# Cybersecurity experiments

B.tech (Dr. A.P.J. Abdul Kalam Technical University)

# IMS ENGINEERING COLLEGE

# PRACTICAL FILE

# CYBER SECURITY WORKSHOP

# (BCS 453)

# B.TECH

# (II YEAR – EVEN SEM)

# (2023-24)

| | |
|---|---|
| **Name** | |
| **Roll No.** | |

## DEPARTMENT OF CSE-AIML

## IMS ENGINEERING COLLEGE

**(Affiliated to Dr A P J Abdul Kalam Technical University, Lucknow )**

Approved by AICTE - Accredited by NAAC – 'A' Grade
NH#24, Adhyatmik Nagar, Ghaziabad, UP, India
www.imsec.ac.in

# INDEX

| S.NO | TITLE OF EXPERIMENT | DATE OF SUBMISSION | FACULTY SIGNATURE |
|---|---|---|---|
| 1 | | | |
| 2 | | | |
| 3 | | | |
| 4 | | | |
| 5 | | | |
| 6 | | | |
| 7 | | | |
| 8 | | | |
| 9 | | | |
| 10 | | | |
| 11 | | | |
| 12 | | | |

# EXPERIMENT-1

**AIM:- Basic packet inspection: Capture Network traffic using Wireshark and analyze basic protocols like HTTP, DNS and SMTP to understand and how data is transmitted and received.**

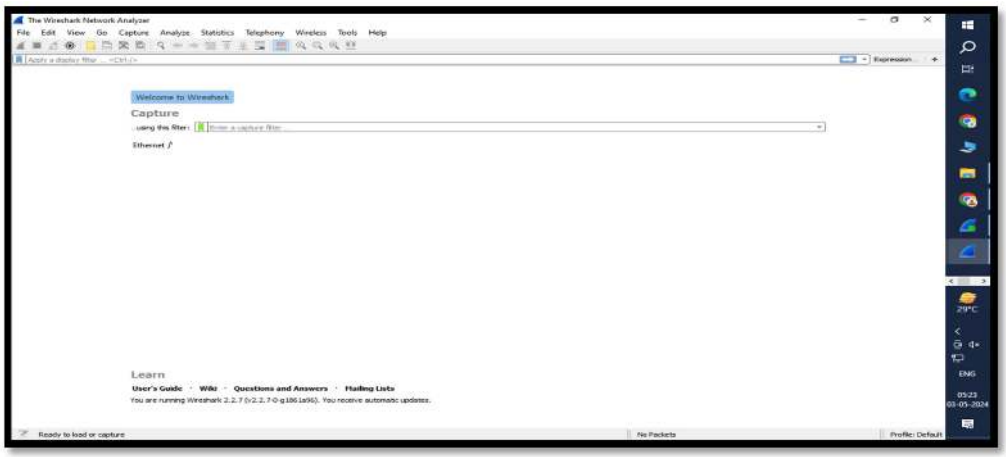
## 1. WIRESHARK:

Wireshark is a widely used, open source network analyzer that can capture and display real-time details of network traffic. It is particularly useful for troubleshooting network issues, analyzing network protocols and ensuring network security. Networks must be monitored to ensure smooth operations and security. Wireshark is a free and open-source packet analyzer. It is used for network troubleshooting, analysis, software and communications protocol development, and education. Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.

### 1.1 Main window:

Wireshark's main window consists of parts that are commonly known from many other GUI programs.

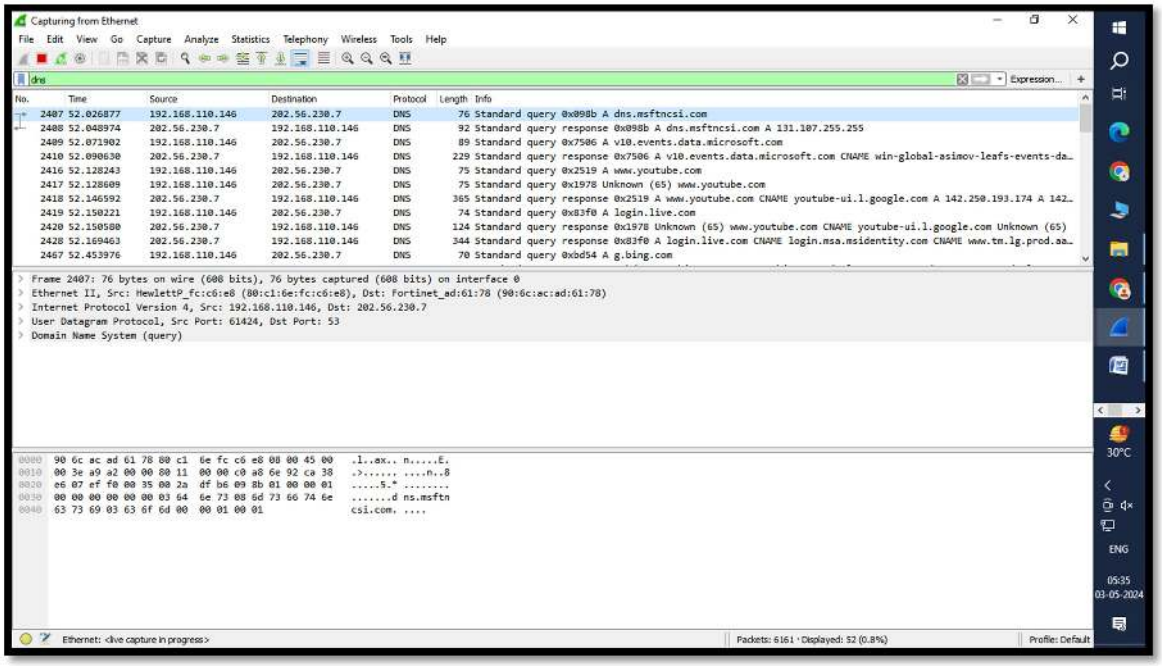
1. The *menu* is used to start actions.
2. The *main toolbar* provides quick access to frequently used items from the menu.
3. The *filter toolbar* allows users to set *display filters* to filter which packets are displayed.
4. The *packet list pane* displays a summary of each packet captured. By clicking on packets in this pane you control what is displayed in the other two panes.
5. The *packet details pane* displays the packet selected in the packet list pane in more detail.
6. The *packet bytes pan* displays the data from the packet selected in the packet list pane, and highlights the field selected in the packet details pane.
7. The *packet diagram pane* displays the packet selected in the packet list as a textbook-style diagram.
8. The *statusbar* shows some detailed information about the current program state and the captured data.



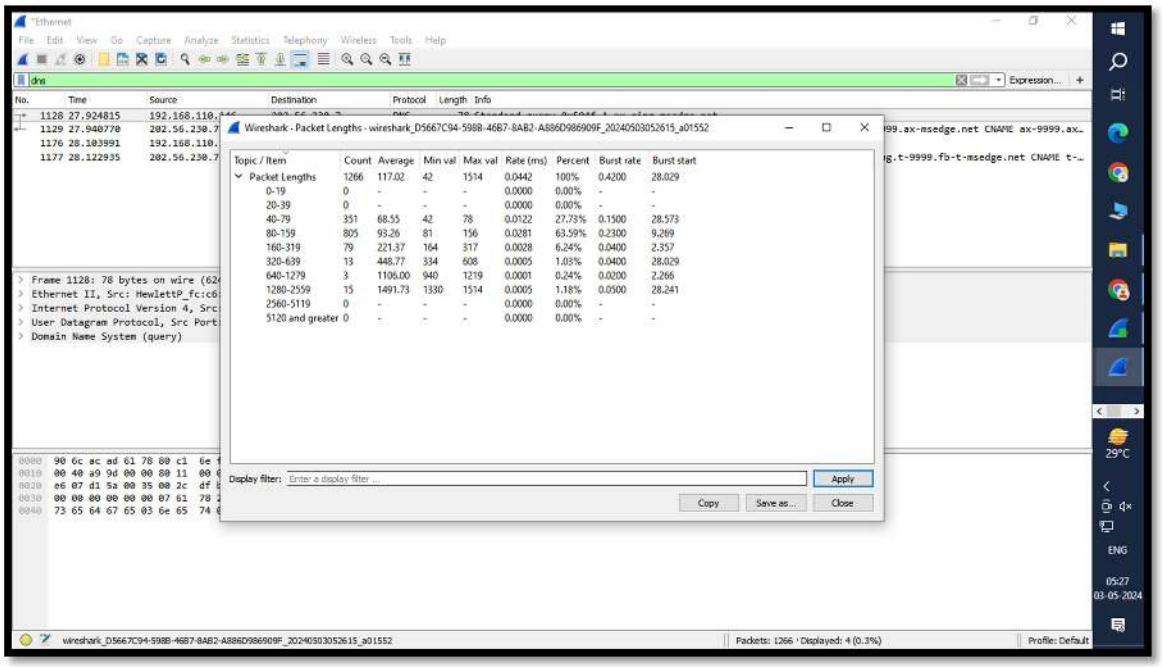

1.1 Main Window

## 1.2 DNS (Domain Name System):

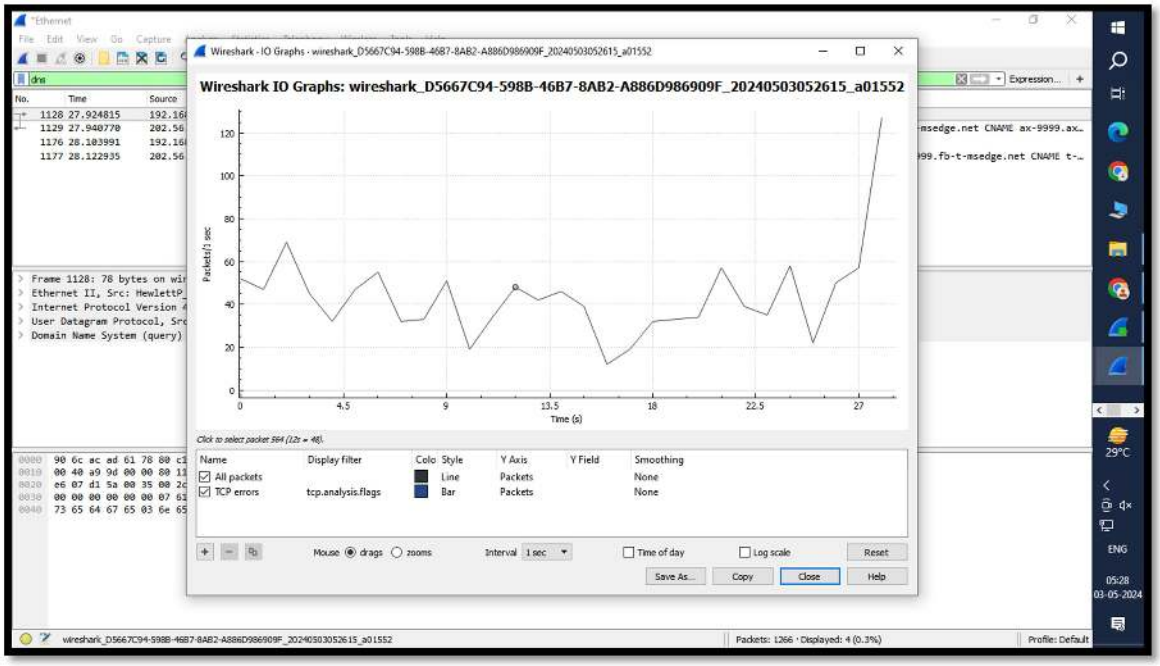
The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources.




1.2.1 DNS Capturing window



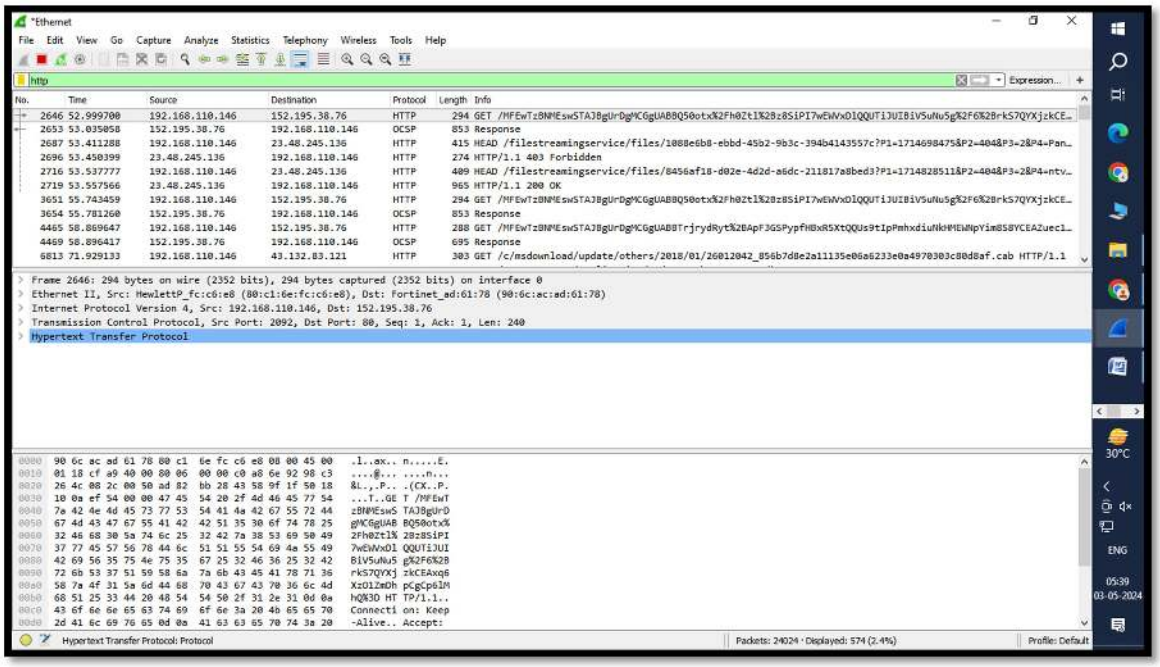

1.2.2 DNS Packet Length



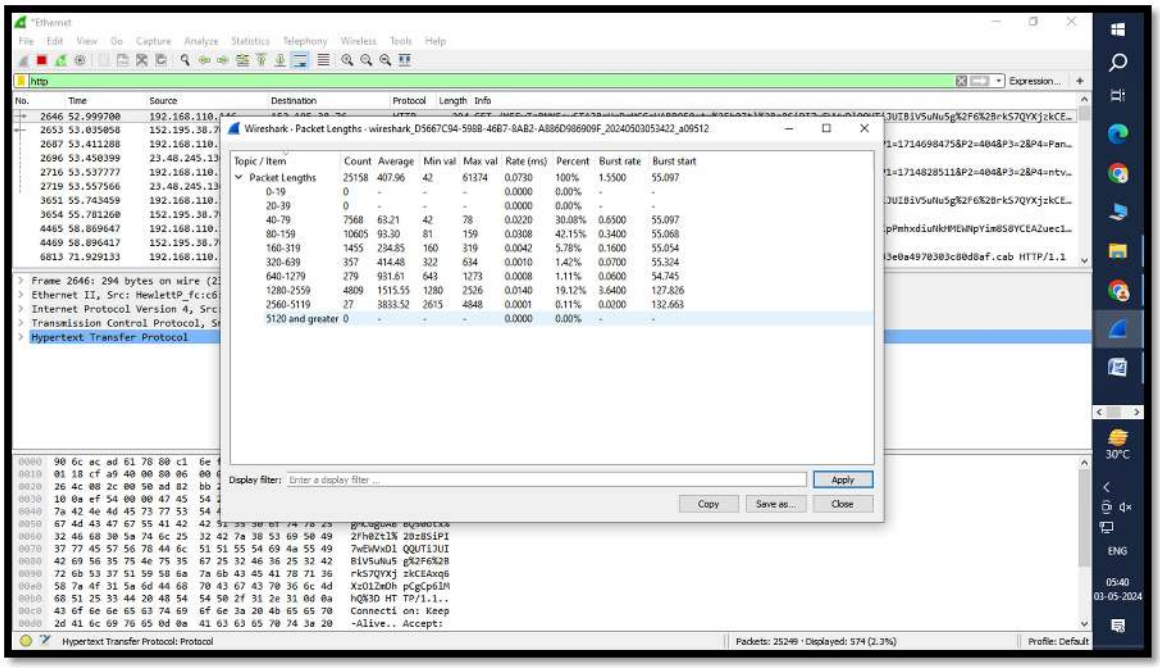

1.2.3 DNS I/O GRAPH

## 1.3 HTTP (Hyper Text Transfer Protocol):

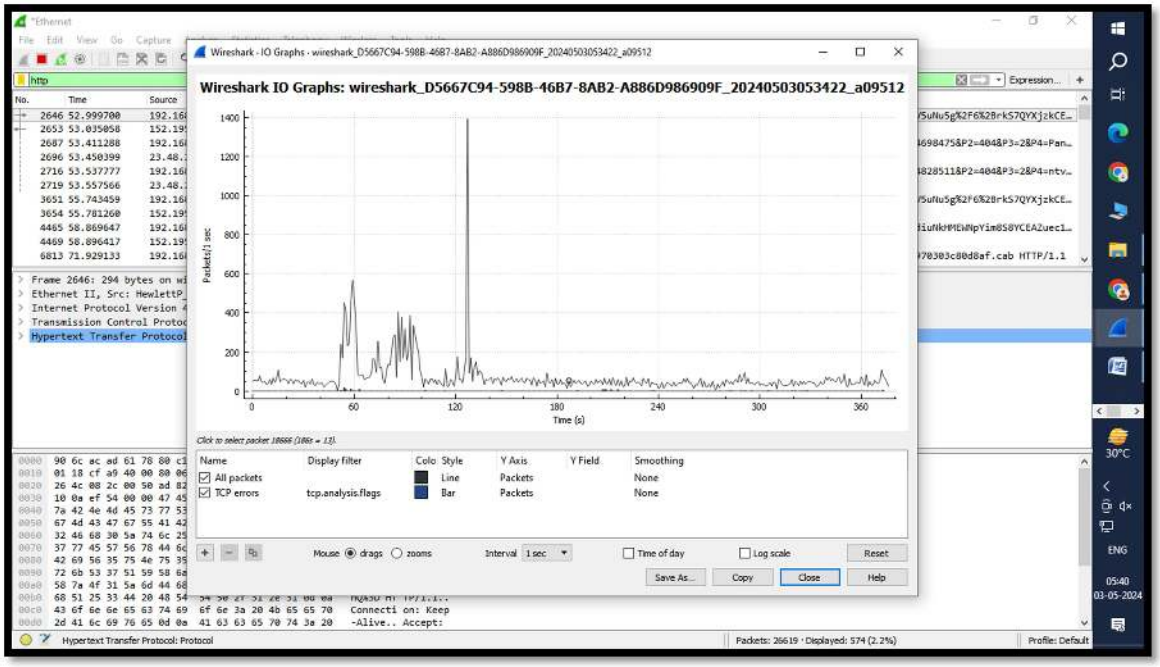
The Hypertext Transfer Protocol (HTTP) is the foundation of the World Wide Web, and is used to load webpages using hypertext links. HTTP is an application layer protocol designed to transfer information between networked devices and runs on top of other layers of the network protocol stack. A typical flow over HTTP involves a client machine making a request to a server, which then sends a response message.



1.3.1 HTTP Capturing Window



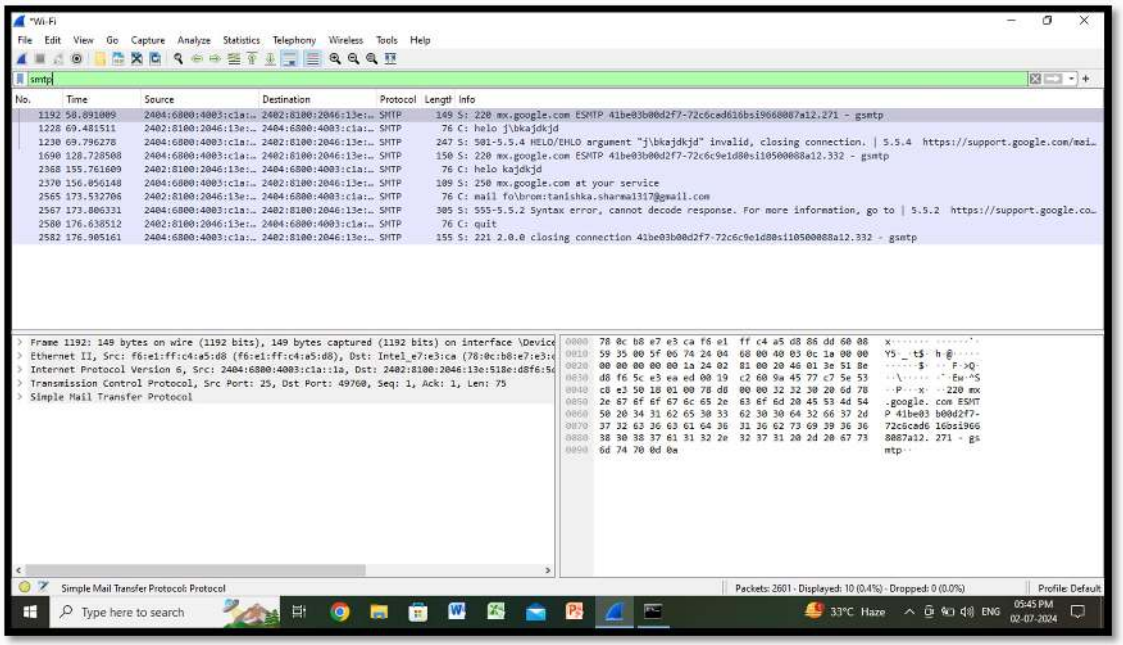

1.3.2 HTTP Packet Lengths



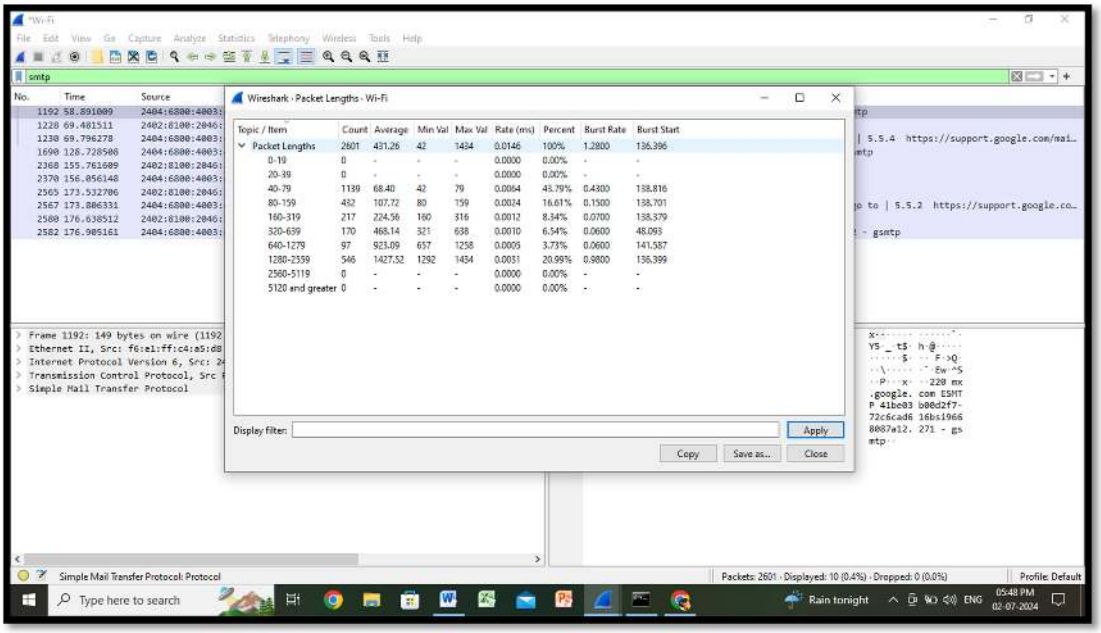

1.3.3 HTTP I/O Graph

## 1.4 SMTP (Simple Mail Transfer Protocol):

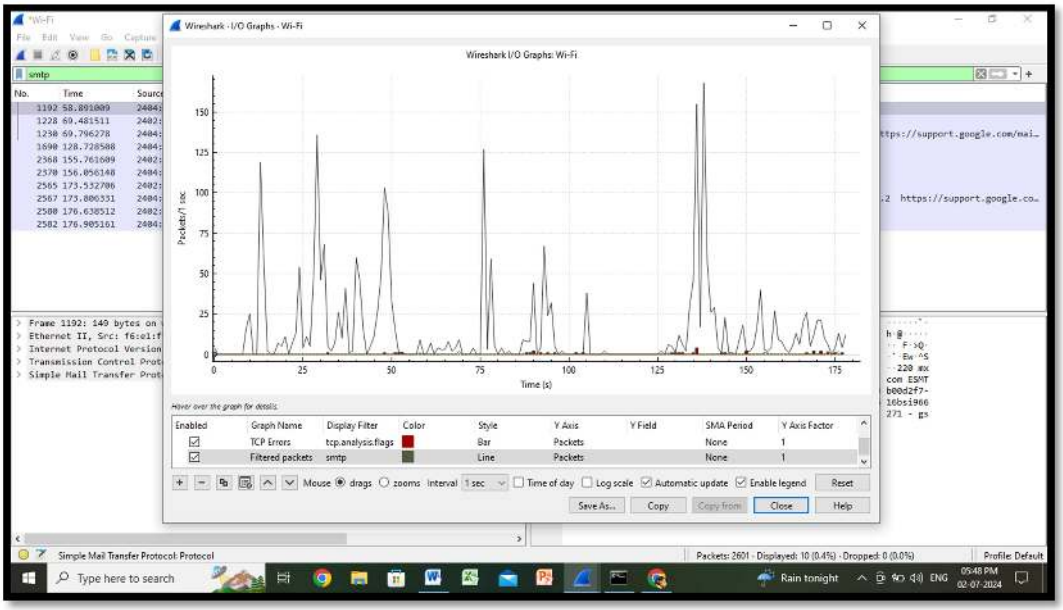
An SMTP server, also known as an outgoing mail server, is a computer or software that handles outgoing email messages. Generally, a mail server refers to a system that gathers, handles, and delivers email. An SMTP server refers specifically to the component of the mail server that uses the Simple Mail Transfer Protocol (SMTP) to send outgoing mail.



1.4.1 SMTP Capturing Windows



1.4.2 SMTP Packet Length




1.4.3 SMTP I/O GRAPH

# EXPERIMENT-2

**AIM: Detecting Suspicious Activity: Analyze network traffic to identify suspicious patterns, such as repeated connection attempts or unusual communication between hosts.**

## THEORY:

1. **Start Capturing Packets:**
    - Click on the 'Start' button or use the Ctrl + E shortcut to commence packet capture.

2. **Analyze Network Traffic:**
    - Wireshark will begin capturing packets in real-time. Observe the captured packets in the main window.

3. **Identify Suspicious Patterns:**

o Look for unusual or suspicious patterns in the network traffic. Some common suspicious activities to watch out for include:

- Unusual volume of traffic: Sudden spikes or unusual patterns in data transfer rates may indicate malicious activity such as a DDoS attack.
- Repeated connection attempts: Numerous connection attempts to a specific host or port could be a sign of port scanning or brute force attacks.
- Unusual protocols: Detection of unfamiliar or uncommon protocols may indicate attempts to evade detection by using non-standard communication methods.
- Unusual packet sizes: Large packets or abnormally small packets may suggest data exfiltration or network scanning.
- Unauthorized access attempts: Look for packets containing login attempts, authentication failures, or access to restricted resources.
- Unusual communication patterns: Analyze the communication between hosts to identify any abnormal behaviors such as communication between hosts that typically do not interact.

4. **Use Filters:**

- Apply filters in Wireshark to focus on specific types of traffic that may be indicative of suspicious activity. For example: o Filter for TCP SYN packets (tcp.flags.syn == 1) to identify TCP connection attempts. o Filter for large packets (frame.len > ) to detect potential data exfiltration attempts.

5. **Follow TCP Streams:**

- Follow TCP streams for suspicious connections to analyze the full conversation between hosts and identify any malicious payloads or commands being transmitted.

6. **Inspect DNS Traffic:**

- DNS traffic can often reveal malicious activity such as domain generation algorithms (DGAs) used by malware. Look for patterns in DNS requests that may indicate malicious domain names.

7. **Stop Capturing Packets**:

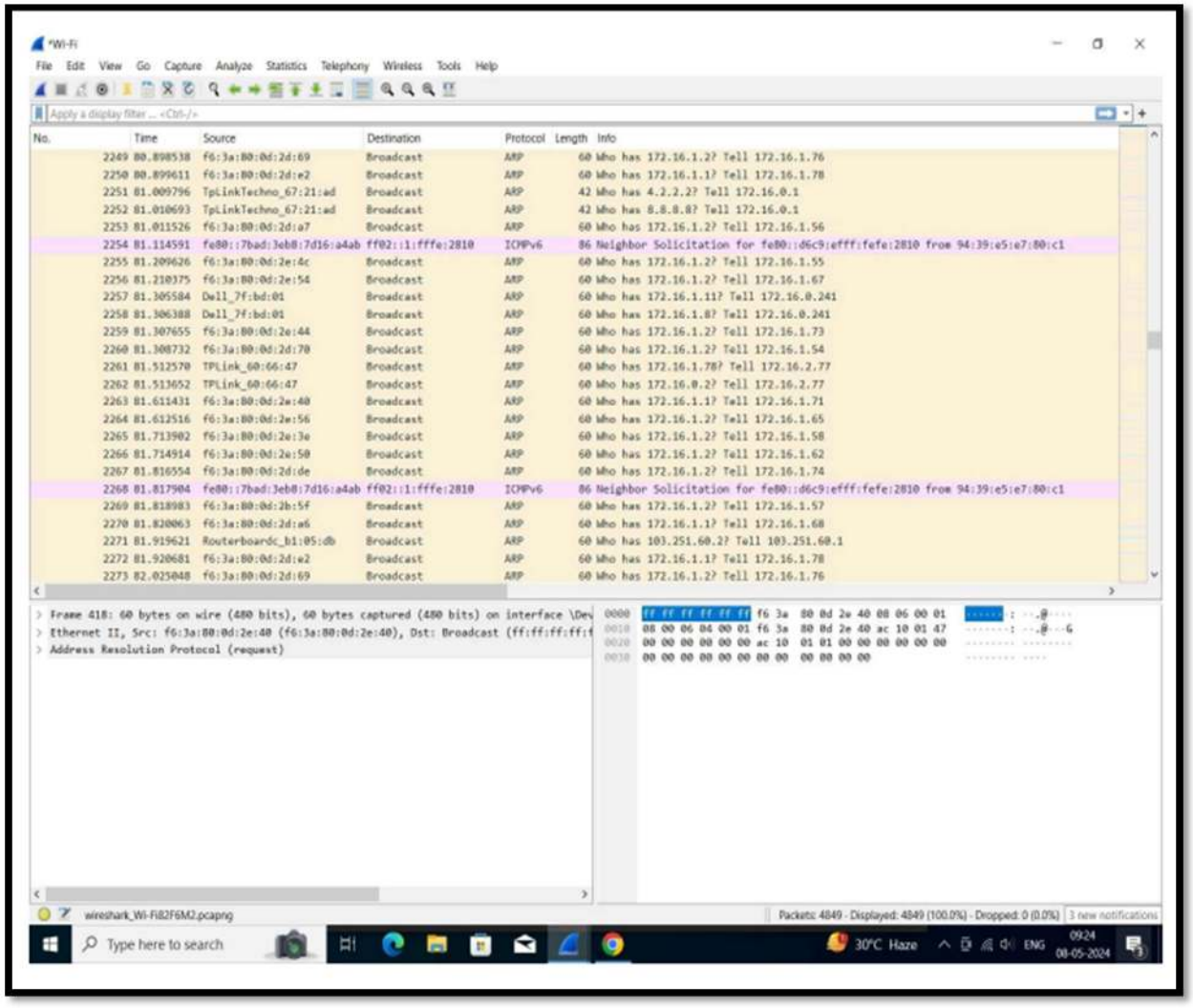
- Once you have gathered sufficient data for analysis, stop the packet capture by clicking on the 'Stop' button or using the Ctrl + E shortcut.

8. **Analyze Captured Data**:
- Review the captured packets and analyze them in detail to confirm any suspicions of malicious activity.

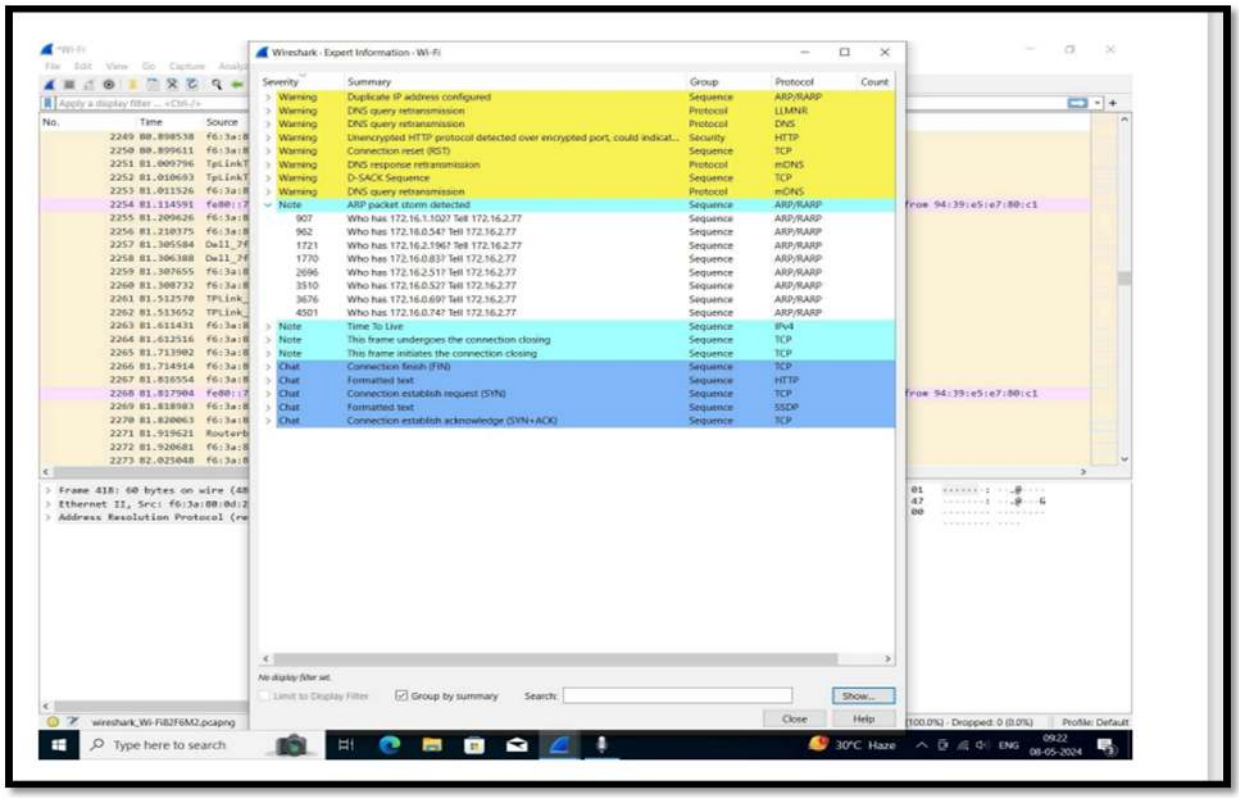
9. **Document Findings**:
- Document your findings, including any suspicious patterns or activities observed during the packet analysis.

By following these steps, detect and analyze suspicious activity on your network.

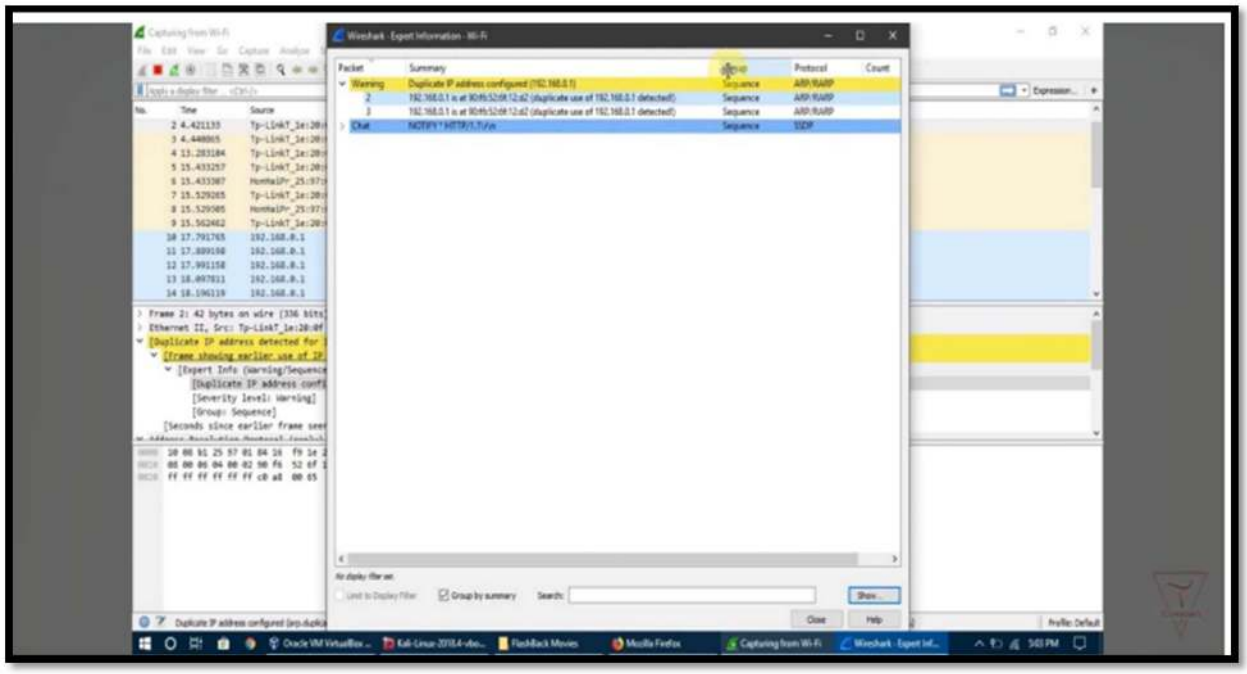
## ACCORDING TO EXPERIMENT:




Start Capturing and Analyzing network packets




Identifying Suspicious Patterns




Inspection of DNS Traffic and Analyzation of captured data

# Experiment - 3

**Malware Traffic Analysis : Analyze captured traffic to identify signs of malware communication, such as command-and-control traffic or data infiltration.**

## 1. Introduction :

Malware traffic analysis is the process of examining captured network traffic to identify signs of malware communication. This can include command-and-control traffic, data infiltration, and other malicious activities.

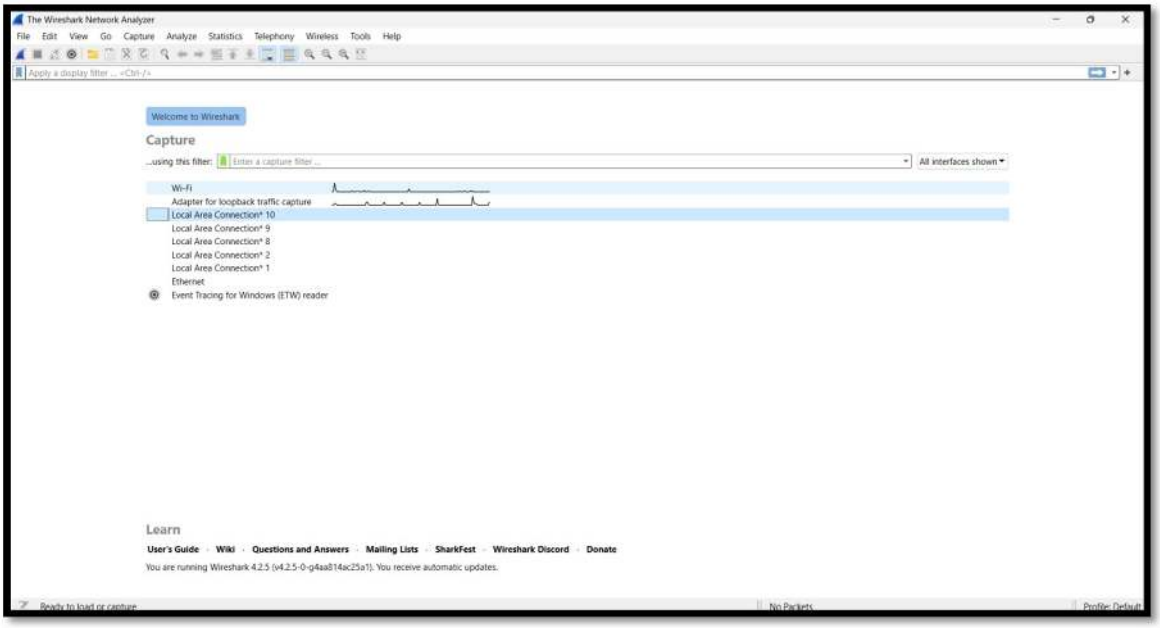
## 2. Use Wireshark :

Analyze captured network packets for patterns, behaviors, and indicators of compromise (IoCs). Examine DNS traffic for suspicious domain names. Look for traffic using non-standard or uncommon protocols. Analyze HTTP and HTTPS traffic for anomalies which include the following things :

- Infected Files
- URL/Domains of the infected site
- IP Address and port of the infected machine
- MAC Address of the infected machine

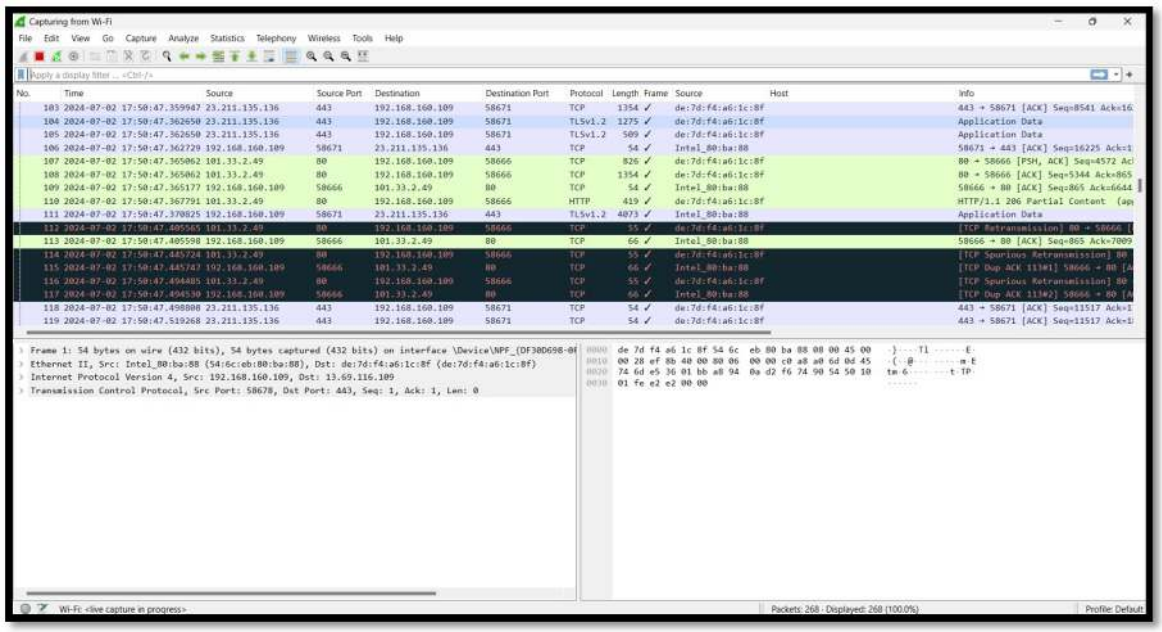
**Step 1 :** Start Wireshark and select the interface whose packet you want to capture.

### Screen 1 : Selecting the Interface




Note : In our case we will be capturing Wi-Fi packets.

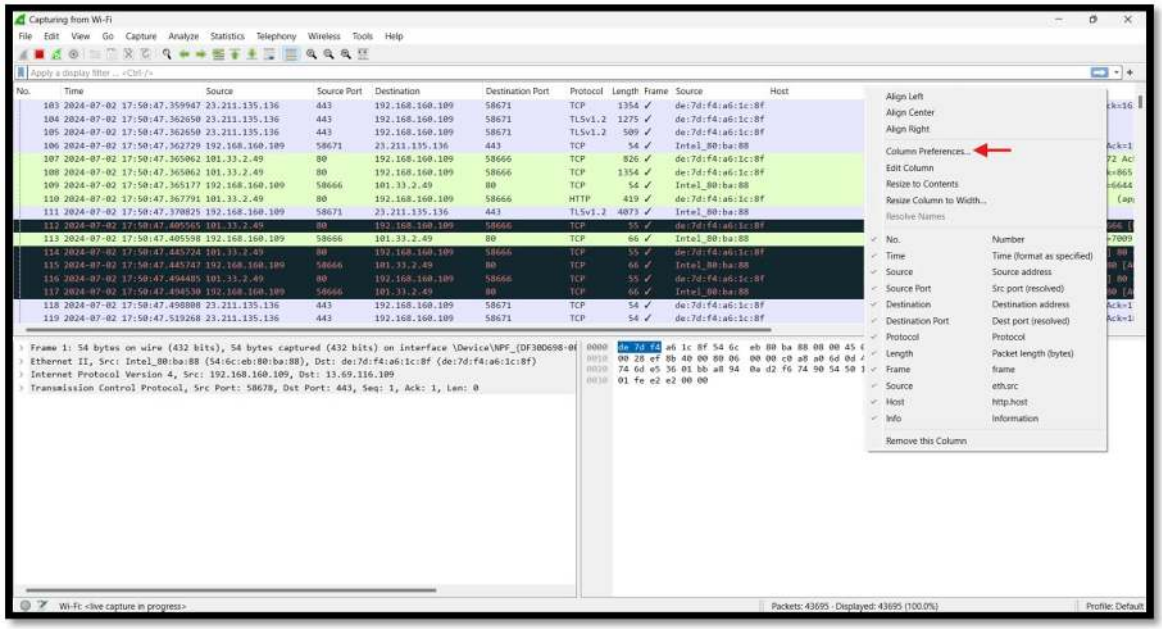
## Screen 2 : Captured packets after selecting interface :



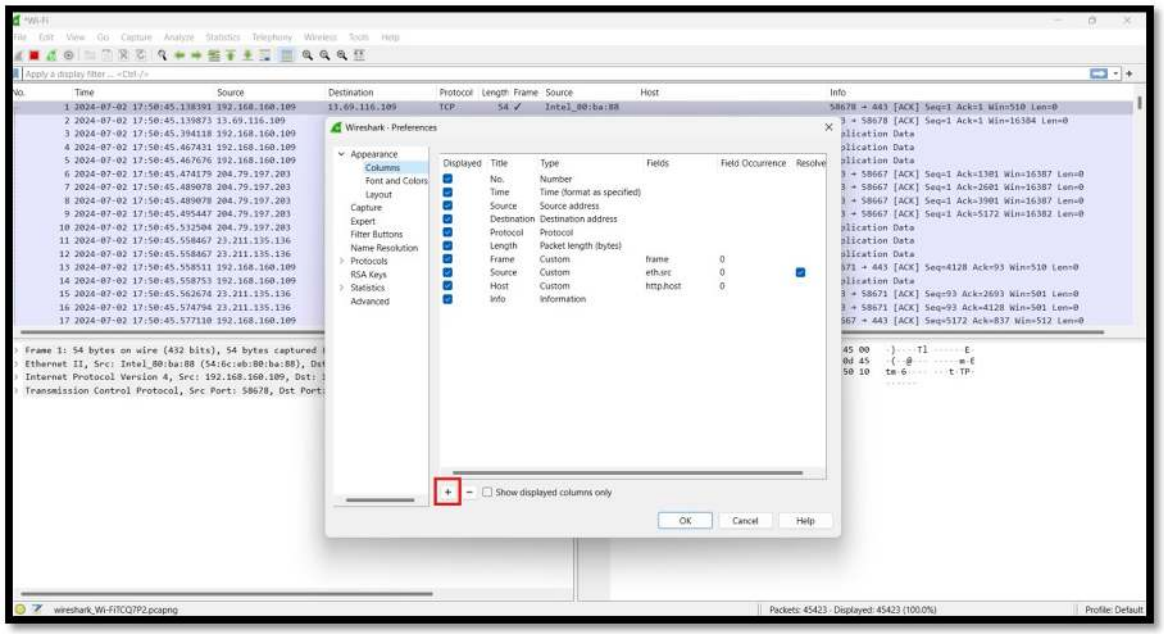
**Step 2 :** Now we will see a whole lot of packets being captured so let's first sort the outputs we are getting and customize the results by adding a column like source port, destination port etc.
To add such Column Headings right-click on Column heading and select

**Column Preferences**.

## Screen 3 : Click on Column Headings to get Column Preferences



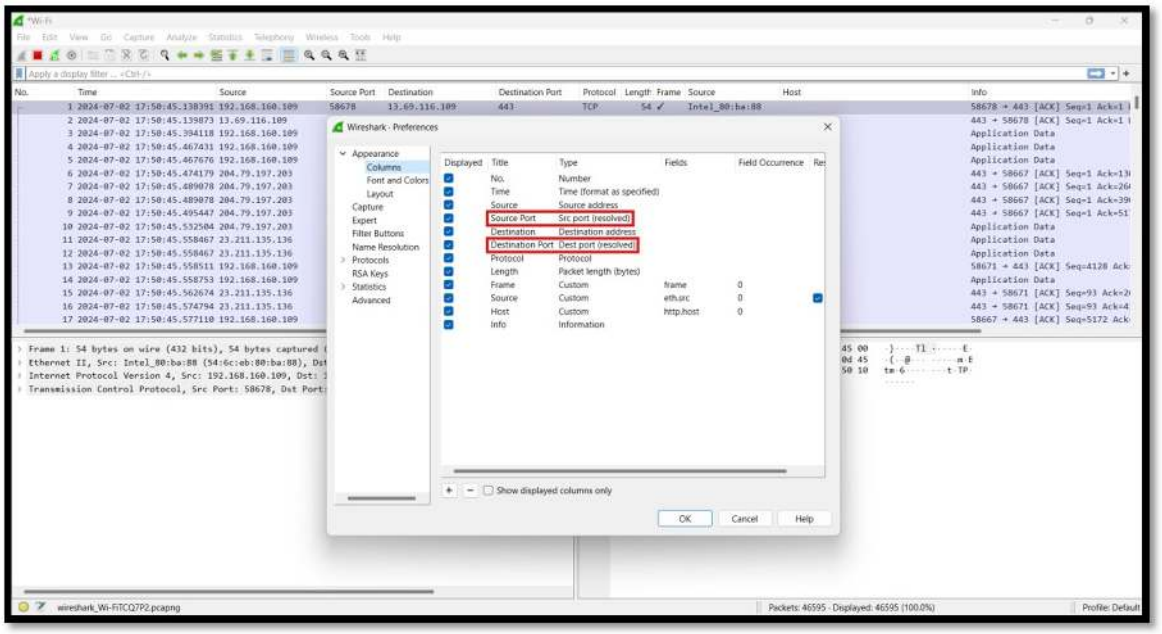
## Screen 4: Click on + to add column



**Step 3 :** Now add two columns naming Source port and Destination port and select type as **src port (resolved)** and **Dest port (resolved)** respectively. Drag them and make them aligned with source IP and destination IP so that it seems more convenient to

identify which port was used and by which IP address. You can even remove the column if you need by simple uncheck option. Then click on OK.

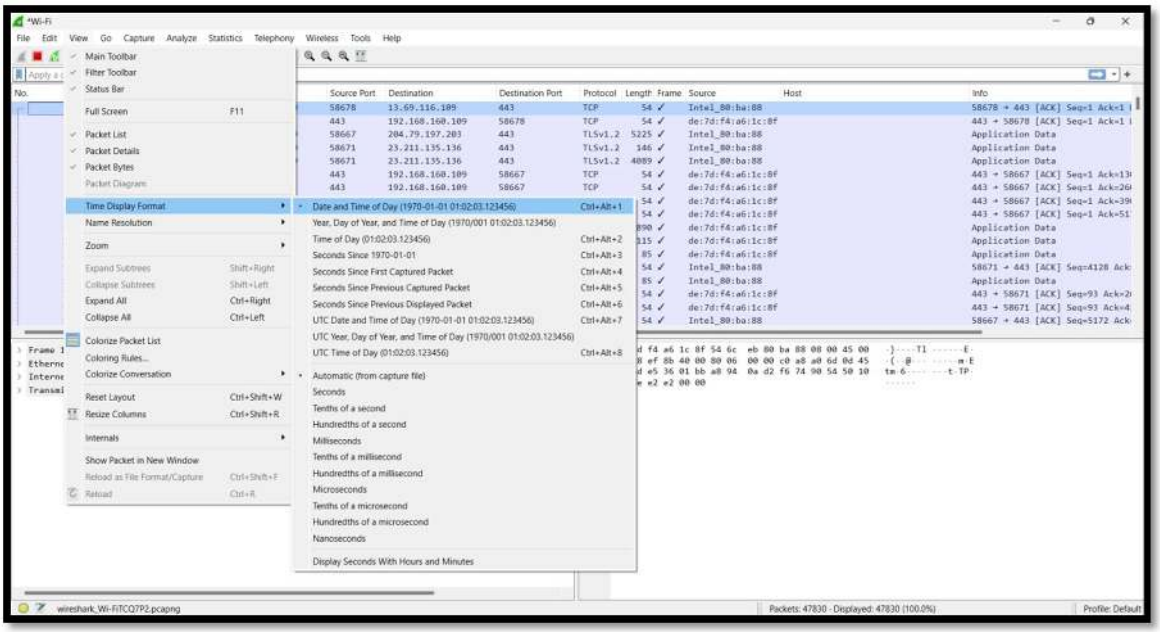
## Screen 5: Final Column Preference



**Step 4:** Change the Time display format in order to identify the Timestamp of the file being flowed over the network. To do this go to,

**View -> Time Display Format -> Date and Time the Day.**

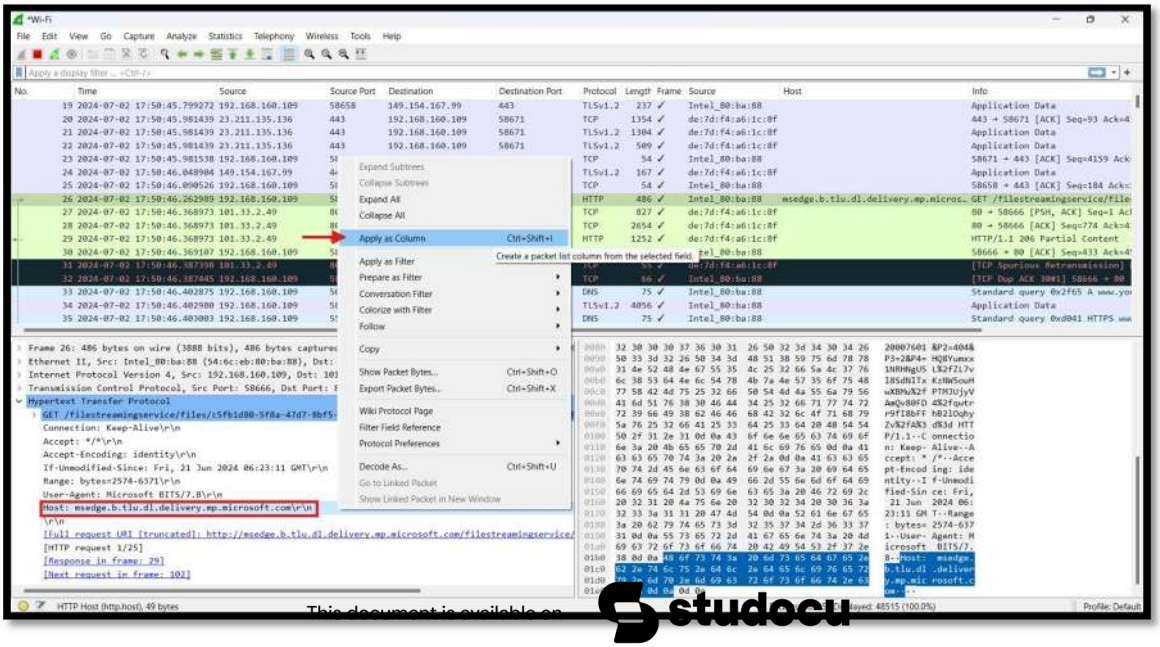
## Screen 6: Selecting Time Display Format



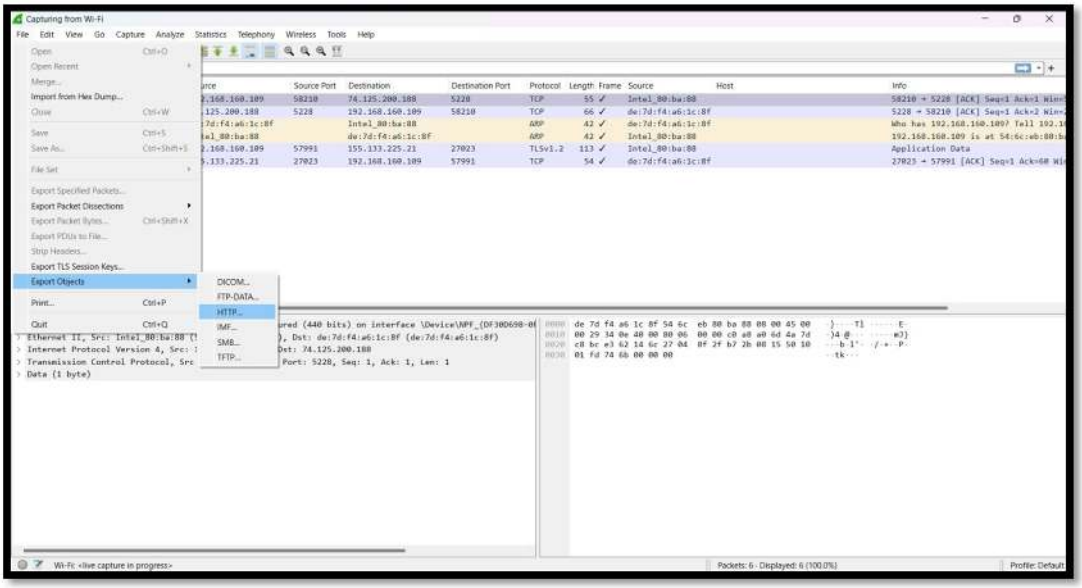
If we look at the panel now in Info we can see the requested URL but we are unable to check for the HOST through which it is being generated. To add that column we can either add it in the same way done in Step 3 or another way is through **Packet Detail Panel.**

**Step 5:** Click on any packet in the Summary Panel then go to its detail panel looked out for **HOST:** right click and select **Apply as Column.**

## Screen 7: Applying HOST name as a column.

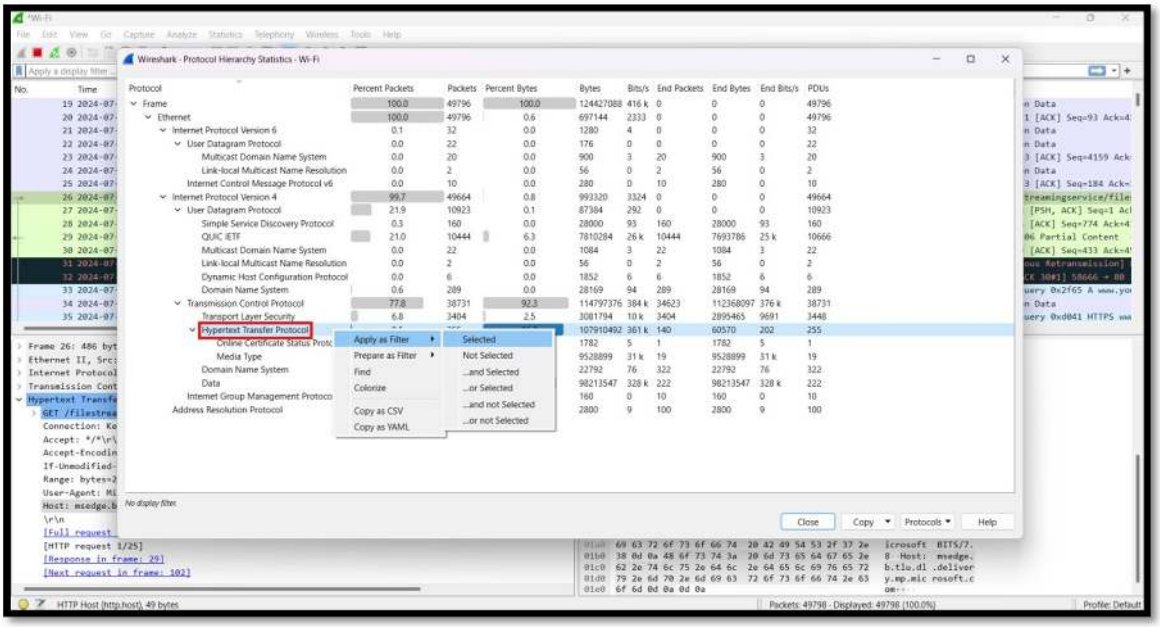


After doing this you will see a new column named as Host now we want to see therequest.



**Step 6:** In the filter bar type HTTP and press enter **OR** click on, **Statistics -> Protocol Hierarchy -> Hypertext Transfer Protocol** right click on it and click on **Apply as Filter -> Selected.** Click on close.

## Screen 8: Applying HTTP as filter.

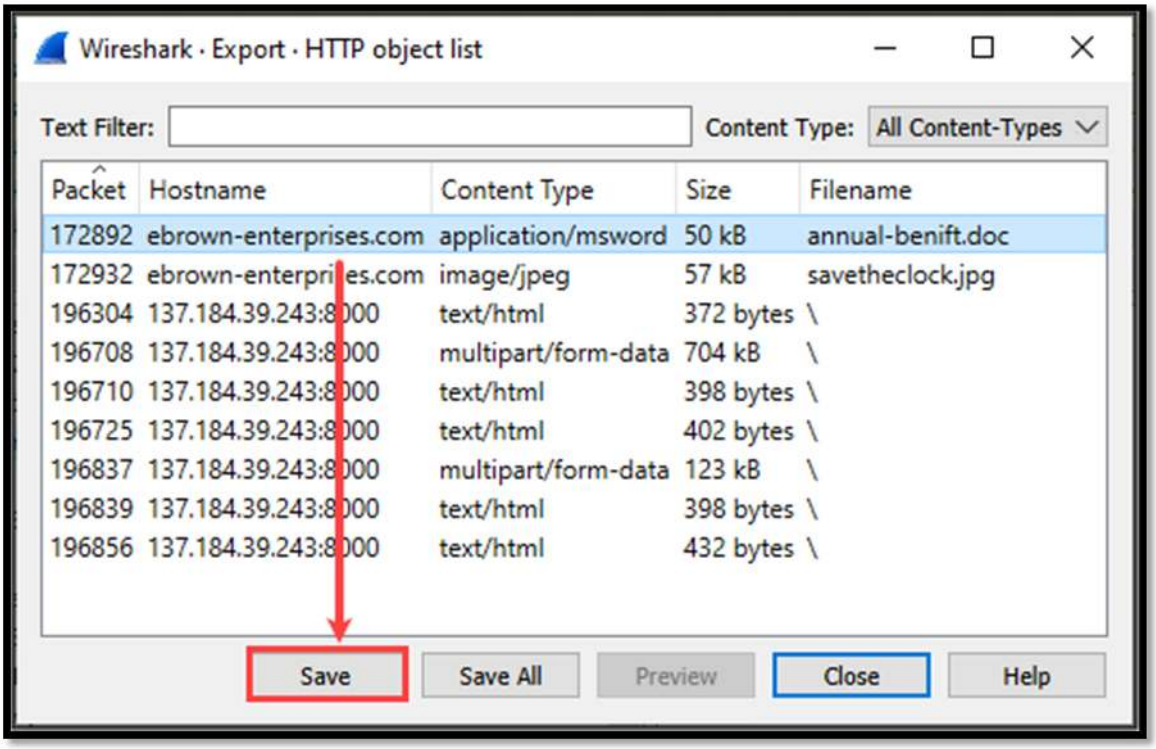


As we can now analyze all the HTTP traffic but we need to check for object or file transmitted.

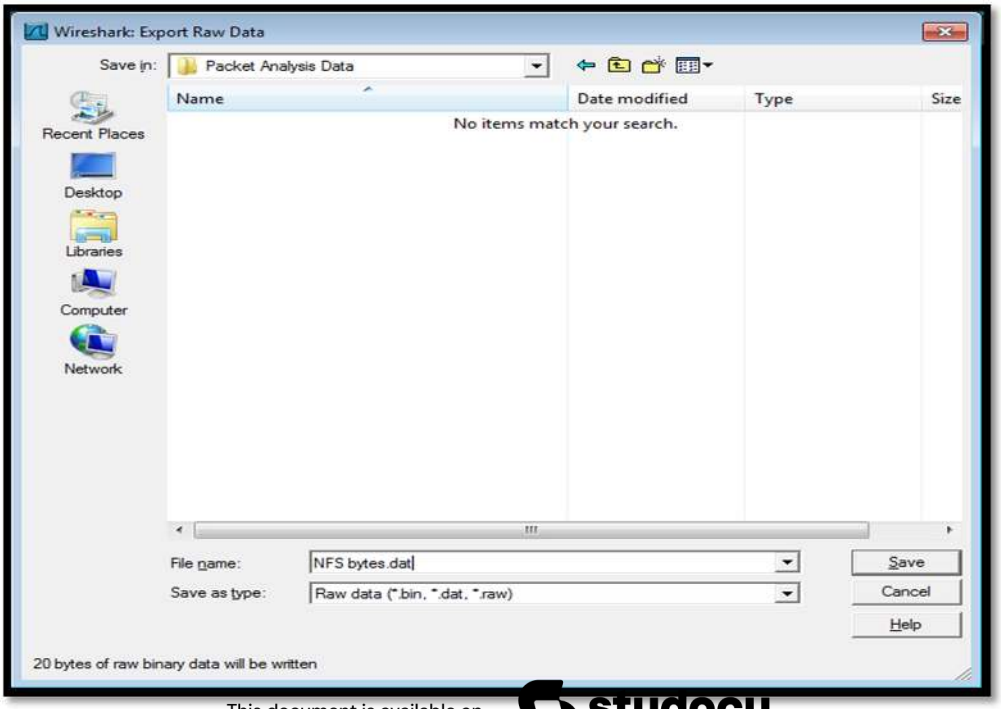
**Step 7:** Go to **File -> Export Objects -> HTTP**

## Screen 9: Exporting all HTTP based objects.

**Step 8:** Select the object you want to download like here in an example I'm going to export to the file that were downloaded on the system, Select any packet. Save it accordingly like the first file in Application/Force download possibly it could be an exe so go to **Save ->** name the file **-> Save.**




## Screen 10: Exporting the object in the desktop for analysis.

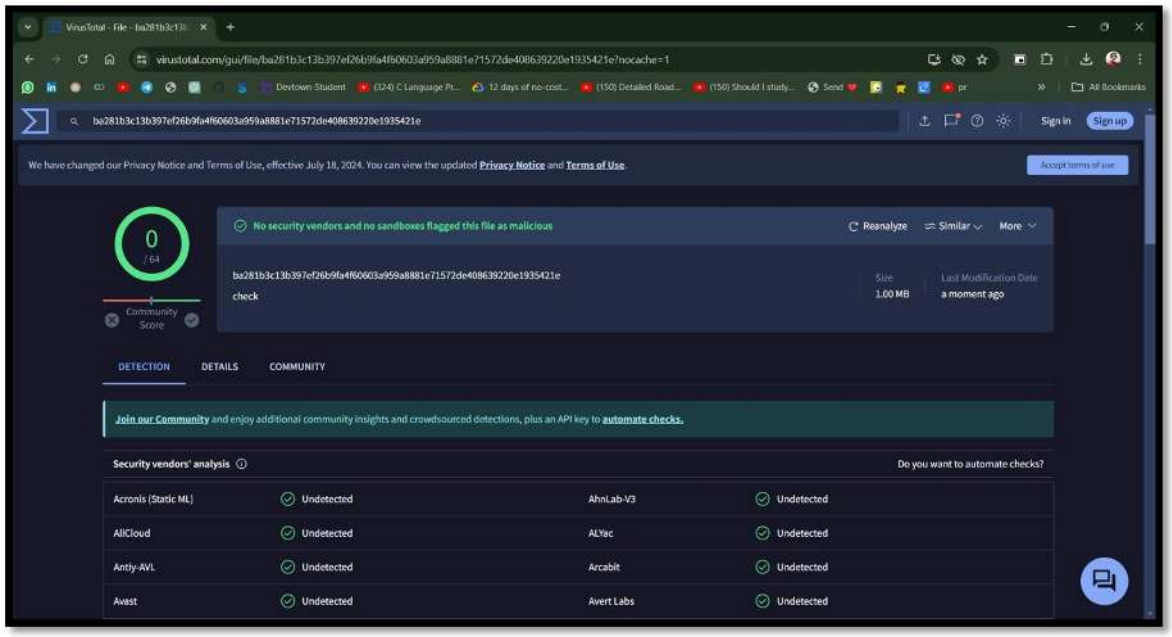


Now we have the file with us what we can do is either check that file through our anti-malware or navigate ourself to Virustotal.com (more preferable as it will show you the behaviour of the file and it will be checked by the knowledge base of several anti-malware.)

**Step 9:** Upload the file in VirusTotal.com it will check for hashes and give you proper analysis.

## Screen 11: Uploading the object/file in VirusTotal.com


## Screen 12: Results found by the calculated hash of that file




So,this is how can check for the objects that are transferred are safe or not.

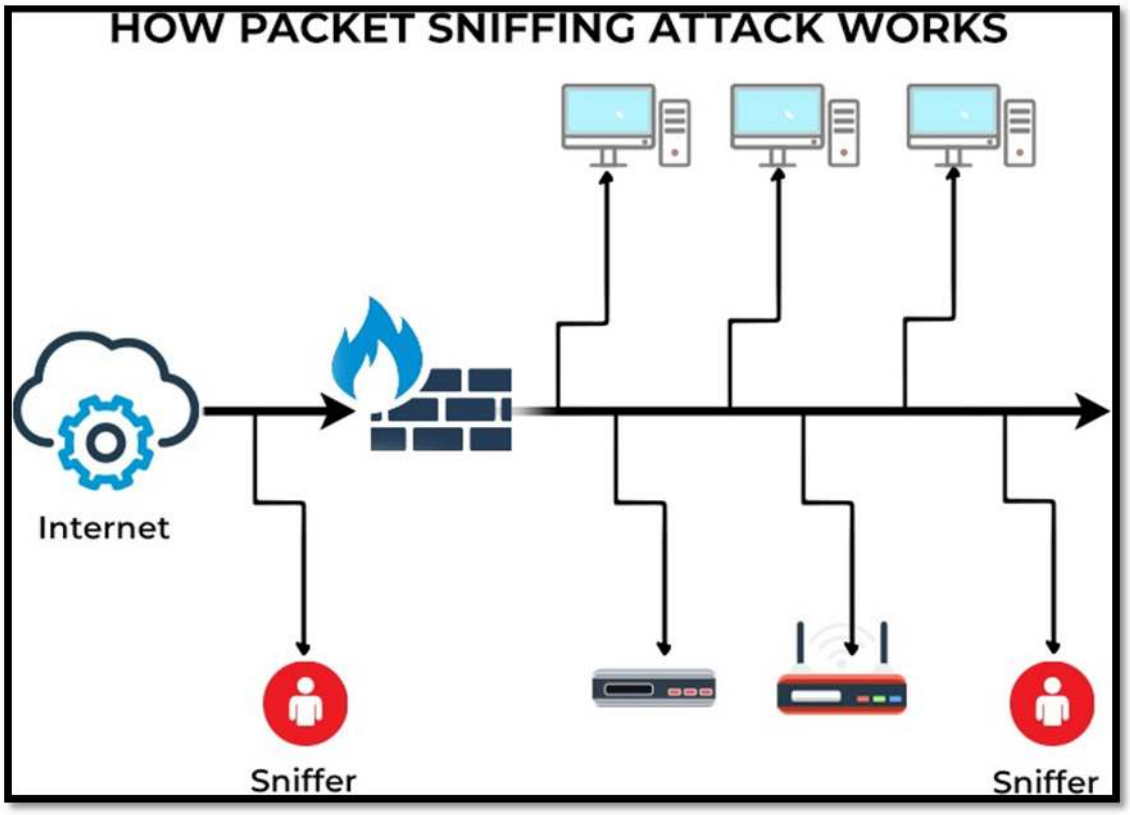
# Experiment – 4

**Password Sniffing : Simulate a scenario where a password is transmitted in plaintext. Use Wireshark to capture and analyze the packets to demonstrate the vulnerability and the importance of encryption.**

The following are protocols that are vulnerable to sniffing

- Telnet
- Rlogin
- HTTP
- SMTP
- NNTP
- POP
- FTP
- IMAP

The above protocol are vulnerable if login details are sent in plain text.



## Passive and Active Sniffing:

- **Passive sniffing :** It is intercepting packages transmitted over a network that uses a hub. It is called passive sniffing because it is difficult to detect. It is also easy to perform as the hub sends broadcast messages to all the computers on the network.
- **Active sniffing :** It is intercepting packages transmitted over a network that uses aswitch. There are two main methods used to sniff switch linked networks, ARP poisoning, and MAC flooding.

## Sniff Network Traffic :

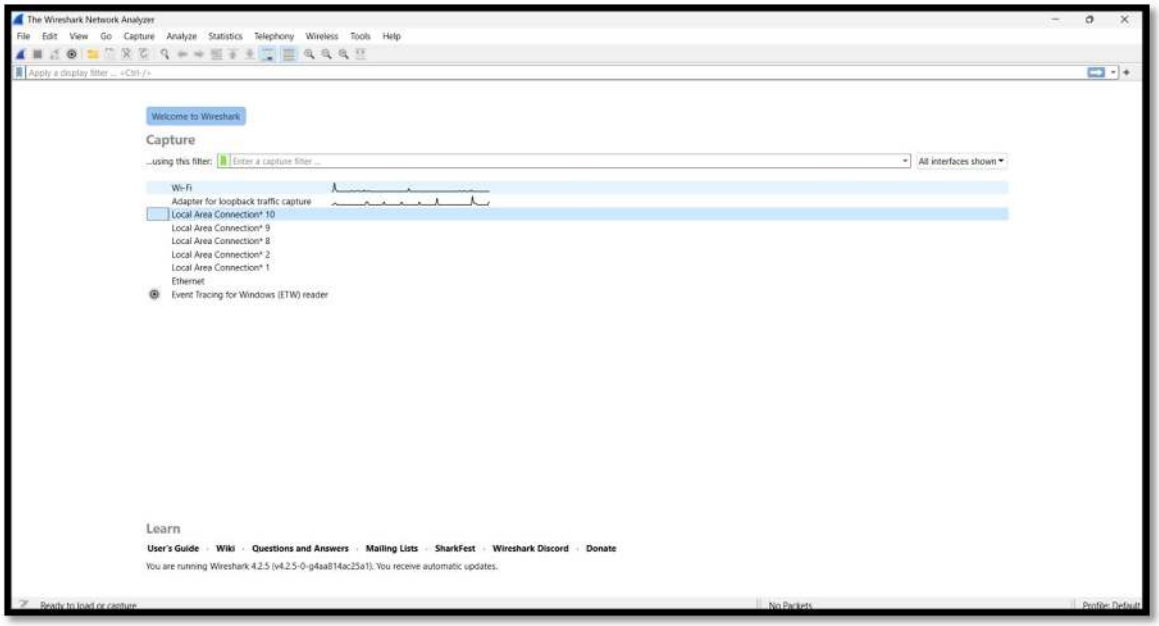
In this practical scenario, we are going to use Wireshark to sniff data packets as they are transmitted over HTTP protocol. For this example, we will sniff the network using Wireshark, then login to a web application that does not use secure communication. We will login to a web on http://testphp.vulnweb.com/login.php

The login address is test.admin@google.com, and the password is password12345.

**Note:** we will login to the web app for demonstration purposes only. The technique can also sniff data packets from other computers that are on the same network as the one that you are using to sniff. The sniffing is not only limited to techpanda.org, but also sniffs all HTTP and other protocols data packets

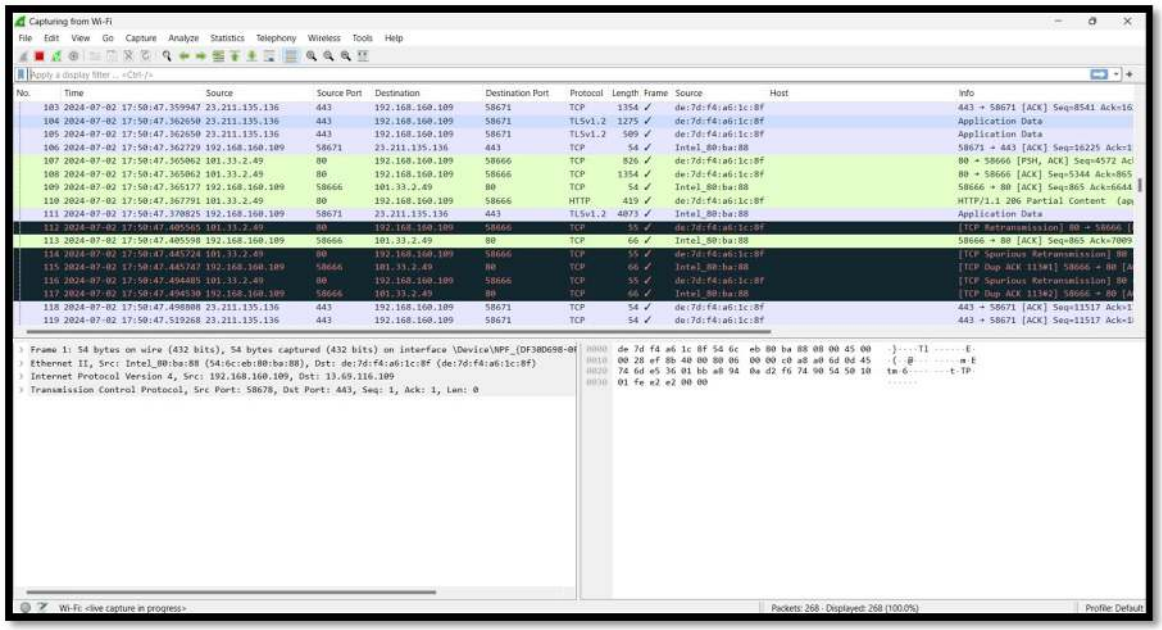
### ❖ Sniffing the Network using Wireshark :

**Step 1 :** Open Wireshark & select the interface



Note : In our case we will be capturing Wi-Fi packets.

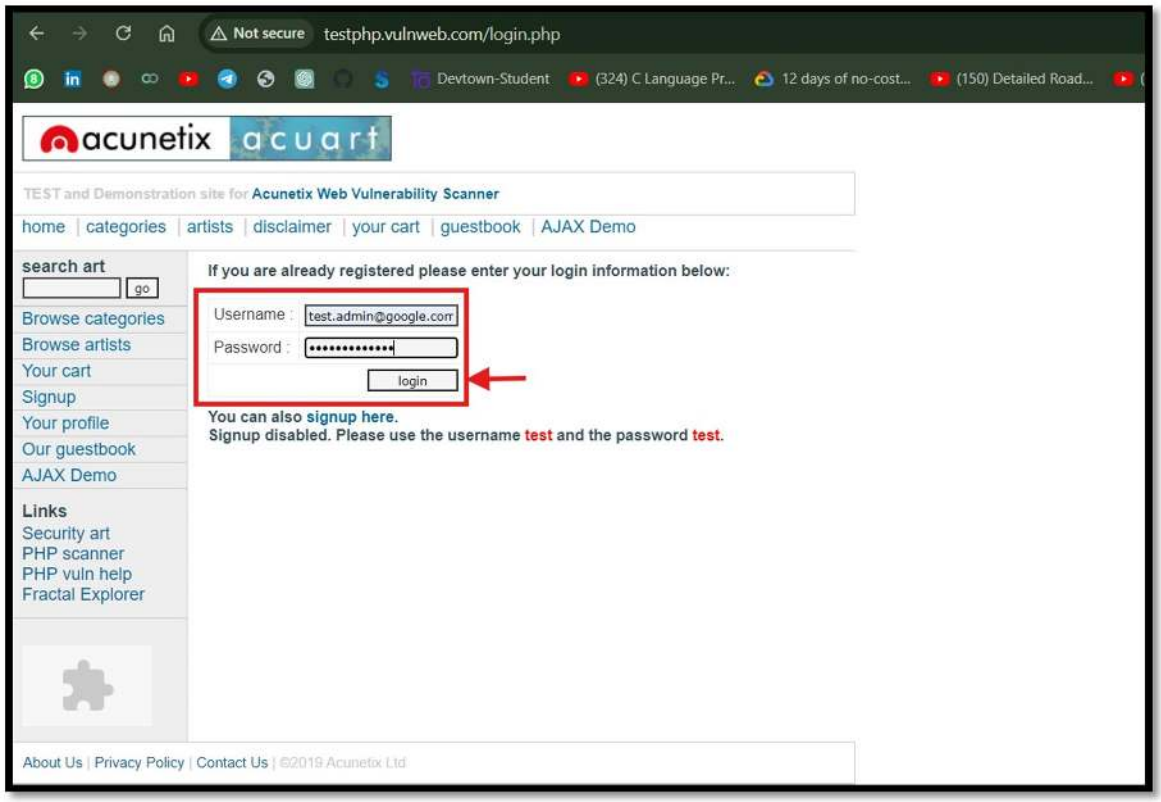
**Step 2:** Captured packets after selecting interface



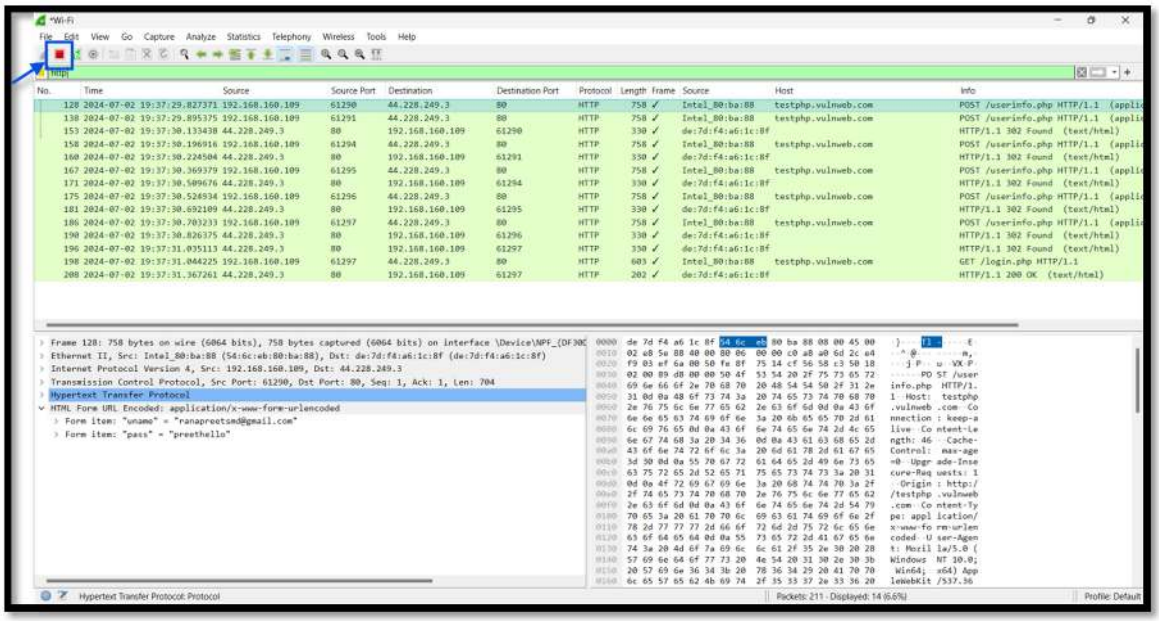
**Step 3 :** Open your web browser and type in http://testphp.vulnweb.com/login.php

The login email is **test.admin@google.com** and the password is **password12345**

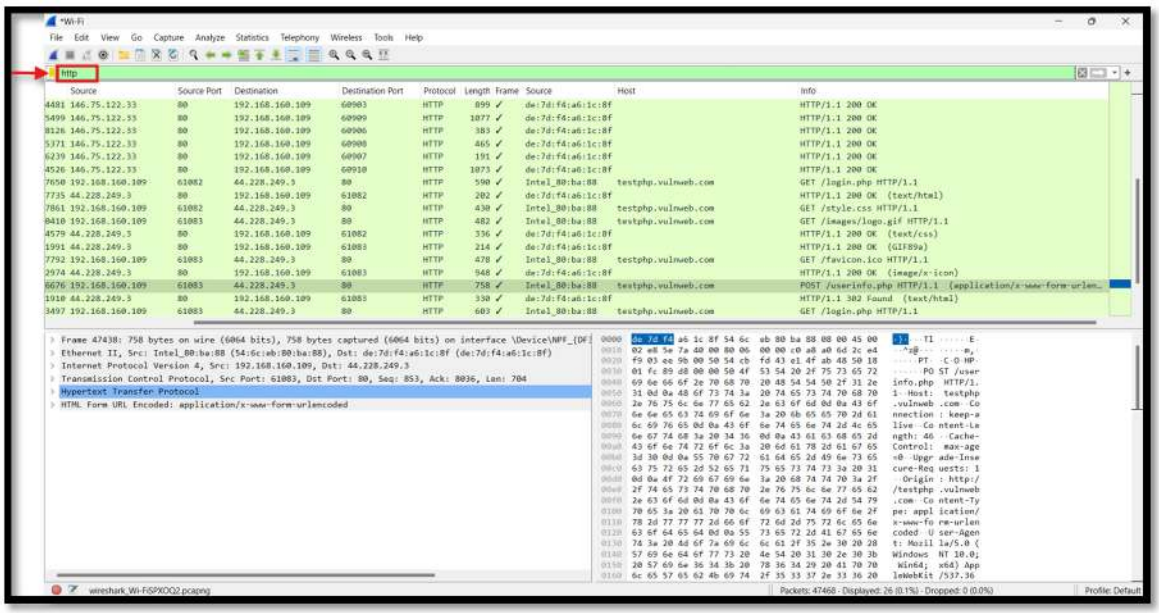
Click on login button.



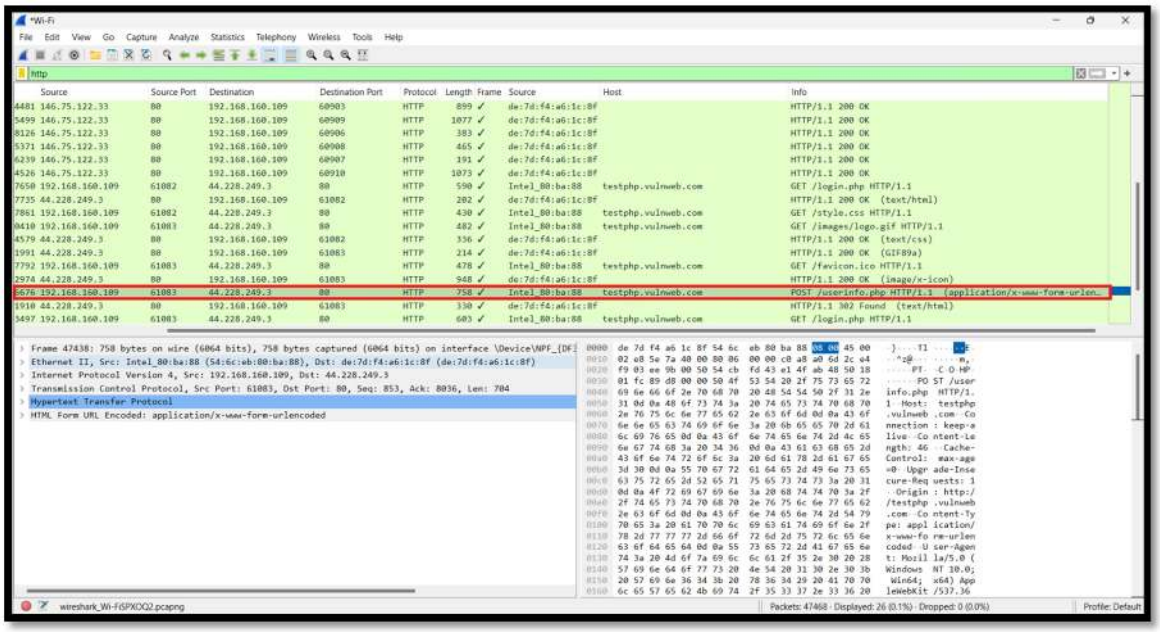
**Step 4 :** Now go back to wireshark, stop the live capture



**Step 5 :** Filter for “http” protocol results only using the filter textbox

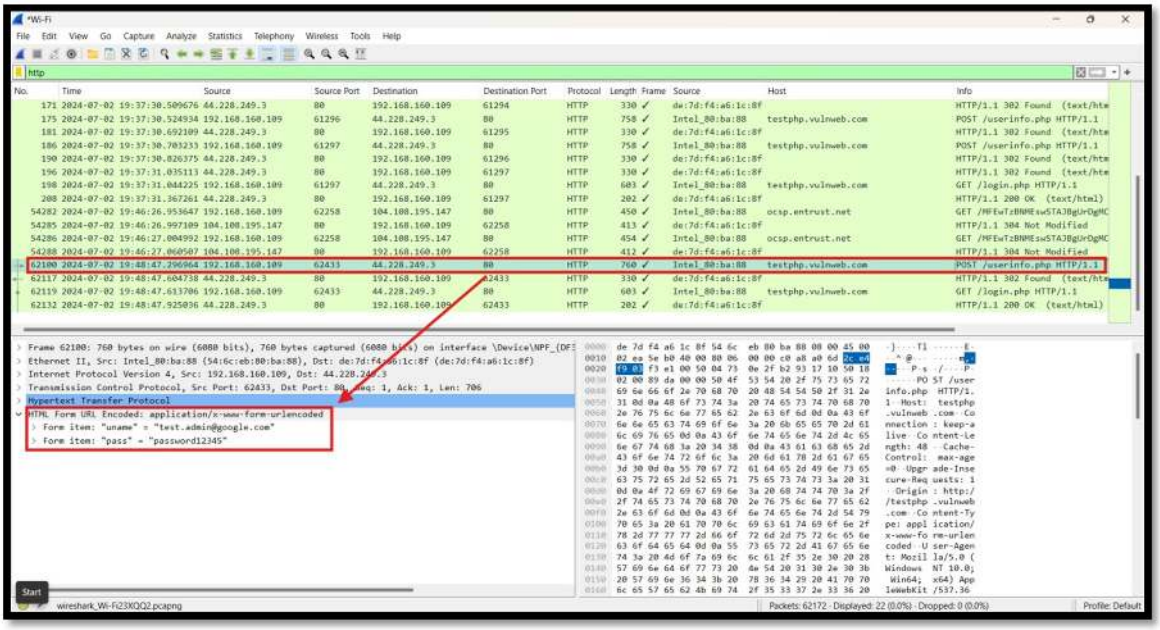


**Step 6 :** Locate the Info column and look for entries with the **HTTP** verb **POST** andclick



on it.

**Step 7 :** Go to Summary Panel and look for the summary that **HTML Form URLEncoded: application/x-www-form-urlencoded**.



You should be able to view the plaintext values of all the POST variables submitted to the server via HTTP protocol.

**So we are able to view the exact username and password we provide to login page,**

**Hence,** we have successfully used Wireshark to sniff data packets that are transmittedover HTTP protocol.

# EXPERIMENT:- 5

**AIM: ARP Poisoning Attack: Set up an ARP poisoning attack using tools like Ettercap. Analyze the captured packets to understand how the attack can lead to a Man-in-the-Middle scenario.**

## Experiment Setup

1. **Prerequisites:**
   a. Two computers in the same network.
   b. Kali Linux installed on one of the computers (Attacker).
   c. Target computer running any operating system (Victim).
2. **Installation:**
   a. Install Ettercap on the Kali Linux machine:
   sudo apt-get update
   sudo apt-get install ettercap-graphical

3. **Network Configuration**:
   a. Ensure both the Attacker and Victim are connected to the same local network.

4. **Enable IP Forwarding:**
   a. Enable IP forwarding on the Attacker machine to ensure it can forward packets between the Victim and the internet.
   sudo sysctl -w net.ipv4.ip_forward=1

## Performing the attack:

1. **Launch Ettercap:**
   - Open a terminal on the Attacker machine and launch Ettercap:
     - sudo ettercap -G

2. **Select Sniffing Interface:**
   - In the Ettercap GUI, select the appropriate network interface to sniff on (usually Ethernet or Wi-Fi).

3. **Scan for Hosts:**
   - Go to 'Hosts' > 'Scan for hosts' to scan the network and identify the Victim's IP address.

4. **ARP Poisoning:**
   - Go to 'Mitm' > 'ARP Poisoning'.
   - Select 'Sniff remote connections'.
   - Enter the Victim's IP address as the target.

5. **Start the Attack:**
   - Click on 'Start' to begin the ARP poisoning attack. Ettercap will now intercept and modify ARP packets, redirecting traffic through the Attacker machine.

## Analyzing captured packets:

1. **Wireshark:**
   - Use Wireshark to capture and analyze network packets.
   - Start Wireshark on the Attacker machine and select the network interface used for ARP poisoning.

2. **Filtering**
   - Apply filters to isolate traffic between the Attacker, Victim, and other network entities.
   - Filter for ARP packets to observe ARP spoofing in action.

3. **Packet Analysis:**
   - Analyze the captured packets to observe the following:
     - ARP requests and responses indicating the spoofed MAC addresses.
     - Victim's traffic being redirected through the Attacker machine.
     - Communication between the Victim and other network entities being intercepted by the Attacker.

## Understanding the MITM Scenario:

1. **Interception of Communication:**
   - The ARP poisoning attack allows the Attacker to intercept and modify all communication between the Victim and other network entities.
   - This includes intercepting sensitive information such as login credentials, financial data, or personal information.

2. **Modification of Data:**
   - The Attacker can modify the intercepted data before forwarding it to its original destination, leading to potential data manipulation or injection attacks.

3. **Identity Theft:**
   - By intercepting communication, the Attacker can impersonate either the Victim or the legitimate network entities, leading to identity theft or unauthorized access to sensitive resources.

# MITM ATTACK SCREENSHOT