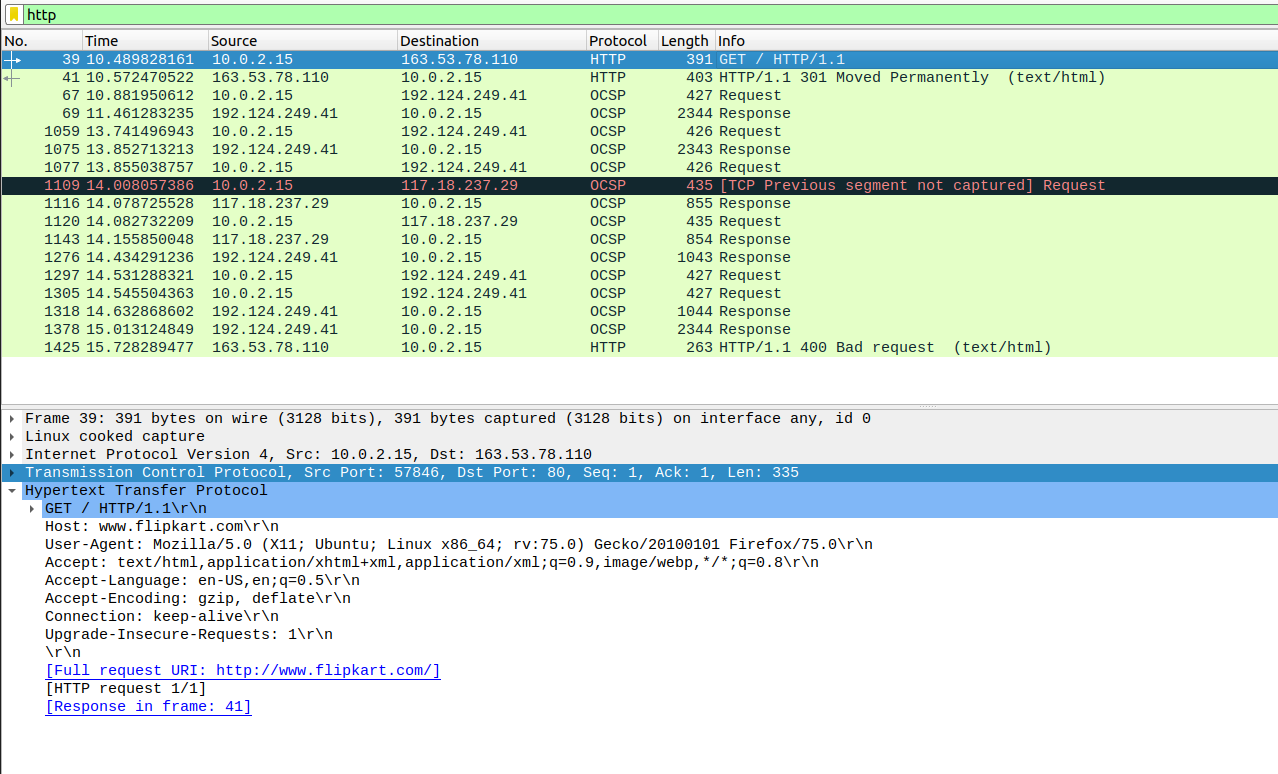
**Step 1:** Launch Wireshark and select ‘any’ interface. On the Filter toolbar, type-in ‘http’ and press enter

**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 | 39 | 41 |
| Source Port | 57846 | 80 |
| Destination Port | 80 | 57846 |
| 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 | PesCompu\_34:8c:be (08:00:27:34:8c:be) | RealtekU\_12:35:02 (52:54:00:12:35:02) |
| Destination Ethernet Address | RealtekU\_12:35:02 (52:54:00:12:35:02) | PesCompu\_34:8c:be (08:00:27:34:8c:be) |



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

|  |  |  |  |
| --- | --- | --- | --- |
| **HTTP Request** | | **HTTP Response** | |
| Get | GET/HTTP/1.1 | Server | nginx |
| Host | [www.flipkart.com](http://www.flipkart.com) | Content-Type | text/html |
| User-Agent | Mozilla/5.0 | Date | Wed, 27 Jan 2021 |
| Accept-Language | en-US | Location | https://www.flipkart.com/ |
| Accept-Encoding | gzip | Content-Length | 178 |
| Connection | keep-alive | Connection |  |

**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.

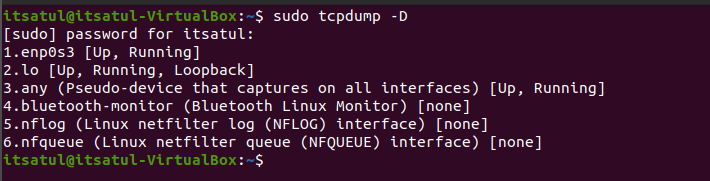
Step 2: Upon following a TCP stream, screenshot the whole window.



# 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.

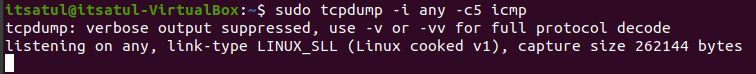
## 

## Observation

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

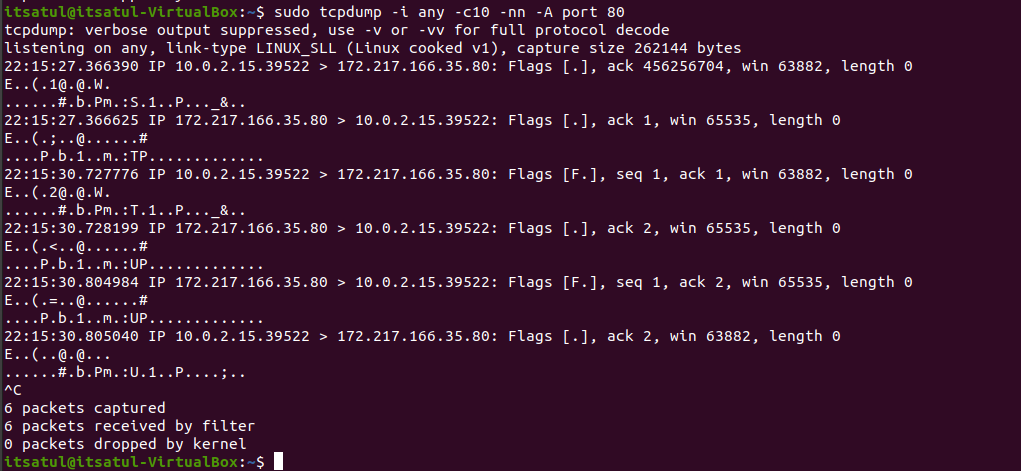
**Step 4:** To filter packets based on protocol, specifying the protocol in the command line. For example, capture ICMP packets only by using this command:

## sudo tcpdump -i any -c5 icmp

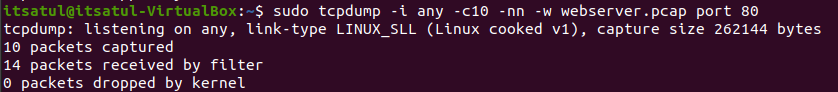


**Step 5:** Check the packet content. For example, inspect the HTTP content of a web request like this:

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



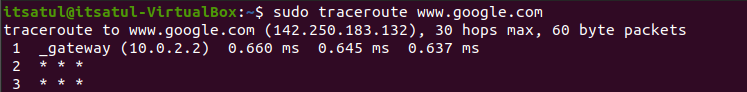
**Step 6:** To save packets to a file instead of displaying them on screen, use the option -w: **sudo tcpdump -i any -c10 -nn -w webserver.pcap port 80**



# Task 5: Perform Traceroute checks

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

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

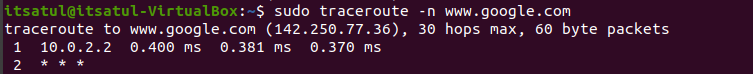


**Step 2:** Analyze destination address of google.com and no. of hops

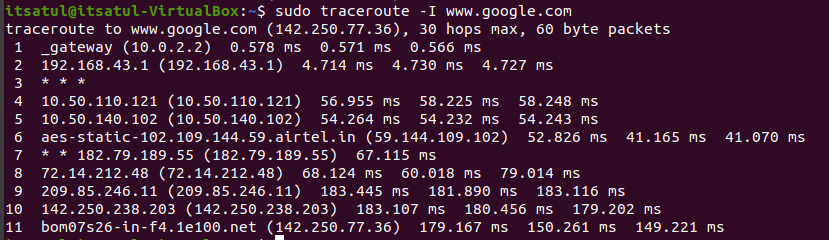
The destination address is **142.250.183.132** and there were **30 hops**.

**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/)

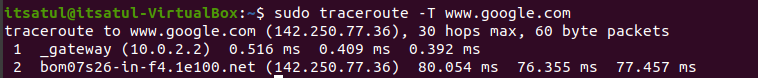


**Step 4:** The -I option is necessary so that the traceroute uses [ICMP.](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) **sudo traceroute -I** [**www.google.com**](http://www.google.com/)



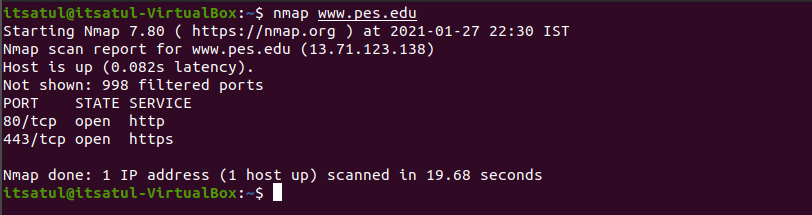
**Step 5:** By default, traceroute uses icmp (ping) packets. If you’d rather test a TCP connection to gather data more relevant to web server, you can use the -T flag.

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



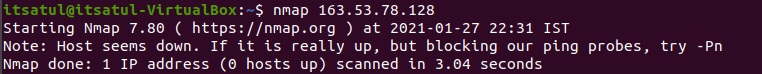
# Task 6: Explore an entire network for information (Nmap)

**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**

