

Computer Networks – LAB 1

Name : Nandan N

SRN : PES1UG21CS361

Section - F

Week #1

Study and understand the basic networking tools - Wireshark, Tcpdump, Ping, Traceroute.

Learn and Understand Network Tools

1. Wireshark

- Perform and analyze Ping PDU capture
- Examine HTTP packet capture
- Analyze HTTP packet capture using filter

2. Tcpdump

- Capture packets

3. Ping

- Test the connectivity between 2 systems

4. Traceroute

- Perform traceroute checks

5. Nmap

- Explore an entire network

IMPORTANT INSTRUCTIONS:

- This manual is written for Ubuntu Linux OS only. You can also execute these experiments on VirtualBox or VMWare platform.
- For few tasks, you may need to create 2 VMs for experimental setup.
- Perform **sudo apt-get update** before installing any tool or utility.
- Install any tool or utility using the command **sudo apt-get install name_of_the_tool**
- Take screenshots wherever necessary and upload it to Edmodo as a single PDF file. (Refer general guidelines for submission requirements).
- To define an IP address for your machine (e.g., Section – ‘a’ & Serial number is 1, then your IP address should be 10.0.1.1. Section – ‘h’ & Serial number is 23, then your IP address should be 10.0.8.23) – applicable only for relevant tasks (which doesn't require internet connectivity to execute the tasks).

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	
Windows IP Configuration	192.168.56.1(Preferred) – IPv4 fe80::6a81:4e2:9de5:e5a%16(Preferred) – IPv6	0A-00-27-00-00-10	
Wireless LAN adapter Wi-Fi:	10.20.207.187(Preferred) – IPv4 fe80::b410:14e2:4bf2:9325%19(Preferred) – IPv6	2C-6D-C1-6B-03-37	

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

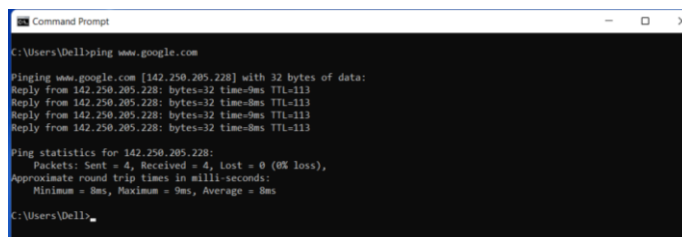
Task 2: Ping PDU (Packet Data Units or Packets) Capture

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**



```
C:\Users\ DELL>ping www.google.com

Pinging www.google.com [142.250.205.228] with 32 bytes of data:
Reply from 142.250.205.228: bytes=32 time=8ms TTL=113
Reply from 142.250.205.228: bytes=32 time=8ms TTL=113
Reply from 142.250.205.228: bytes=32 time=8ms TTL=113
Reply from 142.250.205.228: bytes=32 time=8ms TTL=113

Ping statistics for 142.250.205.228:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 8ms, Maximum = 9ms, Average = 8ms

C:\Users\ DELL>
```

Observations to be made

Step 4: Analyze the following in Terminal

- TTL
- Protocol used by ping
- Time

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	84947	84948
Source IP address	Source Address: 10.20.207.187	162.247.241.14
Destination IP address	Destination Address: 162.247.241.14	10.20.207.187
ICMP Type Value	8	0
ICMP Code Value	0	0
Source Ethernet Address	Source: IntelCor_6b:03:37 (2c:6d:c1:6b:03:37)	Source: HewlettP_d4:86:00 (14:58:d0:d4:86:00)
Destination Ethernet Address	Destination: HewlettP_d4:86:00 (14:58:d0:d4:86:00)	Destination: IntelCor_6b:03:37 (2c:6d:c1:6b:03:37)
Internet Protocol Version	Type: IPv4 (0x0800)	Type: IPv4 (0x0800)
Time To Live (TTL) Value	Time to Live: 128	Time to Live: 62

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

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	Frame 79945	79973
Source Port	Source Port: 50250	Source Port: 80

Destination Port	Destination Port: 80	Destination Port: 50250
Source IP address	Source Address: 10.20.207.187	Source Address: 34.104.35.123
Destination IP address	Destination Address: 34.104.35.123	Destination Address: 10.20.207.187
Source Ethernet Address	Source: IntelCor_6b:03:37 (2c:6d:c1:6b:03:37)	Source: HewlettP_d4:86:00 (14:58:d0:d4:86:00)
Destination Ethernet Address	Destination: HewlettP_d4:86:00 (14:58:d0:d4:86:00)	Destination: IntelCor_6b:03:37 (2c:6d:c1:6b:03:37)

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

HTTP Request		HTTP Response	
Get		Server	server: Google-Edge-Cache\r\n
Host	Host: edgedl.me.gvt1.com\r\n	Content-Type	content-type: application/octet-stream\r\n
User-Agent	User-Agent: Microsoft BITS/7.8\r\n	Date	date: Thu, 19 Jan 2023 20:46:09 GMT\r\n
Accept-Language	en-US	Location	
Accept-Encoding	Accept-Encoding: identity\r\n	Content-Length	content-length: 0\r\n
Connection	Connection: Keep-Alive\r\n	Connection	Connection: keep-alive\r\n

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

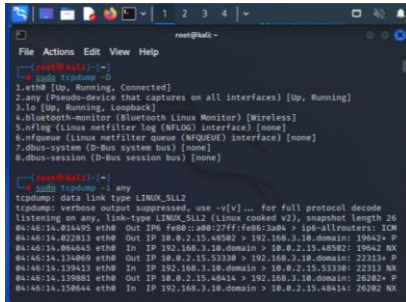
Step 1: Use the command **tcpdump -D** to see which interfaces are available for capture.

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.



```
root@kali:~# sudo tcpdump -i any
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
04:46:14.024495 eth0 Out IP6 fe80::a80:27ff:fe80:3a84 > ip6-allrouters: ICM
04:46:14.022813 eth0 Out IP 10.0.2.15.46502 > 192.168.3.10.domain: 19642v P
04:46:14.004045 eth0 In IP 192.168.3.10.domain > 10.0.2.15.46502: 15642 NX
04:46:14.134009 eth0 Out IP 10.0.2.15.53330 > 192.168.3.10.domain: 22313v P
04:46:14.129423 eth0 In IP 192.168.3.10.domain > 10.0.2.15.53330: 22313 NX
04:46:14.129881 eth0 Out IP 10.0.2.15.48414 > 192.168.3.10.domain: 26202v P
04:46:14.150644 eth0 In IP 192.168.3.10.domain > 10.0.2.15.48414: 26202 NX
```

Observation

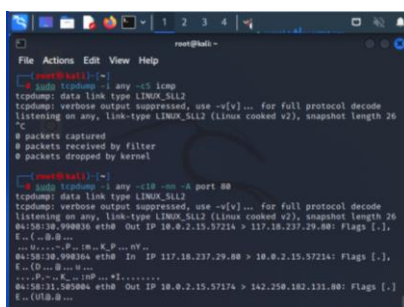
Step 3: Understand the output format.

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

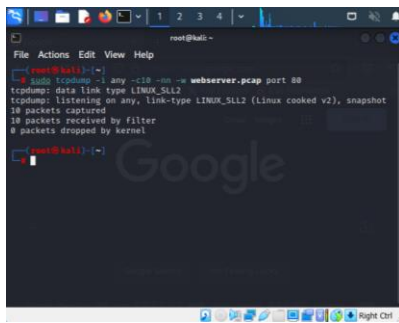


```
root@kali:~# sudo tcpdump -i any -c5 icmp
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
04:46:14.024495 eth0 Out IP6 fe80::a80:27ff:fe80:3a84 > ip6-allrouters: ICM
04:46:14.022813 eth0 Out IP 10.0.2.15.46502 > 192.168.3.10.domain: 19642v P
04:46:14.004045 eth0 In IP 192.168.3.10.domain > 10.0.2.15.46502: 15642 NX
04:46:14.134009 eth0 Out IP 10.0.2.15.53330 > 192.168.3.10.domain: 22313v P
04:46:14.129423 eth0 In IP 192.168.3.10.domain > 10.0.2.15.53330: 22313 NX
04:46:14.129881 eth0 Out IP 10.0.2.15.48414 > 192.168.3.10.domain: 26202v P
04:46:14.150644 eth0 In IP 192.168.3.10.domain > 10.0.2.15.48414: 26202 NX

root@kali:~# sudo tcpdump -i any -c10 -nn -A port 80
tcpdump: data link type LINUX_SLL2
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 26
04:46:14.000000 eth0 Out IP 10.0.2.15.57214 > 117.18.237.29.80: Flags [..],
  E..(..B..0..
...N....P...M...K...F...
04:46:14.000000 eth0 In IP 117.18.237.29.80 > 10.0.2.15.57214: Flags [..],
  E..(D...B...N...
...P...K...M...F...
04:46:14.000000 eth0 Out IP 10.0.2.15.57174 > 142.250.182.131.80: Flags [..],
  E..(U..B..0...
```

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

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

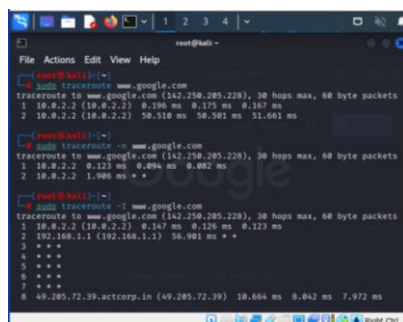
sudo traceroute -n www.google.com

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



```

root@kali ~
File Actions Edit View Help
root@kali ~# traceroute -i www.google.com
traceroute to www.google.com (142.250.205.228), 30 hops max, 60 byte packets
 0 10.0.2.2 (10.0.2.2) 0.147 ms 0.128 ms 0.123 ms
 1 192.168.1.1 (192.168.1.1) 56.981 ms *
 2 *
 3 *
 4 *
 5 *
 6 *
 7 *
 8 69.205.72.39.actcorp.in (69.205.72.39) 10.664 ms 8.862 ms 7.972 ms
 9 72.14.202.242 (72.14.202.242) 8.784 ms 8.869 ms 11.877 ms
10 72.14.232.71 (72.14.232.71) 11.653 ms 11.627 ms 11.595 ms
11 142.251.68.185 (142.251.68.185) 11.561 ms 12.192 ms 12.146 ms
12 maad5128-in-f4.1e100.net (142.250.205.228) 12.297 ms 13.821 ms 13.295 ms
root@kali ~#
root@kali ~# traceroute -i www.google.com
traceroute to www.google.com (142.250.205.228), 30 hops max, 60 byte packets
 0 10.0.2.2 (10.0.2.2) 0.204 ms 0.161 ms 0.162 ms
 1 maad5128-in-f4.1e100.net (142.250.205.228) 56.187 ms 56.155 ms 56.533 ms
root@kali ~#

```

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

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

```

root@kali ~
File Actions Edit View Help
root@kali ~# nmap 163.53.78.128
Nmap done: 1 IP address (1 host up) scanned in 7.36 seconds
root@kali ~#
root@kali ~# nmap 163.53.78.128
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-20 08:26 EST
Nmap scan report for 163.53.78.128
Host is up (0.610s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
Nmap done: 1 IP address (1 host up) scanned in 4.98 seconds
root@kali ~#
root@kali ~# nmap 192.168.1.1 192.168.1.2 192.168.1.3
Starting Nmap 7.92 ( https://nmap.org ) at 2023-01-20 08:28 EST
Warning: 192.168.1.1 giving up on port because retransmission cap hit (10).
root@kali ~#

```

Questions on above observations:

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

HTTP/1.1 200 OK\r\n version 1.1

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

LAST-MODIFIED: Fri, 01 Jan 2016 00:00:10 GMT\r\n

3) How to tell ping to exit after a specified number of ECHO_REQUEST packets?

Use 'ping -c' command

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

Use # nmap -O -v localhost

nmap -O -v server.ip.address

