TCPDump Analysis Report

1. Objective

The objective of this task was to understand and document basic TCPDump usage for network packet capture, including practical testing of packet exchanges using Netcat between two systems (Linux and Windows). This report outlines the steps taken, commands used, and the results observed during the packet capture and analysis process.

2. Tools Used

- **TCPDump** – Command-line packet analyzer for Linux.
- **Netcat/Ncat** – For generating simple TCP connections.
- **Operating Systems** – Kali Linux (ARM64) and Windows 10.

3. Initial Setup and Interface Detection

To verify available network interfaces for capture:

tcpdump -D

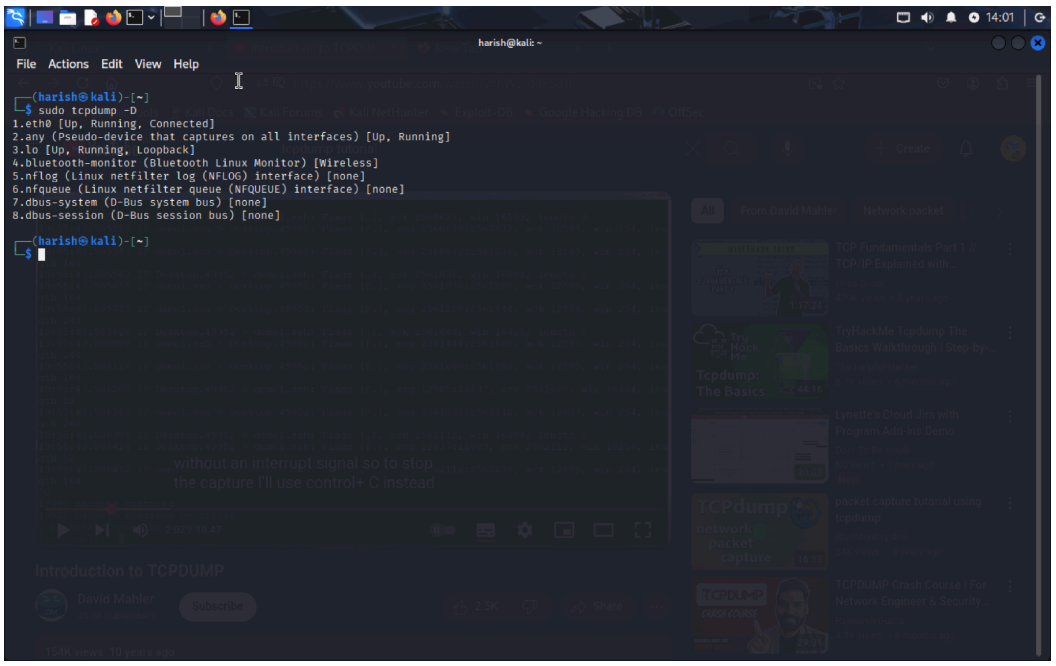


Fig: different network interfaces

This lists all interfaces that can be used with TCPDump.

4. First Capture Attempt – Any Interface

Command used:

tcpdump -i any

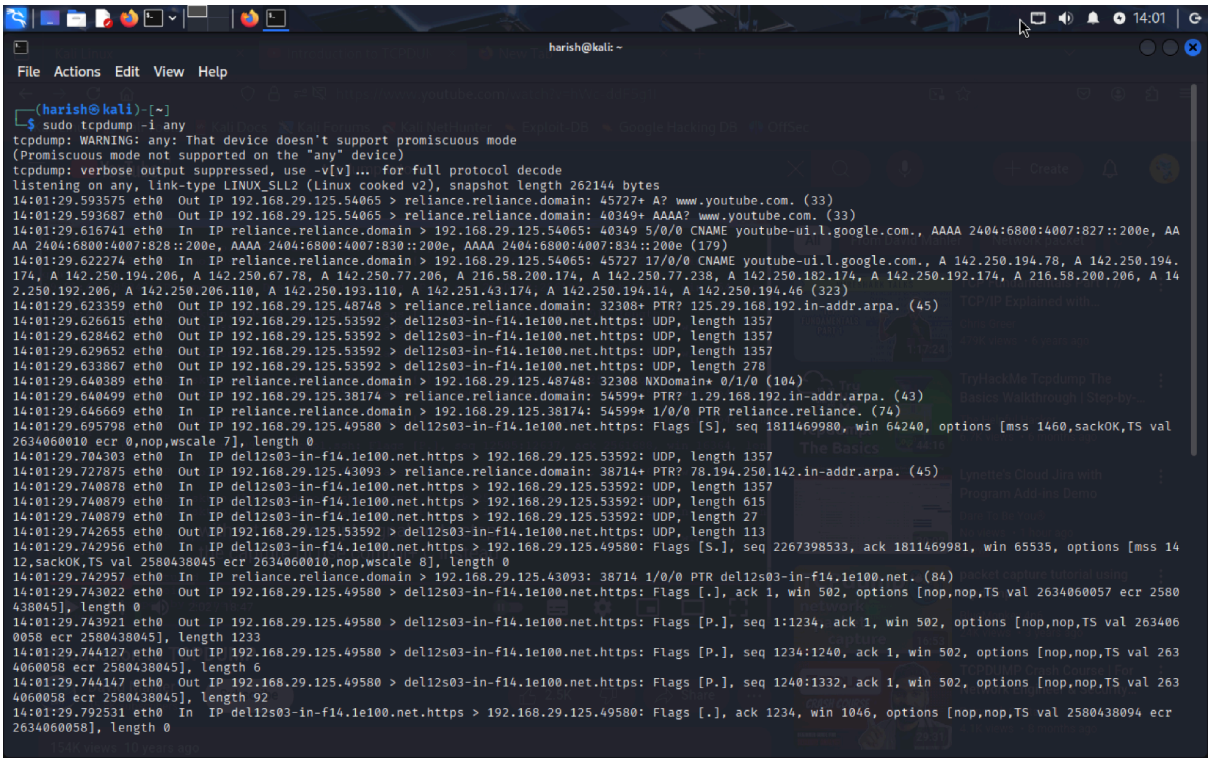


Fig: First Capture

- This command listens on all available interfaces.
- The capture was manually stopped using Ctrl + C as no packet count (-c) was set.
- Purpose: To observe general traffic flowing through any interface.
- **Observation**: This captured all packets on the system until interrupted.

5. Targeted Packet Capture – eth0 Interface

Command used:sudo tcpdump -i eth0 -c 5

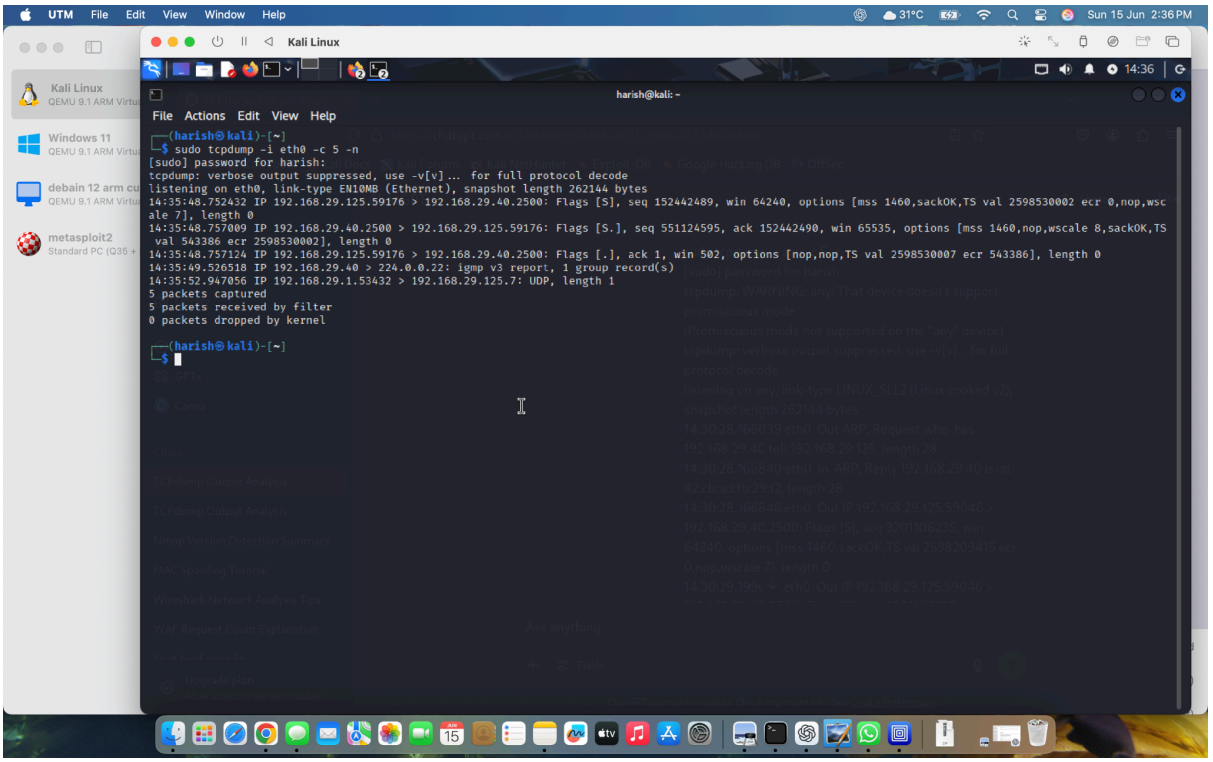


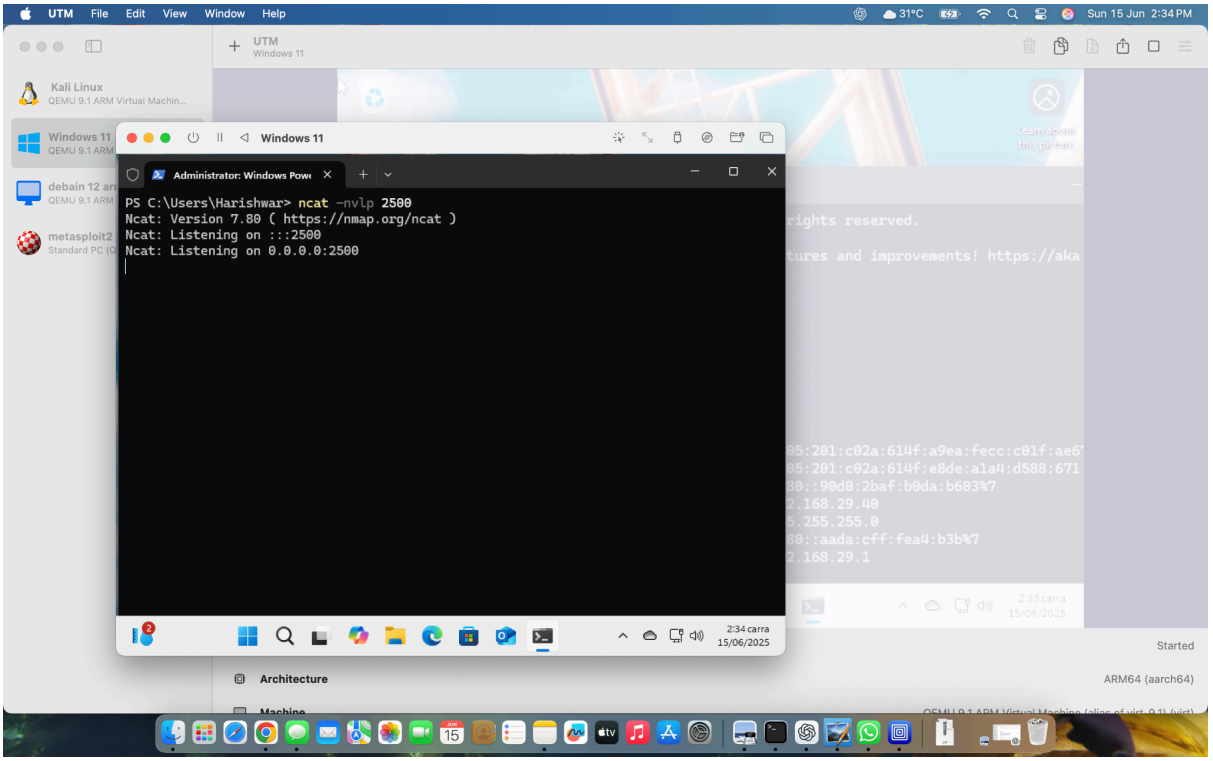
Fig: using eth0 (specified no.of packets)

- This limited the capture to the first **5 packets** on the eth0 interface.
- **Purpose**: To get a quick, focused snapshot of live traffic.
- **Result**: Successfully captured 5 packets for inspection.

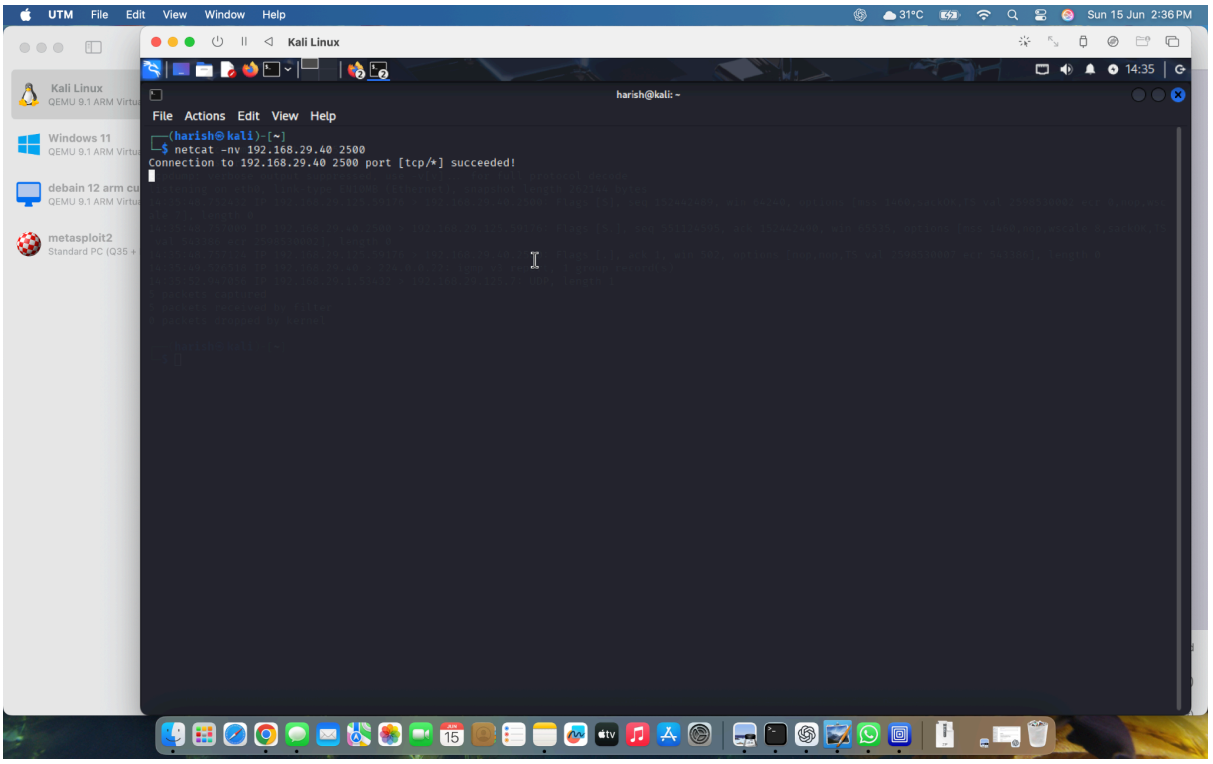
6. Netcat Packet Exchange Testing (Port 2500)

To simulate network communication between Linux and Windows, **Netcat** was used:

- **On Windows (Listener):** ncat -nvlp 2500



- **On Linux (Client):**netcat <Windows_IP> 2500



TCPDump Command: sudo tcpdump -i any -c 5 tcp

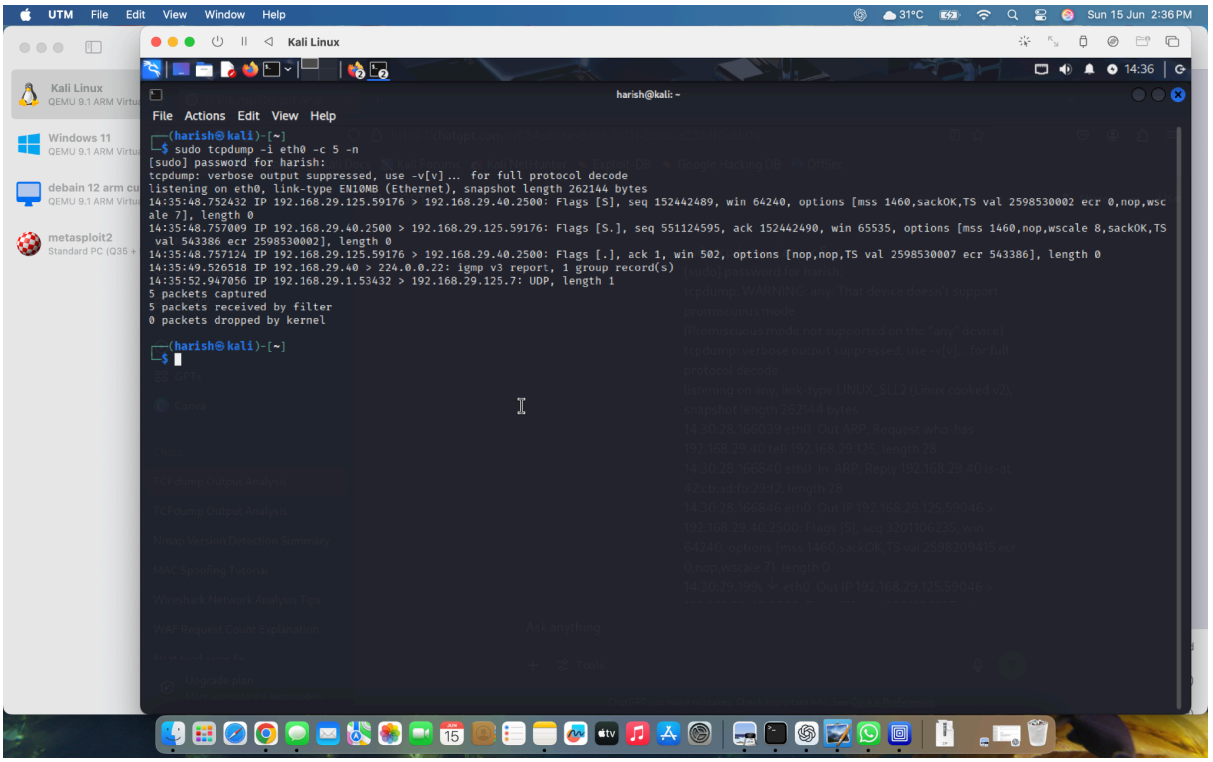


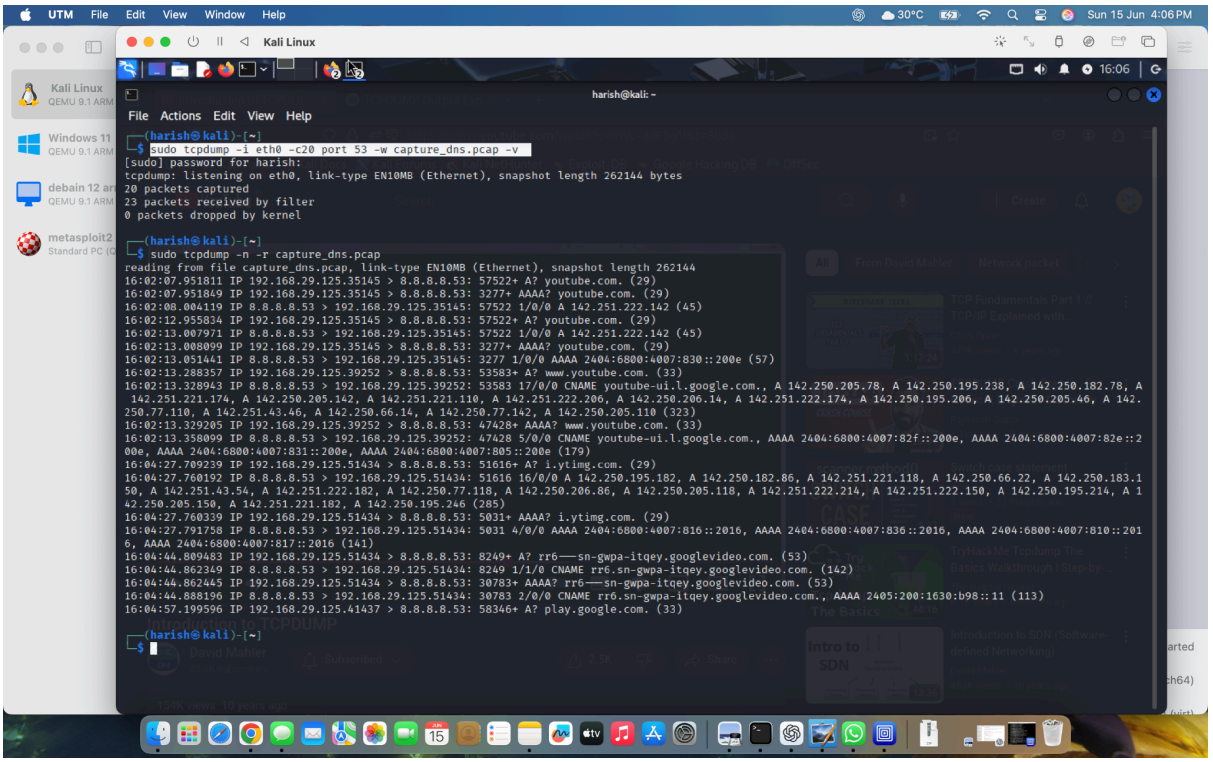
Fig: Packets Captured

- **Goal**: To observe a **3-way handshake (SYN, SYN-ACK, ACK)** as part of the TCP connection on port **2500**.

- **Result**: Successfully observed the handshake in captured packets. Verified basic connectivity between the two machines.

7. Saving Packet Capture to File

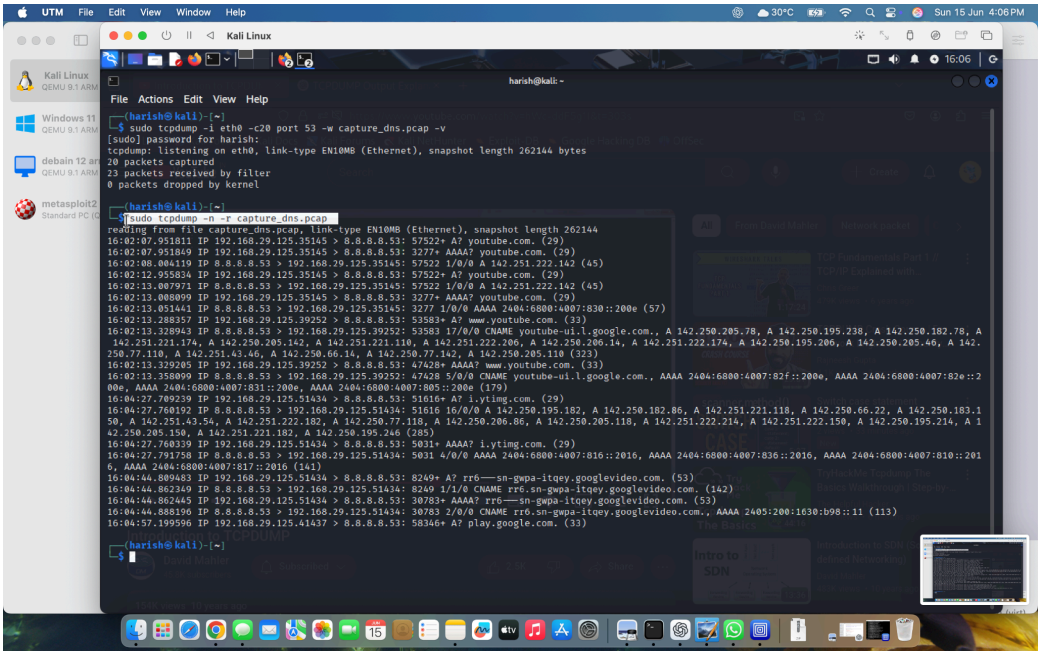
To write the packet capture to a file : tcpdump -i eth0 -w capture.pcap -v



- This saved all captured packets into a file called capture.pcap for future analysis.

8. Reading a Saved Capture File

To read and analyze the saved packet file: tcpdump -r capture.pcap



9. Summary of Key Learnings

- tcpdump -i any captures traffic on all interfaces; useful for generic monitoring.
- Limiting packet count with -c avoids excess data and saves storage.
- Using -w and -r helps store and later review traffic efficiently.
- Netcat was effective in simulating TCP traffic and analyzing the handshake.
- Packet inspection confirms the understanding of TCP behavior.

10. Conclusion

This basic exploration of TCPDump provided foundational knowledge in packet capture, reading raw network data, and observing real-time TCP communication. While these were beginner-level tests, they serve as essential building blocks for more advanced traffic analysis, intrusion detection, or forensic investigation.

Reference:https://youtu.be/hWc-ddF5g1I?si=GiRJFIiwO2PXUvqr