Task 3: Networking Basics for Cyber Security

1.Learn basic networking concepts (IP, MAC, DNS, TCP/UDP).

a) IP Address (Internet Protocol Address) An **IP address** is a unique number given to a device on a network so it can be **identified and communicate** with other devices. **Example:** When your laptop connects to the internet, it may get an IP address like **192.168.1.10**. This address tells the network *where to send data meant for your laptop*.

b) MAC Address (Media Access Control Address) A **MAC address** is a **permanent, physical address** assigned to a network device by the manufacturer. It helps identify the device within a local network. **Example:** Your laptop's Wi-Fi card has a MAC address like **00:1A:2B:3C:4D:5E**, which is used by the router to recognize your laptop on the local network.

c) DNS (Domain Name System) **DNS** converts **human-friendly website names** into IP addresses that computers understand. **Example:** When you type **www.google.com** in a browser, DNS translates it into an IP address like **142.250.190.14** so your browser can reach Google's server.

d) TCP (Transmission Control Protocol) **TCP** is a communication method that ensures data is **sent accurately and in the correct order**. It is reliable. **Example:** When you download a PDF file, TCP makes sure **all parts of the file arrive correctly** and requests missing parts if any data is lost.

e) UDP (User Datagram Protocol) **UDP** is a faster but less reliable communication method. It sends data **without checking if it arrives correctly**. **Example:** During a **video call**, UDP is used because speed is more important than perfect accuracy. If a few packets are lost, the call continues without delay.

f) Router A **router** is a device that connects **different networks** and directs data to the correct destination. **Example:** Your home Wi-Fi router sends data from your laptop to the internet and brings responses back to your laptop.

g) Switch A **switch** connects **multiple devices within the same local network** and sends data only to the intended device. **Example:** In an office, a switch connects computers, printers, and servers so they can communicate efficiently.

h) Gateway A **gateway** acts as a **bridge between two different networks** that use different rules or protocols. **Example:** A router often acts as a gateway between your home network and the internet.

i) Firewall A **firewall** is a security system that **controls incoming and outgoing network traffic** based on rules. **Example:** A firewall blocks unauthorized access while allowing safe traffic like web browsing.

j) Port A **port** is a virtual endpoint that helps a computer **identify which application should receive data**. **Example:** Web traffic uses **port 80 (HTTP)** or **port 443 (HTTPS)**.

k) Protocol A **protocol** is a set of rules that determines **how data is transmitted** over a network. **Example:** HTTP is a protocol used to transfer web pages from a server to a browser.

l) LAN (Local Area Network) A **LAN** is a network that connects devices within a **small area** like a home or office. **Example:** All devices connected to your home Wi-Fi form a LAN.

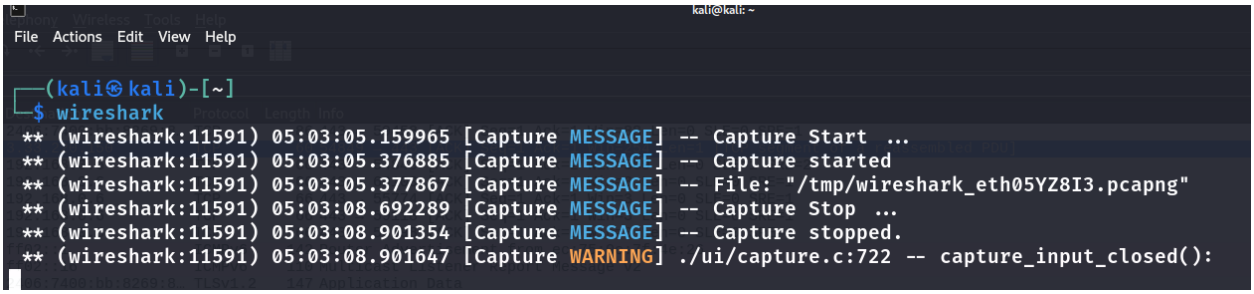
m) WAN (Wide Area Network) A **WAN** connects devices over a **large geographical area**. **Example:** The internet is the largest example of a WAN.

n) Bandwidth: Bandwidth refers to the **maximum amount of data** that can be transferred over a network in a given time. **Example:** A 100 Mbps internet connection can transfer more data per second than a 10 Mbps connection.

o) Latency: Latency is the **delay** between sending data and receiving a response. **Example:** High latency causes noticeable delays in online gaming or video calls.

2.Install Wireshark and capture live network traffic.

The application wireshark comes with inbuilt in kali Linux therefore the step to install is not followed. The second step on capturing the file is displayed below.

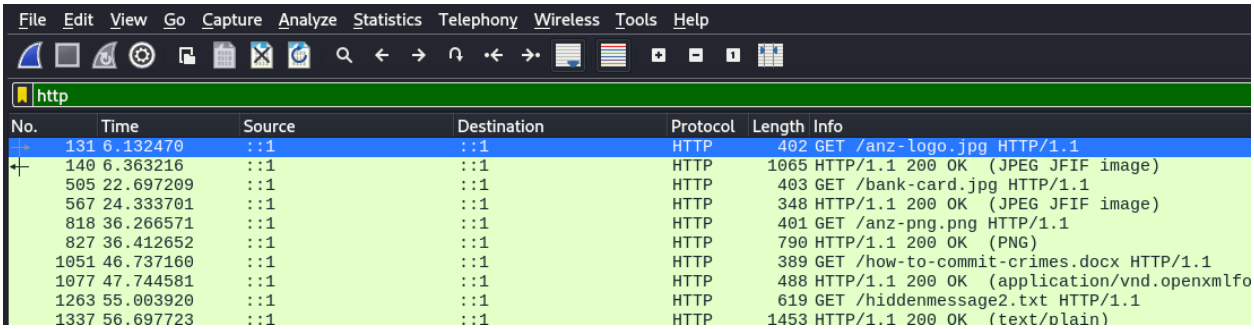


3.Filter packets by protocol (HTTP, DNS, TCP)

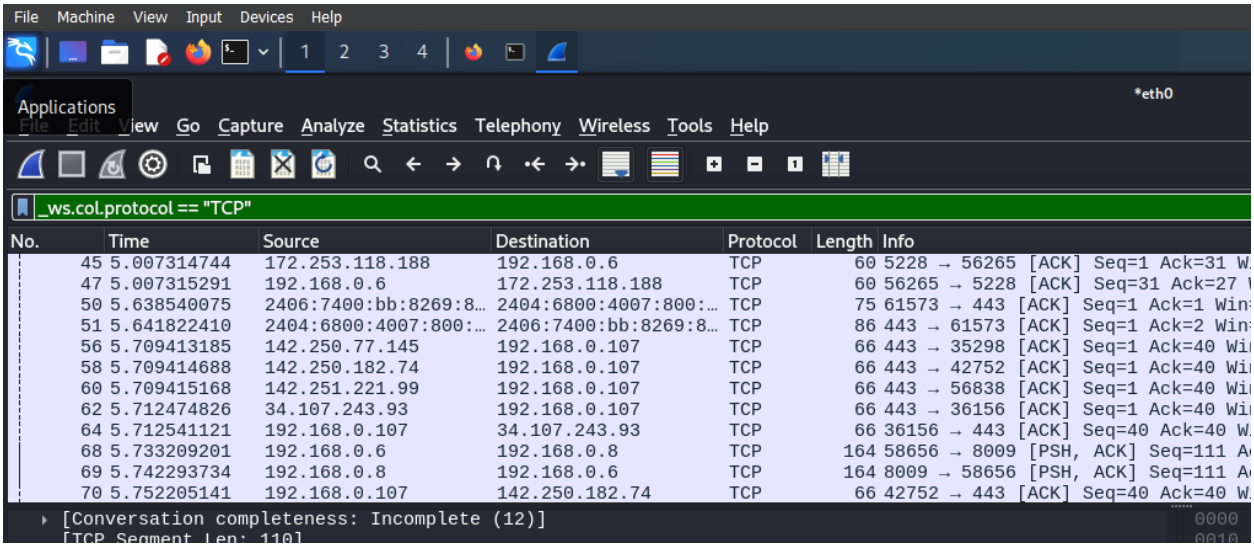
Filtering Packets by Protocol Using Wireshark (Brief Explanation)

Wireshark allows users to **filter and analyze network packets** based on different protocols using **display filters**.

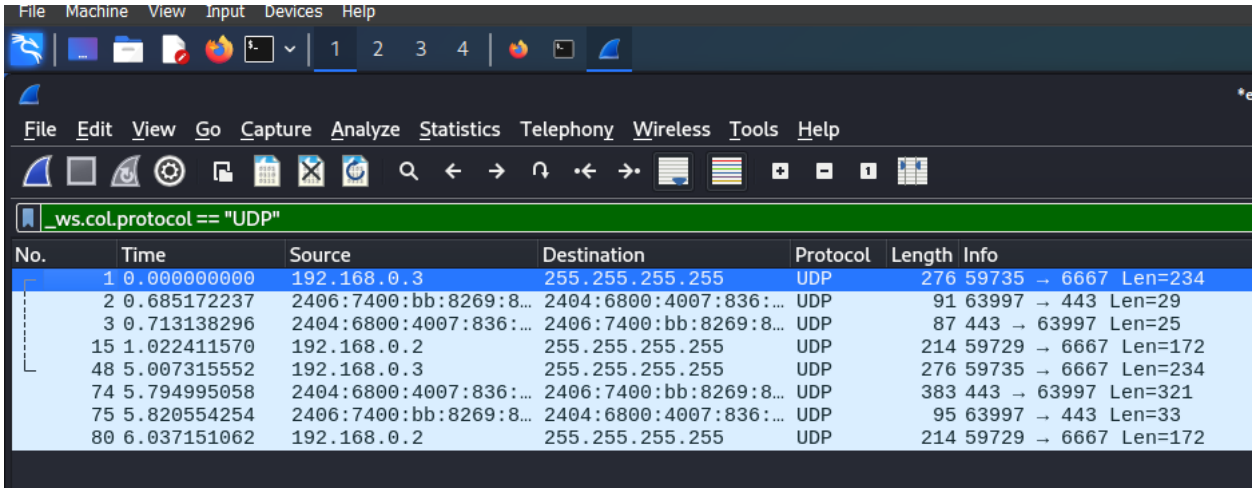
- **HTTP packets:**
  To view only HTTP (web) traffic, type the following in the Wireshark display filter bar:
  http



- **TCP packets:**
  To filter and view TCP packets, use the display filter:
  tcp



- **UDP packets:**
  To filter and view UDP packets, use the display filter:
  udp



After entering the filter, press **Enter**, and Wireshark will display only the packets related to the selected protocol.

👉 This method helps in **network troubleshooting, traffic analysis, and security investigation**.

4.Observe three-way TCP handshake.

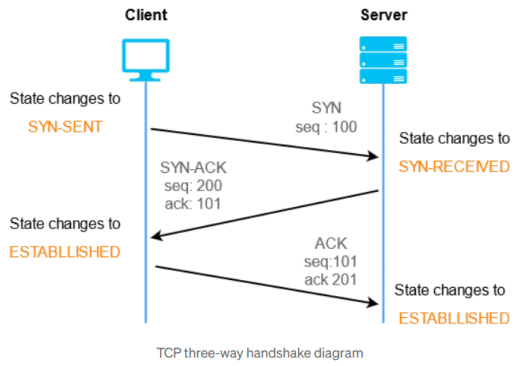
TCP Three-Way Handshake (Short Explanation)

The **TCP three-way handshake** is a process used to **establish a reliable connection** between a client and a server before data transfer begins.

Steps:

1. **SYN** – The client sends a SYN (synchronize) message to the server to request a connection.
2. **SYN-ACK** – The server responds with a SYN-ACK (synchronize-acknowledge) message to accept the request.
3. **ACK** – The client sends an ACK (acknowledge) message back to the server, confirming the connection.

Example:
When you open a website, your browser and the web server perform this three-step process to ensure both sides are ready to communicate.



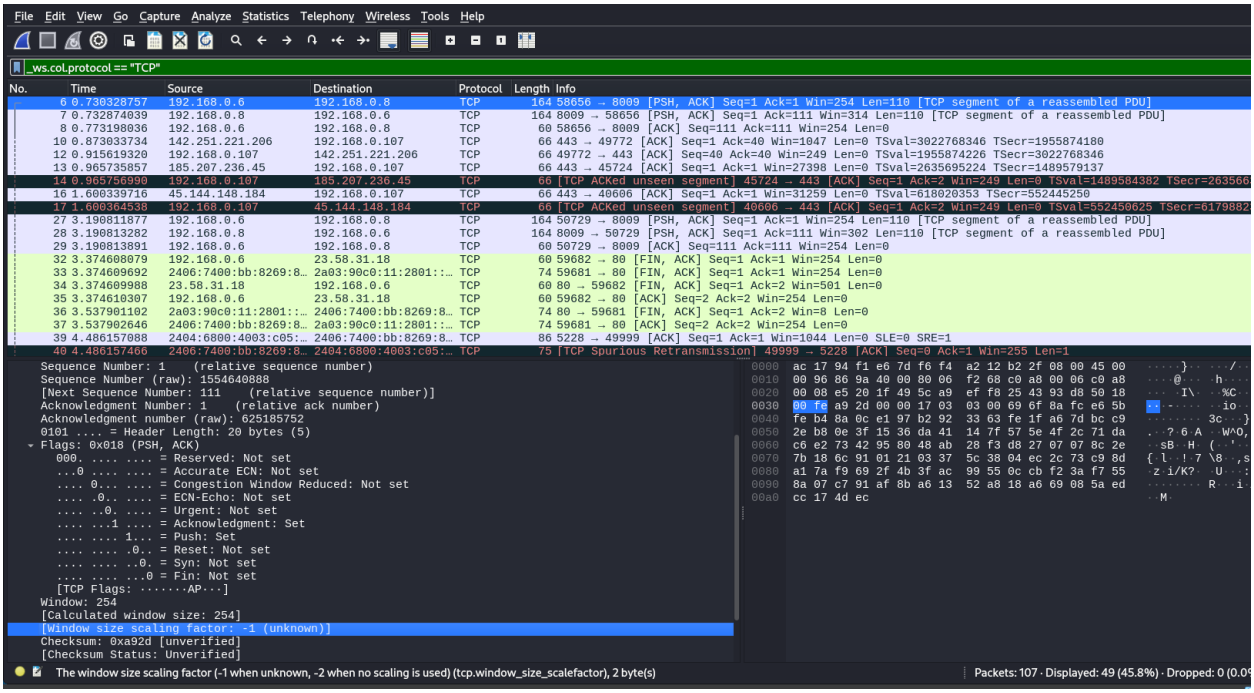
TCP three-way handshake diagram

In the establishment of a TCP connection between a client and a server, a TCP three-way handshake process is performed.

1. The client sends a SYN (synchronize) packet to the server, which has a random sequence number. (Protocol analyzers like wireshark will often use a *relative sequence number of 0* since it's easier to read than some high random number.)

2. The server sends back a SYN-ACK packet, containing a random sequence number and an ACK number acknowledging the client's sequence number.
3. The client sends an ACK number to the server, acknowledging the server's sequence number.
4. The sequence numbers on both ends are synchronized. Both ends can now send and receive data independently

The screenshot given below explains the 3-way handshake using Wireshark.

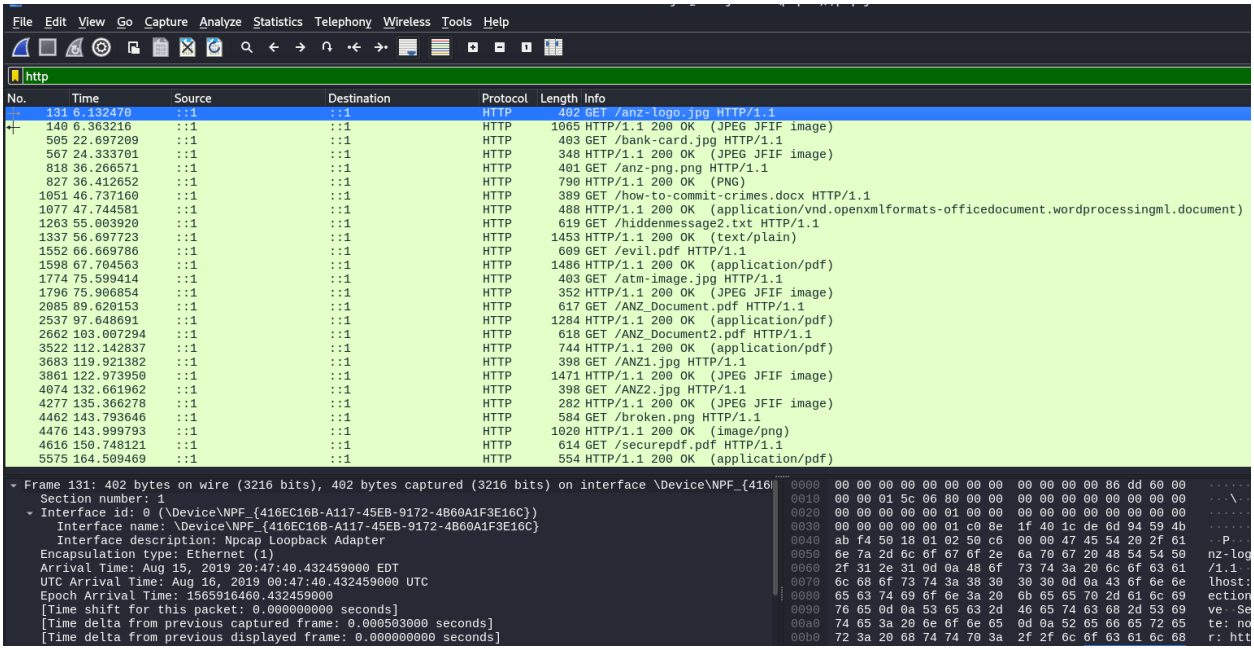


5.Identify plain-text traffic vs encrypted traffic.

Identifying Plain Text Traffic and Encrypted Traffic Using Wireshark

Wireshark can be used to distinguish between **plain text traffic** and **encrypted traffic** by analyzing the protocols and packet contents.

Plain text traffic is data that is transmitted in a readable format. To identify plain text traffic in Wireshark, apply display filters for protocols such as **HTTP**, **FTP**, **Telnet**, or **DNS** (for queries). For example, using the filter http will show HTTP packets. When you click on an HTTP packet and expand the **Packet Details** pane, you can clearly read information such as URLs, headers, and sometimes usernames or data being sent. This visibility confirms that the traffic is not encrypted. As given below:



Encrypted traffic is data that is protected using encryption, making it unreadable to unauthorized users. Common encrypted protocols include **HTTPS**, **TLS**, **SSL**, and **SSH**. To identify encrypted traffic, apply filters such as tls or ssl. When analyzing these packets, the packet content appears as random or unreadable data, and sensitive details like website content or credentials are hidden. For example, HTTPS traffic shows a **TLS handshake** instead of readable web data.

By comparing packet contents, one can easily identify plain text traffic (readable data) and encrypted traffic (unreadable, scrambled data). This analysis is important for **security assessment**, as encrypted traffic helps protect sensitive information from interception.

6.Capture DNS queries and analyze them.

Capturing and Analyzing DNS Queries Using Wireshark

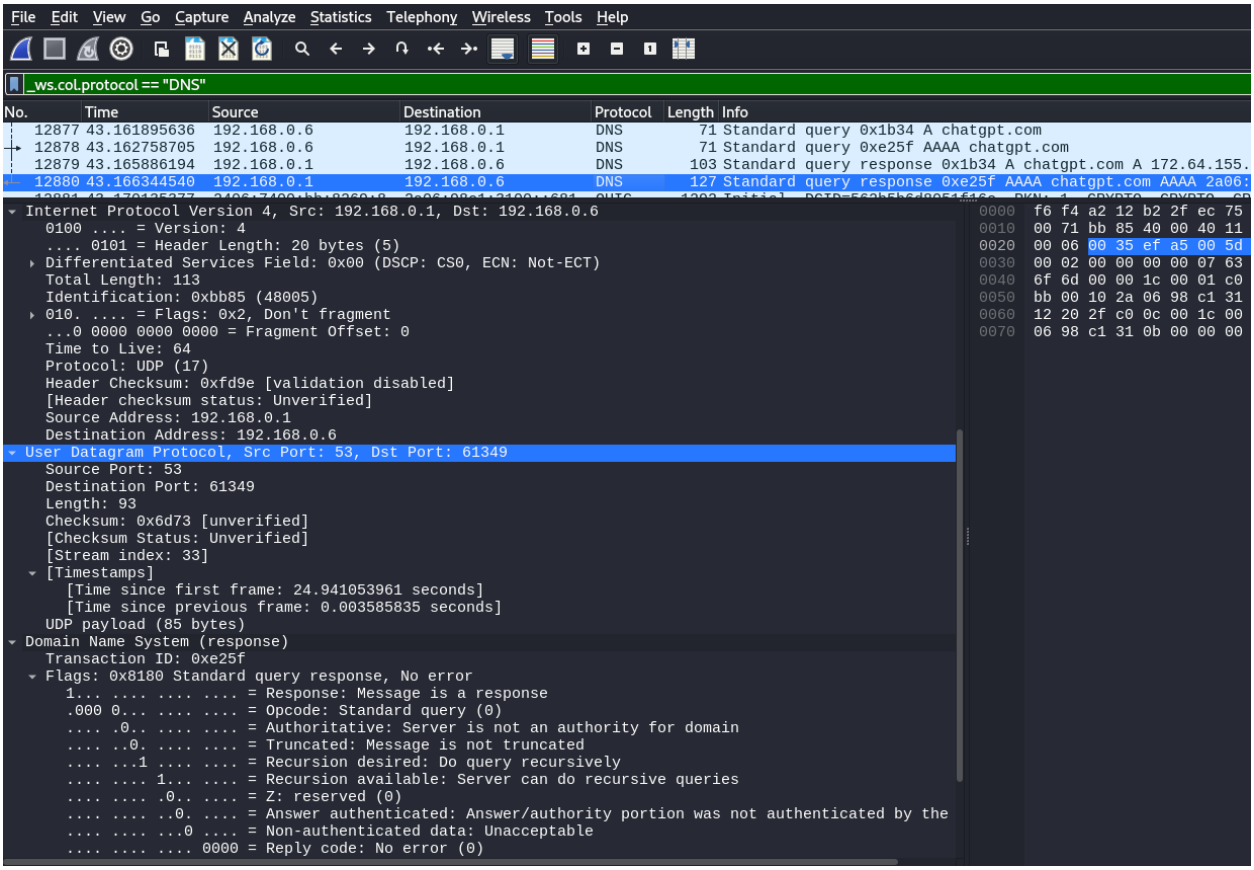
Wireshark can be used to capture and analyze **DNS (Domain Name System) queries**, which helps in understanding how domain names are resolved into IP addresses.

To begin, open **Wireshark** and select the active network interface (such as Wi-Fi or Ethernet). Click the **Start Capture** button to begin capturing network traffic. Once the capture starts, perform an activity that generates DNS traffic, such as opening a web browser and visiting a website (for example, www.google.com).

To view only DNS-related packets, type **dns** in the **display filter bar** and press **Enter**. Wireshark will then display only DNS query and response packets. DNS **queries** are usually sent by the client to request the IP address of a domain name, while DNS **responses** are sent by the DNS server containing the corresponding IP address.

Click on a DNS packet to analyze it in detail. In the **Packet Details** pane, expand the **Domain Name System (DNS)** section. Here, you can observe important information such as the **query name** (requested domain), **query type** (A, AAAA, MX, etc.), **response IP address**, and **time taken for resolution**.

By analyzing DNS queries in Wireshark, one can identify which domains are being accessed, verify correct DNS resolution, and detect suspicious or malicious domain activity. This process is useful for **network troubleshooting, performance monitoring, and security analysis**.

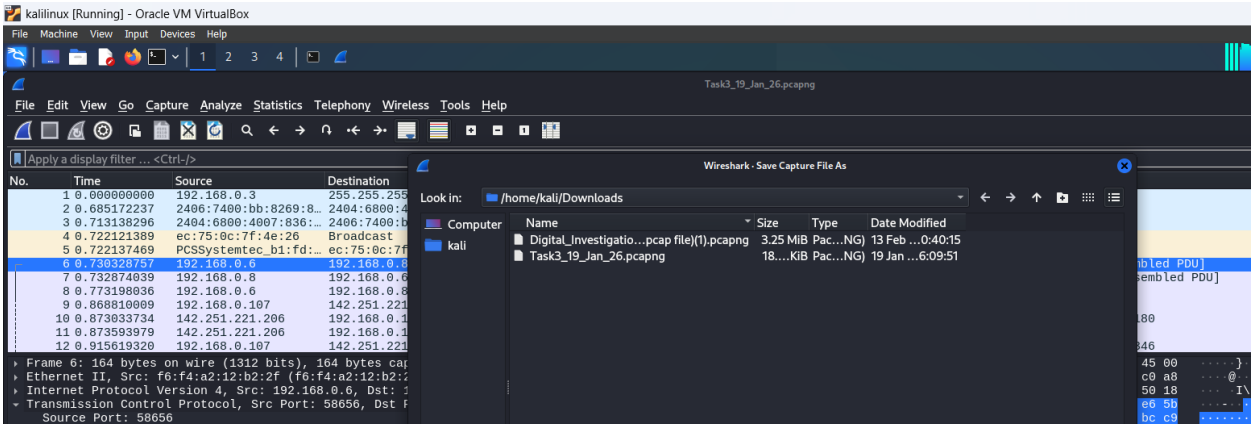


7.Save packet captures for analysis.

Saving the Captured File in Wireshark

After completing packet capture in Wireshark, the captured data can be saved for future analysis or reporting. To save the captured file, first click on the **File** menu located at the top of the Wireshark window. From the dropdown list, select **Save As**. A dialog box will appear asking for the file name and location. The location usually goes to downloads folder. Enter the file name as **Task2_19_Jan_26** and choose an appropriate folder on the system. Ensure the file is saved in the default **.pcapng** format, which preserves all captured packet details. Finally, click **Save** to store the captured network traffic successfully.

This saved file can later be reopened in Wireshark for further examination or submission as part of the assignment.



8.Write observations in simple language

In this experiment, basic networking concepts such as **IP address, MAC address, DNS, and TCP/UDP** were learned and understood. **Wireshark** was successfully installed and used to capture live network traffic from the system. Different types of packets were filtered using protocol filters like **HTTP, DNS, and TCP**, which helped in identifying specific network communications.

The **TCP three-way handshake** was clearly observed, showing how a connection is established between a client and a server. Plain-text traffic such as **HTTP** was identified by readable data, while encrypted traffic like **HTTPS/TLS** was recognized by unreadable, secured data. DNS queries were captured and analyzed to understand how domain names are resolved into IP addresses. Finally, the captured packets were successfully saved for future analysis.