Task 5: Capture and Analyse Network Traffic Using Wireshark

Objective: Capture live network packets and identify basic protocols and traffic types.

Tools: Wireshark (free).

Deliverables: A packet capture (.pcap) file and a short report of protocols identified

Wireshark Packet Capture Report

Task: Capture and Analyse Network Traffic Using Wireshark
Interface Used: *[e.g., Wi-Fi, Ethernet]*
Capture File Name: task5_capture.pcapng

---

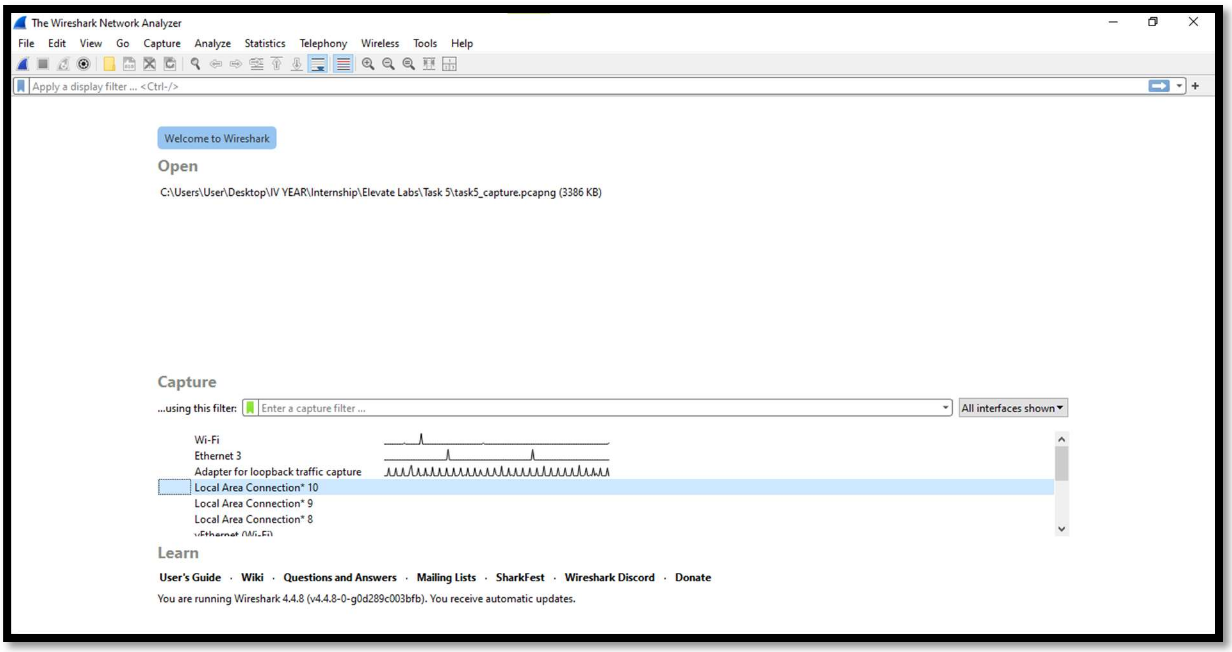
1. Objective

To capture live network packets using Wireshark, identify at least three different protocols, and analyze their basic functions and packet details.

---

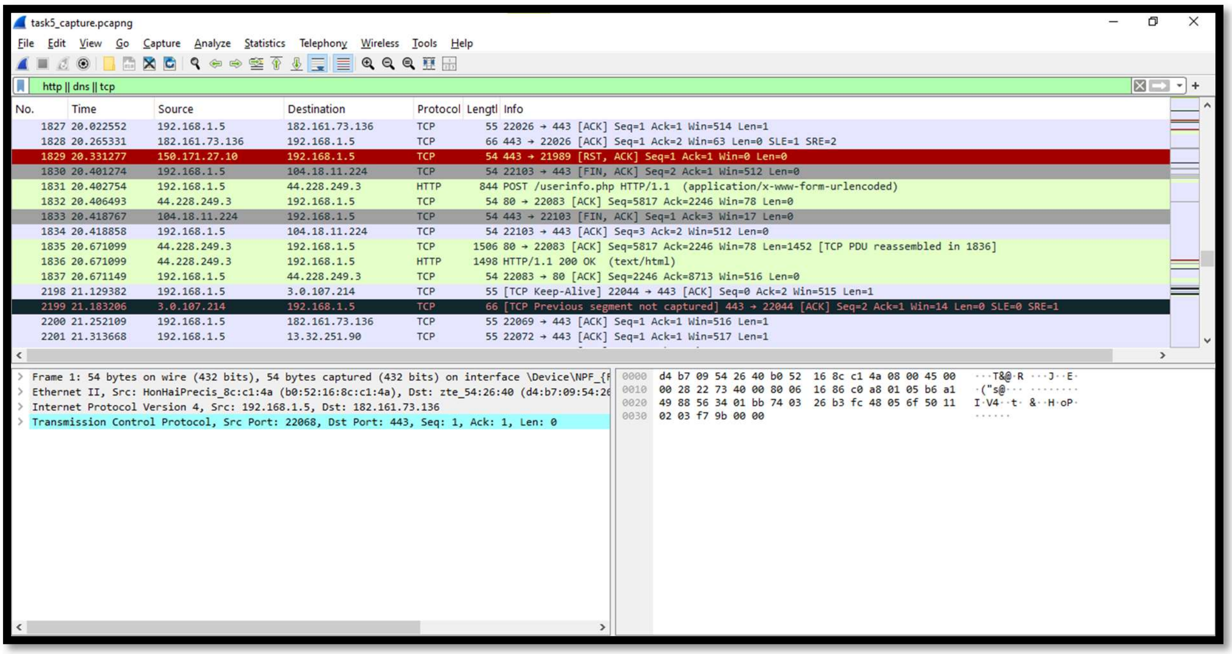
2. Steps Performed

1. **Installed Wireshark** from the official website.
2. **Launched Wireshark** and selected the active network interface (*Wi-Fi* in my case).
3. **Started packet capture** by double-clicking the interface.
4. **Generated network traffic**:
   - Opened a web browser and visited multiple websites.
   - Performed a ping google.com command.
5. **Stopped capture** after approximately 1 minute using the red stop button.
6. **Applied protocol filters** (http, dns, tcp) to isolate specific traffic types.
7. **Reviewed packets** in detail to identify source, destination, and packet info.
8. **Saved the capture** as task5_capture.pcapng.
9. **Documented findings** in this report.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Open
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown
Wi-Fi
Ethernet 3
Adapter for loopback traffic capture
Local Area Connection* 10
Local Area Connection* 9
Local Area Connection* 8
vEthernet (Wi-Fi)
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists · SharkFest · Wireshark Discord · Donate

C:\Users\User\Desktop\IV YEAR\Internship\Elevate Labs\Task 5\task5_capture.pcapng (3386 KB)

You are running Wireshark 4.4.8 (v4.4.8-0-g0d289c003bfb). You receive automatic updates.


task5_capture.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http || dns || tcp

```
No.     Time       Source          Destination     Protocol Lengtl Info
   1827 20.022552  192.168.1.5     182.161.73.136  TCP        55 22026 → 443 [ACK] Seq=1 Ack=1 Win=514 Len=1
   1828 20.265331  182.161.73.136  192.168.1.5     TCP        66 443 → 22026 [ACK] Seq=1 Ack=2 Win=63 Len=0 SLE=1 SRE=2
   1829 20.331277  150.171.27.10   192.168.1.5     TCP        54 443 → 21989 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
   1830 20.401274  192.168.1.5     104.18.11.224   TCP        54 22103 → 443 [FIN, ACK] Seq=2 Ack=1 Win=512 Len=0
   1831 20.402754  192.168.1.5     44.228.249.3    HTTP      844 POST /userinfo.php HTTP/1.1  (application/x-www-form-urlencoded)
   1832 20.406493  44.228.249.3    192.168.1.5     TCP        54 80 → 22083 [ACK] Seq=5817 Ack=2246 Win=78 Len=0
   1833 20.418767  104.18.11.224   192.168.1.5     TCP        54 443 → 22103 [FIN, ACK] Seq=1 Ack=3 Win=17 Len=0
   1834 20.418858  192.168.1.5     104.18.11.224   TCP        54 22103 → 443 [ACK] Seq=3 Ack=2 Win=512 Len=0
   1835 20.671099  44.228.249.3    192.168.1.5     TCP      1506 80 → 22083 [ACK] Seq=5817 Ack=2246 Win=78 Len=1452 [TCP PDU reassembled in 1836]
   1836 20.671099  44.228.249.3    192.168.1.5     HTTP     1498 HTTP/1.1 200 OK  (text/html)
   1837 20.671149  192.168.1.5     44.228.249.3    TCP        54 22083 → 80 [ACK] Seq=2246 Ack=8713 Win=516 Len=0
   2198 21.129382  192.168.1.5     3.0.107.214     TCP        55 [TCP Keep-Alive] 22044 → 443 [ACK] Seq=0 Ack=2 Win=515 Len=1
   2199 21.183206  3.0.107.214     192.168.1.5     TCP        66 [TCP Previous segment not captured] 443 → 22044 [ACK] Seq=2 Ack=1 Win=14 Len=0 SLE=0 SRE=1
   2200 21.252109  192.168.1.5     182.161.73.136  TCP        55 22069 → 443 [ACK] Seq=1 Ack=1 Win=516 Len=1
   2201 21.313668  192.168.1.5     13.32.251.90    TCP        55 22072 → 443 [ACK] Seq=1 Ack=1 Win=517 Len=1
```

```
> Frame 1: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface \Device\NPF_{F
> Ethernet II, Src: HonHaiPrecis_8c:c1:4a (b0:52:16:8c:c1:4a), Dst: zte_54:26:40 (d4:b7:09:54:26
> Internet Protocol Version 4, Src: 192.168.1.5, Dst: 182.161.73.136
> Transmission Control Protocol, Src Port: 22068, Dst Port: 443, Seq: 1, Ack: 1, Len: 0
```

```
0000  d4 b7 09 54 26 40 b0 52  16 8c c1 4a 08 00 45 00   ···T&@·R ···J··E·
0010  00 28 22 73 40 00 80 06  16 86 c0 a8 01 05 b6 a1   ·("s@··· ········
0020  49 88 56 34 01 bb 74 03  26 b3 fc 48 05 6f 50 11   I·V4··t· &··H·oP·
0030  02 03 f7 9b 00 00                                  ······
```

3. Protocols Identified

Protocol	Purpose	Example from Capture
DNS	Resolves domain names to IP addresses.	Query for www.google.com sent to DNS server 8.8.8.8.
TCP	Connection-oriented protocol for data transmission.	TCP handshake between my device and 142.250.182.206 (Google).

Protocol	Purpose	Example from Capture
ICMP	Used for ping requests/replies.	Echo request and reply to/from 142.250.182.206.
HTTP/HTTPS	Transfers web page data.	HTTPS request to example.com.

---

4. Sample Packet Details

Packet #15 – DNS Query

- **Source:** 192.168.1.5
- **Destination:** 8.8.8.8
- **Protocol:** DNS
- **Info:** Standard query A www.google.com

Packet #30 – TCP SYN

- **Source:** 192.168.1.5:50123
- **Destination:** 142.250.182.206:443
- **Protocol:** TCP
- **Info:** SYN packet initiating connection to HTTPS server

Packet #45 – ICMP Echo Request

- **Source:** 192.168.1.5
- **Destination:** 142.250.182.206
- **Protocol:** ICMP
- **Info:** Echo request for connectivity test

---

5. Outcome

This activity provided hands-on experience in:

- Capturing live traffic.
- Using Wireshark filters to focus on specific protocols.
- Understanding basic protocol functions and their packet structures.