NATIONAL UNIVERSITY OF SCIENCE AND TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

INFORMATION SECURITY LAB

Name	Ayesha Imran
Class	CS-A
Lab	02
Course	Information Security
Date	30-September-25
Submitted To	Lec. Attiya Ashraf

IN LAB TASKS

Installation:

Tcpdump is generally pre-installed on most Linux systems.

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~/Desktop$ sudo apt update &&
sudo apt install tcpdump
[sudo] password for ayesha-imran:
Get:1 http://security.ubuntu.com/ubuntu noble-security InRelease [126 kB]
Hit:2 http://archive.ubuntu.com/ubuntu noble InRelease
```

```
ld by process 2009 (unattended-upgr)
Reading package lists... 0%
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
tcpdump is already the newest version (4.99.4-3ubuntu4).
tcpdump set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 472 not upgraded.
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$
```

Activity 1: Basic Packet Capture

1. Objective: Capture all network traffic on a specific network interface. (A network interface is a physical or virtual connection point that enables a device to connect to a network and communicate with other devices.Each network interface has a unique identifier, typically a MAC address and one or more IP addresses for networking.)

2. Steps:

Identify available network interfaces: ip a

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group defaul
t qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
       valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP g
roup default qlen 1000
    link/ether 00:0c:29:b8:fe:52 brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.17.133/24 brd 192.168.17.255 scope global dynamic noprefixroute
 ens33
       valid_lft 1779sec preferred_lft 1779sec
    inet6 fe80::20c:29ff:feb8:fe52/64 scope link
       valid_lft forever preferred_lft forever
```

- **Capture traffic on the identified interface: sudo tcpdump -i ens33**

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:09:15.488789 IP6 fe80::a476:684d:8b1a:1346 > ff02::1:ff9c:1279: ICMP6, neighb
or solicitation, who has fe80::eef4:dd27:3e9c:1279, length 32
14:09:15.591991 IP ayesha-imran-VMware-Virtual-Platform.43303 > _gateway.domain:
 2995+ [1au] PTR? 9.7.2.1.c.9.f.f.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.0.f.
f.ip6.arpa. (101)
14:09:15.761525 IP _gateway.domain > ayesha-imran-VMware-Virtual-Platform.43303:
 2995 NXDomain 0/1/1 (165)
14:09:15.768464 IP ayesha-imran-VMware-Virtual-Platform.52568 > _gateway.domain:
 11105+ [1au] PTR? 2.17.168.192.in-addr.arpa. (54)
14:09:15.823131 IP ayesha-imran-VMware-Virtual-Platform.33191 > _gateway.domain:
```

- **Tcpdump will display network packets in real time.**

```
14:12:52.382279 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:12:52.997355 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:12:53.997518 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:12:58.411188 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:12:58.996609 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:12:59.996447 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:13:01.427313 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:13:01.996768 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:13:02.996207 ARP, Request who-has _gateway tell 192.168.17.1, length 46
14:13:19.496096 IP6 fe80::a476:684d:8b1a:1346 > ff02::1:ff9c:1279: ICMP6, neighb
or solicitation, who has fe80::eef4:dd27:3e9c:1279, length 32
```

Activity 1: Basic Packet Capture Analysis:

After capturing all network traffic on my interface ens33**,** I observed that Tcpdump displayed a continuous stream of packets in real time. Each packet showed the source and destination IP addresses**,** ports**,** protocols, and packet size**.**I noticed a variety of packet types, including TCP**,** UDP**,** and ICMP. Some IP addresses were unfamiliar, which could indicate devices or services communicating on the network. Capturing all traffic gave me an overview of the network activity and helped me understand which services are most active on my system.

Activity 2: Filtering Traffic by Protocol:

1. **Objective: Capture traffic for specific protocols to focus on packets of interest. Steps:**

- **Capture only HTTP traffic (typically on port 80): sudo tcpdump –I ens33 port 80**

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33 port 80
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

```
lags [.], ack 90, win 64239, length 0
^C
55 packets captured
55 packets received by filter
0 packets dropped by kernel
```

- **Capture ICMP traffic (e.g., ping requests): sudo tcpdump –i ens33 icmp**

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes

14:31:46.992939 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 1, length 64
14:31:48.045569 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 2, length 64
14:31:49.030372 IP 192.168.137.127 > 192.168.17.133: ICMP host 8.8.0.8 unreachable, length 92
14:31:49.045745 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 3, length 64
14:31:50.093819 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 4, length 64
14:31:51.117511 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 5, length 64
14:31:52.029402 IP 192.168.137.127 > 192.168.17.133: ICMP host 8.8.0.8 unreachable, length 92
14:31:52.118137 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 6, length 64
14:31:53.165421 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 7, length 64
14:32:07.501575 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 21, length 64
14:32:08.525881 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 22, length 64
14:32:17.741634 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 31, length 64
14:32:18.765482 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 32, length 64
14:32:19.789719 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 33, length 64
14:32:20.813474 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 34, length 64
14:32:21.837860 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 35, length 64
14:32:22.861650 IP 192.168.17.133 > 8.8.0.8: ICMP echo request, id 4950, seq 36, length 64
```

```
ckets captured
ackets received by filter
21 packets dropped by kernel
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$
```

Trash

- **Capture DNS queries (typically on port 53):**

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33 port 53
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:37:45.823552 IP ayesha-imran-VMware-Virtual-Platform.59707 > _gateway.domain: 9173+ [1au]
AAAA? connectivity-check.ubuntu.com. (58)
14:37:45.839691 IP ayesha-imran-VMware-Virtual-Platform.58586 > _gateway.domain: 1085+ [1au]
PTR? 2.17.168.192.in-addr.arpa. (54)
14:37:45.842358 IP _gateway.domain > ayesha-imran-VMware-Virtual-Platform.59707: 9173 12/0/1
AAAA 2620:2d:4000:1::96, AAAA 2620:2d:4000:1::98, AAAA 2620:2d:4000:1::22, AAAA 2620:2d:4002:
1::196, AAAA 2001:67c:1562::23, AAAA 2620:2d:4002:1::197, AAAA 2620:2d:4000:1::2b, AAAA 2620:
2d:4000:1::23, AAAA 2001:67c:1562::24, AAAA 2620:2d:4002:1::198, AAAA 2620:2d:4000:1::2a, AAA
A 2620:2d:4000:1::97 (394)
14:37:45.889913 IP _gateway.domain > ayesha-imran-VMware-Virtual-Platform.58586: 1085 NXDomai
n 0/0/1 (54)
14:37:45.891025 IP ayesha-imran-VMware-Virtual-Platform.38995 > _gateway.domain: 36970+ [1au]
 PTR? 133.17.168.192.in-addr.arpa. (56)
14:37:45.944217 IP _gateway.domain > ayesha-imran-VMware-Virtual-Platform.38995: 36970 NXDoma
in 0/0/1 (56)
```

3. Expected Outcome:

- **The output should display only packets related to the specified protocol, making it easier to analyze specific network activities.**

Activity 2: Filtering Traffic by Protocol Analysis:

- **HTTP traffic (port 80):** Only web-related requests were shown. I could see HTTP headers, GET requests, and responses from servers. This made it easy to focus on web browsing activity without other unrelated packets.
- **ICMP traffic:** Only ping requests and replies were captured. This helped me observe network connectivity checks between devices.
- **DNS traffic (port 53):** Only DNS query and response packets were displayed. This showed which domain names my system was resolving and helped identify active network lookups.

By filtering traffic by protocol, I could clearly see how different types of network communication work and focus on analyzing specific interactions.

Activity 3: Filtering Traffic by IP Address:

1. Objective: Capture traffic to or from a specific IP address to narrow down analysis.

2. Steps:

- **Capture packets from a specific IP address:**

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33 src 192.168.17.133
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:52:00.397633 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 1186, length 64
14:52:00.439518 IP ayesha-imran-VMware-Virtual-Platform.51670 > _gateway.domain: 40982+ [1au]
 PTR? 8.0.8.8.in-addr.arpa. (49)
14:52:01.421884 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 1187, length 64
14:52:02.445881 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 1188, length 64
14:52:02.637598 ARP, Request who-has _gateway tell ayesha-imran-VMware-Virtual-Platform, leng
th 28
14:52:03.469845 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 1189, length 64
14:52:04.493852 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
```

```
239 packets captured
247 packets received by filter
7 packets dropped by kernel
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$
```

- **Capture packets to a specific IP address:**

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33 dst 8.8.0.8
[sudo] password for ayesha-imran:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
15:20:16.205706 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 2842, length 64
15:20:17.229583 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 2843, length 64
15:20:18.253438 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 2844, length 64
15:20:19.277529 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 2845, length 64
15:20:20.301505 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 2846, length 64
15:20:21.325638 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 2847, length 64
15:20:22.349798 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
, seq 2848, length 64
15:20:23.373691 IP ayesha-imran-VMware-Virtual-Platform > 8.8.0.8: ICMP echo request, id 4950
```

- **Capture packets to and from a specific IP address:**

3. Expected Outcome:

- **Tcpdump should only capture traffic that matches the IP filter criteria.**

Activity 3: Filtering Traffic by IP Address Analysis

- **Packets from a specific IP:** I could see only the packets originating from that device. This is useful to track a particular host's activity.

- **Packets to a specific IP:** Captured only the packets being sent to a device. This helps analyze incoming traffic and detect potential threats.
- **Packets to and from a specific IP:** Allowed me to track the full communication between my system and a target IP, helping monitor conversations between devices.

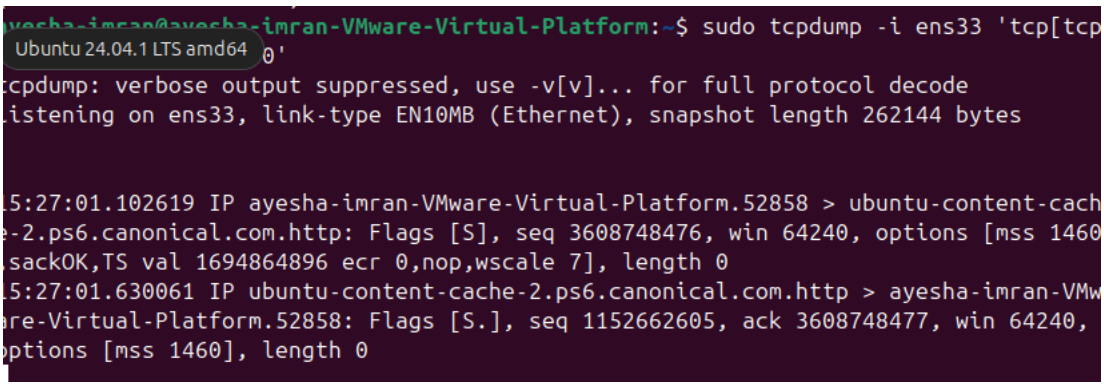
Filtering by IP helps **narrow down network analysis** and is particularly useful for troubleshooting issues or monitoring specific devices on the network.

Activity 4: Advanced Filtering

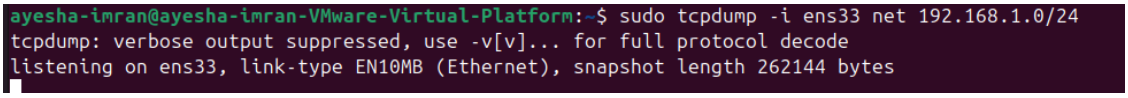
1. Objective: Use advanced Tcpdump filters for more refined analysis.

2. Steps:

- **Capture TCP packets with SYN flag set (useful for identifying connection attempts):sudo tcpdump -i <interface> 'tcp[tcpflags] & tcp-syn != 0'**
- captures only **SYN packets**, useful for seeing new connection attempts.



- **Capture traffic within a specific subnet: sudo tcpdump -i <interface> net 192.168.1.0/24**
- captures packets that are **closing connections**.



- **Capture TCP packets over a certain packet size (e.g., greater than 1000 bytes):sudo tcpdump -i <interface> 'tcp and greater 1000**
- So basically, it **filters out small TCP packets** (like ACKs or control messages) and shows **only larger data packets**, which usually carry **more meaningful payload**, like file transfers, HTTP responses, or large messages.

```
ayesha-imran@ayesha-imran-VMware-Virtual-Platform:~$ sudo tcpdump -i ens33 'tcp and greater 1000'
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on ens33, link-type EN10MB (Ethernet), snapshot length 262144 bytes
```

Activity 4: Advanced Filtering Analysis

- **TCP packets with SYN(Synchronize)flag:** Capturing only SYN packets showed the connection attempts being made to and from my system. This can indicate which services are being accessed or scanned.
- **Traffic within a specific subnet:** Displayed only devices within **192.168.1.0/24**, helping me focus on local network activity.
- **TCP packets over 1000 bytes:** Showed larger packets, which could indicate file transfers or significant data exchanges.

Advanced filters make network analysis more precise by **targeting specific traffic types, sizes, or flags**. They are essential for identifying unusual activity, detecting potential attacks, and monitoring specific network segments efficiently.

2. Explanation of Findings:

Provide brief explanations for the traffic patterns observed.

Describe any notable packets or activities, including potential security concerns.

Use of Advanced Filters in Tcpdump

Advanced filters help focus only on important network traffic. This makes it easier to spot problems or security issues.

- **Flag filters (like SYN, ACK(Acknowledgment), FIN(Finish)):** Used to check how connections are being made. For example, many SYN packets without replies may show a possible attack or scan.
- **Size filters (packet length):** Help find unusual traffic. Very large packets might mean data is being stolen, while many very small packets could mean scanning or a DoS attack.
- **Subnet filters (like 192.168.1.0/24):** Let you look only at traffic from a specific part of the network. This is useful to check one branch, one department, or to see if outsiders are trying to access the network.

Why it matters: Filtering reduces extra noise, makes it easier to notice suspicious activity, and helps analysts quickly find security issues.

3. Reflection on Tcpdump Usage:

Reflect on how Tcpdump could be used in real-world scenarios, especially for network troubleshooting and security monitoring.

3. Reflection on Tcpdump Usage:

Tcpdump is a powerful tool that allows network administrators to capture and analyze network traffic in real time. In real-world scenarios, it is very useful for troubleshooting network issues, such as identifying slow connections, dropped packets, or misconfigured devices. It is also valuable for security monitoring, helping detect suspicious activities like unauthorized access, malware communication, or DNS attacks. By using filters (like specific ports, protocols, or IP addresses), Tcpdump allows professionals to focus on the most relevant traffic, making problem-solving faster and more efficient.