Network Packet Analysis using Wireshark

Introduction

The primary objective of this project is to analyze network packets to understand data transmission within a network. Using Wireshark, a robust network protocol analyzer, we will capture and dissect the packets to gain insights into network behaviors and potential security issues.

Tools and Environment

Wireshark: Version 3.6.6

Operating System: Windows 11

Network Environment: Local Area Network (LAN)

Methodology

1. Installing Wireshark

Download: Download the latest version of Wireshark from [<wireshark.org>](<https://www.wireshark.org/>).

Installation: Follow the installation wizard to install Wireshark on your system.

2. Capturing Network Packets

Starting a Capture: Open Wireshark and start a capture on the relevant network interface.

Network Activities: Perform various network activities such as browsing websites, file transfers, and using network services to generate traffic.

3. Applying Filters

Filter Syntax: Use Wireshark’s filter syntax to isolate specific types of traffic.

UDP Packets: udp

TCP Packets: tcp

HTTP Packets: http

-Custom Filters: Create custom filters for more detailed analysis

(e.g., ip.addr == 192.168.1.1).

4. Analyzing Packets

Packet Details: Inspect packet details including headers, payload, and metadata.

Protocol Hierarchy: Use Wireshark’s protocol hierarchy statistics to get an overview of protocol distribution.

Follow Streams: Use the “Follow TCP/UDP Stream” feature to view entire conversations.

5. Saving Captures

Saving Files\*: Save the packet captures for documentation and further analysis.

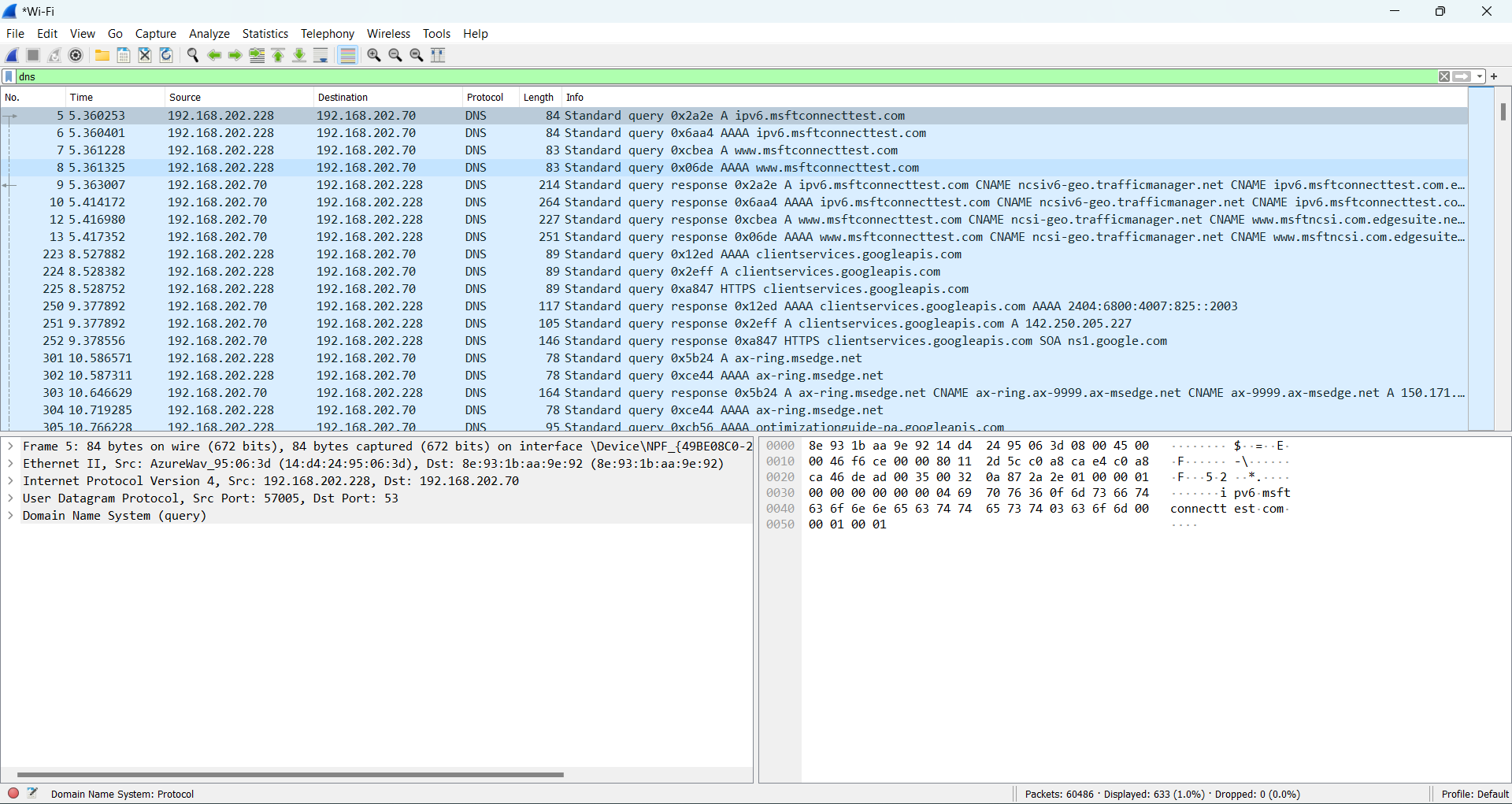
Navigate to File -> Save As.

Choose the desired location, file name, and format (e.g., .pcapng).

Screenshots and Analysis

1.DNS Packets Capture

The Domain Name System (DNS) is a fundamental component of the internet, responsible for translating human-readable domain names (like www.example.com) into machine-readable IP addresses (like 192.0.2.1). This process is crucial for routing traffic on the internet as it allows users to access websites using easy-to-remember domain names instead of numerical IP addresses.



In the captured DNS packets, we can observe the following key elements:

Query Types: The types of queries sent by clients, such as A, AAAA, MX, and CNAME records.

Response Codes: The status of the response, such as No Error (successful resolution), NXDOMAIN (non-existent domain), or SERVFAIL (server failure).

TTL (Time to Live): The duration for which the DNS response is cached by the client or intermediate DNS servers.

Authority and Additional Sections\*: Information about authoritative DNS servers and additional resource records that may be included in the response.

DNS Queries: These are requests sent by a client (such as a web browser or an application) to a DNS server to resolve a domain name to its corresponding IP address. The query typically includes the domain name and the type of record being requested (e.g., A record for IPv4 addresses).

DNS Responses: These are replies from the DNS server containing the requested information. The response includes the resolved IP address if the DNS server has the information, or it may indicate that the domain name does not exist. Responses can also include other types of records, such as CNAME (canonical name) or MX (mail exchange) records.

Analysis:

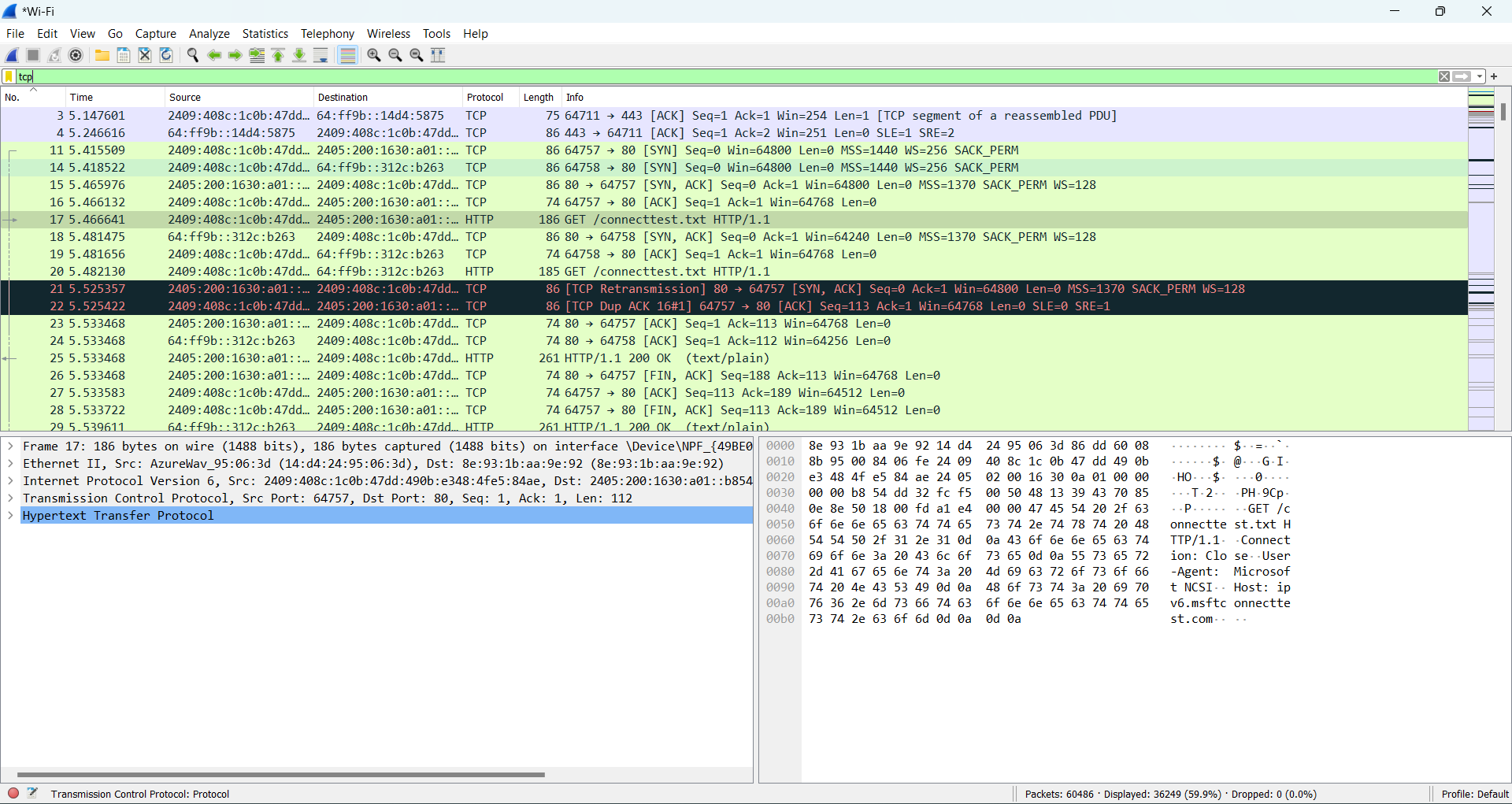
By analyzing DNS traffic, we can gain insights into the browsing habits and domain access patterns of network users.

Monitoring DNS traffic helps in detecting unusual or malicious activities, such as DNS spoofing or DDoS attacks targeting DNS infrastructure.

Understanding DNS traffic is essential for troubleshooting network issues related to domain resolution.

2. TCP Packets Capture

Transmission Control Protocol (TCP) is one of the core protocols of the Internet Protocol Suite. TCP provides reliable, ordered, and error-checked delivery of data between applications running on hosts communicating via an IP network. It is used extensively by many internet applications, including the World Wide Web, email, and file transfer.



Key Observations:

Connection Establishment: The three-way handshake process, involving SYN, SYN-ACK, and ACK packets.

Data Transfer: Continuous exchange of data packets with sequence and acknowledgment numbers.

Connection Termination: The four-step process involving FIN and ACK packets.

Error Handling: Retransmissions and adjustments in window size to handle network congestion and ensure reliable data delivery.

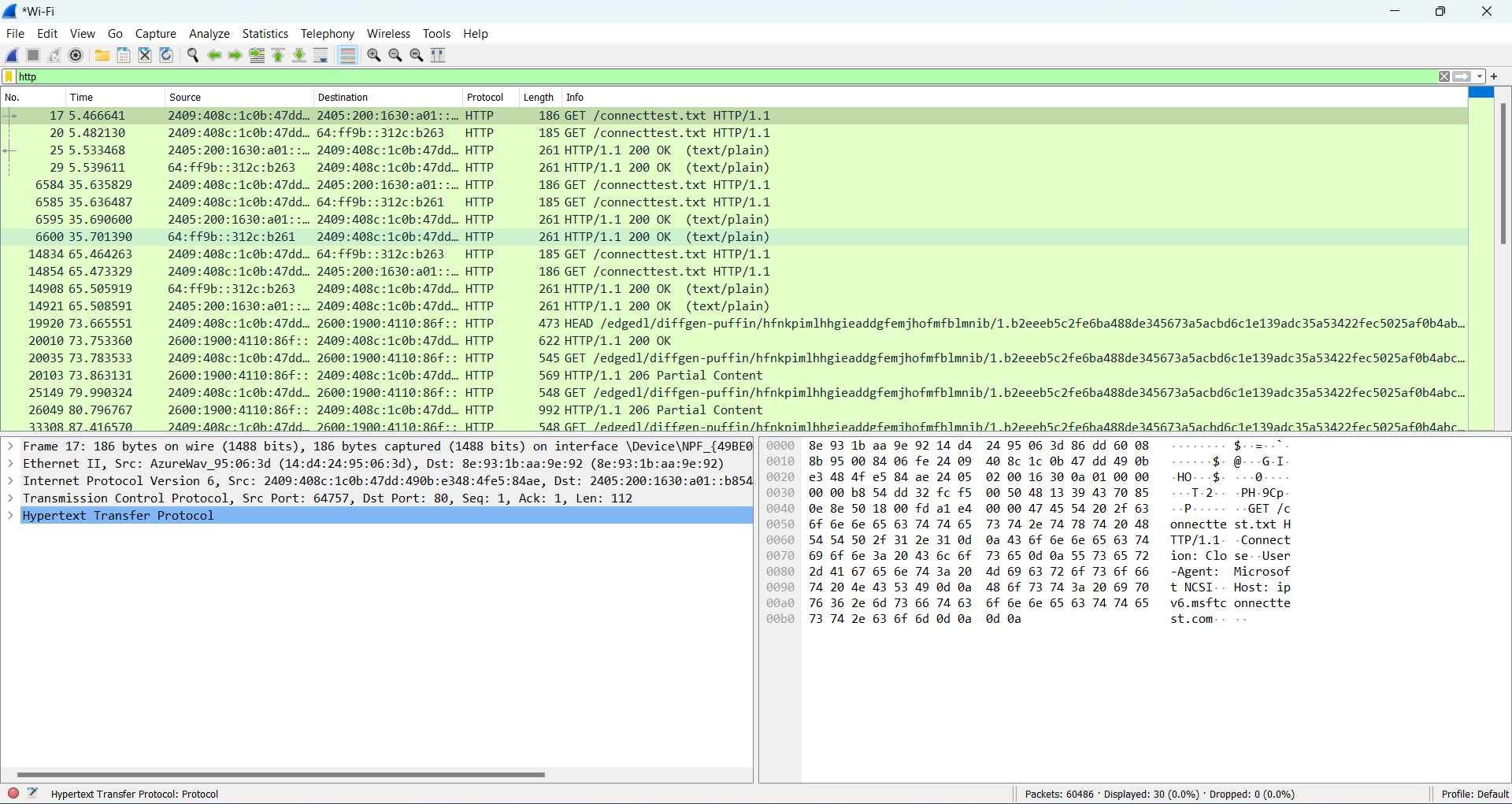
Analysis:

Analyzing TCP traffic helps in understanding the reliability and efficiency of data transfer between hosts.

It allows detection of issues such as packet loss, retransmissions, and network congestion.

Monitoring TCP connections is essential for identifying potential security threats such as TCP SYN flood attacks and ensuring the robustness of network communications.

3. HTTP Packets Capture



Hypertext Transfer Protocol (HTTP) is the foundation of any data exchange on the Web, and it is a protocol used for transmitting hypertext requests and information on the internet. HTTP is a client-server protocol, where the client sends a request to the server, and the server responds with the requested resource.

In the captured HTTP packets, we can observe the following key elements:

HTTP Requests: These are sent by the client to request resources from a server. Key components include the request method (e.g., GET, POST), URL, headers, and optionally, a message body.

HTTP Responses: These are sent by the server in response to client requests. Key components include the status line (e.g., 200 OK), headers, and the response body containing the requested resource.

Status Codes: These indicate the result of the HTTP request. Common status codes include 200 (OK), 404 (Not Found), and 500 (Internal Server Error).

Headers: These provide additional information about the request or response, such as content type, content length, and caching policies.

key Observations:

Request Methods: Different types of request methods such as GET (retrieves data), POST (submits data to be processed), PUT, DELETE, etc.

Status Code: Responses from the server with status codes indicating the success or failure of the request.

Headers: Important headers that provide metadata about the request and response.

Payload: The content of the request or response, such as HTML pages, JSON data, images, etc.

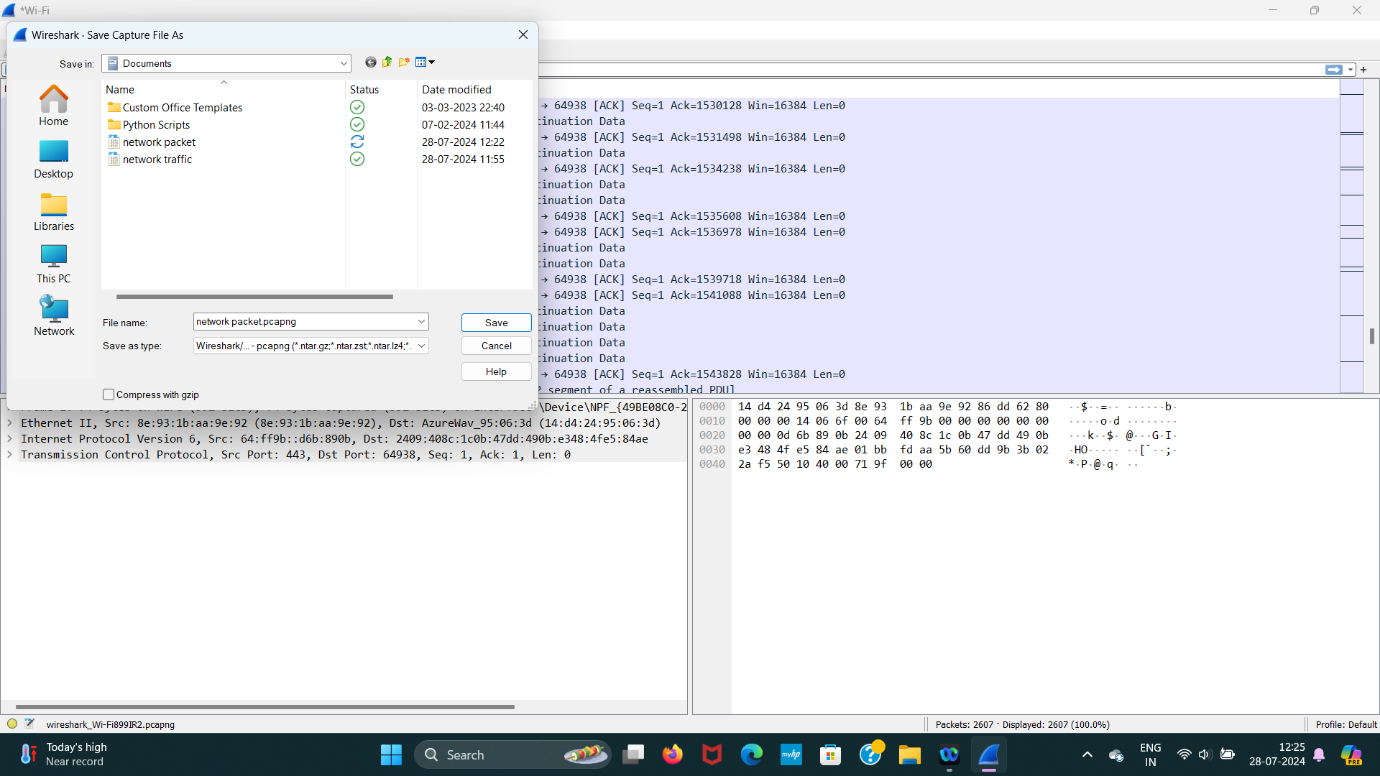
Analysis:

Analyzing :HTTP traffic helps in understanding the interactions between web clients and servers.

It allows identification of issues such as broken links (404 errors), server errors (500 errors), and performance bottlenecks.

Monitoring HTTP traffic is essential for web application security, enabling detection of attacks such as SQL injection, XSS (Cross-Site Scripting), and unauthorized access attempts.

4. Save Capture File



Description: Process of saving the captured packets for future reference.

Steps:

Navigate to File -> Save As.

Choose location, file name, and format.

Save the capture.

Conclusion

Through this project, we have demonstrated the use of Wireshark for network packet analysis. Key insights include understanding the network communication process, identifying potential issues, and enhancing network security. This analysis underlines the importance of monitoring and analyzing network traffic in cybersecurity and network management.

Recommendations

Regular Monitoring: Continuously monitor network traffic to detect anomalies early.

Security Measures: Implement security measures based on findings from packet analysis.

Training: Train network administrators on using tools like Wireshark for better network management.

Future Work

Advanced Protocol Analysis: Extend the analysis to more complex protocols like TLS/SSL.

Automated Analysis: Integrate automated tools and scripts to analyze packet captures efficiently.

Real-time Monitoring: Set up real-time monitoring systems using Wireshark and other tools.

References

Wireshark Documentation: [[Wireshark User Guide](Wireshark%20User%20Guide)](<https://www.wireshark.org/docs/wsug_html_chunked/>)

TCP/IP Illustrated, Volume 1 by W. Richard Stevens.

Practical Packet Analysis by Chris Sanders.