**Wireshark Network Traffic Analysis Report**

**Task: Task 5 – Capture and Analyze Network Traffic Using Wireshark  
Submitted By: Aryanshu Saha  
Date: 30/06/2025  
Organization: Elevate Labs  
Deliverables: capture.pcap, Technical Report**

1. **Objective**

To capture live network traffic using Wireshark, identify and filter key internet protocols (such as DNS, ICMP, and HTTP), and analyze the nature of traffic flows between client and servers.

1. **Steps to Perform the Task**

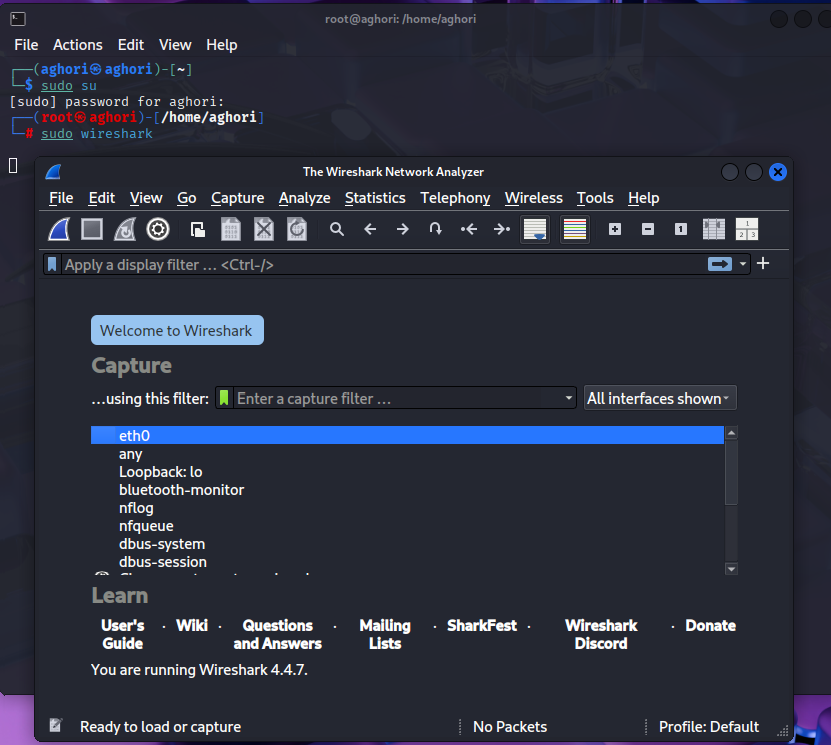
**🛠️ Prerequisites**

* Kali Linux (or any Linux distro)
* Wireshark installed (sudo apt install wireshark)
* Root privileges for packet capture

**Execution Steps**

**1. Start Wireshark**

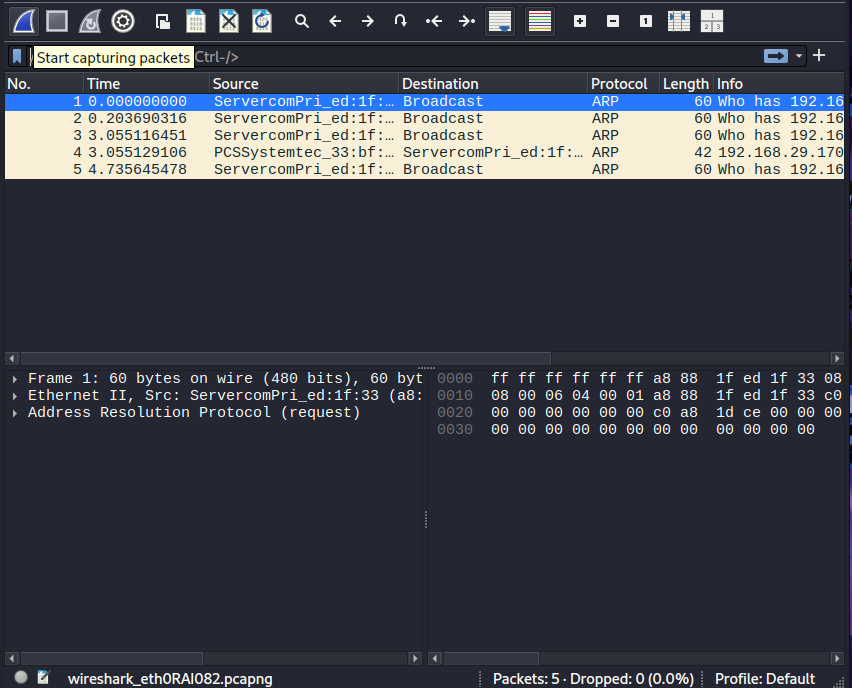
* Launch Wireshark with: sudo wireshark



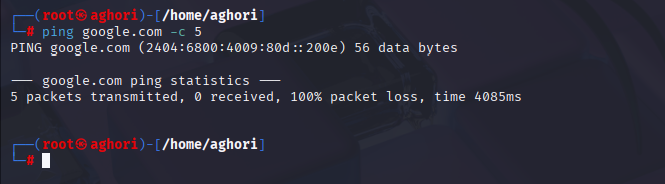
* Select your active network interface (e.g., wlan0 or eth0).

**2. Begin Packet Capture**

* Click **Start** to begin capturing packets on the selected interface.

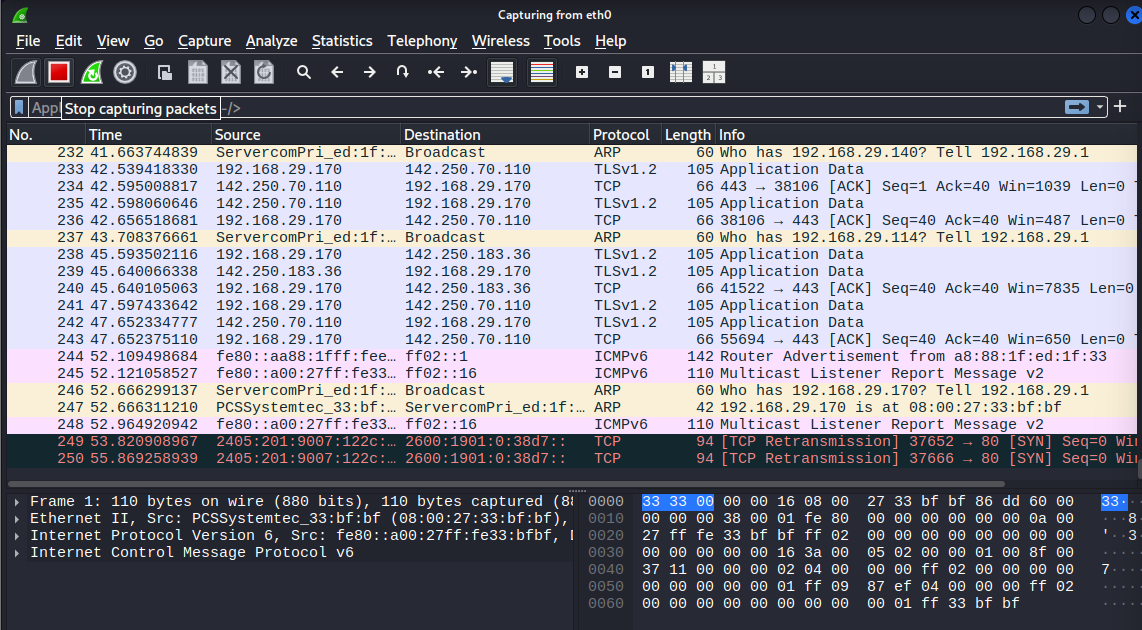


**3. Generate Network Traffic**

* Open a terminal and run: ping google.com -c 5
* Open a web browser and visit: <http://google.com>

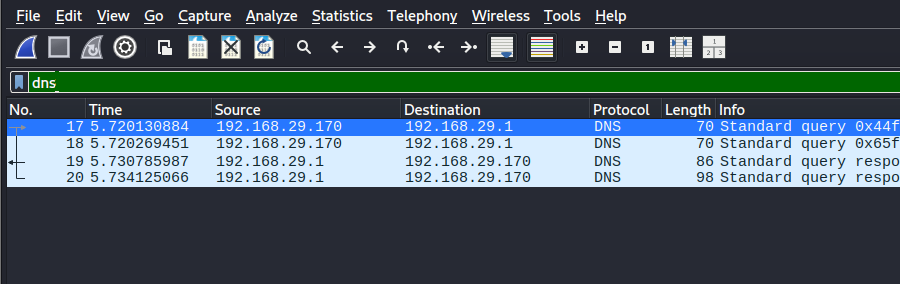
**4. Stop the Capture**

* Let it run for about one minute.
* Click the **Stop** button (red square) in Wireshark.

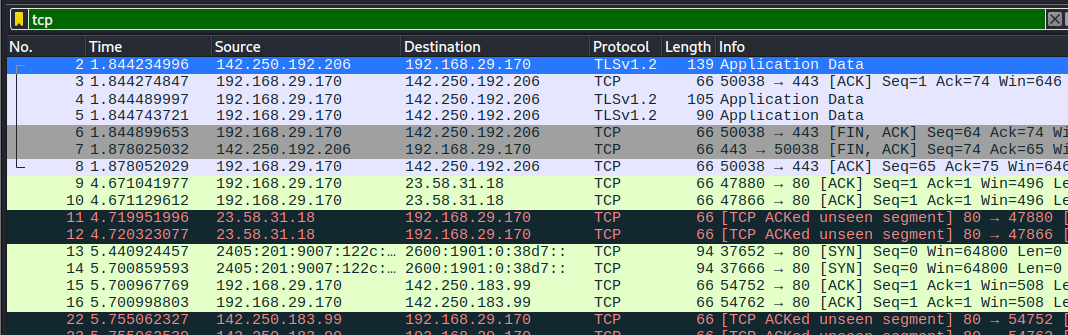


**5. Apply Protocol Filters**

* Use Wireshark’s filter bar to isolate specific protocols:
  + DNS: dns

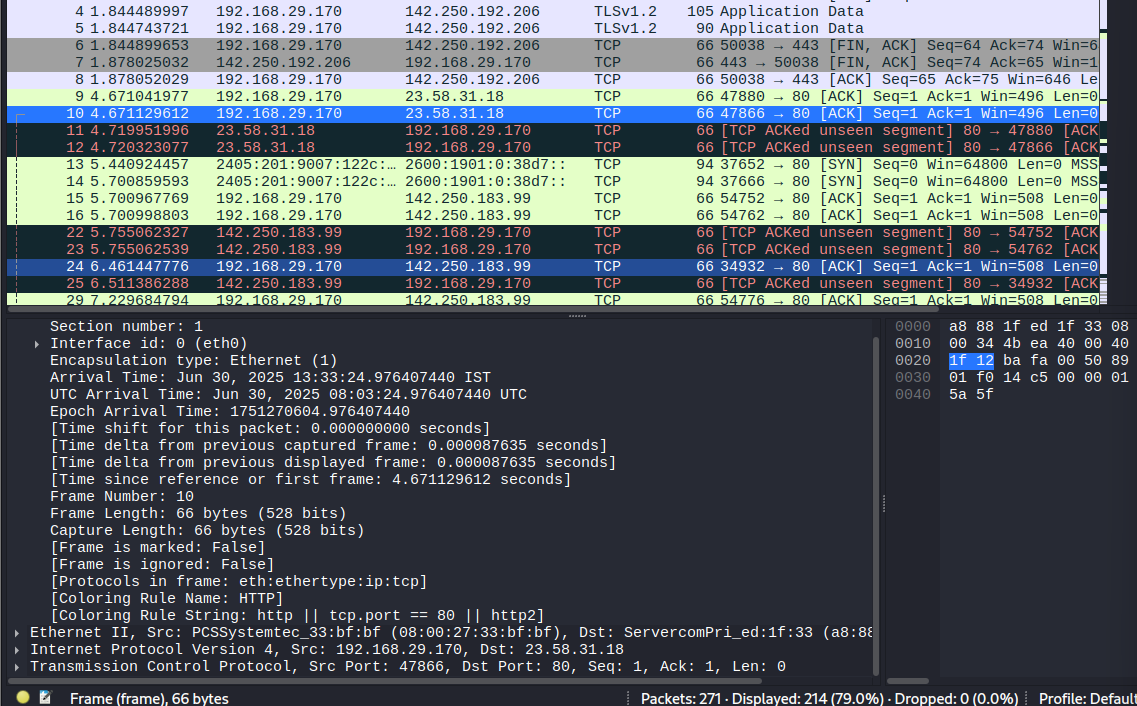


* + ICMP: icmp
  + HTTP: http
  + TCP: tcp



**6. Review Packet Details**

* Expand frame and protocol layers to inspect:
  + IP addresses
  + Port numbers
  + Request/response content



**7. Save the Capture**

* Export your work: File > Save As > capture.pcap

**8. Create This Report**

* Summarize identified protocols and key packet data (as detailed below).

**Protocols Identified & Analysis**

| **Protocol** | **Function** | **Example Observed** |
| --- | --- | --- |
| **DNS** | Resolves domain names to IPs | Query: google.com → DNS response from 8.8.8.8 |
| **ICMP** | Ping test traffic | Echo Request/Reply to/from Google |
| **HTTP** | Web communication | HTTP GET request to example.com |
| **TCP** | Session establishment | TCP SYN and ACK for HTTP |
| **ARP** | MAC/IP resolution | Local ARP queries to default gateway |

**Packet Highlights**

1. **DNS Query (Frame 19)**
   * **Source:** 192.168.1.10 → **Destination:** 8.8.8.8
   * **Query:** A record for google.com
2. **ICMP Packet (Frame 32)**
   * **Ping Request:** to 142.250.182.206
   * **Size:** 98 bytes
3. **HTTP GET (Frame 117)**
   * **Host:** example.com
   * **Request Method:** GET
   * **Response Code:** 200 OK

**Deliverables**

| **File** | **Description** |
| --- | --- |
| capture.pcap | Raw packet capture file (includes DNS, ICMP, HTTP) |
| Wireshark\_Report.pdf | This report with step-by-step analysis and findings |

**Conclusion**

This task demonstrated practical usage of Wireshark for live traffic analysis. The protocols identified represent key layers of network communication. These findings contribute to stronger understanding of packet behavior, protocol functions, and network security awareness.