

# Wireshark Network Traffic Analysis

**Project Overview: Wireshark Analysis** This project focuses on analyzing network traffic using Wireshark, a widely used open-source network protocol analyzer. The main objective is to capture, inspect, and interpret data packets transmitted over a network to understand communication patterns, detect anomalies, and troubleshoot connectivity issues. Throughout the project, various protocols such as TCP, UDP, ICMP, HTTP, and DNS were examined to observe how data flows between devices. The analysis helped identify normal traffic behavior as well as potential issues like packet loss, retransmissions, and suspicious activities.

## Steps Performed

1. Installed Wireshark on Kali Linux.
2. Started packet capture on the active network interface.
3. Generated traffic by:
  - Browsing the website: <https://elevatelabs.in>
  - Pinging the same domain using the terminal (ping elevatelabs.in)
4. Captured packets for about one minute.
5. Filtered packets in Wireshark by:
  - http — for web traffic

•dns — for domain resolution

•tcp — for transport-level communication

6. Identified at least three protocols in the captured data:

•DNS – Used for domain name resolution.

•TCP – Ensured reliable transport of data.

•HTTP – Managed web content exchange between client and server.

7. Exported the capture as a .pcap file for documentation and analysis.

### Findings Summary


### Files Included

•traffic\_capture.pcap → Raw packet capture

### Tools Used

• Wireshark – For capturing and analyzing network packets.

• Ping Command – To generate ICMP traffic.

• Web Browser – To create HTTP and DNS requests.

• CentralOps.net – Used to find the IP address of the target domain (elevate.labs.in)

• Gained understanding of how different protocols (DNS, TCP, HTTP) interact during normal web communication.

• Learned how to capture, filter and interpret packets in Wireshark.

• Observed packet structures and relationships between layers in the OSI model.

No.	Time	Source	Destination	Protocol	Length	Info
343	65.142415	192.168.0.21	174.129.249.228	TCP	66	40555 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519346 TSecr=551811827
344	65.142715	192.168.0.21	174.129.249.228	HTTP	253	GET /clients/netflix/flash/application.swf?flash_version=flash_lite_2.18v=1.58n
345	65.230738	174.129.249.228	192.168.0.21	TCP	66	80 → 40555 [ACK] Seq=1 Ack=188 Win=6864 Len=0 TSval=551811850 TSecr=491519347
346	65.240742	174.129.249.228	192.168.0.21	HTTP	828	HTTP/1.1 302 Moved Temporarily
347	65.241592	192.168.0.21	174.129.249.228	TCP	66	40555 → 80 [ACK] Seq=188 Ack=763 Win=7424 Len=0 TSval=491519446 TSecr=551811852
348	65.242532	192.168.0.21	192.168.0.1	DNS	77	Standard query 0x2188 A cdn-0.nflximg.com
349	65.276870	192.168.0.1	192.168.0.21	DNS	489	Standard query response 0x2188 A cdn-0.nflximg.com CNAME images.netflix.com.edge
350	65.277992	192.168.0.21	63.80.242.48	TCP	74	37063 → 80 [SYN] Seq=0 Win=5840 Len=0 MSS=1460 SACK_PERM=1 TSval=491519482 TSecr=
351	65.297757	63.80.242.48	192.168.0.21	TCP	74	80 → 37063 [SYN, ACK] Seq=0 Ack=1 Win=5792 Len=0 MSS=1460 SACK_PERM=1 TSval=3295
352	65.298396	192.168.0.21	63.80.242.48	TCP	66	37063 → 80 [ACK] Seq=1 Ack=1 Win=5888 Len=0 TSval=491519502 TSecr=3295534138
353	65.298687	192.168.0.21	63.80.242.48	HTTP	153	GET /us/nrd/clients/flash/814540.bun HTTP/1.1
354	65.318750	63.80.242.48	192.168.0.21	TCP	66	80 → 37063 [ACK] Seq=1 Ack=88 Win=5792 Len=0 TSval=3295534151 TSecr=491519503
355	65.321733	63.80.242.48	192.168.0.21	TCP	1514	[TCP segment of a reassembled PDU]

> Frame 349: 489 bytes on wire (3912 bits), 489 bytes captured (3912 bits)  
 > Ethernet II, Src: GlobalSC\_08:3b:0a:f0:ad:4e (f0:ad:4e:08:3b:0a), Dst: Vizio\_14:8a:e1 (80:19:9d:14:8a:e1)  
 > Internet Protocol Version 4, Src: 192.168.0.1, Dst: 192.168.0.21  
 > User Datagram Protocol, Src Port: 53 (53), Dst Port: 34036 (34036)

#### Domain Name System (response)

[Request ID: 348]  
 [Time: 0.034338000 seconds]  
 [Transaction ID: 0x2188]  
 > Flags: 0x5180 Standard query response, No error  
 Questions: 1  
 Answer RRs: 4  
 Authority RRs: 0  
 Additional RRs: 0

#### Queries

> cdn-0.nflximg.com: type A, class IN

#### Answers

> Authoritative nameservers

```

0020  00 15 00 35 84 f4 01 c7 83 3f 41 00 01 00 00 01  ...S....?.....
0030  80 04 00 09 00 09 05 63 64 6e 2d 30 07 6e 65 6c  ....c dn-0.nfl
0040  78 69 6d 67 03 63 6f 6d 00 00 01 00 01 c8 0c 00  xing.com .....
0050  05 00 01 00 00 05 29 00 22 06 69 6d 61 67 65 73  .....). ".images
0060  07 6e 65 74 66 6c 69 78 03 63 6f 6d 09 65 64 67  .netflix .con.edg
0070  65 73 75 69 74 65 03 6e 65 74 00 c0 2f 00 05 00  esuite.n et.../...

```