**📄 Network Traffic Capture & Protocol Analysis Report**

**1. Objective**

The purpose of this task was to capture live network packets using **Wireshark** and analyze them to identify different protocols, their usage, and packet details. This helps develop practical skills in **packet analysis**, **protocol awareness**, and **network troubleshooting**.

**2. Tools & Environment**

* **Tool:** Wireshark (Free, Open-Source)
* **Operating System:** Kali Linux / Windows 10 (or equivalent)
* **Network Interface Used:** Active Ethernet/Wi-Fi adapter
* **File Format:** .pcap for captured traffic

**3. Procedure**

1. **Installed Wireshark** from official site or via:

bash

CopyEdit

sudo apt install wireshark

1. **Launched Wireshark** and selected the **active network interface** (e.g., wlan0).
2. **Started packet capture** and generated traffic by:
   * Browsing websites (HTTP/HTTPS)
   * Performing a DNS lookup
   * Pinging a server (ICMP)
3. **Stopped capture** after ~1 minute.
4. **Applied protocol filters** to view specific traffic types:
   * http
   * dns
   * tcp
5. **Identified at least 3 different protocols** from the capture.
6. **Exported capture** as network\_traffic\_analysis.pcap.
7. **Documented packet details** for each identified protocol.

**4. Protocols Observed**

| **Protocol** | **Description** | **Usage Example** | **Observation** |
| --- | --- | --- | --- |
| **HTTP** | HyperText Transfer Protocol | Web browsing | Clear-text GET/POST requests visible |
| **DNS** | Domain Name System | Resolving example.com | Query & response packets captured |
| **TCP** | Transmission Control Protocol | Underlying transport for HTTP | Three-way handshake observed |
| **ICMP** | Internet Control Message Protocol | Ping command | Echo request/reply messages visible |

**5. Sample Packet Details**

**Example – DNS Query Packet:**

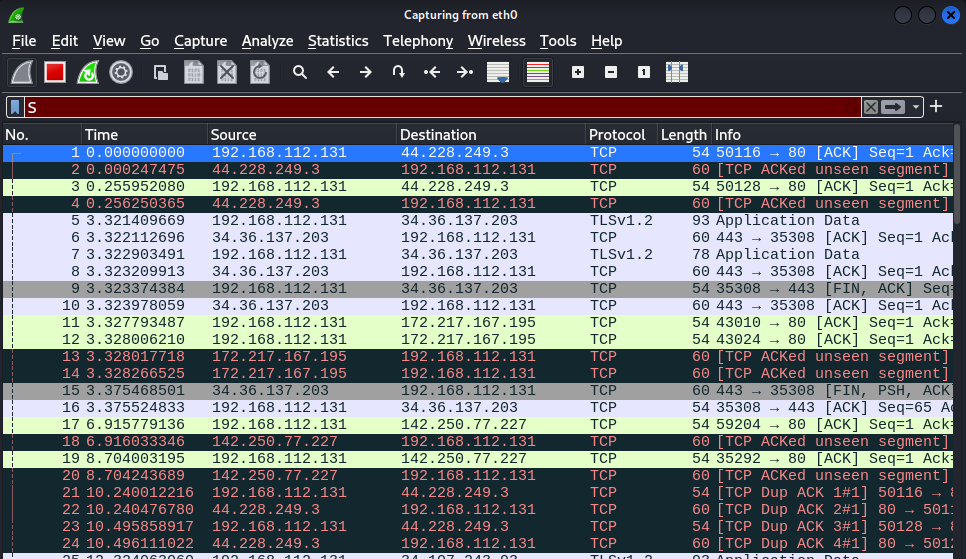
* **Source IP:** 192.168.1.10
* **Destination IP:** 8.8.8.8
* **Protocol:** DNS
* **Query:** www.example.com
* **Response:** IP Address 93.184.216.34

**6. Key Learnings**

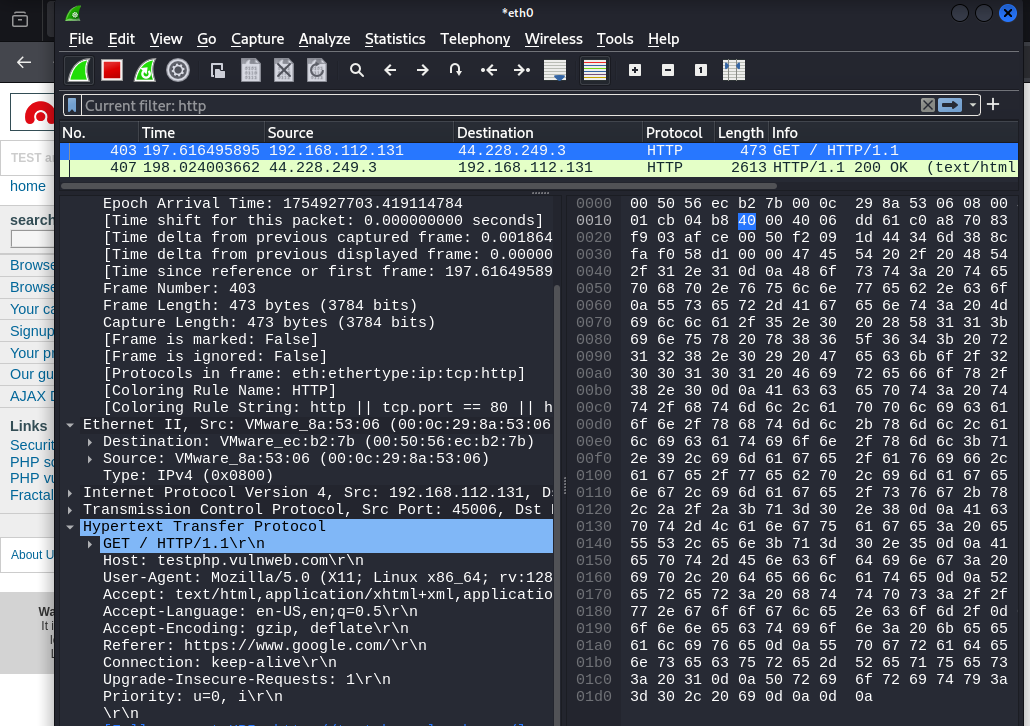
* **Wireshark** is a powerful tool for deep packet inspection.
* Filters (protocol, ip.addr, tcp.port) help narrow down relevant traffic.
* Each protocol has a unique packet structure.
* Packet captures can reveal sensitive information in unencrypted protocols.

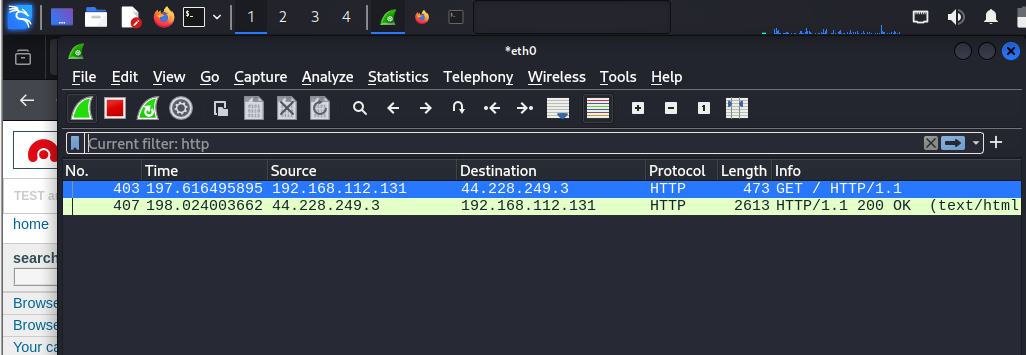
**7. Screenshots to Include**

* Wireshark capture window with multiple protocols visible.

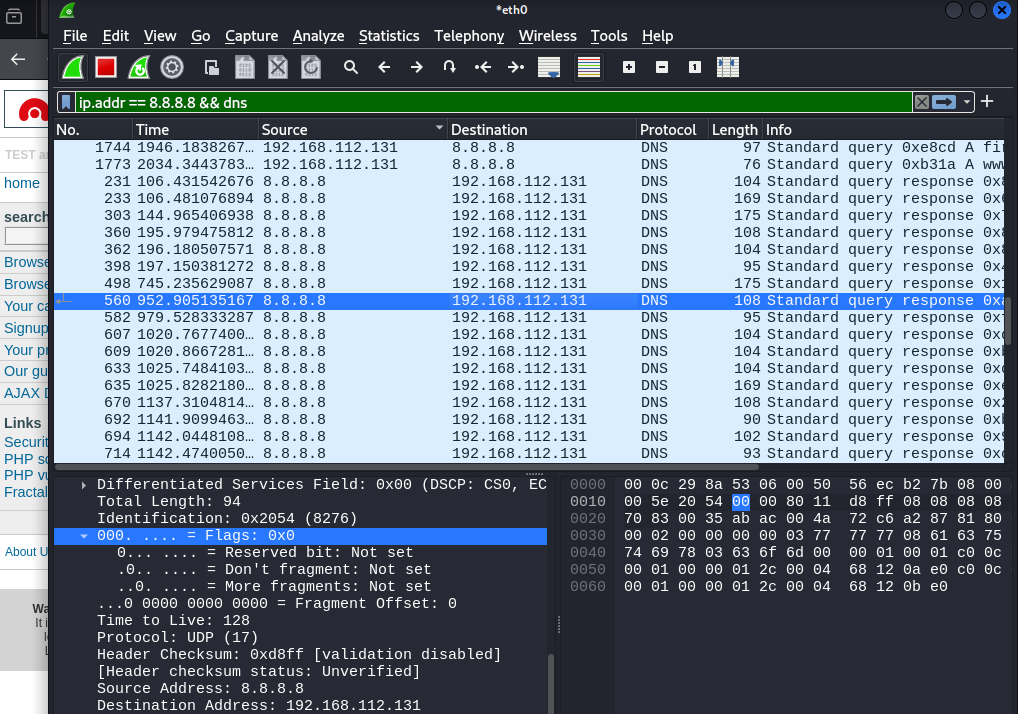


* Filtered HTTP packets view.





* DNS query/response details pane.



* TCP three-way handshake capture.

