EXPERIMENT NO. 5

Introduction to Wireshark and Traffic analysis (packet headers) using Wireshark

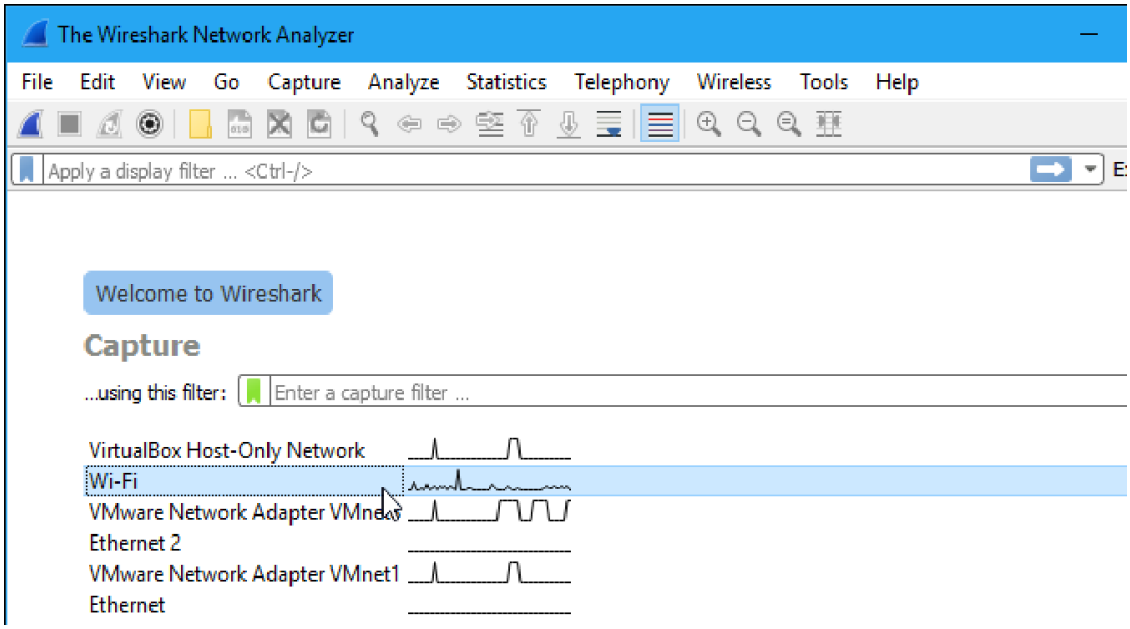
Aim: To study Wireshark and its applications in traffic analysis and password sniffing.

Procedure:

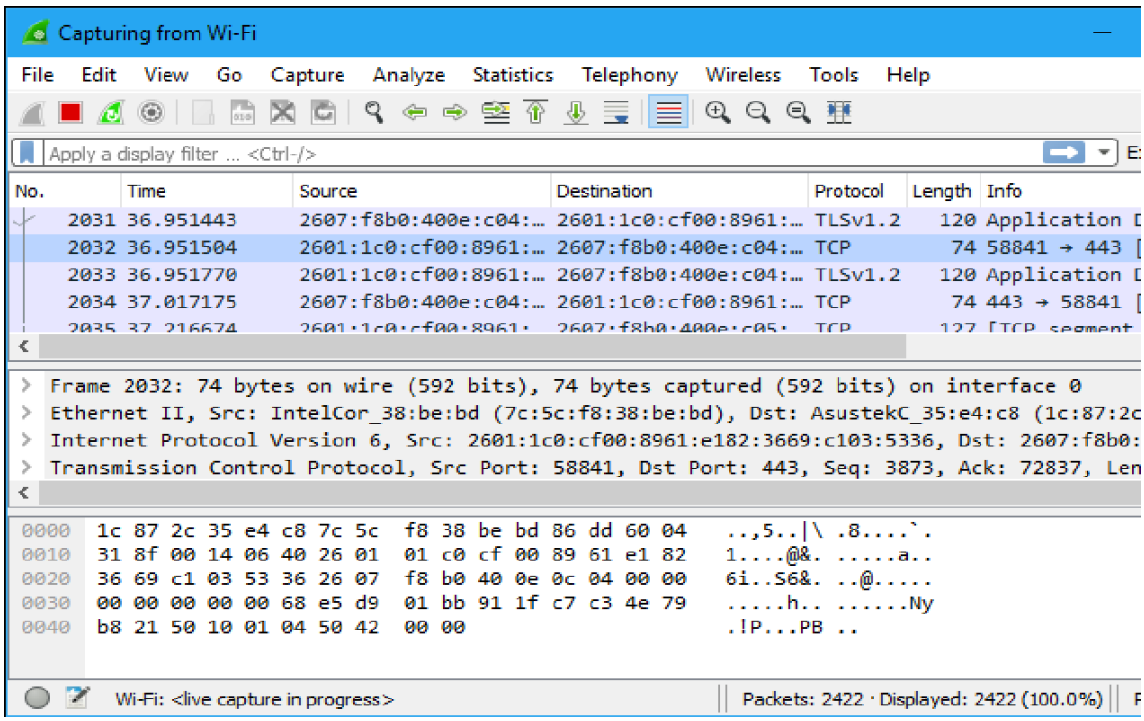
1. Open Wireshark.
2. Study Wireshark GUI and components (display filter, packet listing, packet header and packet content)
3. To begin packet capture, select the Capture pull down menu and select Options. This will cause the "Wireshark: Capture Options"
4. Click Start. Packet capture will now begin. All packets being sent (received) from (by) your computer are now being captured by Wireshark.
5. Once you begin packet capture, a packet-capture-summary window will appear. It will display all information relevant to that packet while Wireshark is running, enter the following URL in a web browser (whichever is installed on your system) http://gaia.cs.umass.edu/wireshark-labs/INTRO-wireshark-file1.html
6. After your browser has displayed the INTRO-wireshark-file1.html page, stop Wireshark packet-capture by selecting STOP in the Wireshark capture window.
7. Type in "http" (without the quotes, and in lower case – all protocol names are in lower case in Wireshark) into the display filter specification window at the top of the main Wireshark window.
8. Then select Apply (to the right of where you entered "http"). This will cause only HTTP message to be displayed in the packet-listing window.
9. Select the first http message shown in the packet-listing window. This should be the HTTP-GET message which was sent from your computer to the http://gaia.cs.umass.edu HTTP server. When you select the HTTP GET message, the Ethernet frame, IP datagram, TCP segment, and HTTP message header information will be displayed in the packet-header window.
10. By clicking plus and minus boxes to the left side of the packet details window, minimize the amount of Frame, Ethernet, Internet Protocol, and Transmission Control Protocol information displayed. Maximize the amount information displayed about the HTTP protocol).

Procedure for Capturing Data Packets

After downloading and installing Wireshark, you can launch it and double-click the name of a network interface under capture to start capturing packets on that interface. For example, if you want to capture traffic on your wireless network, click your wireless interface.



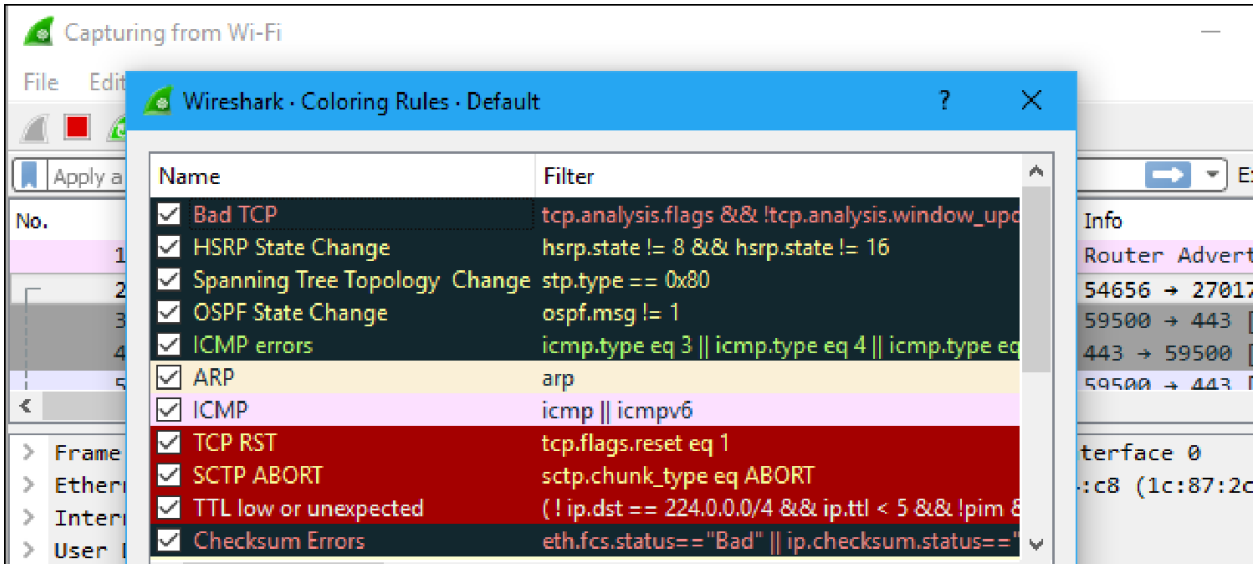
As soon as you click the interface's name, you'll see the packets start to appear in real time. Wire-shark captures each packet sent to or from your system.



Click the red "Stop" button near the top left corner of the window when you want to stop capturing traffic.

Color Coding

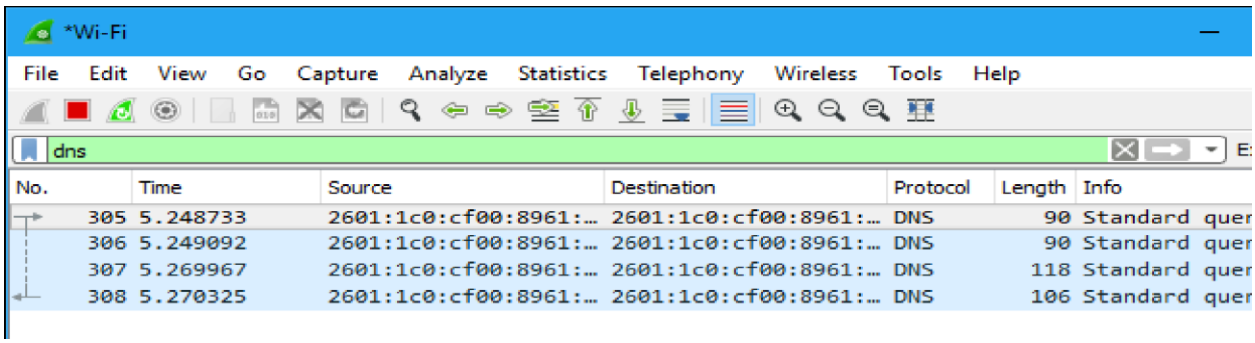
Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors .To view exactly what the colour codes mean, click View > Coloring Rules. You can also customize and modify the colouring rules from here, if you like.



Procedure for Filtering Data Packets

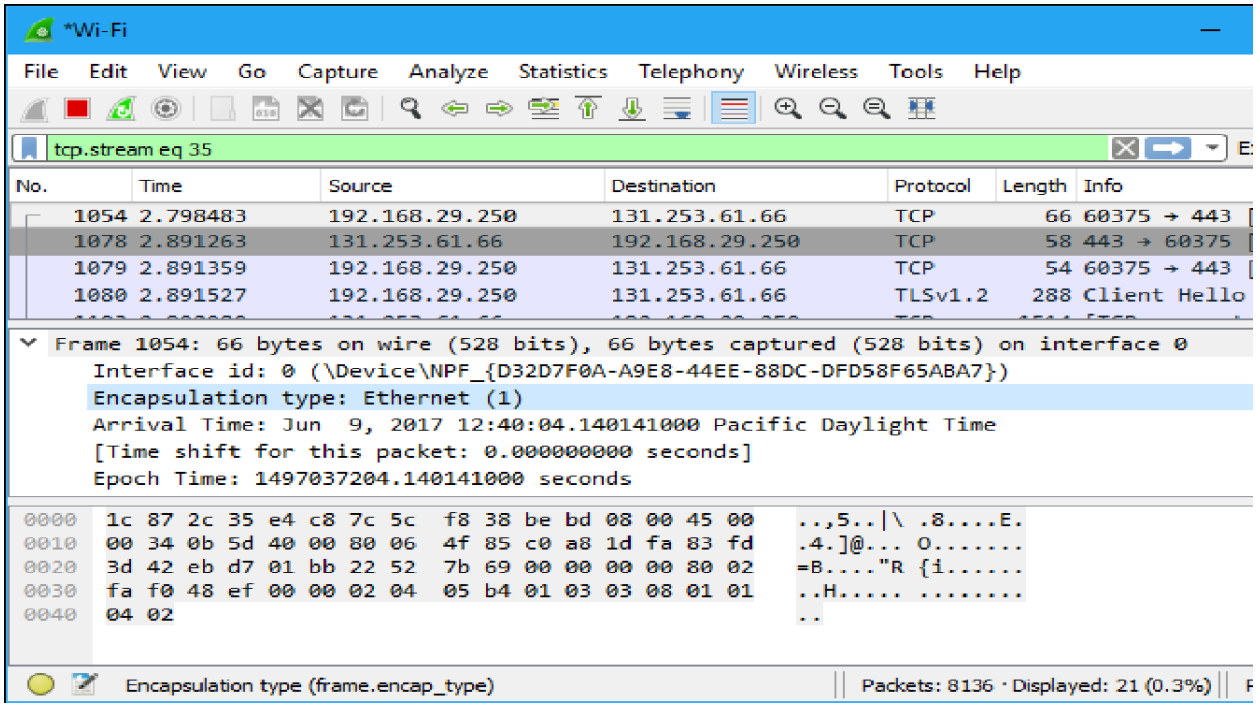
The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking Apply (or pressing Enter). For example, type “dns” and you will see only DNS packets. When you start typing, Wireshark will help you autocomplete your filter.

You can also click Analyze > Display Filters to choose a filter from among the default filters included in Wireshark. From here, you can add your own custom filters and save them to easily access them in the future.



Procedure for inspecting Data Packets

Click a packet to select it and you can dig down to view its details as shown below,



Output:

Attach Screenshots of following; (a brief explanation of each screenshot is required.)

1. Wireshark Interface window
2. Wireshark packet sniffing window
3. Wireshark window with different filters (arp, http, tcp, icmp, with ip address etc)
4. Wireshark window with Ethernet protocol details.
5. Wireshark window with arp protocol details.
6. Login window of an insecure webpage.
7. Wireshark window with http packet filtering for password sniffing.
8. TCP stream data window showing login id and password details.

Post Experimental Exercise-

Using Wireshark answer the following questions-

For Ethernet Protocol Frame	
1	Frame length of the packet: 945 bytes (7650 bits)
2	Frame number: 142011
3	Type of frame (Ether-type Number): IPv4 (0x0800)
4	Can you say that there is padding in the payload? No
5	Destination link-layer address: ASUSTekCOMPU_16:bc:7a (c8:7f:54:16:bc:7a)
6	Source link-layer address: Sophos_c8:66:82 (7c:5a:1c:c8:66:82)

7	Upper layer protocol: Internet Protocol Version 4
8	Is destination link-layer address unicast or broadcast? Unicast

For ARP Protocol Frame	
1	Hardware type: Ethernet
2	Protocol type: IPv4 (0x0800)
3	Hardware size: 6
4	Protocol size: 4
5	Source hardware address. ASUSTekCOMPU_16:bc:7a (c8:7f:54:16:bc:7a)
6	Source IP address? 192.168.6.106
7	Destination hardware address: EardaTechnol_b6:b5:e4 (54:77:87:b6:b5:e4)
8	Destination IP address: 192.168.3.80
9	Number of bytes of padding in a frame: 0
10	Is it arp request or reply packet? reply
11	Identify two differences between arp request and reply packets. 1 ARP reply packet has opcode (2) ,ARP request Packet have opcode (1) 2 ARP reply packet are always unicaste whereas ARP replace packet are always broadcasted

Conclusion: *(To be hand written on journal sheet)*

Attach Screenshots of following; (a brief explanation of each screenshot is required.)

1. Wireshark Interface window

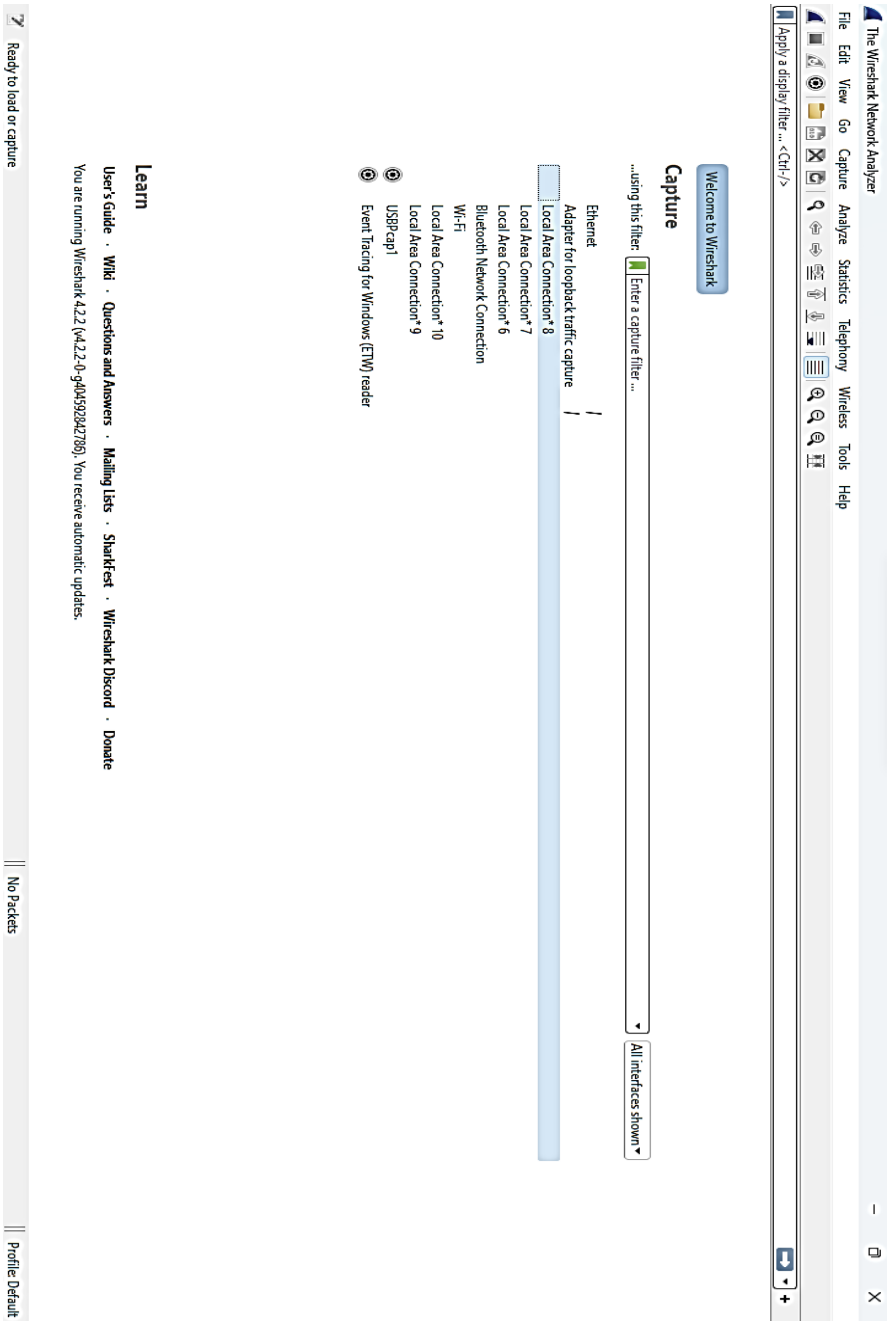


Fig.1 Wireshark Interface window

Explanation:

Wireshark presented a user-friendly interface with clearly defined sections for efficient packet analysis. The primary window functioned as the packet sniffing display, providing a live overview of network activity.

2. Wireshark packet sniffing window

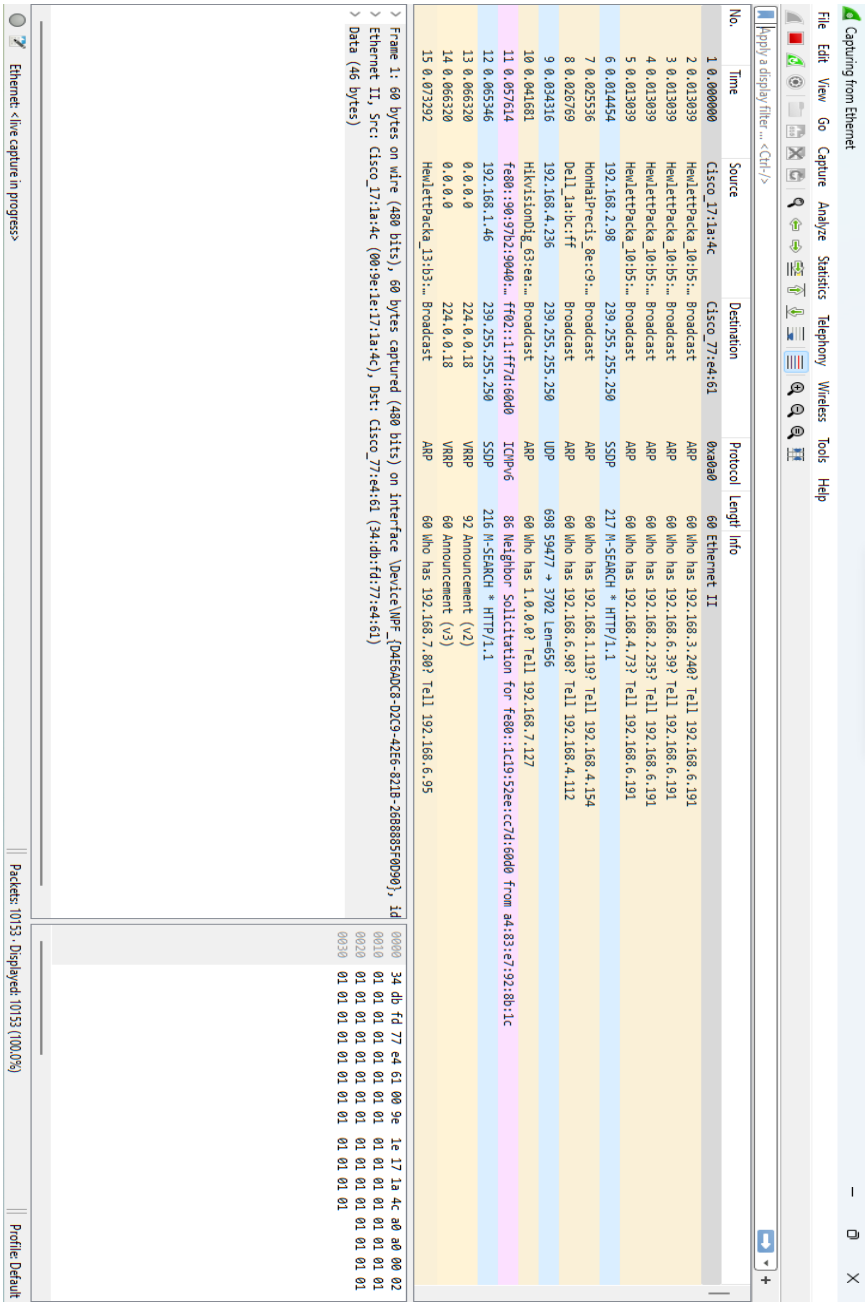


Fig. 2 Wireshark packet sniffing window

Explanation:

The top window is the packet sniffing window.
The bottom left window is the packet header details window.
The bottom right window is the packet data details window.

3. Wireshark window with different filters (arp, http, tcp, icmp, with ip address etc)

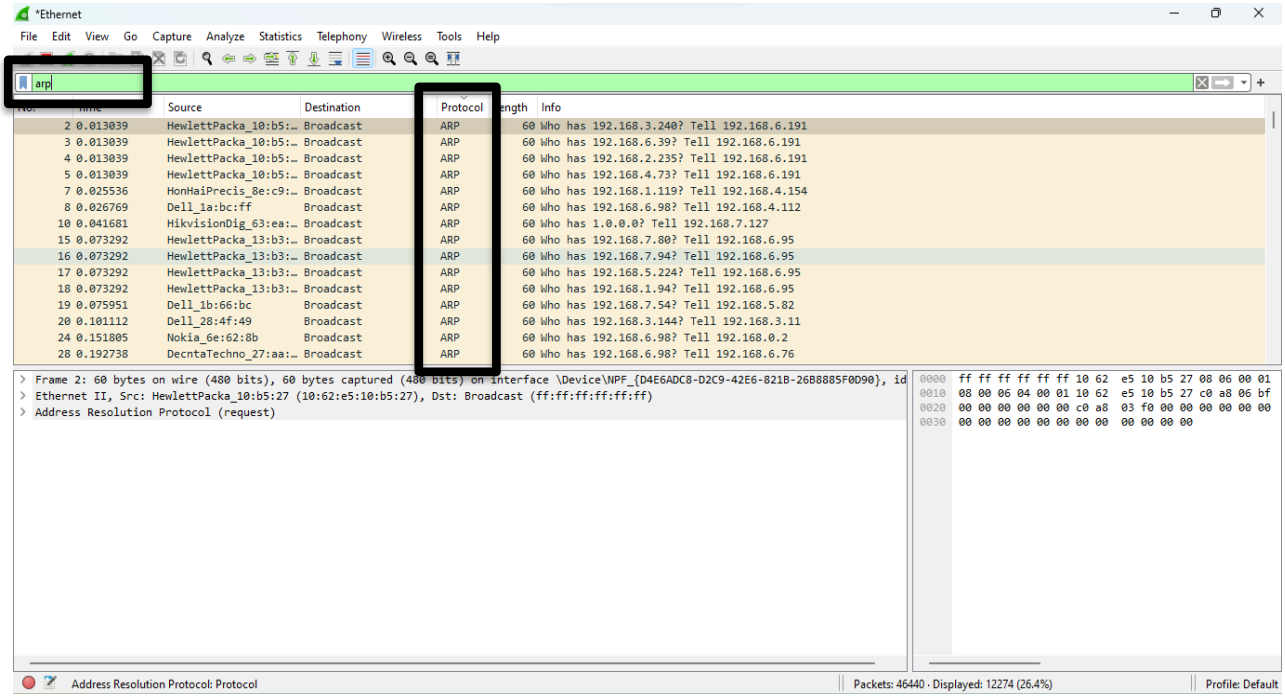


Fig.3.1 Wireshark window with different filters (arp)

To display only the packets of ARP protocol, we applied a filter here.

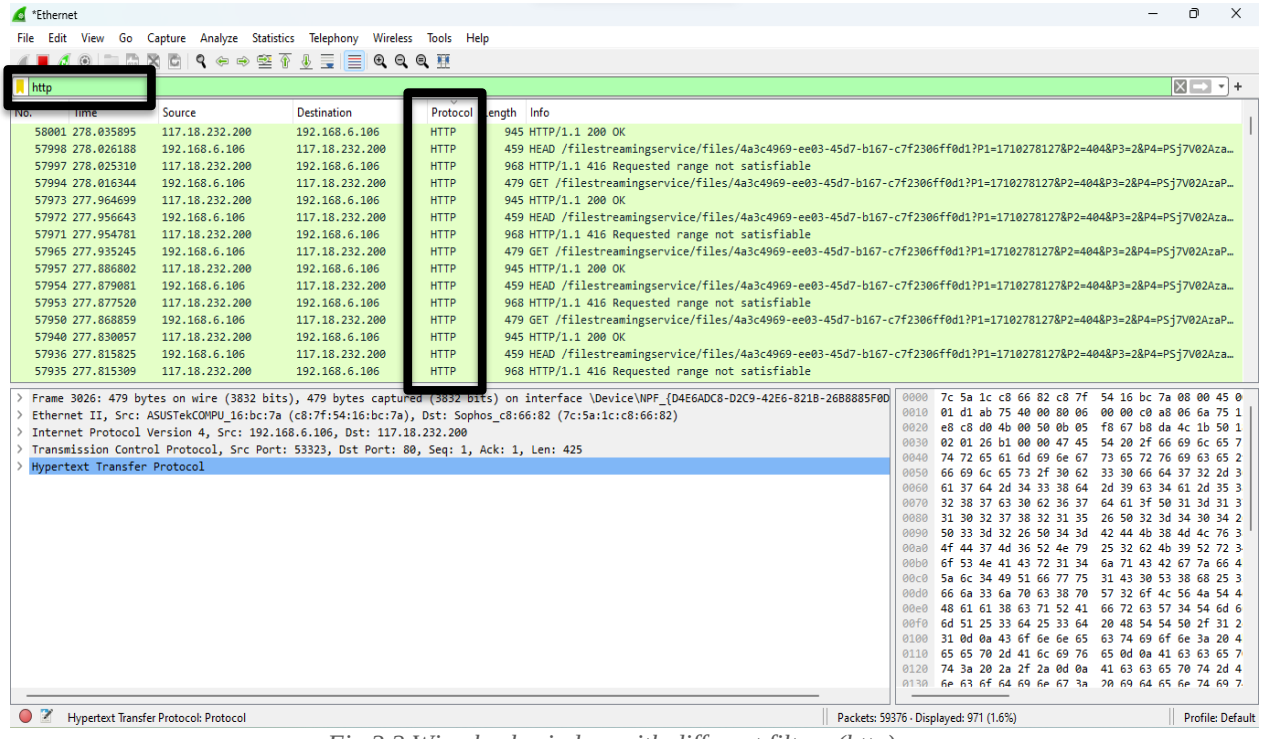


Fig.3.2 Wireshark window with different filters (http)

Similarly, here we applied a filter for packets for HTTP protocol.

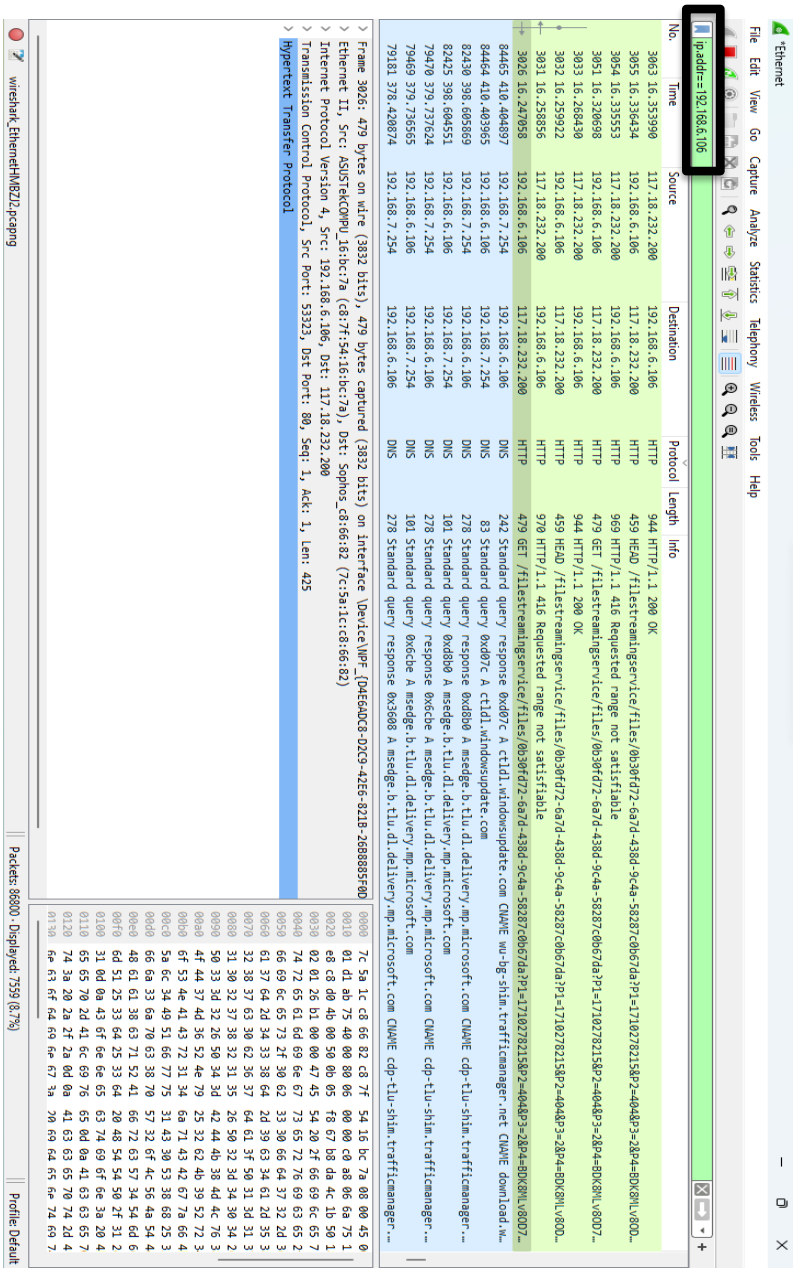


Fig.3.3 Wireshark window with different filters (ip.addr)

Here we applied a filter for IP address 192.168.6.106 and all the packets that have this IP address either in its source or destination are displayed. That means all the packets that are sent or received by the specified IP address are displayed.

4. Wireshark window with Ethernet protocol details.

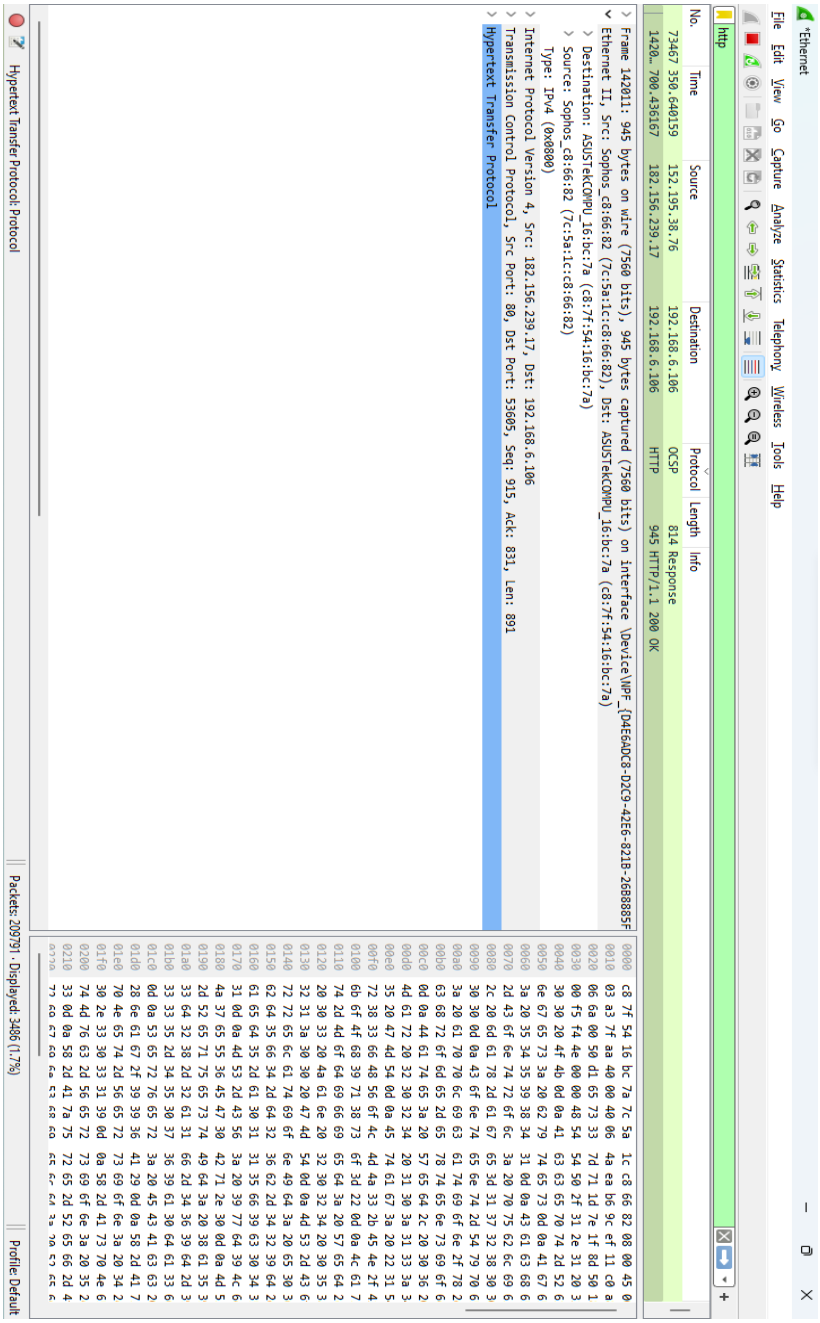


Fig.4 Wireshark window with Ethernet protocol details

Within the Ethernet protocol details window, essential information such as frame length, frame number, and link-layer addresses was prominently displayed. This data played a pivotal role in comprehending the intricacies of the data link layer in network communication.

5. Wireshark window with arp protocol details.

Fig.5 Wireshark window with ARP protocol details

The ARP packet details display includes information such as hardware type, protocol type, sender and target hardware addresses, as well as ARP operation codes.

6. Login window of an insecure webpage.

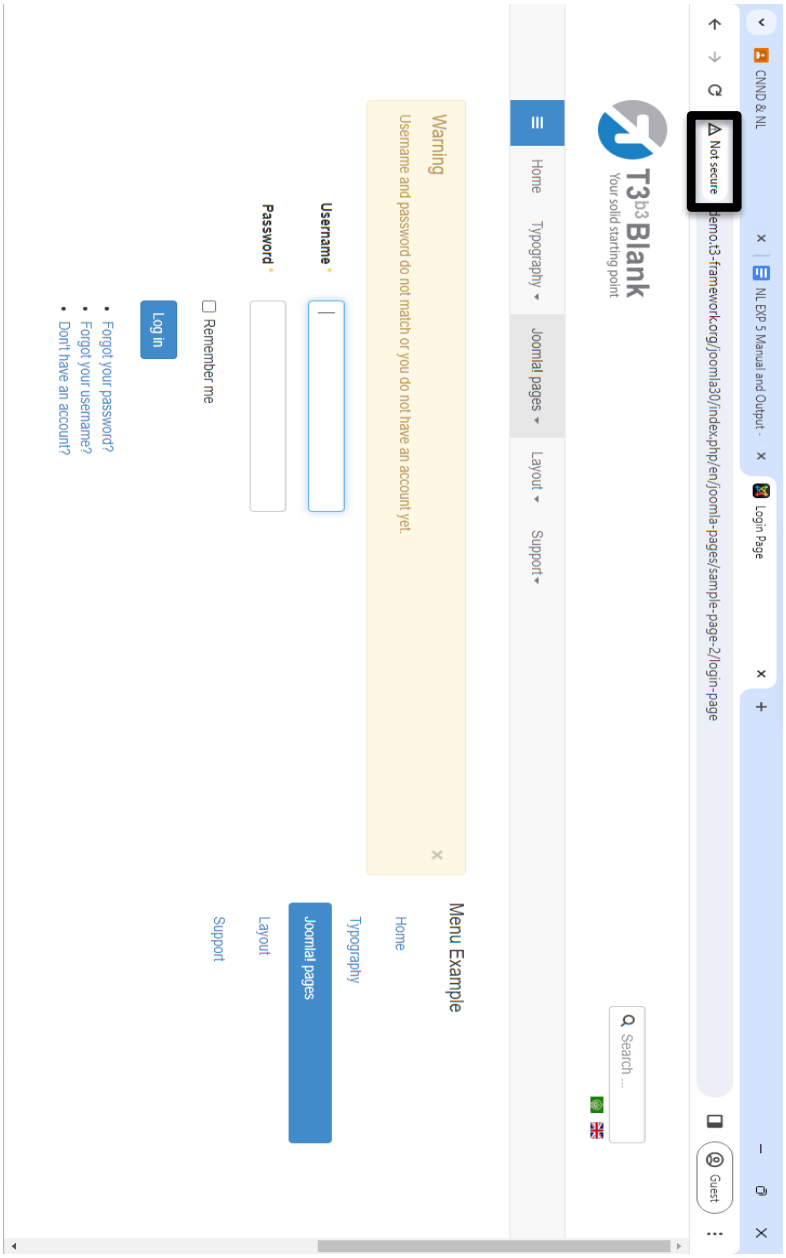


Fig.6. Login window of an insecure webpage

The login interface of an insecure web page exposes potential vulnerabilities in the website's security, underscoring the danger of transmitting sensitive information over unencrypted connections.

We used google dork *"inurl http login page"* to find this website

7. Wireshark window with http packet filtering for password sniffing.

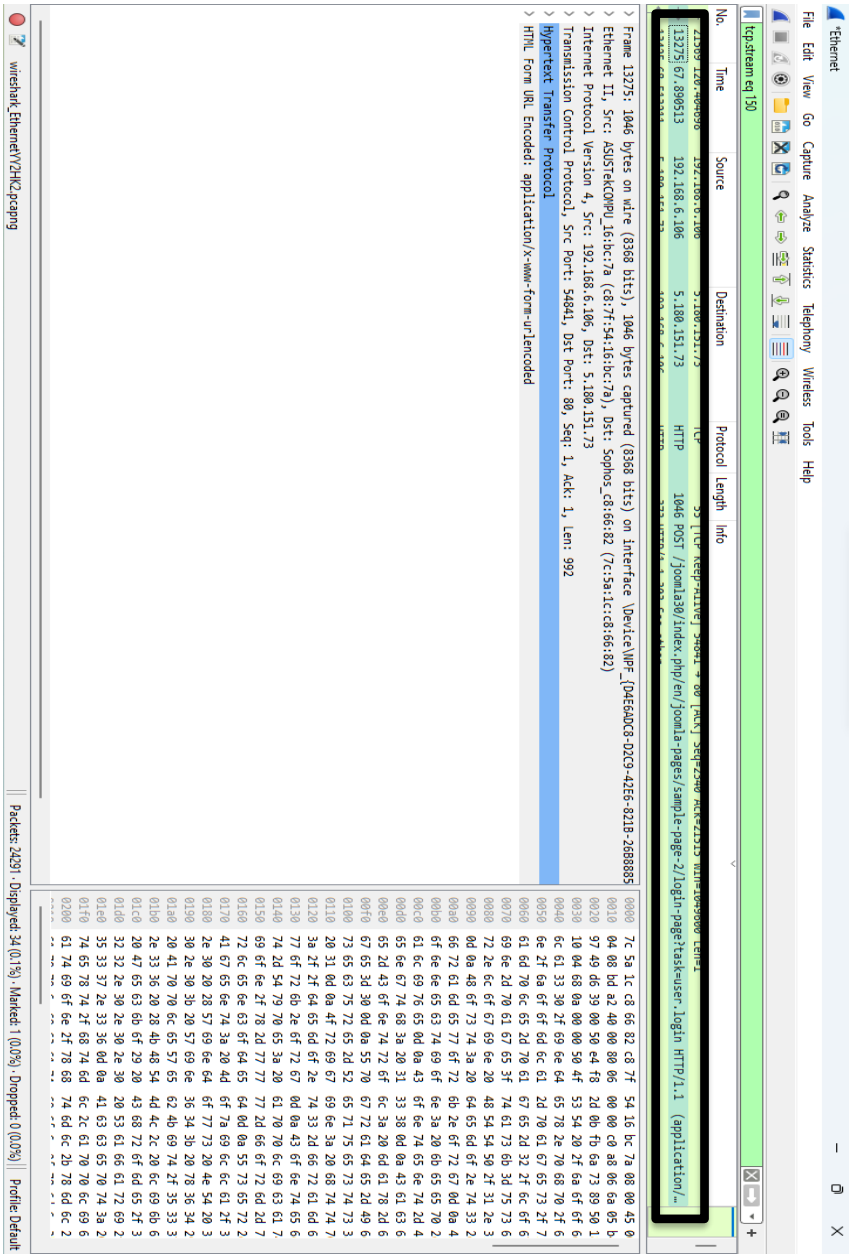


Fig.7. Wireshark Window with http packet filtering for password sniffing

The Wireshark interface, when configured with HTTP packet filtering for password sniffing, can detect plaintext passwords transmitted over unencrypted HTTP connections. This underscores the security risks associated with insecure communication protocols

8. TCP stream data window showing login id and password details.

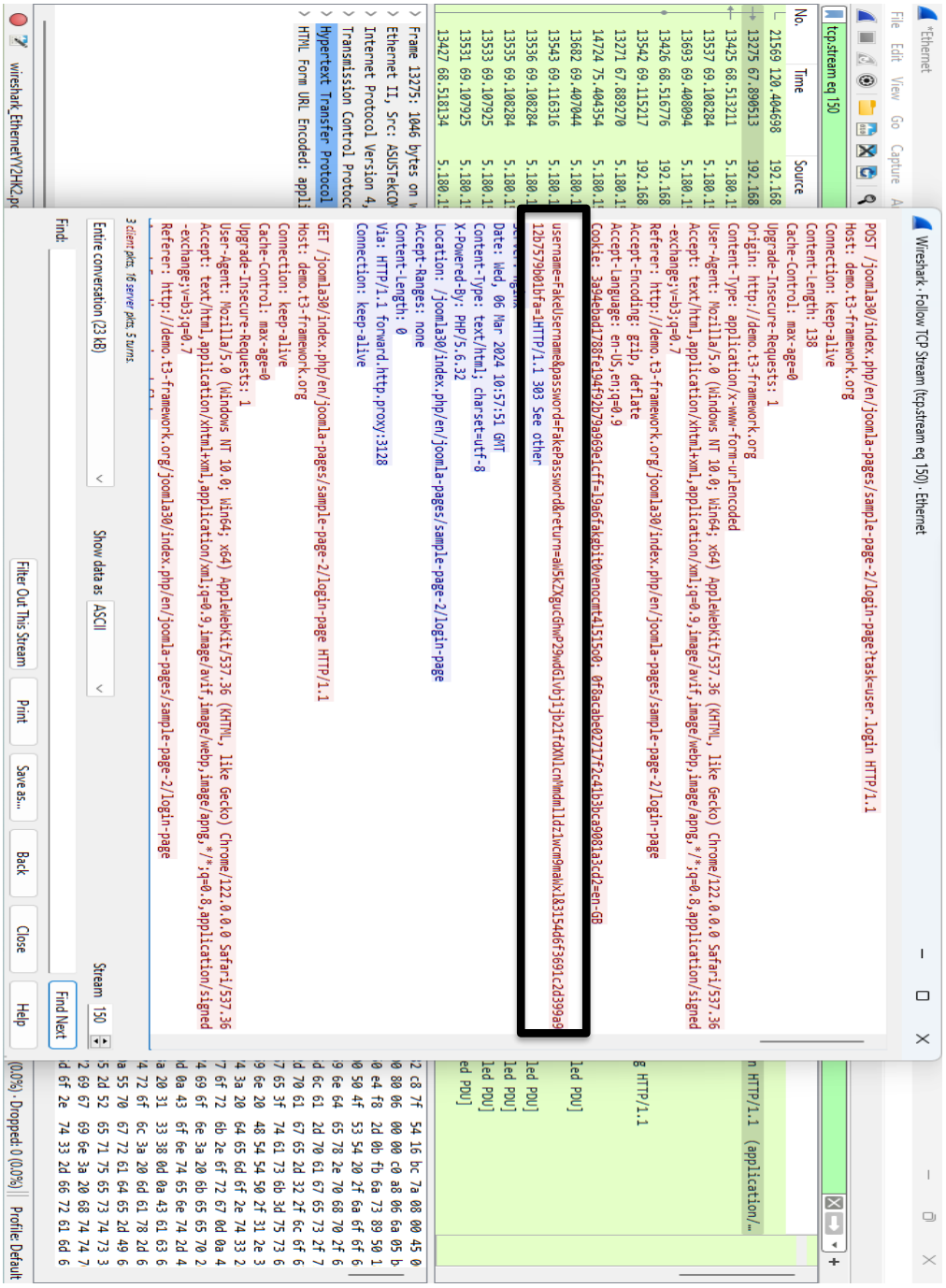


Fig.8. TCP stream data windows showing login id and password details

The TCP stream data window, displaying login ID and password details, exposes intercepted login credentials from network traffic, underscoring the critical need for secure communication protocols to safeguard sensitive information.