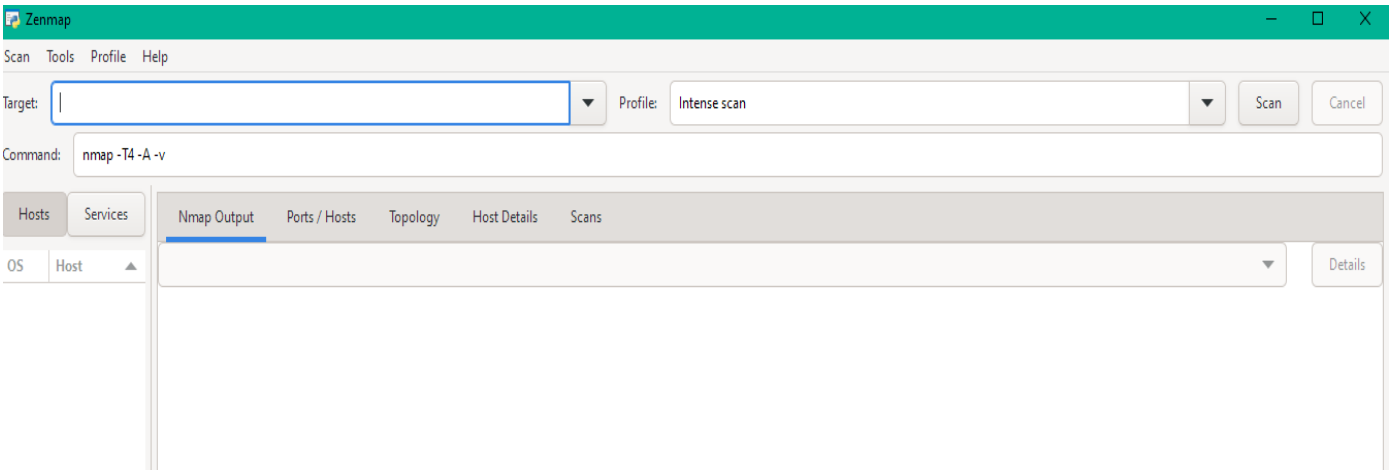
Nmap Port scanning:

Initially Nmap was used to see the network ports whether they are open/closed.

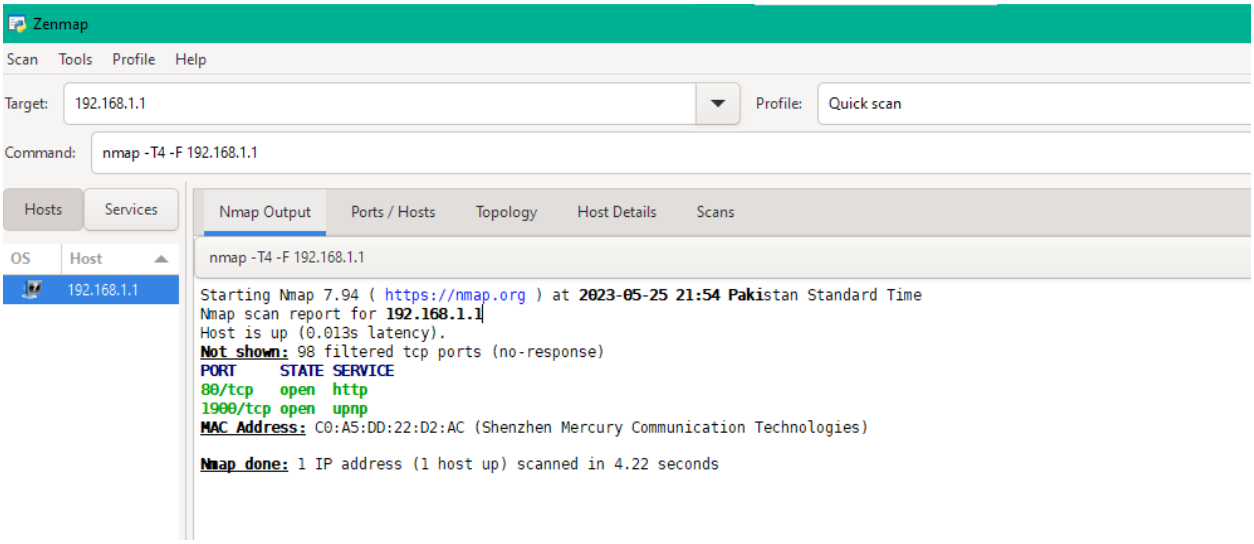
1. **Scan for service and version detection:**

   In addition to checking whether network ports are open or closed, Nmap can also identify the services running on those ports and provide information about the version of the service. This can be useful in understanding the specific software or application running on each port and its associated vulnerabilities.



2. **Vulnerability scanning:**

   Nmap has the capability to perform vulnerability scanning by using NSE (Nmap Scripting Engine) scripts. These scripts can help identify specific vulnerabilities present on the target system or network. You can utilize Nmap's extensive collection of NSE scripts to scan for common vulnerabilities, misconfigurations, or weaknesses in network services. This involves entering targets IP in the initial stage.

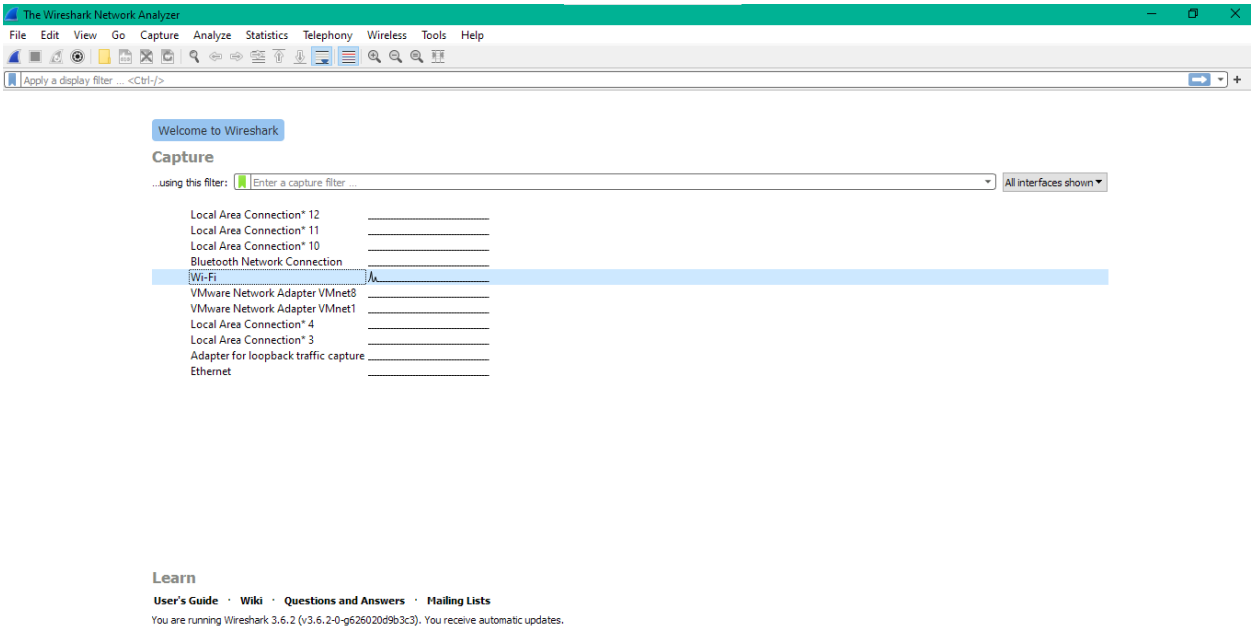


Wireshark Network sniffing:

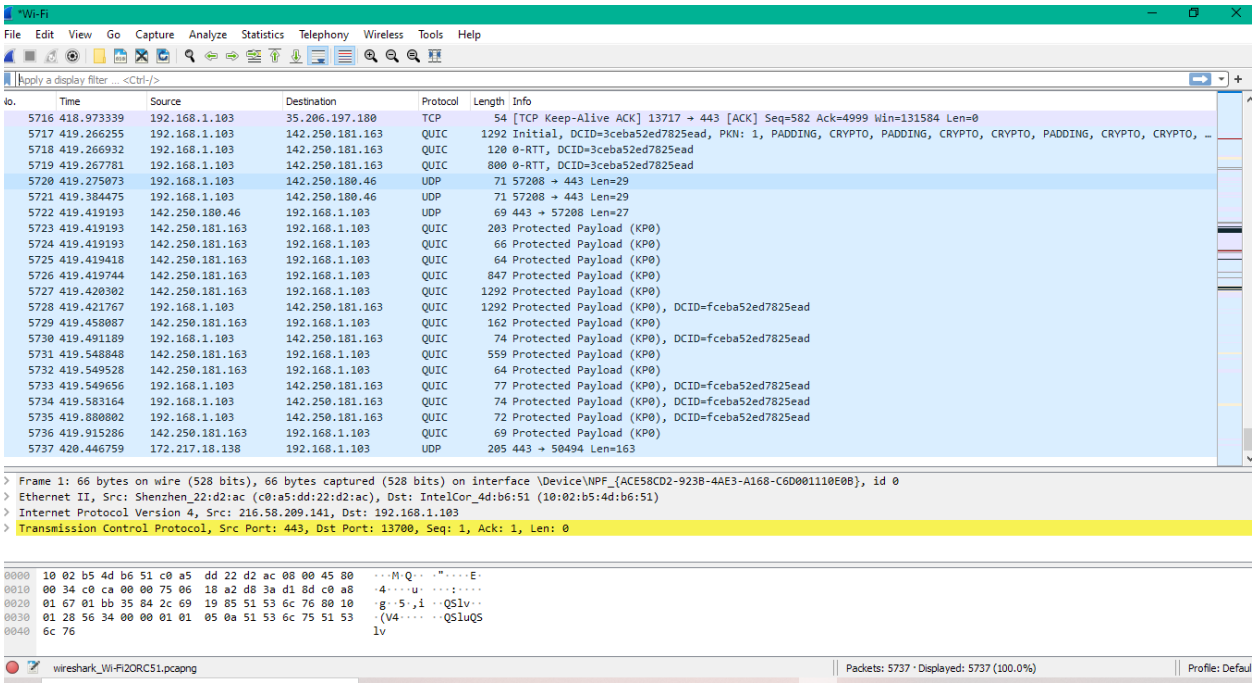
After the initial network vulnerability check using Nmap, we employed the use of a network sniffing tool (Wireshark) for capturing data packets on a network.

1. **Select a specific channel:**

   Wi-Fi networks operate on different channels, and by selecting a specific channel in Wireshark, you can focus your packet capturing on a particular Wi-Fi channel of interest. This can help in narrowing down the analysis and capturing relevant packets from a specific channel.

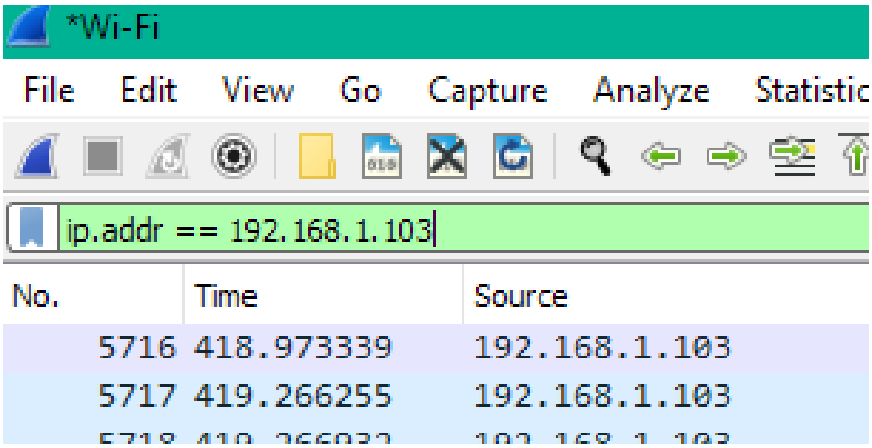


By double clicking on the selected channel, all the network traffic of that specific channel becomes visible.

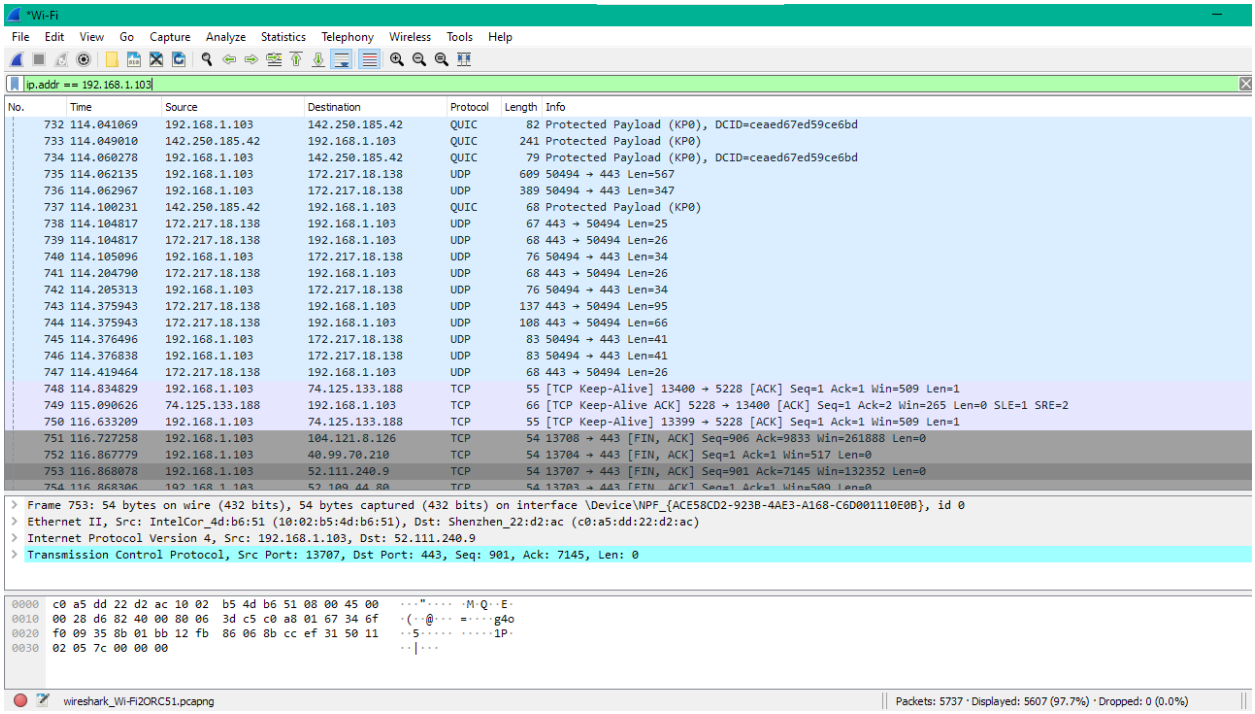


2. **Select a specific target:**

Wireshark provides a "Filter" field where we can enter filter expressions to selectively display packets that meet specific criteria. In this case, we'll use a filter expression to match packets with the desired IP address. The filter expression ip.addr == 192.168.1.103 is an example of a filter syntax. It uses the ip.addr field to filter for packets with a source or destination IP address that matches the specified IP address (192.168.0.103 in this example).



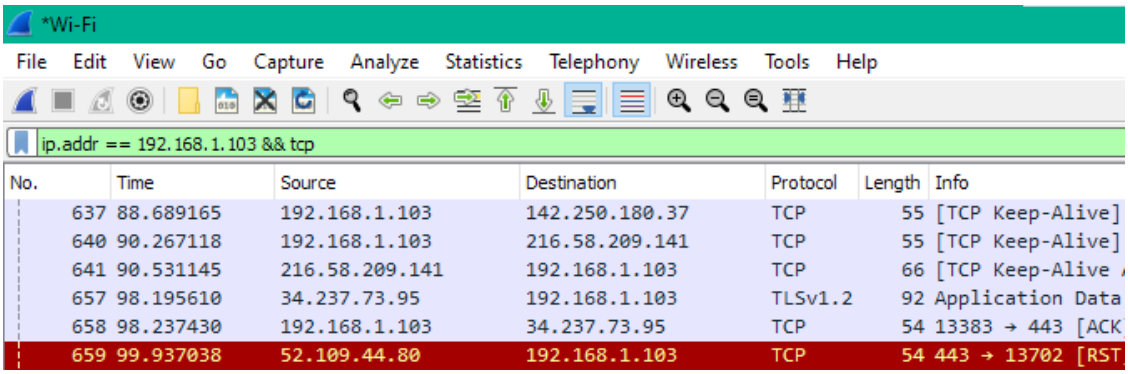
Now we can see the network traffic specific to the selected IP address.

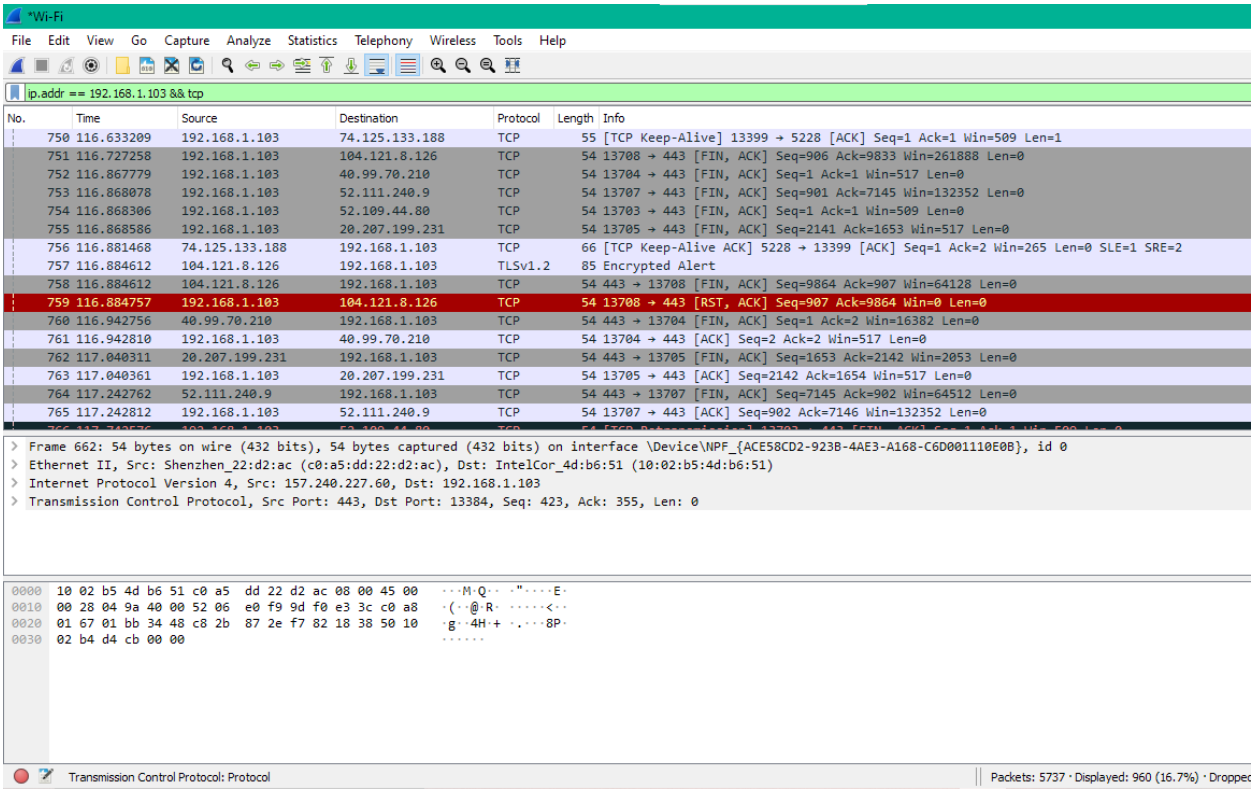


3. Filter packets by specific protocols:

Wireshark allows us to filter captured packets based on specific protocols. By applying protocol filters, such as HTTP, DNS, TCP, or FTP, we can focus on analyzing the traffic of protocols within the selected Wi-Fi channel. This can help us gain a deeper understanding of the specific protocols' behavior and potential security implications.

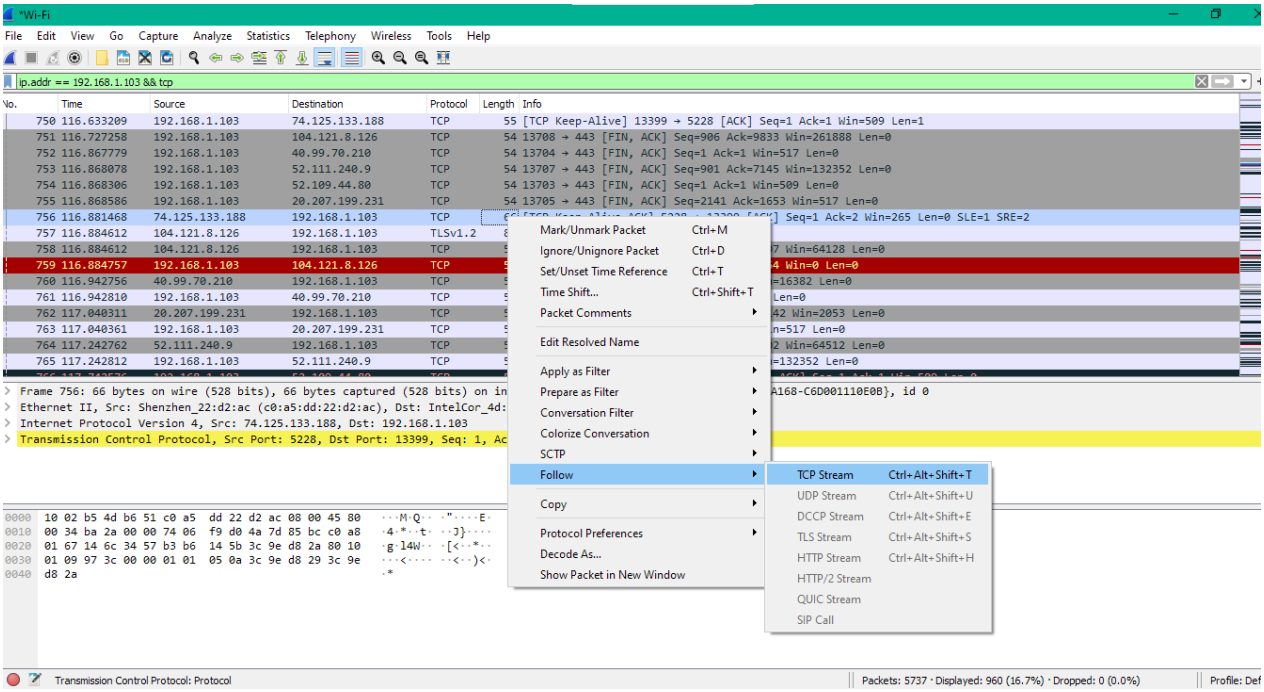
Enter the desired protocol filter expression to display packets related to a specific protocol. For example, to filter for TCP traffic, we can use the filter expression && tcp after the pervious expression ip.addr == 192.168.1.103 to filter for TCP packets



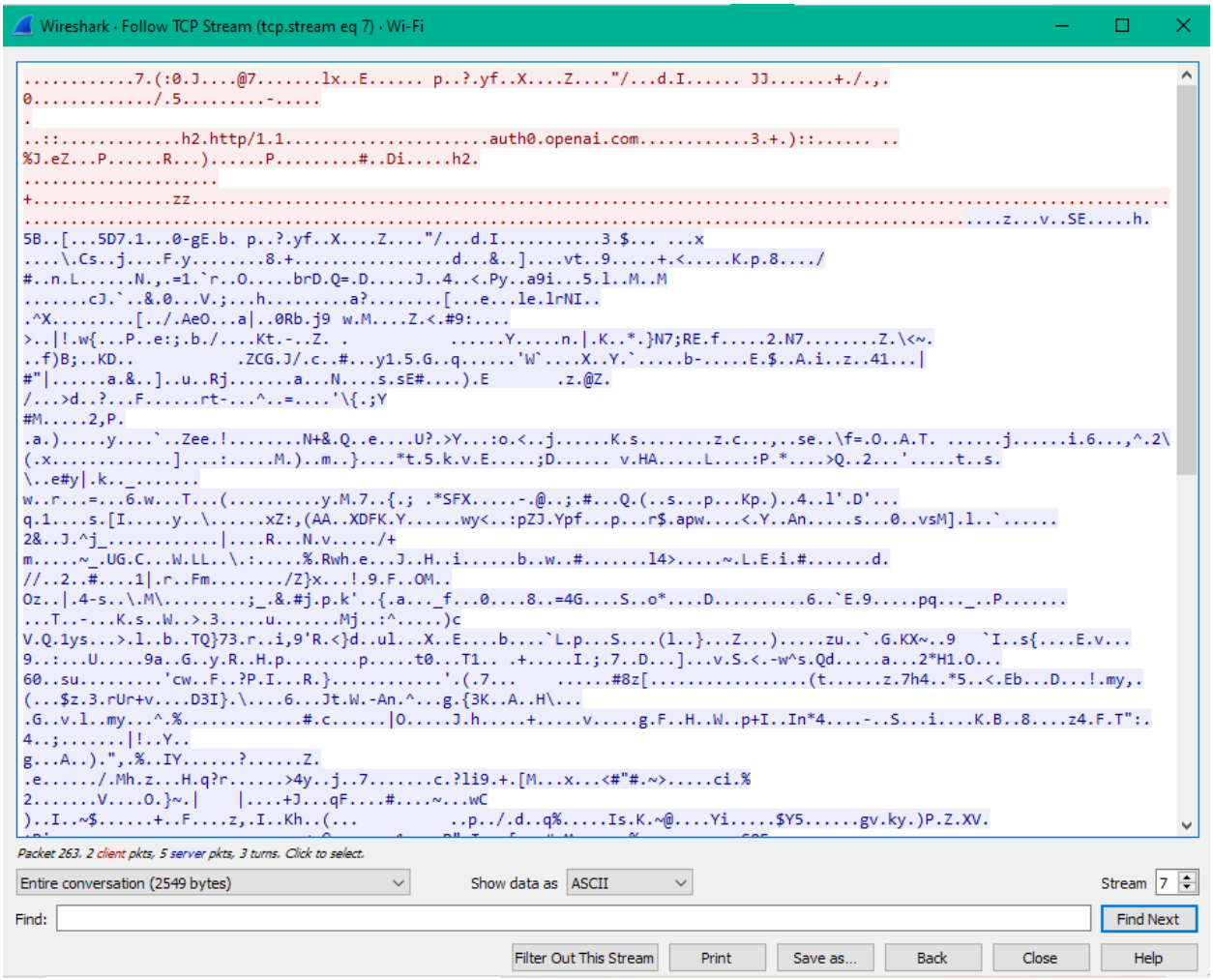


4. Analyzing Packet:

On selecting a specific packet we can view the encrypted contents. Initially by right clicking on the specific packet, selecting Follow option in the menu and then click on specific protocol (TCP).



The highlighted encryption (in Red) indicates the request sent by the user, and the encryption (in Blue) indicates the reply in response to the request.



Further Analysis:

The data received from the captured packet can be decrypted to obtain information using the website www.apackets.com .Where you can upload the saved packet for decryption.