International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Exploring Wireshark for Network Traffic Analysis

Brij Mala[1], Sanskar Agrawal[2], Aditya Sharma[3], Rupinder Kaur[4]

[1,2,3]B. E Scholars, Dept. of Computer Science and Engineering Chandigarh University- Mohali, India
[4]Professor, Deptartment of Computer Science and Engineering Chandigarh University-Mohali, India

Abstract
In today's world security and privacy are the most needed qualities of any network. From calling to online transactions all occur on the network on daily basis. So there is the need to analyse the network in which all these activities occur. Network analysis thus plays a vital role in maintaining and optimizing network performance, security and troubleshooting. This paper researches on the topic network traffic analysis using Wireshark. Wireshark is an open source network packet analyser and it can be used by security administrators to analyse the traffic, analyse data packets and their role in shaping network behaviour. The network analysis using Wireshark will help to know any suspicious or malicious traffic in our network so that timely action can be taken to avoid attacks like DDOS, Man-in-the middle etc.

Keywords: Network traffic analysis, Wireshark, Network Packet Analyzer

1 Introduction

1.1 Network Traffic Analysis

It is a process of capturing the traffic or data packets in the network and analyse them [3]. It also includes monitoring and evaluating the data that is transmitted over the network from the computers to the servers and back. It involves the examination of the captured data packets that flow through the network in order to gain insights into the network's behaviour, performance and security. The analyses helps the network administrators to track the utilization of the network, identify and suspicious activity, bottlenecks. In the event of breach, network traffic analysis provides valuable evidence for investigating, understanding the reasons for the attack. Network traffic analysis can be performed using various tools and techniques, like intrusion detection systems, packet sniffers, Wireshark etc.

1.2 Wireshark

Wireshark is an network packet analyzer used for network analysis by security administrators, network administrators in order to understand the flow of packets over the network [1]. Wireshark provides a platform to capture, inspect and analyse the packets and the traffic. It was originally known as Ethereal. It came into existence in order to better understand the inner working of network communication protocols like TCP, UDP, HTTP etc. Over time it has evolved as a versatile and indispensable resource to drive insights and reveal hidden dynamics of the network behaviour [2]. Wireshark thus helps in troubleshooting network issues, analysing packets, analysing various protocols and their role in the network traffic. Wireshark also provide various statistical methods and graphs to clearly visualize and understand the network traffic and thus helping the network security administrators to detect various active and pas-

sive attacks over the network. With the help of Wireshark we can get the IP addresses and the Port numbers of sender and receiver.

1.3 Packet Capturing

Packet capturing refers to the process of recording and intercepting individual data packets that flow over the network. These packets are the fundamental units of data transmission. The packets includes data like the IP address, the port numbers, length of the data, data itself and much more. Capturing allows the network administrators, security professionals, analysts to inspect, analyse and examine the network traffic in real time or offline.

2 Literature Review

Table 1. Review Summary

Author/ Creator	Title	Purpose	Source	Summary
Muhammed Alfawareh	A Deeper Look into Network Traffic Analysis using Wireshark	This paper discusses the optimization of traffic analysis performance, detection of network forensics and spam, network proofing with penetration testing, policy formation, and data delivery in integrated systems, while also discussing countermeasures to reduce this risk [3].	Research Paper	The paper discussed the use of wireshark for network traffic analysis, its role in addressing network forensics, and the risks associated with obtaining useful information for attacks or stealing data, while also addressing solutions [3].
G Jain, Anubha	Application of SNORT and Wireshark in Network Traffic Analysis.	Intrusion detection systems analyze packets to secure data transmission in networks. Wireshark is a popular tool for detecting intrusions, but it can intercept and analyze network encrypted traffic. SNORT captures live packets from the internet, generating a log file. These log files are exported to Wireshark, where the captured network packets are analyzed. The I/O graph displays the packet flow, total traffic, TCP errors [2].	Research Paper	This paper shows how to use Wireshark tool to generate all the possible details of log file. The I/O graph provides an overview of packet flow showing total traffic, measured in bytes or packets per second [2].
Bindu Dodiya, Umesh Kumar Singh	Malicious Traffic Analysis using Wireshark by collection of	This paper demonstrated the application of Wireshark in diagnosis of network protocol and identify malware by collecting compromise indicators [1].	Research Paper	This paper showcases packet analysis using indicators of compromise, demonstrating its usefulness in network

	Indicators of Compromise			forensics. WireShark's analysis can identify various security threats and attacks on networked computer systems [1].

3 Methodology

Wireshark uses several phases for network traffic analysis. The steps are as follows for capturing the network packet:

- Select the network on which you want to perform traffic analysis from various available options like Ethernet, Wi-Fi , LAN connection etc.
- If you wish to apply filters before starting the capture you can do so by typing the same in the text field provided by the Wireshark. Filters like choosing a specific protocol or port and many more can be applied.
- Traffic would be captured and be available for analysis. We can also save the captured traffic for future work if we wish to do so.
- The packets are then decoded, examined, and analysed by the users.

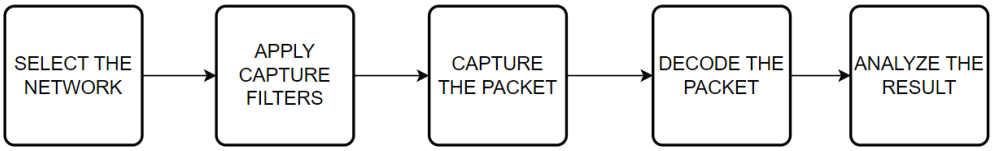

Fig 1. Flowchart

4 Transmission Control Protocol

TCP (Transmission Control Protocol) is a transport layer protocol that is used to transmit packets from sender to receiver over the network. It is a connection oriented protocol, and thus uses three-way handshake for connecting devices before sending data over the Internet. This connection remains established till the communication is completed.

TCP HEADER:

- **Source Port:** It includes the source or sender's unique port number. The field has 16 bits[5].
- **Destination Port:** It includes the receiver's or destination's precise port number. The field has 16 bits[5].
- **Sequence Number:** This identifies the amount of data transferred throughout the TCP Session. It's a 32-field here. Initial sequence number for a new connection is a random 32-bit value. Utilizing this sequence number, the recipient replies with an acknowledgment to the sender. To make it simpler to comprehend, Wireshark utilizes relative sequence numbers beginning with 0[5].
- **DO (Data Offset):** The header length is another name for the 4 bit data offset field. It provides the length of the TCP header to identify when the actual data starts [5].
- **RSV:** The reserved field's three bits are set to 0 and not used[5].

- **Flags:** There are nine flag bits, are also known as control bits. This field is used to create connections, transmit data and break connections:
  - **Urgent Pointer: (URG**) The data should be viewed as having priority over other data when this bit is set [5].
  - **ACK**: abbreviation for acknowledgement [5].
  - **PSH:** This stands for PUSH. This instructs a program to send the data right away rather than waiting for it to fill the whole TCP segment [5].
  - **RST**: This resets the connection. We must immediately cut off the connection, if we receive it. This is not the typical technique to close a TCP connection; it is only done when there are irrecoverable faults [5].
  - **SYN:** This is used to establish the first sequence number and for the initial three-way handshake [5].
  - **FIN:** The TCP connection is ended using this finish bit. TCP is full duplex, so in order to terminate a connection; both parties must use the FIN bit. This is how we typically cut off a connection [5].
- **Window:** The window field is of 16-bit. It is used to represent the maximum length of bytes that a receiver will accept. It does this by informing the number of bytes after the sequence number in the acknowledgment field [5].
- **Checksum:** Checksum is a 16 bit field which is used to tell whether the TCP header is in a good situation or not [5].
- **Urgent Pointer:** URG is a 16 bit field and when this bit is set then it acts as a marker for the end of the urgent data [5].
- **Options:** This is an optional field. It's size lies between 0 to 320 bits [5].

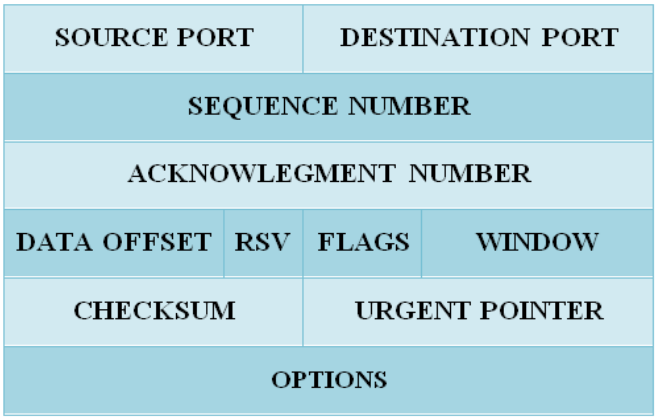

Fig 2. TCP Header

5 Wireshark: Network Traffic Capturing

The Wireshark has **easy to use interface**. Users can easily capture the packets and analyse those packets. It is available for both **UNIX and Windows**. Captured packets can also be exported and imported in a number of capture file formats for further analysis. Colour coding is also provided by Wireshark, helping the user to analyse the packets. Below table shows the various colouring rules.

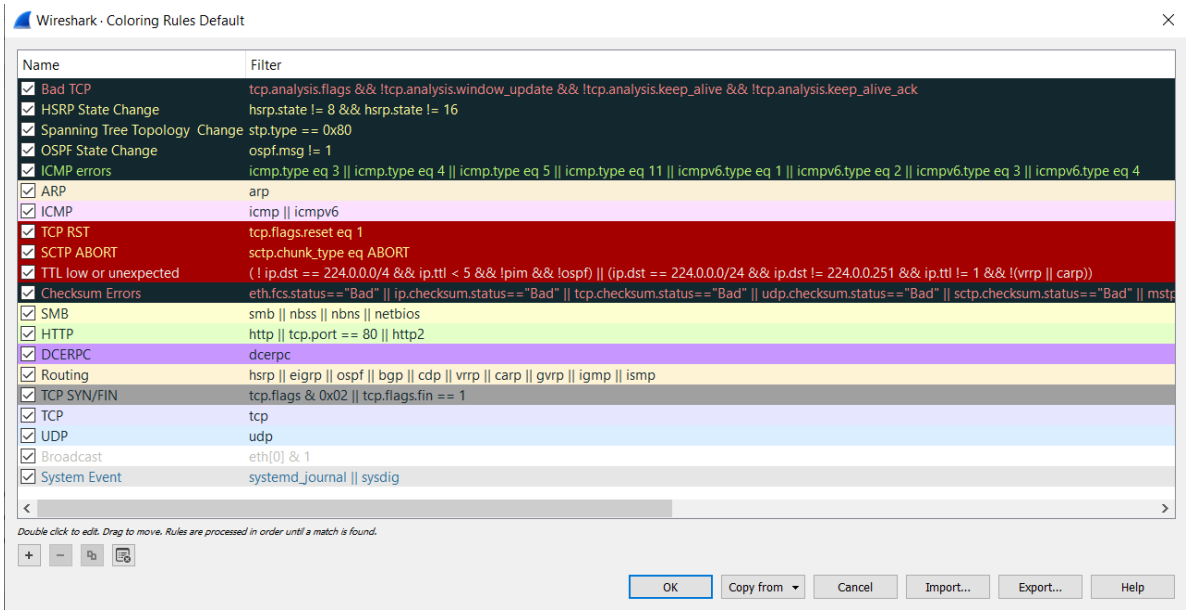

Fig 3. Coloring Rules

Interface to capture traffic:

Wireshark provides easy to use interface to capture the traffic.The below image shows the interface provided by Wireshark to capture the traffic. To start the traffic capture one needs to select the network like Wi-Fi, local area connection, adapter etc. We can also select various interfaces like wired, Bluetooth, wireless from the drop down menu. If we wish to apply filters like specific port number or specific protocol before starting the capture, we can do so by typing the same in **using the filter text box.** If we have captured any traffic before we can also open that traffic from the recent files available on the very first interface of Wireshark. As soon as a network is chosen, Wireshark starts the network traffic capturing and whatever is searched on various platforms, like Google Chrome, traffic pertaining to those platforms is captured for traffic analysis.

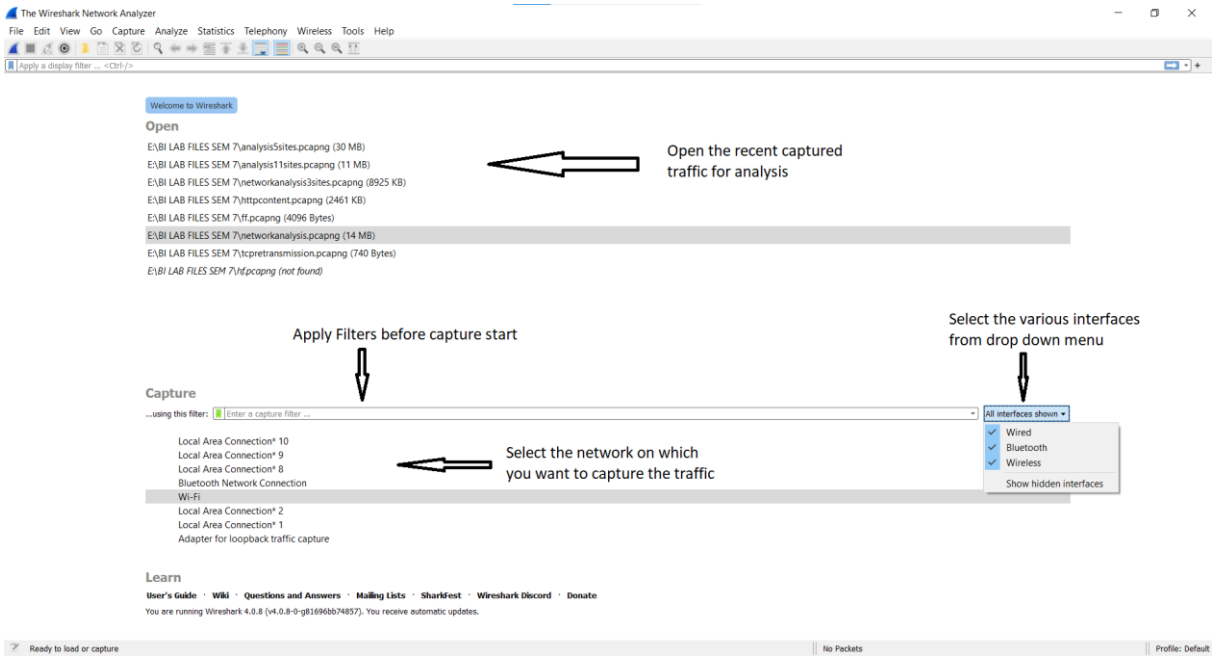

Fig 4. Wireshark Interface

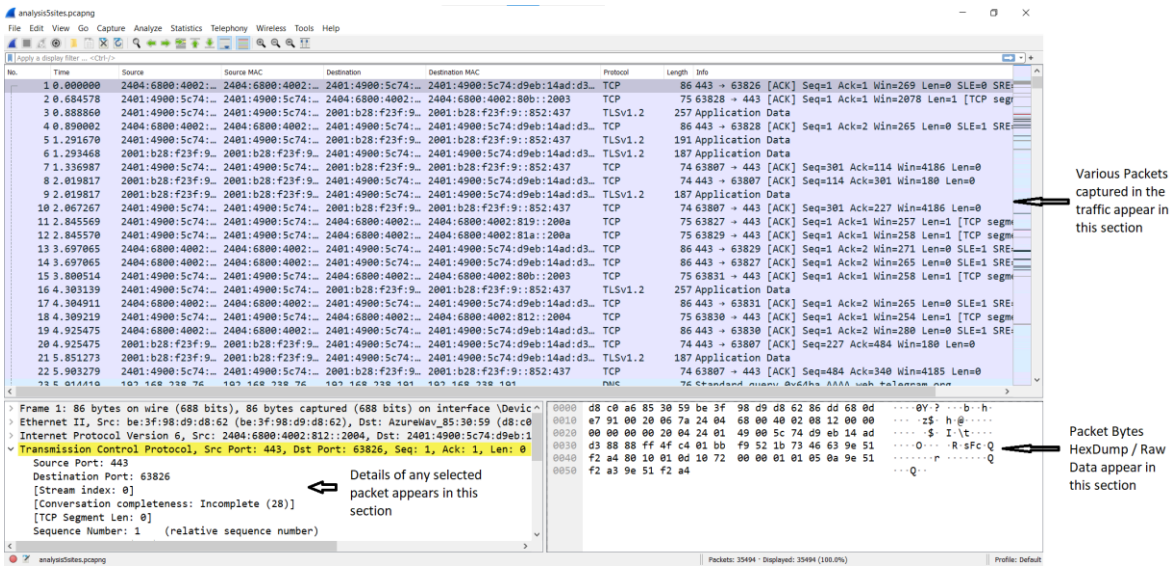

Fig 5. Packet Capturing Interface

Wireshark captured traffic can be seen as in the above figure. It mainly has the following sections:

1. **Menu Options**: From here various menus can be selected like file menu, edit menu, analyze menu etc. to help in analysis , importing and exporting of captured traffic.
2. **Filters:** In this section filters can be applied while capturing the packets. Specific protocol like http, tcp ,udp etc can be selected. Applying the filters helps in analyzing specific and relevant packets instead of all packets.
3. **Packet Traffic List:** This section shows the various packets captured in the traffic with details like time, source address, destination address, protocol, descriptions etc.
4. **Packet Details:** This section shows details regarding a particular selected packet like the frame details, TCP header details etc. From here various details can be seen regarding any packet like destination port , source port, length of packets, Bytes in the packets etc to help analyze the packet.
5. **Packet Bytes Hex Dump:** This sections shows the hex dump of the packets or any raw data in human readable form.

6 Results and Discussion

During the traffic analysis, traffic was captured from different sites and for a particular duration. This was done in order to check the scalability of the Wireshark depending on the number of sites searched, amount of activity done on those sites etc.

Following images show the summary of the captured traffic depending on the number of sites and amount of activity performed. It shows information regarding the capture, such as time remaining, packet and byte counts, and similar data [4].

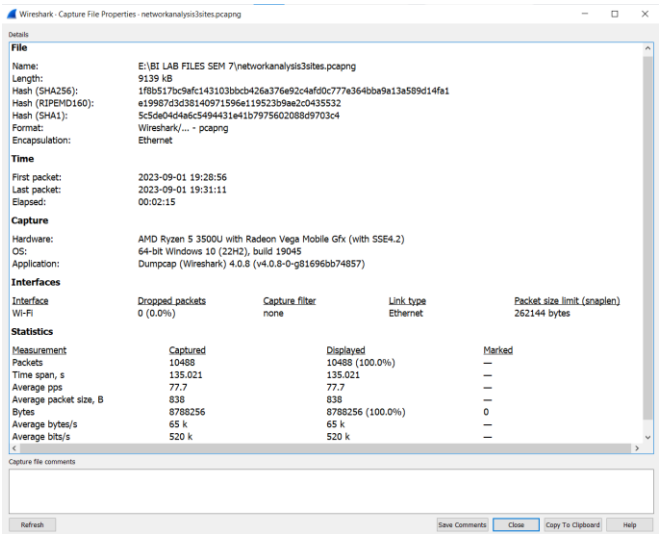

Fig 6. Capture File Properties – 1

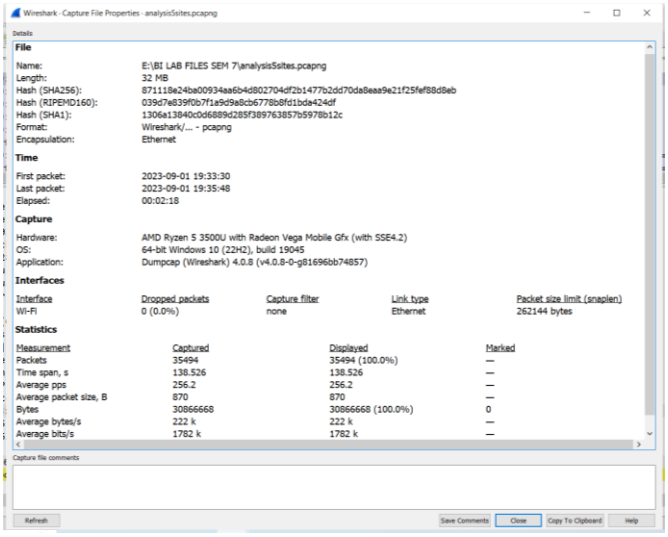

Fig 7. Capture File Properties – 2

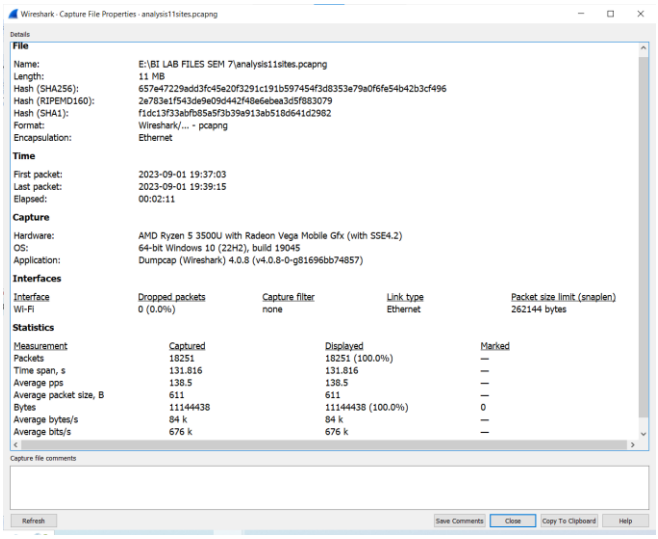

Fig 8. Capture File Properties – 3

Below table shows the comaprison of the above captured traffic :

Table 2. Comparison Of Above Captured Traffic

Sr. No.	No of Sites	Amount of Activity	Duration (in minutes)	Size of the file	Dropped Packets	Average pps	Captured traffic packets
1	3	Low	02:15	8925KB	0	77.7	10488
2	5	Moderate	02:18	30MB	0	256.2	35494
3	11	Low	02:11	11MB	0	138.5	18251

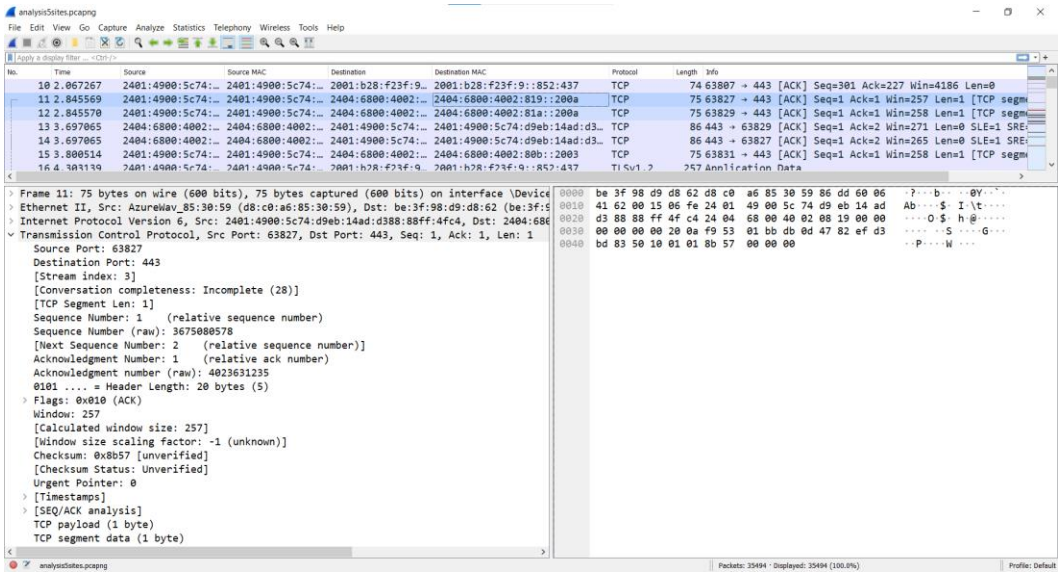

Fig 9. TCP Header Details

The above image shows the TCP Header details for the **packet number 11** captured.
Clearly from above:
Source Port: 63827
Destination Port: 443
Sequence Number: 3675080578
Acknowledgment Number: 4023631235
Data Offset: 0101
Flag: ACK
Window: 257

Applying filters and analysing the packets:
In this traffic, protocol filter of HTTP was applied, which resulted in showing the packets related to http protocol only as shown in below figure.

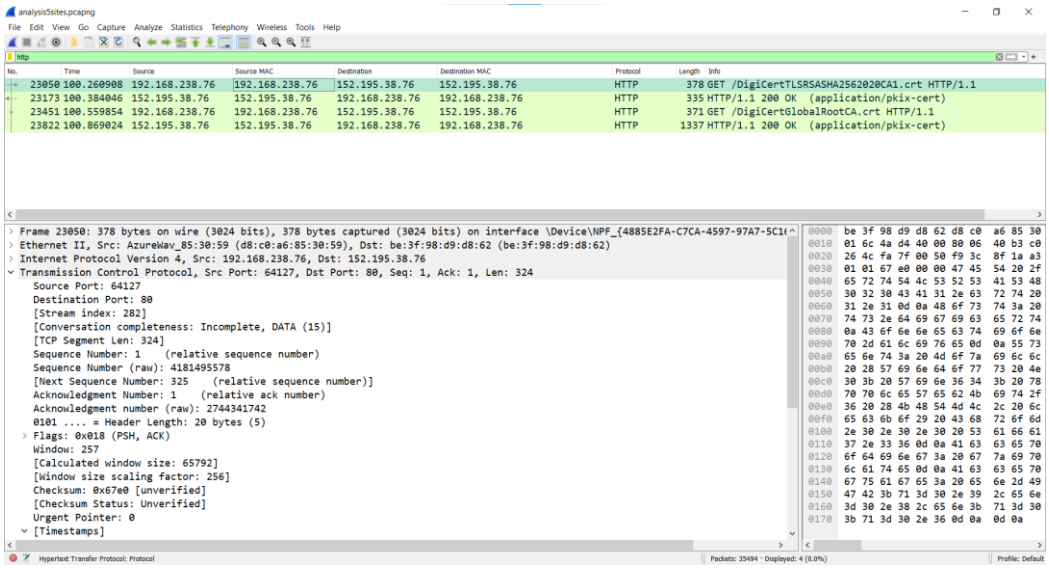

Fig 10. HTTP Protocol Captured Packets

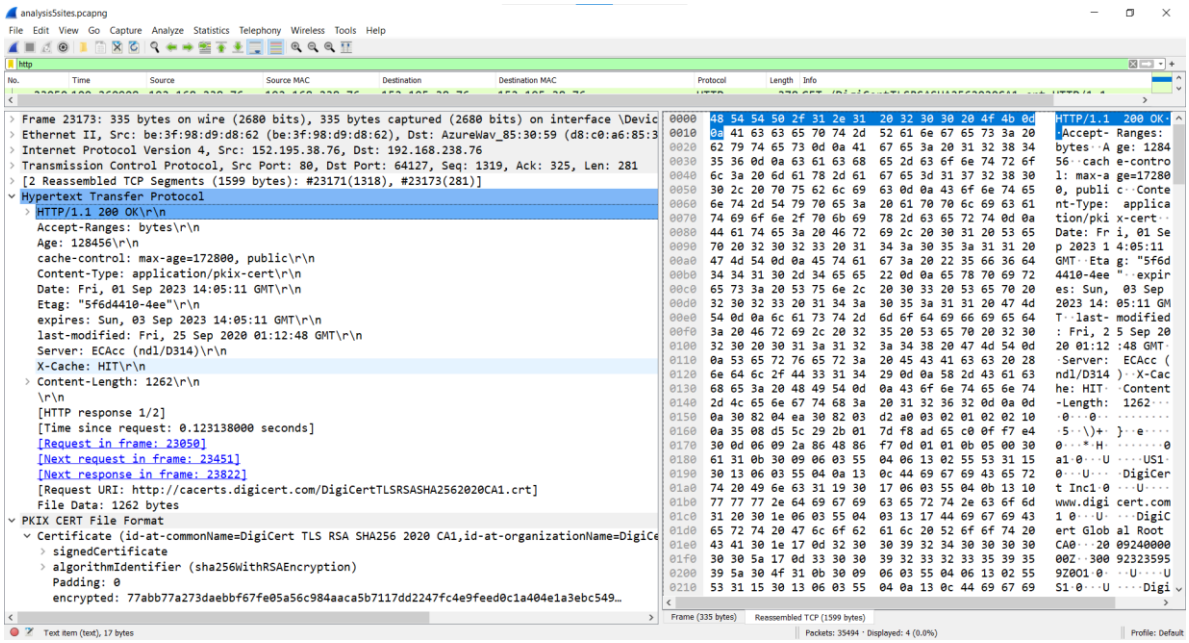

Fig 11. HTTP Protocol Packet Details

On selecting the Hypertext Transfer Protocol in Packet Details section on **packet number 23173** it shows the HTTP header details. From above it can be seen that this packet contain details regarding the certificate.

From above following details can be figured out:

- **Request URI:** http://cacerts.digicert.com/DigiCertTLSRSASHA2562020CA1.crt
- **Algorithm used for encryption:** SHA 256 along with RSA
- **serialNumber:** 0x0a3508d55c292b017df8ad65c00ff7e4
- **Algorithm Id**: 1.2.840.113549.1.1.11 (in this SHA256 along with RSA encryption is used)

Such field can be used by attackers in wrong way which can also result in various attacks. This shows that if any http site is used by any user, and if any attacker had access to their network they can easily trace these packets and manipulate the data within the packets, **resulting in the Man –in – the- middle or redirecting attack.**

Exploring the Statistics on the Captured Traffic:

Protocol Hierarchy: Protocol Hierarchy display the number of packets and number of bytes in those packets for various protocols that were captured during for network analysis all the protocols are arranged in the same hierarchy as they were found in the traffic. It provides the count of packets in which the protocol is present and the packet in which it is the last protocol in the stack. These last-protocol counts let you know how many packets—along with the corresponding byte count—ended in a certain protocol. They are listed under "End Packets" and "End Bytes" in the table[4].

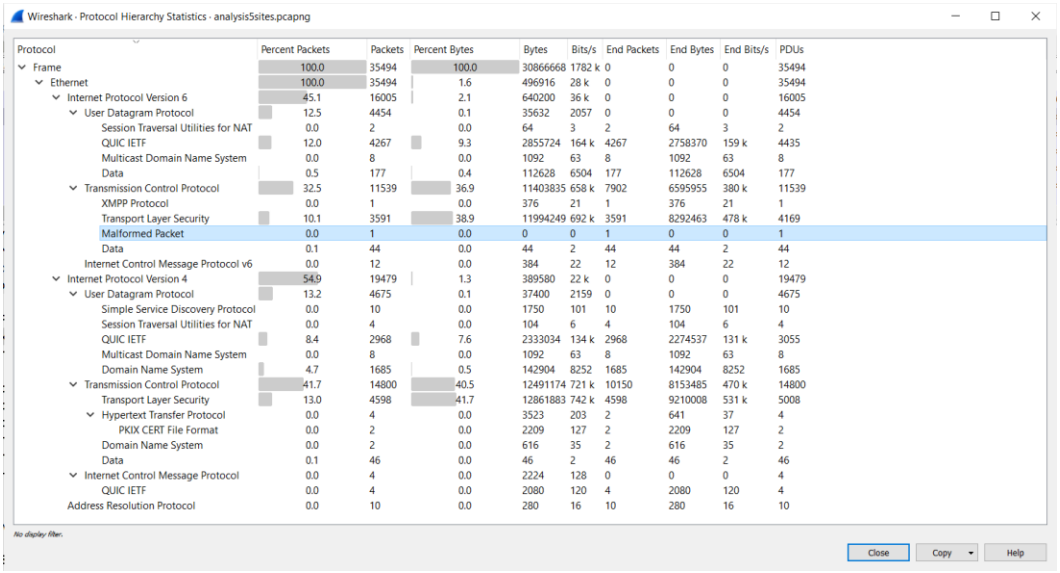

Fig 12. Protocol Hierarchy Statistics

Flow Graph: Flow graph shows the connection between the hosts. For each connection that was captured it shows the packet timing, direction, ports, and comments. It provides filters like ICMP (Internet Control Message Protocol) flows, ICMPv6 flows, UIM flows, and TCP flow [4]. The flow graph window provides different controls based on that. With the help of flow graph you can easily figure out various port numbers and IP addresses and thus can easily get to get know if any unusual port number or IP address occurs in the traffic [4]. The below figure shows the flow graph for the captured traffic.

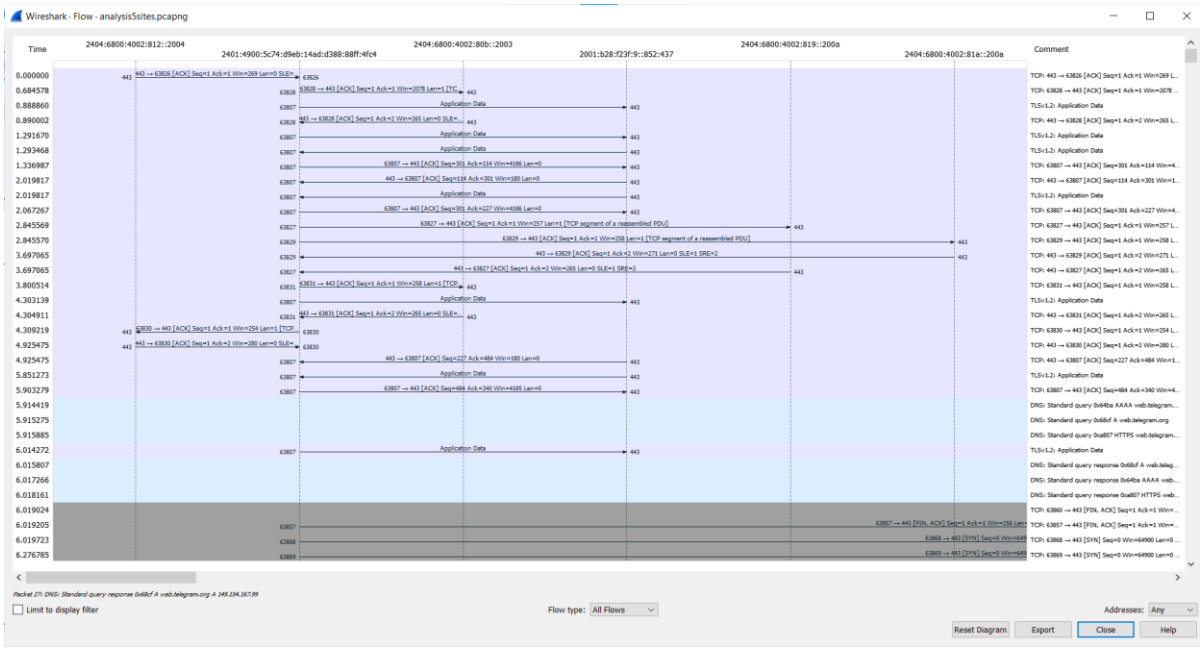

Fig 13. Flow Diagram

IO GRAPH: Display the number of packets or the amount of bytes per second for all packets that match the chosen filter. By default, only one graph displaying the number of packets per second will be shown [4].

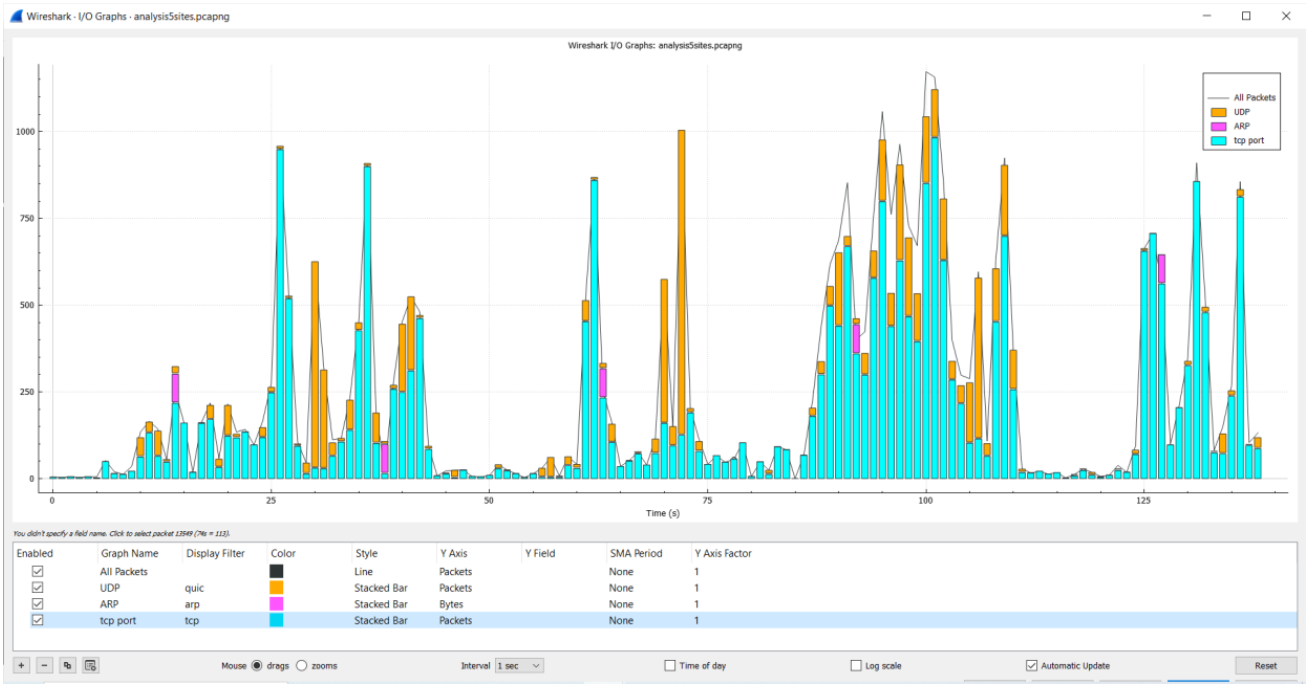

Fig 14. I/O Graphs

HTTP -> Packet Counter: The packet counter the data regarding the HTTP packets. From here, we can analyse if there was any redirection or any kind of error. It helps in knowing if any attack like DDOS attack took place, or were any packets redirected to any other unusual address. From the below figure, we analysed that all the packets have 2xx Response, indicating that all packets were transmitted successfully and no packet was dropped.

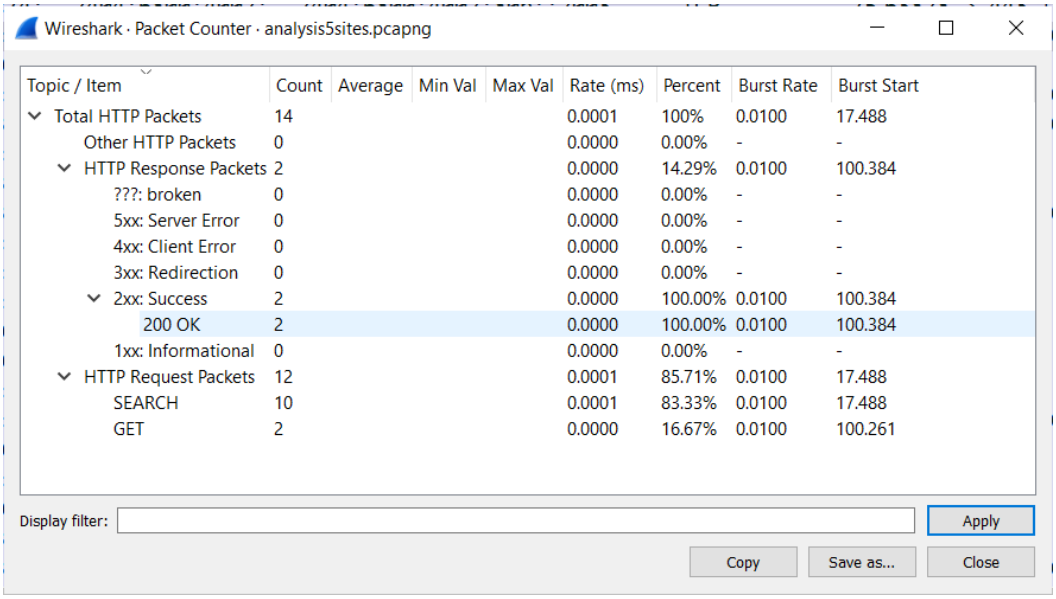

Fig 15. Packet Counter

7 Conclusion

In this paper, we examined network traffic analysis and its significance. The document also includes information and instructions for capturing traffic with Wireshark. The paper discusses how Wireshark can assist security and network administrators with packet capture and analysis. This paper demonstrates how

graphs such as the flow graph and the IO graph may be plotted and used to investigate captured traffic. We learned about the TCP header and how to trace IP addresses, port numbers, sequence numbers, and other information from captured traffic. Wireshark's efficiency was examined as traffic increased. Wireshark is a very significant tool in network traffic analysis, and if used properly, it may assist administrators notice any suspicious or anomalous activity in the network in real time, allowing them to take appropriate action to prevent any attacks.

References

1. Dodiya, Bindu, and Umesh Kumar Singh. "Malicious Traffic analysis using Wireshark by collection of Indicators of Compromise." *Int J Comput Appl* 183.53 (2022): 1-6.
2. Jain, G. "Application of snort and wireshark in network traffic analysis." *IOP Conference Series: Materials Science and Engineering*. Vol. 1119. No. 1. IOP Publishing, 2021.
3. Alfawareh, Muhamed. "A deeper Look into Network Traffic Analysis Using Wireshark." (2015).
4. *Wireshark(1) Manual Page*. wireshark(1). (n.d.). https://www.wireshark.org/docs/man-pages/wireshark.html
5. Molenaar, R. (2019, October 21). *TCP header*. NetworkLessons.com. https://networklessons.com/cisco/ccie-routing-switching-written/tcp-header