Task 8 – Network Traffic Capture Using Wireshark

1. Introduction

Network traffic analysis is a critical component of cybersecurity, particularly in areas such as security monitoring, incident response, digital forensics, and network troubleshooting. By analyzing network packets, security professionals can understand how data flows across a network, detect abnormal behavior, and identify potential security threats.

Wireshark is one of the most widely used packet capture and analysis tools. It allows analysts to capture live network traffic, inspect individual packets, and apply filters to focus on specific protocols or activities. This task demonstrates the use of Wireshark to capture and analyze web-based HTTP traffic in a controlled lab environment.

---

2. Objective

The objectives of this task are:

- To capture live network traffic using Wireshark
- To identify the active network interface used by the system
- To generate and observe HTTP-based web traffic
- To apply display filters to analyze specific protocols
- To understand the security implications of unencrypted web traffic

---

3. Environment Setup

The task was performed in an isolated virtual lab environment to ensure safe and ethical analysis.

- Operating System: Kali Linux
- Platform: Oracle VirtualBox
- Tool Used: Wireshark
- Network Interface: eth0

Using a virtualized environment ensures that packet capture activities do not interfere with external or production networks.

---

4. Tool Used – Wireshark

Wireshark is an open-source network protocol analyzer that enables deep inspection of hundreds of network protocols. It is commonly used by network administrators and security analysts for:

- Network troubleshooting
- Security monitoring
- Malware and intrusion analysis
- Protocol learning and debugging

Wireshark captures packets in real time and provides detailed information such as source and destination IP addresses, protocols, headers, payloads, and timing information.

---

5. Capture Setup

The following steps were followed to set up packet capture:

1. Wireshark was launched on the Kali Linux virtual machine.
2. The active network interface (eth0) was identified and selected.
3. Packet capture was started to monitor live network traffic.

Once capture began, Wireshark continuously recorded packets flowing through the selected interface.

---

6. Traffic Generation

To ensure meaningful packet capture, web traffic was manually generated:

- The Firefox web browser was opened.
- An HTTP-based website (http://neverssl.com) was accessed.
- The webpage was refreshed multiple times to generate sufficient HTTP packets.

The use of an HTTP website was intentional to observe unencrypted traffic for analysis purposes.

---

7. Packet Analysis

After capturing traffic, packet analysis was performed using Wireshark's display filters.

7.1 Display Filter Applied

- Filter Used: http

This filter limits the displayed packets to HTTP traffic only, making analysis easier and more focused.

7.2 Observations

The following HTTP-related packets were observed:

- HTTP GET requests sent by the client
- HTTP responses returned by the web server
- Clear-text URLs and request headers

Packet details such as source IP address, destination IP address, protocol type, and request method were examined using the packet details and packet bytes panes.

---

8. Security Significance

This task highlights several important security concepts:

- Visibility: Packet capture provides full visibility into network communication.
- Risk of HTTP: Unencrypted HTTP traffic exposes sensitive information such as URLs, headers, and session data.
- Incident Detection: Traffic analysis helps detect suspicious activity, malicious connections, and policy violations.
- Skill Development: Wireshark is a core tool for SOC analysts, DFIR professionals, and network engineers.

Understanding packet-level data is essential for responding to network-based attacks and incidents.

---

9. Limitations of the Task

- Encrypted HTTPS traffic cannot be easily inspected without decryption keys.
- Only HTTP traffic was analyzed; other protocols were not covered.
- The task focuses on basic filtering rather than advanced analysis.

Despite these limitations, the task provides a strong foundation in packet capture and analysis.

---

10. Conclusion

This task successfully demonstrated how to capture live network traffic using Wireshark and analyze HTTP packets through display filtering. By observing web requests and responses, the exercise emphasized how data travels across a network and how unencrypted traffic can expose sensitive information.

The knowledge gained from this task is essential for cybersecurity professionals involved in network monitoring, incident response, and forensic investigations.