

# 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.

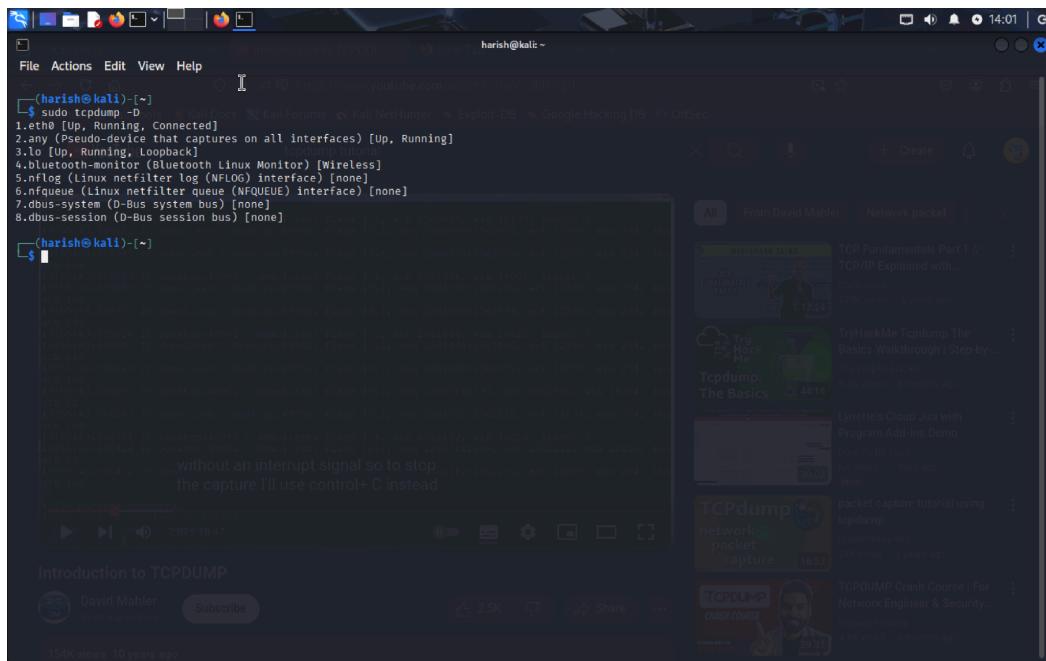
## 2. Tools Used

- **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
```



```
(harish@kali)-[~] $ sudo tcpdump -D
[sudo] password for harish: 
1.eth0 [Up, Running, Connected]
2.any [Pseudo-device that captures on all interfaces] [Up, Running]
3.wlan0 [Up, Running, Not-Block]
4.bluetooth-monitor [Bluetooth Linux Monitor] [Wireless]
5.wifilog [Linux wifilog filter log (NFLOG) interface] [none]
6.nfqueue [Linux nfqueue queue (NFQUEUE) interface] [none]
7 dbus-system (D-Bus system bus) [none]
8 dbus-session (D-Bus session bus) [none]
```

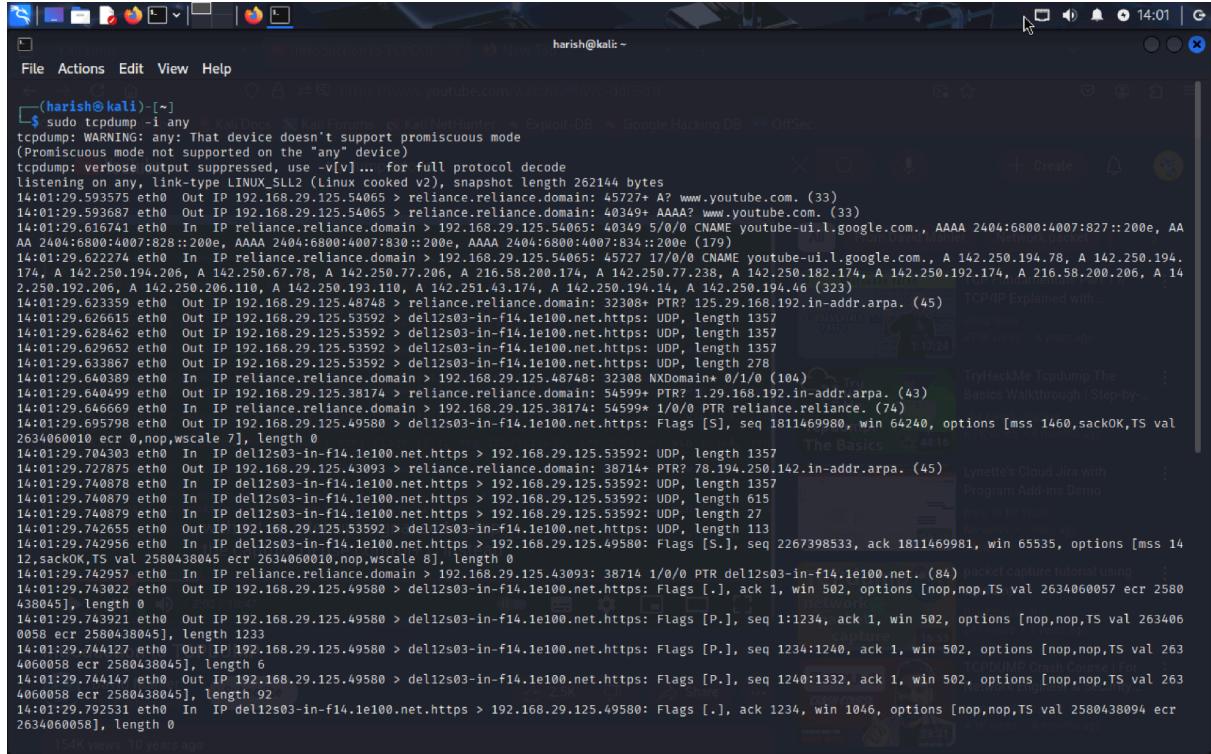
**Fig:** different network interfaces

This lists all interfaces that can be used with TCPDump.

## 4. First Capture Attempt – Any Interface

Command used:

tcpdump -i any



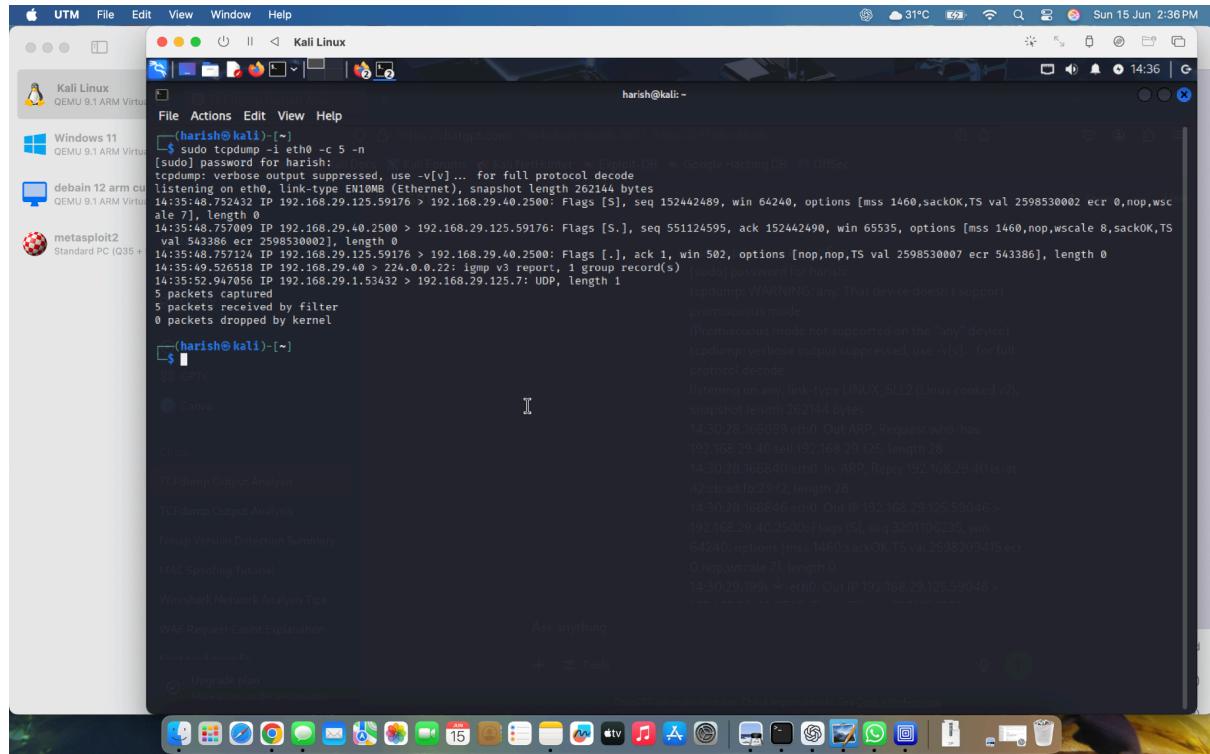
```
(harish㉿kali)-[~]
$ sudo tcpdump -i any
tcpdump: WARNING: any: That device doesn't support promiscuous mode
(Promiscuous mode not supported on the "any" device)
tcpdump: verbose output suppressed, use -vff... for full protocol decode
listening on any, link-type LINUX_SLL2 (Linux cooked v2), snapshot length 262144 bytes
14:01:29.593575 eth0 Out IP 192.168.29.125.54065 > reliance.reliance.domain: 45727: A? www.youtube.com. (33)
14:01:29.593667 eth0 Out IP 192.168.29.125.54065 > reliance.reliance.domain: 40349: AAAA? www.youtube.com. (33)
14:01:29.616741 eth0 In  IP reliance.reliance.domain > 192.168.29.125.54065: 40349 5/0/0 CNAME youtube-ui.l.google.com., AAAA 2404:6800:4007:830::200e, AA
AA 2404:6800:4007:828::200e, AAAA 2404:6800:4007:834::200e (179)
14:01:29.622274 eth0 In  IP reliance.reliance.domain > 192.168.29.125.54065: 45727 17/0/0 CNAME youtube-ui.l.google.com., A 142.250.194.78, A 142.250.194.174, A 142.250.194.206, A 142.250.206.110, A 142.250.193.110, A 142.251.43.174, A 142.250.194.14, A 142.250.194.46 (323)
14:01:29.623359 eth0 Out IP 192.168.29.125.48748 > reliance.reliance.domain: 32308+ PTR? 125.29.168.192.in-addr.arpa. (45) TCP/IP Explained with...
14:01:29.626615 eth0 Out IP 192.168.29.125.53592 > del12s03-in-f14.1e100.net.https: UDP, length 1357
14:01:29.628462 eth0 Out IP 192.168.29.125.53592 > del12s03-in-f14.1e100.net.https: UDP, length 1357
14:01:29.629652 eth0 Out IP 192.168.29.125.53592 > del12s03-in-f14.1e100.net.https: UDP, length 1357
14:01:29.633867 eth0 Out IP 192.168.29.125.53592 > del12s03-in-f14.1e100.net.https: UDP, length 278
14:01:29.640389 eth0 In  IP reliance.reliance.domain > 192.168.29.125.48748: 32308 NXDomain+ 0/1/0 (104)
14:01:29.640499 eth0 Out IP 192.168.29.125.38174 > reliance.reliance.domain: 54599+ PTR? 1.29.168.192.in-addr.arpa. (43) TryHackMe Tcpdump The Basics
14:01:29.646669 eth0 In  IP reliance.reliance.domain > 192.168.29.125.38174: 54599+ 1/0/0 PTR reliance.reliance. (74)
14:01:29.695798 eth0 Out IP 192.168.29.125.49580 > del12s03-in-f14.1e100.net.https: Flags [S], seq 1811469980, win 64240, options [mss 1460,sackOK,TS val 2634060010 ecr 0,nop,wscale 7], length 0
14:01:29.704303 eth0 In  IP del12s03-in-f14.1e100.net.https > 192.168.29.125.53592: UDP, length 1357
14:01:29.727875 eth0 Out IP 192.168.29.125.43093 > reliance.reliance.domain: 38714+ PTR? 78.194.250.142.in-addr.arpa. (45) Lynette's Cloud Jira with...
14:01:29.740878 eth0 In  IP del12s03-in-f14.1e100.net.https > 192.168.29.125.53592: UDP, length 1357
14:01:29.740879 eth0 In  IP del12s03-in-f14.1e100.net.https > 192.168.29.125.53592: UDP, length 615
14:01:29.740879 eth0 In  IP del12s03-in-f14.1e100.net.https > 192.168.29.125.53592: UDP, length 27
14:01:29.742655 eth0 Out IP 192.168.29.125.53592 > del12s03-in-f14.1e100.net.https: UDP, length 113
14:01:29.742956 eth0 In  IP del12s03-in-f14.1e100.net.https > 192.168.29.125.49580: Flags [S.], seq 2267398533, ack 1811469981, win 65535, options [mss 14
12,sackOK,TS val 2580438045 ecr 2634060010,nop,wscale 8], length 0
14:01:29.742957 eth0 In  IP reliance.reliance.domain > 192.168.29.125.43093: 38714 1/0/0 PTR del12s03-in-f14.1e100.net. (84) hacked capture tutorial using...
14:01:29.743022 eth0 Out IP 192.168.29.125.49580 > del12s03-in-f14.1e100.net.https: Flags [.], ack 1, win 502, options [nop,nop,TS val 2634060057 ecr 2580
438045], length 0
14:01:29.743921 eth0 Out IP 192.168.29.125.49580 > del12s03-in-f14.1e100.net.https: Flags [P.], seq 1:1234, ack 1, win 502, options [nop,nop,TS val 263406
058 ecr 2580438045], length 1233
14:01:29.744127 eth0 Out IP 192.168.29.125.49580 > del12s03-in-f14.1e100.net.https: Flags [P.], seq 1234:1240, ack 1, win 502, options [nop,nop,TS val 263
406058 ecr 2580438045], length 6
14:01:29.744147 eth0 Out IP 192.168.29.125.49580 > del12s03-in-f14.1e100.net.https: Flags [P.], seq 1240:1332, ack 1, win 502, options [nop,nop,TS val 263
406058 ecr 2580438045], length 92
14:01:29.792531 eth0 In  IP del12s03-in-f14.1e100.net.https > 192.168.29.125.49580: Flags [.], ack 1234, win 1046, options [nop,nop,TS val 2580438094 ecr
2634060058], length 0
```

**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



```
[harish@kali:~] $ sudo tcpdump -i eth0 -c 5 -n
[sudo] password for harish:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:35:48.752432 IP 192.168.29.125.59176 > 192.168.29.40.2500: Flags [S], seq 152442489, ack 152442490, options [mss 1460,sackOK,TS val 2598530002 ecr 0,nop,wscale 8,sackOK,TS val 543886 ecr 2598530002], length 0
14:35:48.757124 IP 192.168.29.125.59176 > 192.168.29.40.2500: Flags [S.], seq 551124595, ack 152442490, win 65535, options [mss 1460,nop,wscale 8,sackOK,TS val 543886 ecr 2598530002], length 0
14:35:49.526512 IP 192.168.29.40 > 224.0.0.2: igmp v3 report, 1 group record(s)
14:35:52.947056 IP 192.168.29.1.53432 > 192.168.29.125.7: UDP, length 1
5 packets captured
5 packets received by filter
0 packets dropped by kernel
[harish@kali:~] $
```

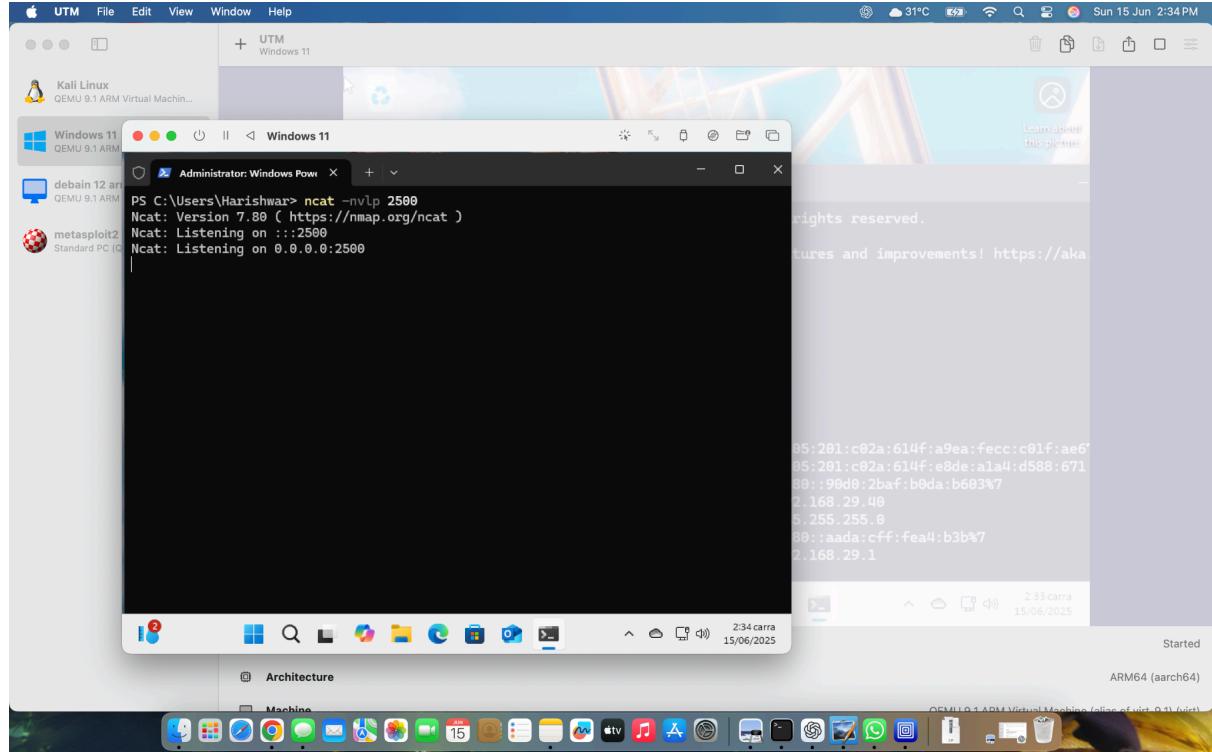
Fig: using eth0 (specified [no.of](#) packets)

- 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.

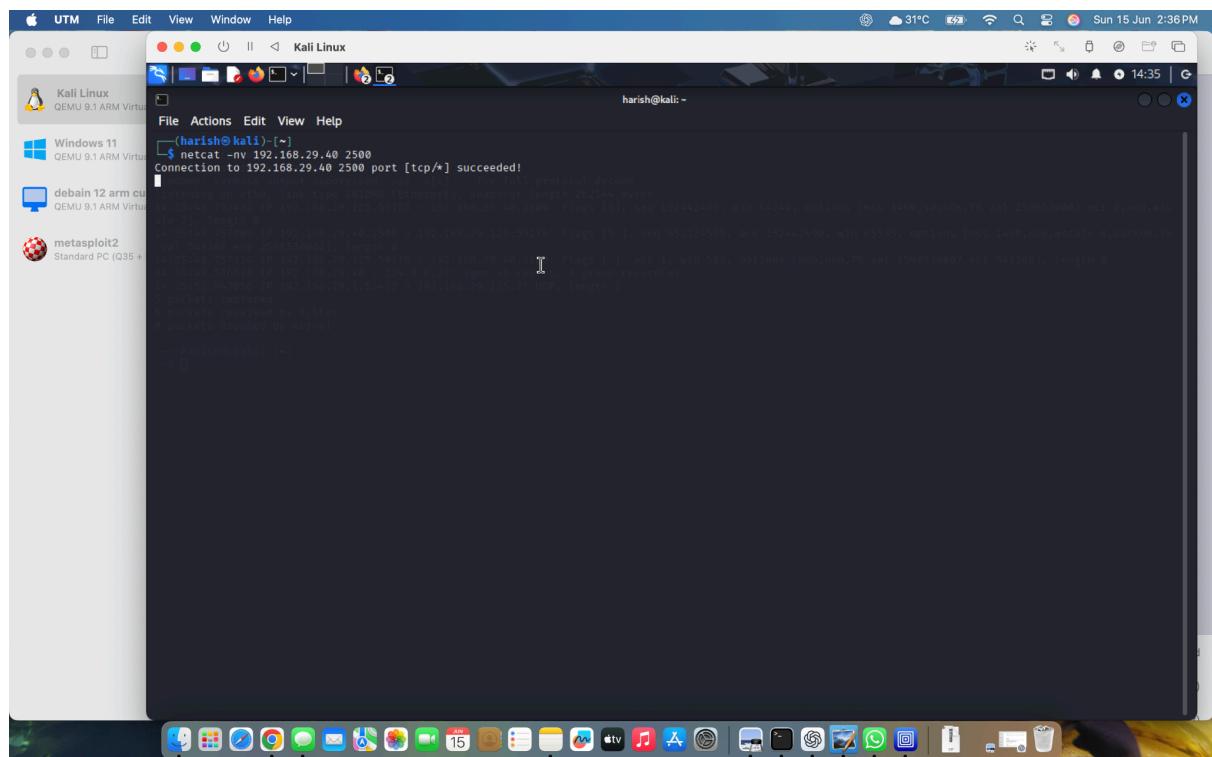
## 6. Netcat Packet Exchange Testing (Port 2500)

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

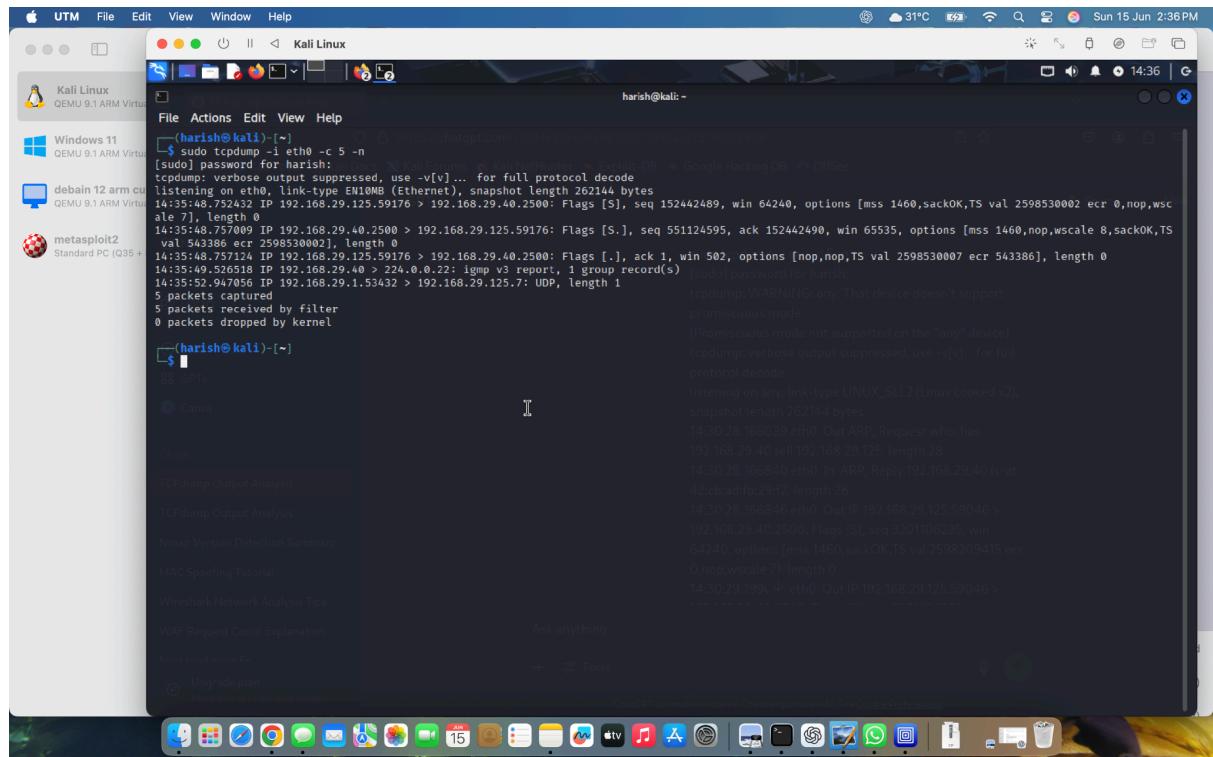
- **On Windows (Listener):** ncat -nvlp 2500



- **On Linux (Client):** netcat <Windows\_IP> 2500



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



The screenshot shows a terminal window on a Kali Linux desktop environment. The terminal output displays the results of a TCPdump command:

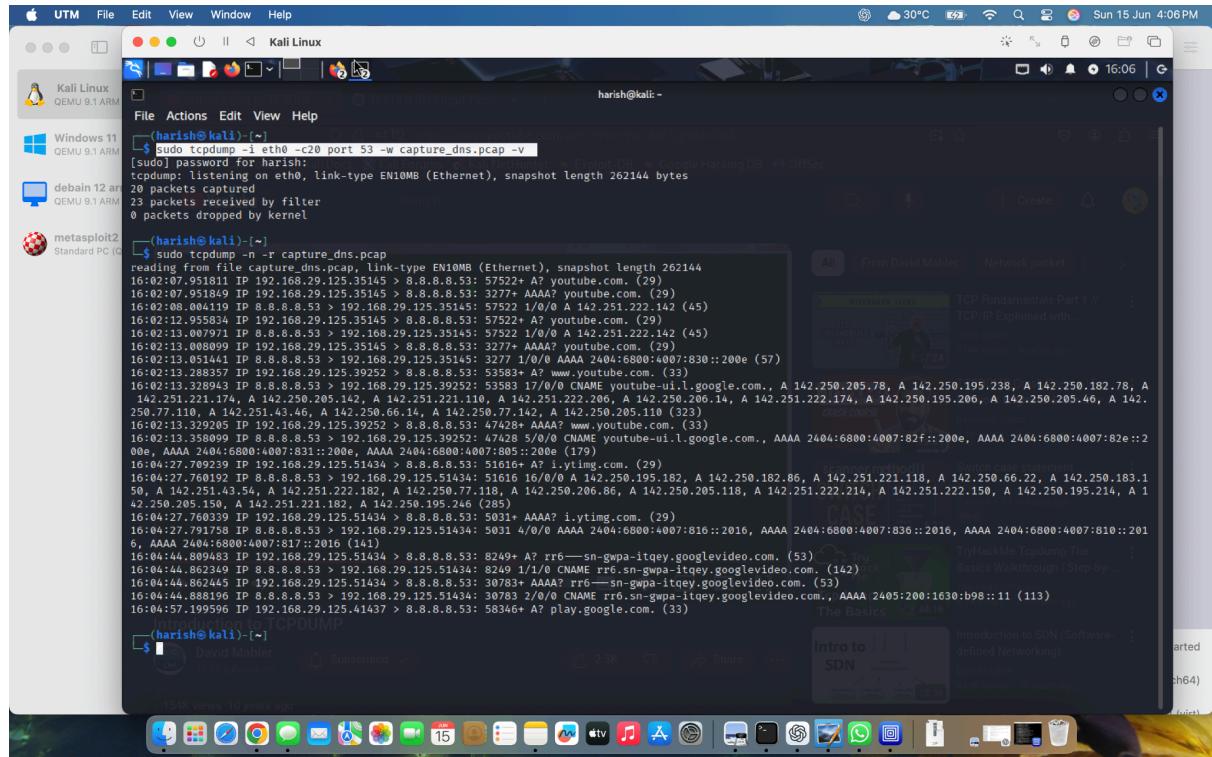
```
[harish@kali:~] $ sudo tcpdump -i eth0 -c 5 -n
[sudo] password for harish:
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on any, link-type EN10MB (Ethernet), snapshot length 262144 bytes
14:35:48.757242 IP 192.168.29.125.59176 > 192.168.29.40.2500: Flags [S], seq 152442489, win 64240, options [mss 1460,sackOK,TS val 2598530002 ecr 0,nop,wscale 8,sackOK,TS val 7], length 0
14:35:48.757099 IP 192.168.29.40.2500 > 192.168.29.125.59176: Flags [S.], seq 551124595, ack 152442490, win 65535, options [mss 1460,nop,wscale 8,sackOK,TS val 2598530007 ecr 543386], length 0
14:35:49.526518 IP 192.168.29.40.2500 > 224.0.0.22: igmp v3 report, 1 group record(s)
14:35:52.947056 IP 192.168.29.1.53432 > 192.168.29.125.7: UDP, length 1
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

**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`



```
harish@kali:~$ sudo tcpdump -i eth0 -c 20 port 53 -w capture_dns.pcap -v
[sudo] password for harish:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20 packets captured
23 packets received by filter
0 packets dropped by kernel

(harish@kali):~$ sudo tcpdump -n -r capture_dns.pcap
reading from file capture_dns.pcap, link-type EN10MB (Ethernet), snapshot length 262144
16:02:07.951811 IP 192.168.29.125.35145 > 8.8.8.53: 5752+ A? youtube.com. (29)
16:02:07.951849 IP 192.168.29.125.35145 > 8.8.8.53: 3277+ AAAA? youtube.com. (29)
16:02:08.004419 IP 8.8.8.53 > 192.168.29.125.35145: 5752 1/0/0 A 142.251.222.142 (45)
16:02:12.955834 IP 192.168.29.125.35145 > 8.8.8.53: 5752+ A? youtube.com. (29)
16:02:13.007971 IP 8.8.8.53 > 192.168.29.125.35145: 5752 1/0/0 A 142.251.222.142 (45)
16:02:13.007971 IP 8.8.8.53 > 192.168.29.125.35145: 3277 1/0/0 AAAA? 2404:6800:4:007:830::200e (57)
16:02:13.051441 IP 8.8.8.53 > 192.168.29.125.35145: 3277 1/0/0 AAAA? 2404:6800:4:007:830::200e (33)
16:02:13.328943 IP 8.8.8.53 > 192.168.29.125.39252: 5358+ 17/0/0 CNAME youtube.ul.google.com., A 142.250.205.78, A 142.250.195.238, A 142.250.182.78, A 142.251.221.174, A 142.250.205.102, A 142.251.222.206, A 142.251.222.174, A 142.250.195.206, A 142.250.205.46, A 142.250.205.110, A 142.251.43.46, A 142.250.66.14, A 142.250.77.142, A 142.250.205.110 (323)
16:02:13.329295 IP 192.168.29.125.39252 > 8.8.8.53: 4742+ AAAA? www.youtube.com. (33)
16:02:13.358099 IP 8.8.8.53 > 192.168.29.125.39252: 4742 5/0/0 CNAME youtube.ul.google.com., AAAA 2404:6800:4:007:82f::200e, AAAA 2404:6800:4:007:82e::200e, AAAA 2404:6800:4:007:831::200e, AAAA 2404:6800:4:007:805::200e (179)
16:02:27.709239 IP 192.168.29.125.51434 > 8.8.8.53: 51616+ A? i.ytimg.com. (29)
16:04:27.760339 IP 192.168.29.125.51434 > 8.8.8.53: 51616 1/0/0 A 142.251.222.118, A 142.250.182.86, A 142.251.221.118, A 142.250.66.22, A 142.250.183.1, A 142.251.221.174, A 142.250.205.102, A 142.251.222.206, A 142.251.222.174, A 142.250.205.110, A 142.251.222.214, A 142.251.222.150, A 142.250.195.214, A 142.250.205.150, A 142.251.221.182, A 142.250.195.246 (285)
16:04:27.791758 IP 8.8.8.53 > 192.168.29.125.51434: 8.8.8.53: 5031+ AAAA? i.ytimg.com. (29)
16:04:27.791758 IP 8.8.8.53 > 192.168.29.125.51434: 30783 2/0/0 CNAME rr6.sn-gwpa-itqey.googlevideo.com., AAAA 2405:200:1630:b98::11 (113)
16:04:57.199596 IP 192.168.29.125.41437 > 8.8.8.53: 58346+ A? play.google.com. (33)

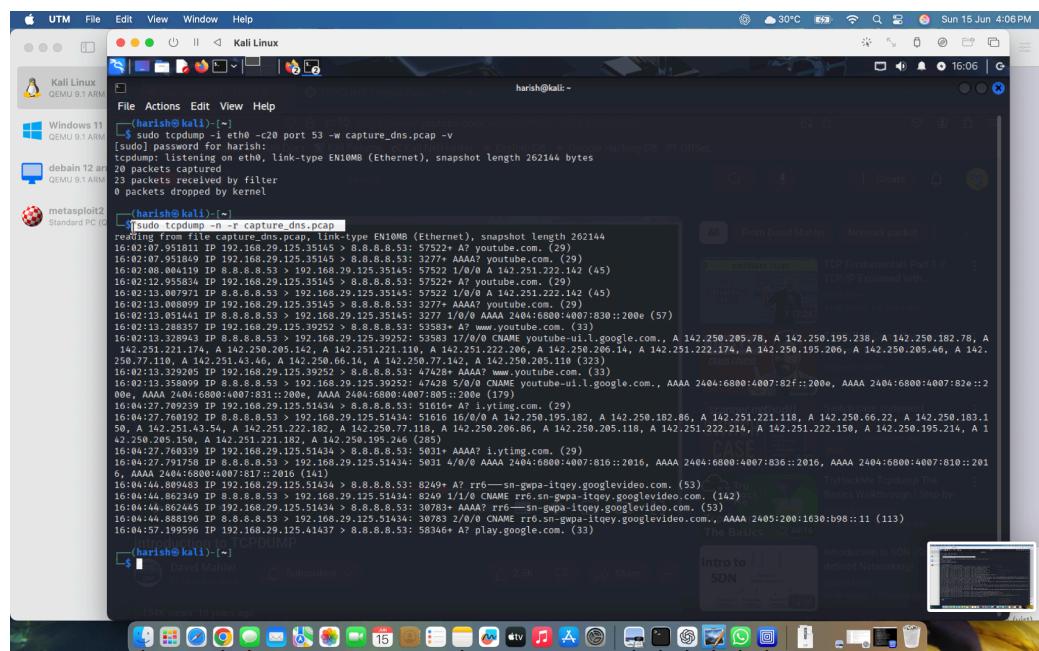
Introduction to TCPDUMP
David Maher
Submitted
10K views 10 years ago

(harish@kali):~$
```

- 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`



```
harish@kali:~$ sudo tcpdump -i eth0 -c 20 port 53 -w capture_dns.pcap
[sudo] password for harish:
tcpdump: listening on eth0, link-type EN10MB (Ethernet), snapshot length 262144 bytes
20 packets captured
23 packets received by filter
0 packets dropped by kernel

(harish@kali):~$ sudo tcpdump -n -r capture_dns.pcap
reading from file capture_dns.pcap, link-type EN10MB (Ethernet), snapshot length 262144
16:02:07.951811 IP 192.168.29.125.35145 > 8.8.8.53: 5752+ A? youtube.com. (29)
16:02:07.951849 IP 192.168.29.125.35145 > 8.8.8.53: 3277+ AAAA? youtube.com. (29)
16:02:08.004419 IP 8.8.8.53 > 192.168.29.125.35145: 5752 1/0/0 A 142.251.222.142 (45)
16:02:12.955834 IP 192.168.29.125.35145 > 8.8.8.53: 5752+ A? youtube.com. (29)
16:02:13.007971 IP 8.8.8.53 > 192.168.29.125.35145: 3277 1/0/0 AAAA? 2404:6800:4:007:830::200e (57)
16:02:13.051441 IP 8.8.8.53 > 192.168.29.125.35145: 3277 1/0/0 AAAA? 2404:6800:4:007:830::200e (33)
16:02:13.328943 IP 8.8.8.53 > 192.168.29.125.39252: 5358+ 17/0/0 CNAME youtube.ul.google.com., A 142.250.205.78, A 142.250.195.238, A 142.250.182.78, A 142.251.221.174, A 142.250.205.102, A 142.251.222.206, A 142.251.222.174, A 142.250.195.206, A 142.250.205.46, A 142.250.205.110, A 142.251.43.46, A 142.250.66.14, A 142.250.77.142, A 142.250.205.110 (323)
16:02:13.329295 IP 192.168.29.125.39252 > 8.8.8.53: 4742+ AAAA? www.youtube.com. (33)
16:02:13.358099 IP 8.8.8.53 > 192.168.29.125.39252: 4742 5/0/0 CNAME youtube.ul.google.com., AAAA 2404:6800:4:007:82f::200e, AAAA 2404:6800:4:007:82e::200e, AAAA 2404:6800:4:007:831::200e, AAAA 2404:6800:4:007:805::200e (179)
16:02:27.709239 IP 192.168.29.125.51434 > 8.8.8.53: 51616+ A? i.ytimg.com. (29)
16:04:27.760339 IP 192.168.29.125.51434 > 8.8.8.53: 51616 1/0/0 A 142.251.222.118, A 142.250.182.86, A 142.251.221.118, A 142.250.66.22, A 142.250.183.1, A 142.251.221.174, A 142.250.205.102, A 142.251.222.206, A 142.251.222.174, A 142.250.205.110, A 142.251.222.214, A 142.251.222.150, A 142.250.195.214, A 142.250.205.150, A 142.251.221.182, A 142.250.195.246 (285)
16:04:27.791758 IP 8.8.8.53 > 192.168.29.125.51434: 8.8.8.53: 5031+ AAAA? i.ytimg.com. (29)
16:04:27.791758 IP 8.8.8.53 > 192.168.29.125.51434: 30783 2/0/0 CNAME rr6.sn-gwpa-itqey.googlevideo.com., AAAA 2405:200:1630:b98::11 (113)
16:04:57.199596 IP 192.168.29.125.41437 > 8.8.8.53: 58346+ A? play.google.com. (33)

Introduction to TCPDUMP
David Maher
Submitted
10K views 10 years ago

(harish@kali):~$
```

## **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=GiRJFliwO2PXUvqr>