DHANALAKSHMI.C **241901023**

Exp no: 12

To Capture, save, and analyze network traffic on TCP / UDP / IP / HTTP / ARP /DHCP /ICMP/DNS using Wireshark Tool

Aim:

To capture, save, and analyze network traffic for various protocols such as TCP, UDP, IP, HTTP, ARP, DHCP, ICMP, and DNS using the Wireshark tool.

Introduction:

Wireshark is a widely used open-source network protocol analyzer that allows users to capture and interactively browse the traffic running on a computer network. It is a powerful tool for network troubleshooting, protocol development, and educational purposes.

Network protocols operate at different layers of the OSI model and each protocol serves a unique function:

- **IP (Internet Protocol)**: Handles addressing and routing of packets.
- **TCP (Transmission Control Protocol)**: Ensures reliable delivery of packets.
- **UDP (User Datagram Protocol)**: Provides a faster but unreliable transmission method.
- **HTTP (HyperText Transfer Protocol)**: Used for web communication.
- **ARP (Address Resolution Protocol)**: Resolves IP addresses to MAC addresses.
- **DHCP (Dynamic Host Configuration Protocol)**: Assigns IP addresses to hosts.
- **ICMP (Internet Control Message Protocol)**: Used for error messages and diagnostics (e.g., ping).
- **DNS (Domain Name System)**: Translates domain names to IP addresses.

Wireshark allows users to inspect packets for these protocols in real time or from saved capture files, making it easier to analyze communication between devices.

Algorithm / Procedure:

Step 1: Install Wireshark

- Download and install Wireshark from the official website: https://www.wireshark.org/

Step 2: Start Wireshark

- Open the application.
- Select the appropriate network interface (e.g., Wi-Fi or Ethernet) to begin capturing packets.

Step 3: Begin Packet Capture

- Click on the interface to start capturing live traffic.
- Optionally, apply **capture filters** to capture specific types of traffic.

Step 4: Generate Network Traffic

- Open a browser, access websites (for HTTP/HTTPS).
- Use ping command (for ICMP).
- Use nslookup or access websites (for DNS).
- Connect to other devices or services to generate TCP, UDP, DHCP, ARP traffic.

Step 5: Apply Display Filters in Wireshark

Use filters to isolate traffic:

- ip → View all IP traffic.
- tcp → View TCP packets.
- udp → View UDP packets.
- http → View HTTP traffic.
- arp → View ARP requests and replies.
- dhcp → View DHCP messages (use bootp filter as DHCP uses BOOTP protocol).
- icmp → View ICMP traffic (e.g., ping).

- dns → View DNS queries and responses.

Step 6: Analyze Packets

- Click on any packet to view detailed headers.
- Expand each protocol layer to understand source/destination, flags, port numbers, checksums, etc.

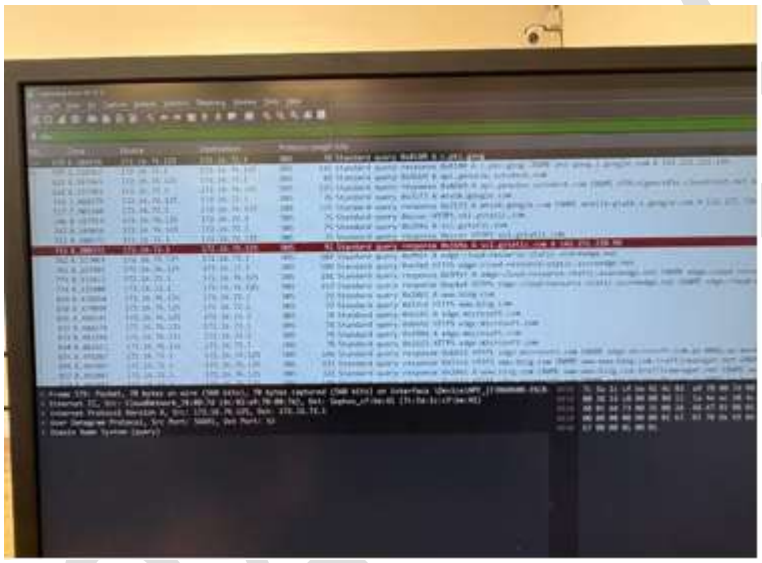
Step 7: Save the Capture

- Go to File > Save As, and save the capture in .pcapng format for future analysis.

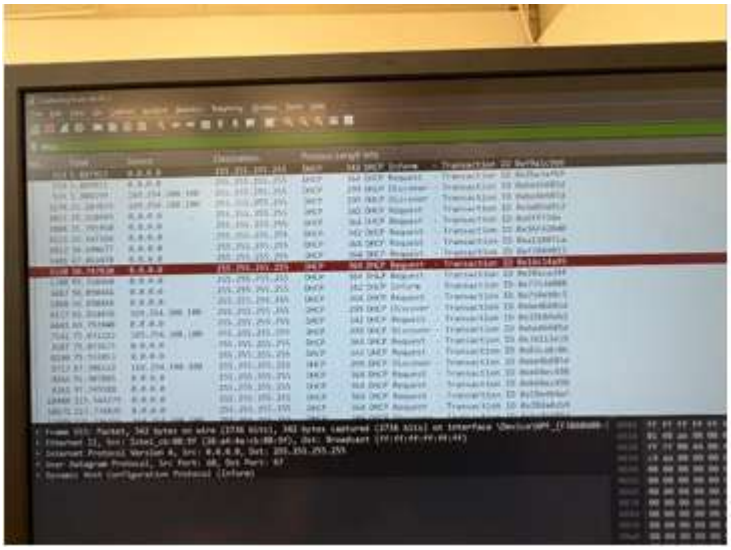
Step 8: Stop Capture

- Click the red square (stop) button once you have collected enough data.

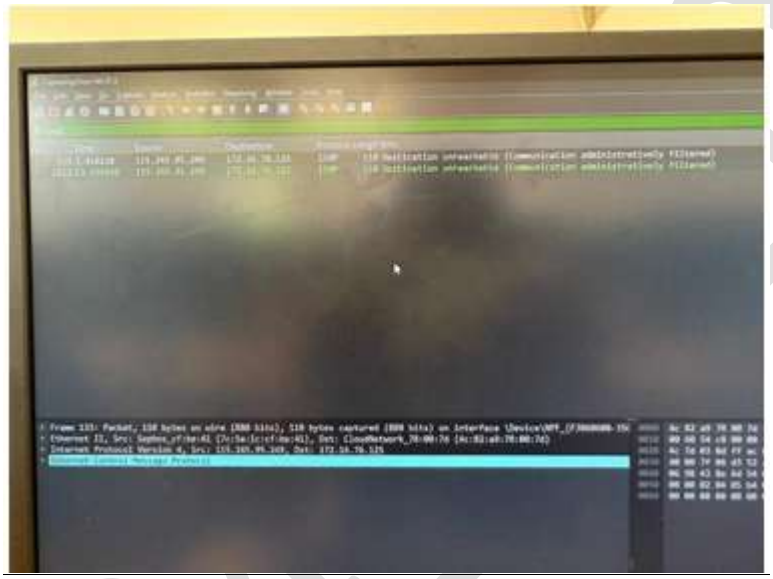
DNS:



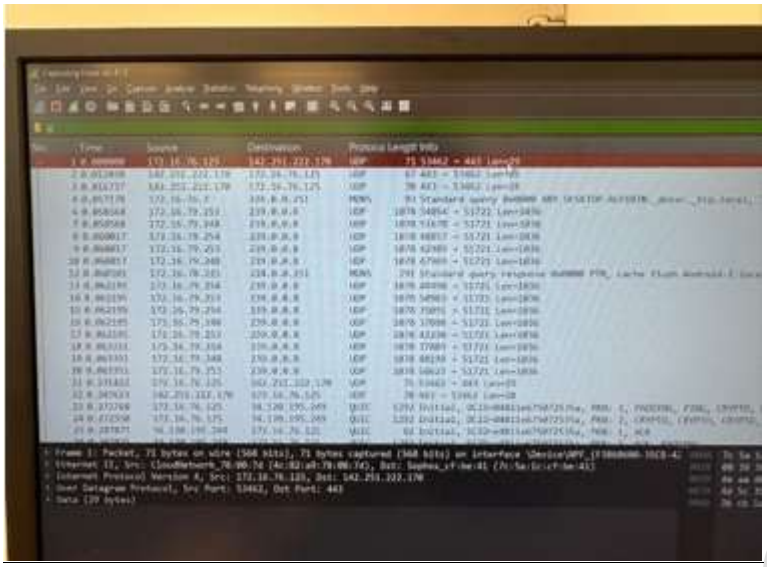
DHCP:



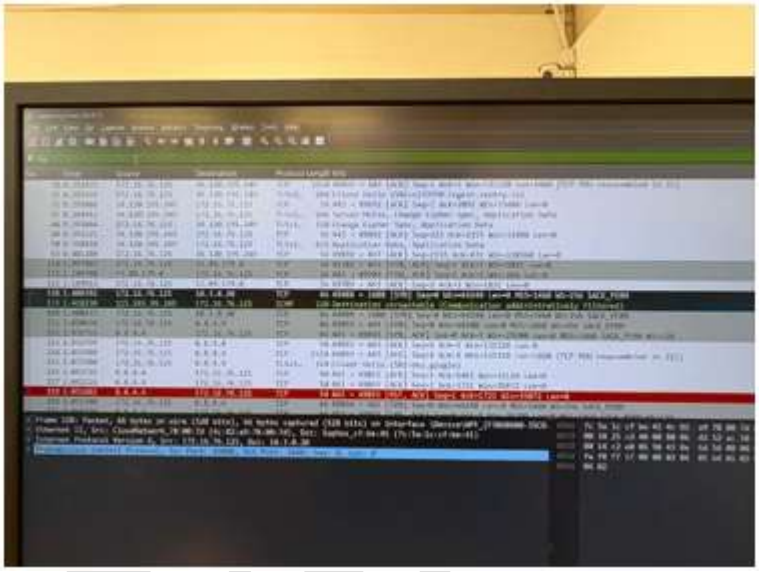
ICMP:



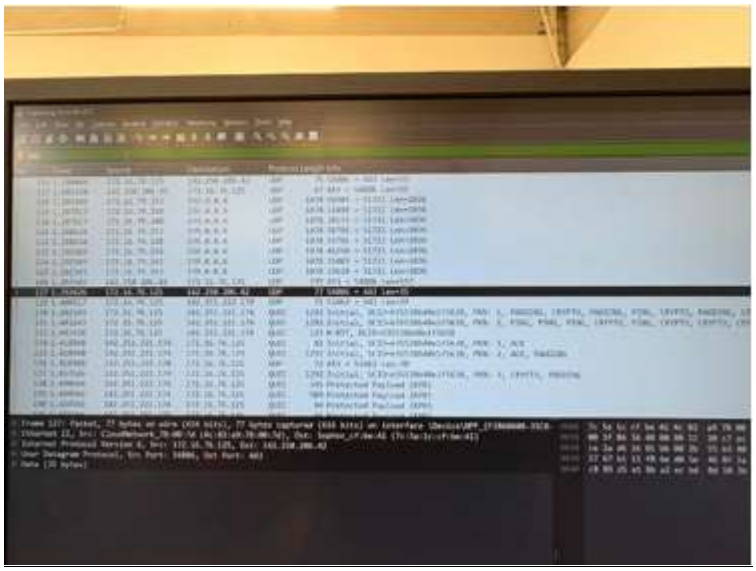
IP:



TCP:



UDP:



Result:

- Successfully captured and analyzed live network traffic using Wireshark.
- Observed and analyzed protocol-specific information for:
    - **TCP/UDP**: Source and destination ports, sequence and acknowledgment numbers.
    - **IP**: Source and destination IP addresses, TTL values.
    - **HTTP**: Request methods (GET, POST), status codes.
    - **ARP**: IP-to-MAC address resolution.
    - **DHCP**: IP lease offers and requests.
    - **ICMP**: Echo requests and replies.
    - **DNS**: Name resolution queries and responses.