Task 5: Capture and Analyze Network Traffic Using Wireshark
Objective: Capture live network packets and identify basic protocols and traffic types.
Tool Used: Wireshark

---

1. Installation of Wireshark

Wireshark was downloaded and installed from the official website https://www.wireshark.org/. It is a free and open-source packet analyzer used for network troubleshooting, analysis, and protocol development.

---

2. Starting Packet Capture

After launching Wireshark:

- The active network interface (e.g., Ethernet or Wi-Fi) was selected.
- Packet capture was started by clicking the "Start Capturing Packets" button.

---

3. Generating Network Traffic

To generate traffic:

- A website (e.g., www.google.com) was opened in a browser.
- This triggered web (HTTP), DNS, and TCP traffic.

---

4. Stopping Packet Capture

After around 1 minute of activity:

- The capture was stopped using the red "Stop" button in Wireshark.

---

5. Filtering Packets by Protocol

Wireshark provides a filter bar to view specific protocols. The following filters were applied:

- http to view HTTP traffic
- tcp to view TCP-level communication

- dns to view DNS queries and responses
- arp to observe ARP requests and replies within the local network

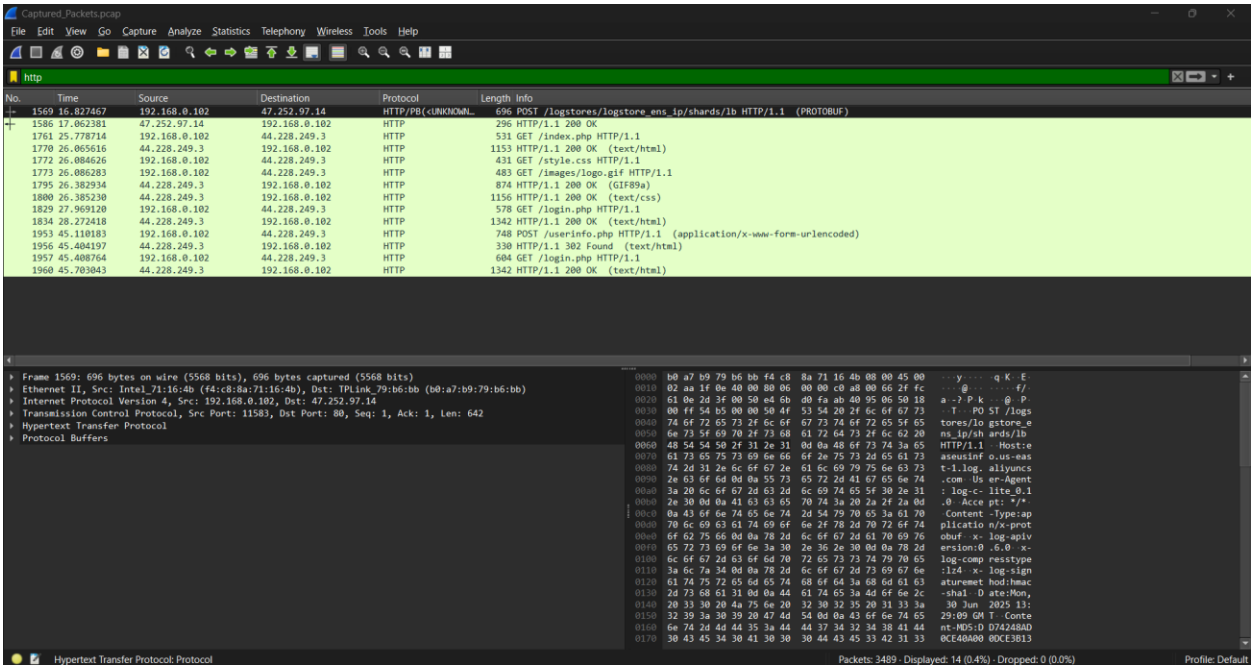
---

6. Protocols Identified

From the capture, the following protocols were identified:

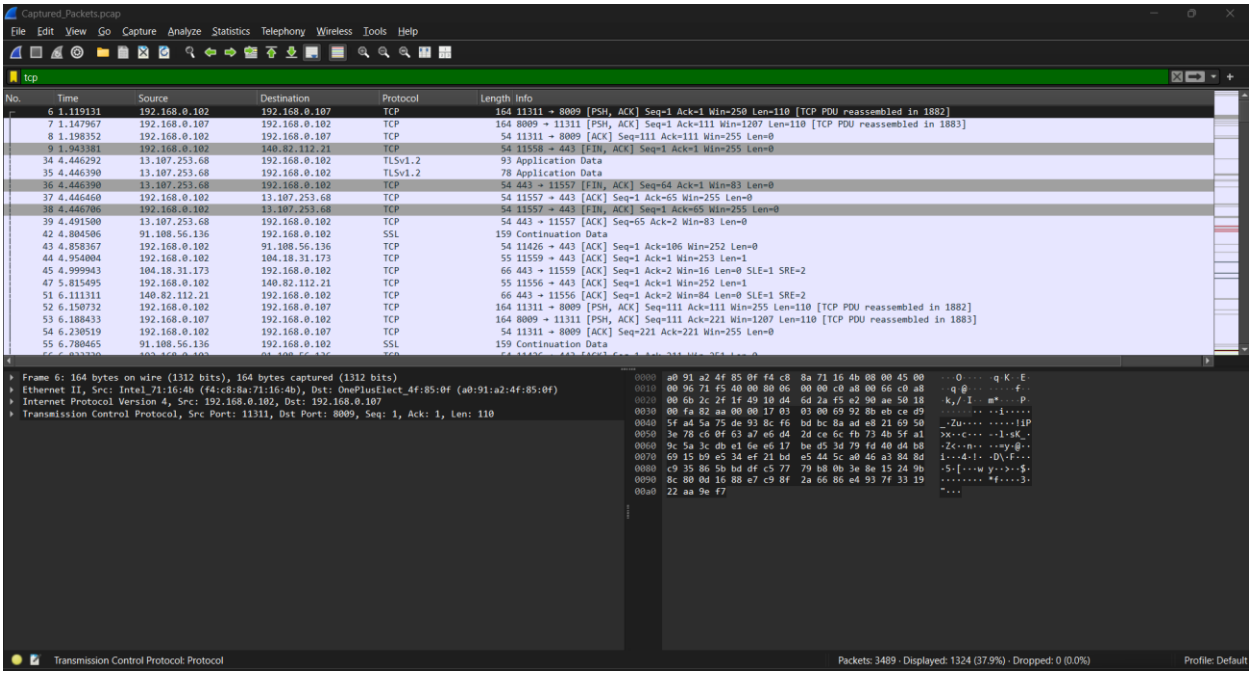
a. HTTP (Hypertext Transfer Protocol)

- Used for web browsing.
- Example: GET request to www.google.com.
- Contains headers like Host, User-Agent, Accept, etc.



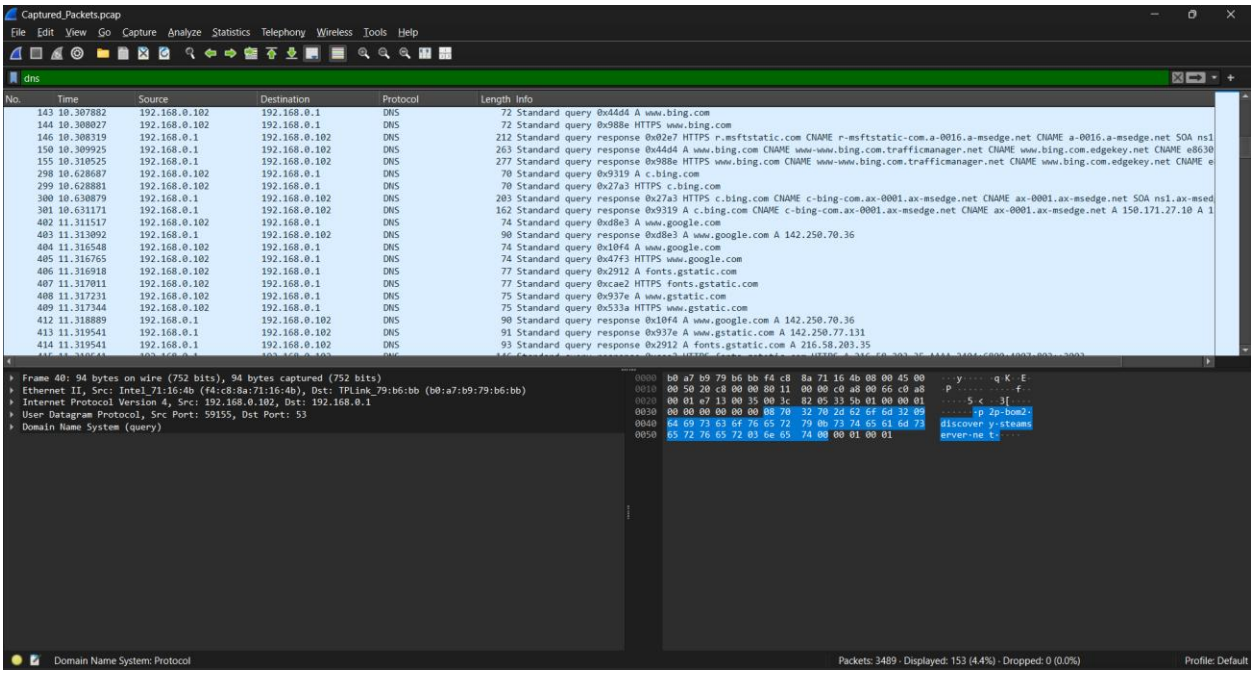
b. TCP (Transmission Control Protocol)

- Underlying transport protocol for HTTP.
- Responsible for connection setup (SYN), data transfer, and teardown (FIN).
- TCP handshakes and data segments were captured.



c. DNS (Domain Name System)

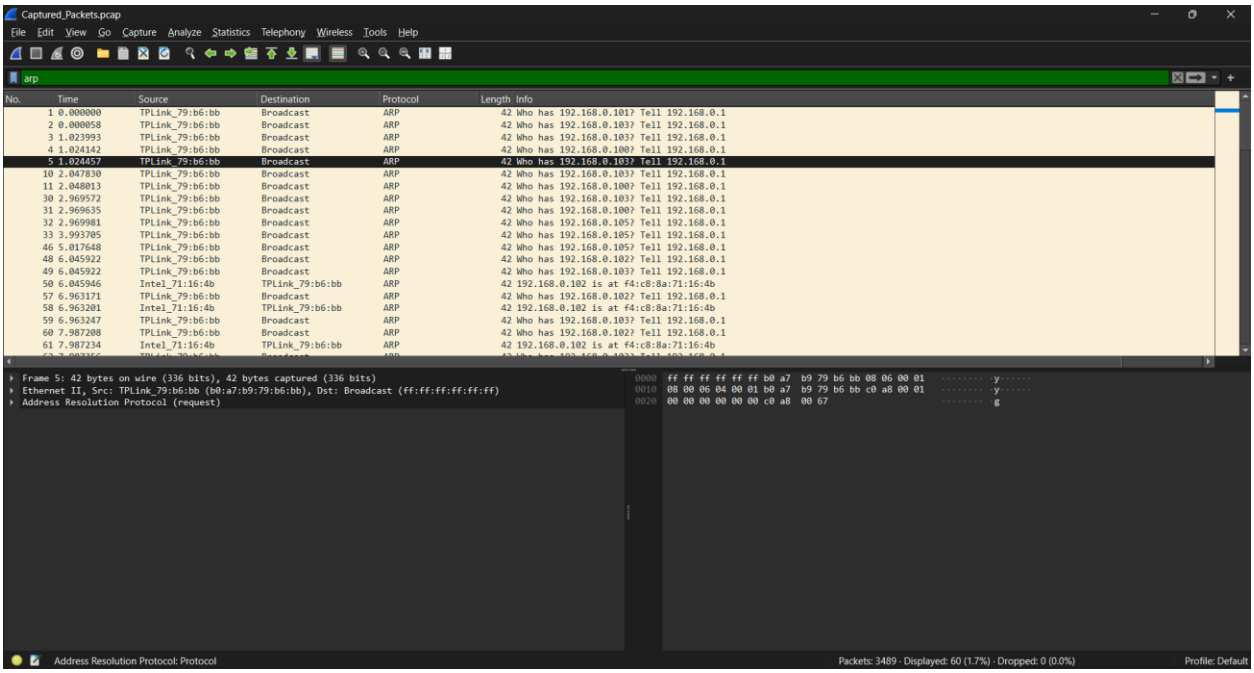
- Resolves domain names to IP addresses.
- Example: DNS query and response for www.google.com.



d. ARP (Address Resolution Protocol)

- Resolves IP addresses to MAC addresses in the local network.

- Captured ARP requests like “Who has 192.168.0.103? Tell 192.168.0.1”.



---

7. Exporting the Capture

The capture file was saved as a .pcap file using:

- File > Save As
- File name: Captured_Packets.pcap

---

8. Summary of Findings

Protocol	Function	Example Packet Details
HTTP	Web traffic	GET request to www.google.com
TCP	Data transport	SYN, ACK, FIN packets
DNS	Name resolution	Query for www.google.com
ARP	MAC resolution	Request: Who has 192.168.1.1?

Conclusion

The task demonstrated successful capture and analysis of network traffic using Wireshark. Four protocols (HTTP, TCP, DNS, and ARP) were filtered and analyzed. The exported .pcap file contains detailed records of captured packets and supports further inspection.