INTERNSHIP PROJECT REPORT

on

Network Traffic Analysis Using Wireshark

Internship Domain: Ethical Hacking

Submitted in partial fulfilment of the requirements for the completion of Ethical Hacking

From:

Name: Pavan Kumar V
Intern ID: ITID4887

Submitted to:
Inlighnx Global Pvt Ltd (Inlighnx Tech)

From: 1/10/2025 to 01/01/2026

Duration: 3 months

1. Abstract:

Network security is essential for safeguarding digital communications. Analysing network traffic is vital for comprehending how data moves within a network and for identifying potentially harmful activities. This project centres on examining network traffic through Wireshark, a popular tool for packet sniffing and analysis. In this undertaking, live network packets are collected and examined to investigate various network protocols, including TCP, UDP, HTTP, DNS, and ICMP. The project also highlights how unencrypted information can reveal confidential data and how attackers might take advantage of inadequate network setups. By scrutinizing packet behaviour and protocol statistics, it becomes possible to pinpoint potential security vulnerabilities. This project offers practical experience in packet capture, filtering, protocol analysis, and traffic scrutiny, which are key competencies for cybersecurity experts.

2. INTRODUCTION

2.1 Network Traffic Analysis

Network traffic analysis is the process of capturing, inspecting, and analyzing network packets to understand communication patterns, detect anomalies, and identify security threats. It is widely used in network troubleshooting, intrusion detection, and forensic investigations.

2.2 About Wireshark

Wireshark is a widely used **network protocol analyzer** that allows users to capture and inspect network traffic in real time. It helps in understanding how data packets flow across a network. Wireshark supports hundreds of protocols, including TCP, UDP, HTTP, DNS, and ICMP. The tool displays detailed packet information such as headers, payloads, and protocol fields. It is commonly used by network administrators, cybersecurity professionals, and ethical hackers. Wireshark provides powerful filtering options to isolate specific traffic. It is an open-source and cross-platform tool. Wireshark is widely used for troubleshooting network issues. It also plays an important role in security analysis and forensic investigations. Overall, Wireshark is an essential tool for network traffic analysis.

2.3 Importance of the Project

- Helps understand real network communication
- Used in Security Operations Centres (SOC)
- Detects suspicious or malicious activities
- Improves cybersecurity skills

3.OBJECTIVES & SCOPE

3.1 Objectives

- To capture live network traffic using Wireshark
- To analyze common network protocols
- To identify potential security risks
- To understand packet-level communication

3.2 Scope of the Project

- Analysis of TCP, UDP, HTTP, DNS, and ICMP traffic
- Identification of unencrypted data
- Study of protocol hierarchy and traffic flow
- Basic anomaly detection

This project is limited to basic traffic analysis and does not include advanced malware reverse engineering.

4.SYSTEM REQUIREMENTS

4.1 Hardware Requirements

- Laptop / Desktop Computer
- Minimum 4 GB RAM
- Network Interface Card

4.2 Software Requirements

- Operating System: Windows / Linux
- Wireshark Tool
- Internet Browser
- Npcap Driver (for Windows)

4.3 Tools Used

- Wireshark

- Command Prompt / Terminal
- Web Browser

5.METHODOLOGY

5.1 Installation of Wireshark

Wireshark was installed from the official website. During installation, Npcap was installed to enable packet capturing.

5.2 Packet Capture Process

1. Launch Wireshark
2. Select active network interface (Wi-Fi/Ethernet)
3. Start packet capture
4. Perform internet activities (browsing, ping, login)
5. Stop capture after few minutes

5.3 capturing packets

Here we start sending and analysing packet using TryHackMe website. Upon sending a connection request to the TryHackMe platform, a well-organized and secure communication process is activated between the client and the server. When we enter the website name it convert the domain name into the server IP address and server send the connection request to the client.

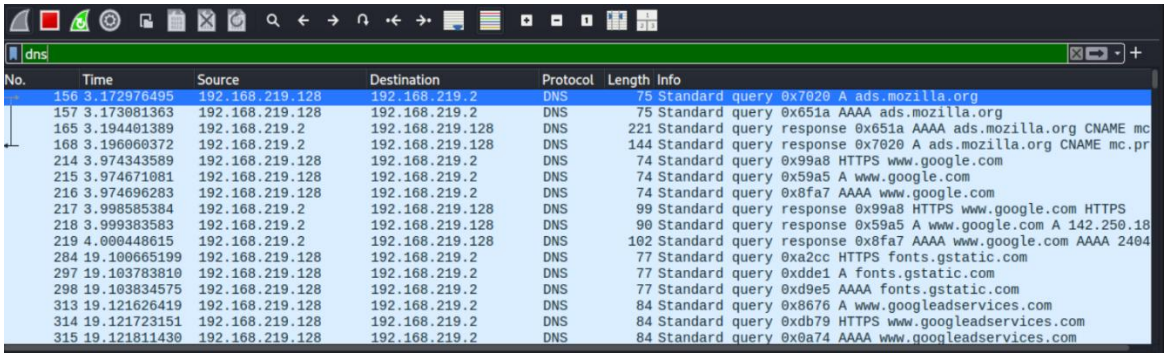

Fig 5.3.1 DNS data Packets

This process commences with the TCP three-way handshake, a crucial method employed to create a dependable connection. The handshake entails the interchange of SYN, SYN-ACK, and

ACK packets, verifying that both the client and server are prepared to send data and that the connection settings are aligned. After the TCP connection is successfully formed, ongoing data transmission occurs between the client and the server. Throughout this stage, numerous packets are exchanged, including TCP, UDP, and various protocol-specific packets. TCP packets play a key role in ensuring dependable data delivery by maintaining packet order, performing error checks, and resending packets if any are lost. UDP packets may also be present, generally utilized for swift, connectionless data transfer where minimal delay is desired.

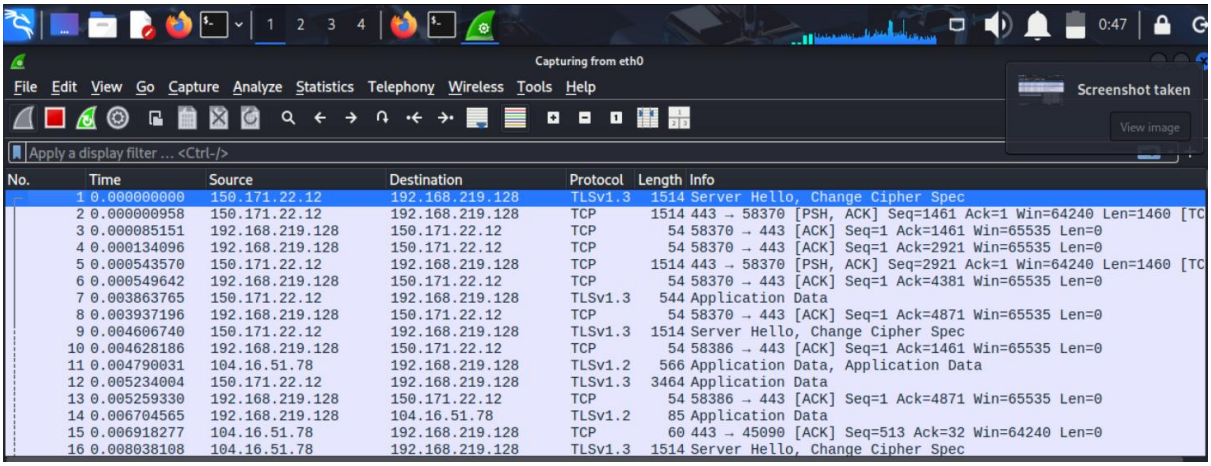

Fig 5.3.2 TCP data Packets

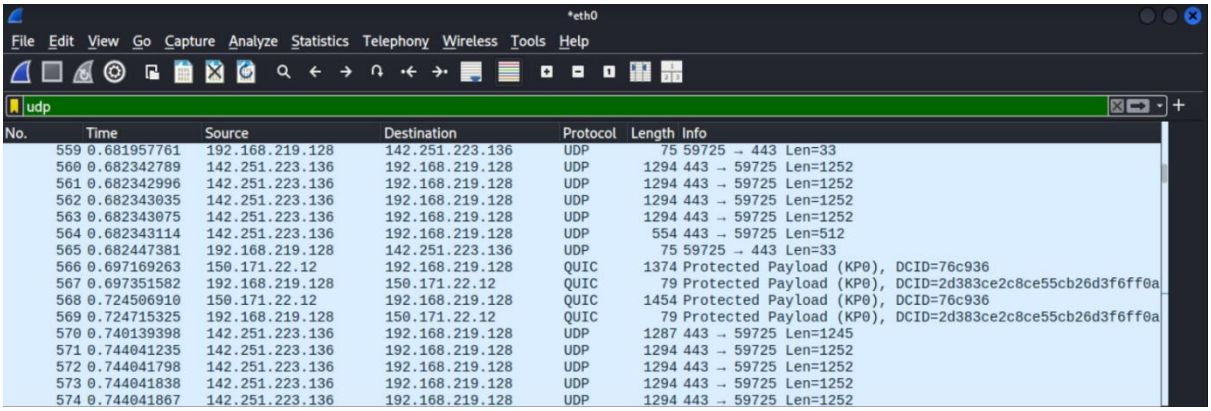

Fig 5.3.3 UDP data Packets

Here we are getting TCP and UDP packets after sending the connection request to the server with different secure encryption protocols. After sending and receiving TCP and UDP packets, the communication between the client and server continues in an organized manner to ensure successful data exchange. TCP packets maintain reliability through acknowledgments, sequencing, and retransmissions, while UDP packets enable faster, connectionless data transfer where required. These packets carry application-level data as well as control information necessary for session management. Secure protocols may then encrypt the transmitted data to

protect it from unauthorized access. Overall, this packet exchange ensures efficient, reliable, and secure network communication

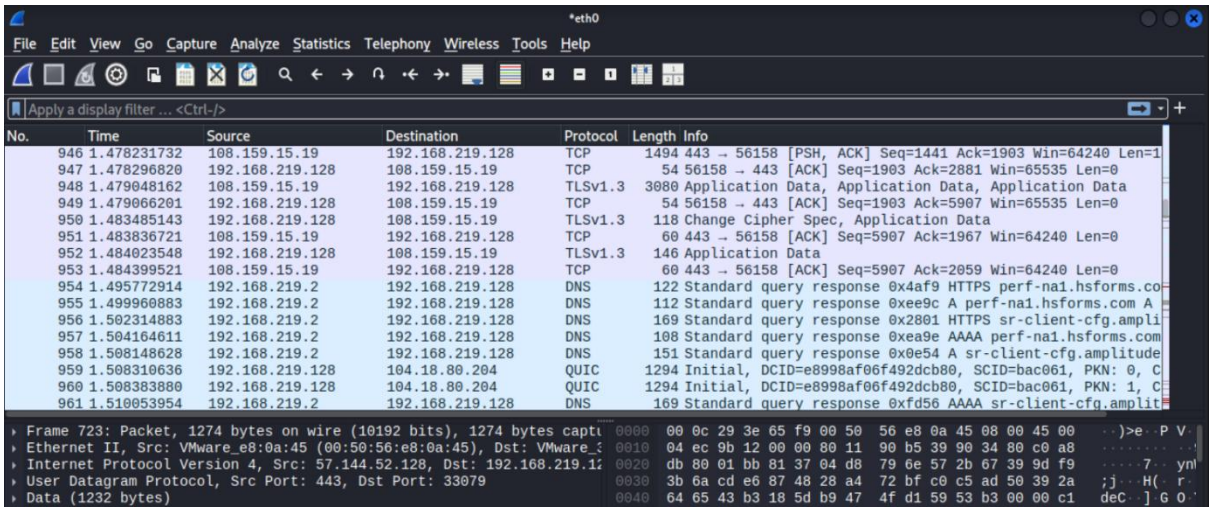

Fig 5.3.4 TLSv1.3 Secure encryption Protocal

This entire process takes place not only when the user logs in but also when they communicate with the TryHackMe platform or send text messages. Every action taken by the user creates encrypted data packets that are safely sent and acknowledged. The significance of encryption and protocol layering in contemporary network security is highlighted by packet analysis using programs like Wireshark, which aids in understanding how secure client-server communication is created, maintained, and shielded against interception. **TLS v1.3 (Transport Layer Security version 1.3)** is the **latest and most secure encryption protocol** used to protect data during communication over the internet.

It ensures that data exchanged between a client and a server remains **confidential, authentic, and tamper-proof**. TLS 1.3 improves security by removing outdated cryptographic algorithms and reducing the number of handshake steps. This results in **faster connection establishment** and **lower latency** compared to older versions like TLS 1.2. It is widely used in **HTTPS websites, online banking, cloud services, and secure applications**. Overall, TLS 1.3 provides stronger encryption, better performance, and enhanced privacy for modern network communications.

Then, I tested the http connection using the **TestPHP website**, which is an HTTP-based web application commonly used by ethical hackers for practice and learning. Since it uses the HTTP protocol, the data is transmitted in plaintext without encryption. This makes it suitable for observing request and response packets during network analysis. Ethical hackers use this website to understand web vulnerabilities and traffic behavior. The connection helps in analyzing HTTP

methods, headers, and server responses. It is widely used for educational and security testing purposes.

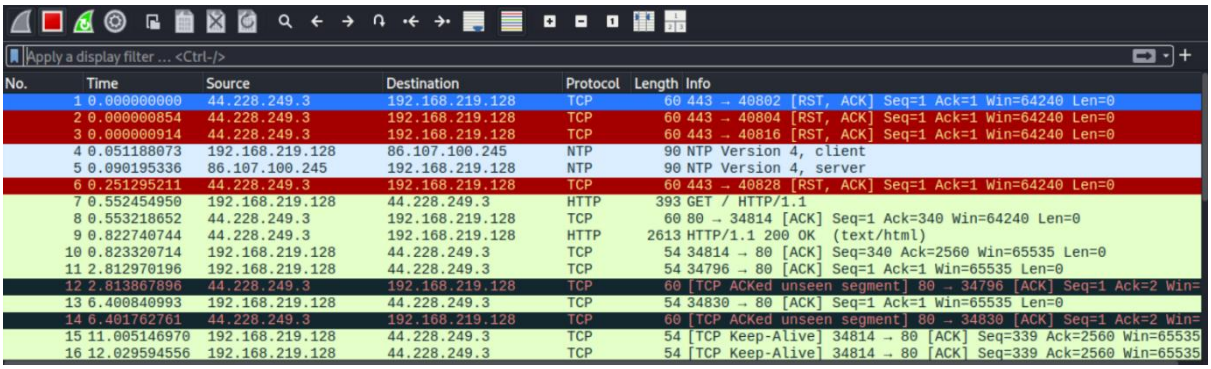

fig 5.3.5 HTTP packets

I captured HTTP packets in Wireshark during the network traffic analysis process. These packets show the communication between the client and the web server using the HTTP protocol. The captured data includes HTTP request methods such as GET and POST along with server responses. Since HTTP is not encrypted, the request headers and transmitted data are visible in plain text. This makes it easier to analyze URLs, parameters, and server responses. Observing HTTP packets helps in understanding how web communication works at the application layer.

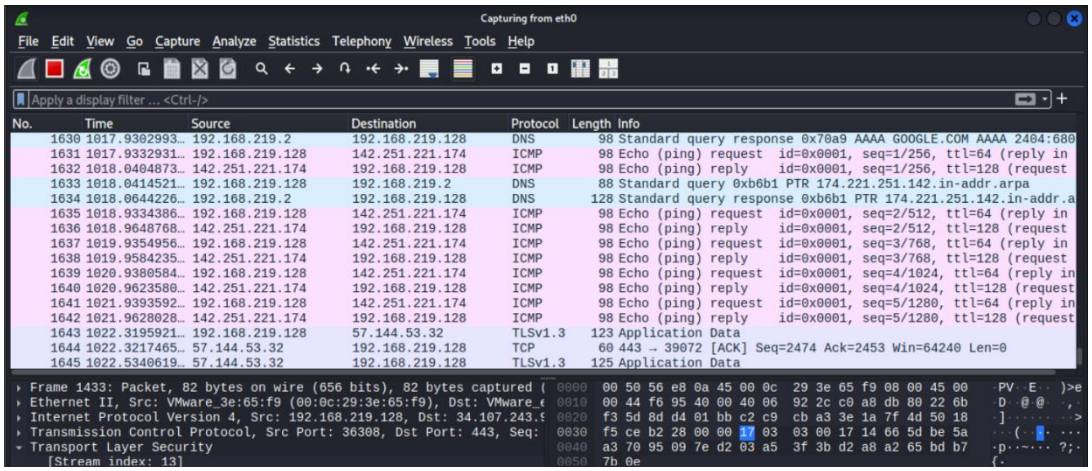

Fig 5.3.6 ICMP Packets

Using Ping to ping the google.com domain system generated an iterative process to retrieve the associated IP address, first determining the address via DNS resolution, then sending Echo Request via ICMP packets to the remote host, and receiving ICMP Echo Replies from the remote server, thereby verifying the remote server was reachable via IP address using ICMP. The subsequently captured packets in Wireshark contained the following additional details: source and destination

addresses (IP), a packet's sequence number and Time To Live (TTL) values. This capability to capture and analyse ICMP packets provides network connectivity verification, identifies latency in the network, and is an example of 'connectless communications' – ICMP does not provide a session; instead ICMP uses 'connectionless communication'. Therefore, it should be noted that the above captured packets provide examples of request-to-response communication. In general, ICMP packet analysis is a common method for diagnosing network connectivity and performance issues.

6. Conclusion

This project successfully demonstrated how network traffic can be captured and analyzed using Wireshark. It provided practical exposure to real network packets and highlighted the importance of secure communication. The project enhanced understanding of protocols and network security threats.

6.1 Learning Outcomes

- Hands-on packet analysis
- Understanding of network protocols
- Awareness of cybersecurity risks
- Practical security investigation skills

6.2 Future Scope

- Malware traffic analysis
- Integration with IDS/IPS
- Advanced forensic analysis
- SIEM-based traffic monitoring

******