Wireshark Packet Capture Analysis Documentation

2024-01-05

Project: Simple Chat Application

Overview

The Wireshark packet capture analysis documentation provides detailed insights into the communication between the client and server components of the Simple Chat Application. The capture was performed using Wireshark on **January 5, 2024, at 15:30:00**.

Capture Details

- **Date and Time:** January 5, 2024, at 15:30:00
- **Captured By:** Rejen Thompson
- **Capture Duration:** Approximately 10 minutes

Tools Used

- Wireshark (Version: 4.0.7 (Git v4.0.7 packaged as 4.0.7-1))

Server Details

Server Script

```
import socket

def start_server():
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    server_socket.bind(('127.0.0.1', 12345))
    server_socket.listen(1)

    print("Server listening on port 12345...")

    client_socket, client_address = server_socket.accept()
    print("Connection from:", client_address)

    while True:
        data = client_socket.recv(1024).decode('utf-8')
        if not data:
            break
        print("Received:", data)

        if "malicious" in data.lower():
            response = "Malicious activity detected. Disconnecting..."
            client_socket.send(response.encode('utf-8'))
            client_socket.close()
            break
        else:
            response = f"Server received: {data}"
```

```
            client_socket.send(response.encode('utf-8'))

    client_socket.close()
    server_socket.close()

if __name__ == "__main__":
    start_server()
```

Client Details

Client Script

```
import socket

def start_client():
    client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    client_socket.connect(('127.0.0.1', 12345))

    while True:
        message = input("Enter a message (or 'exit' to quit): ")
        client_socket.send(message.encode('utf-8'))

        if message.lower() == 'exit':
            break

        response = client_socket.recv(1024).decode('utf-8')
        print("Server response:", response)

    client_socket.close()

if __name__ == "__main__":
    start_client()
```

Analysis

The Wireshark packet capture analysis focuses on the communication between the client and server. The primary objective is to verify the understanding and knowledge of Wireshark capabilities in capturing and analyzing network traffic.

1. Establishing Connection

- The communication begins with a three-way handshake (SYN, SYN-ACK, ACK) between the client and server to establish a connection.

2. Client Sending Messages

- Messages sent by the client are captured, and the server responds accordingly.

3. Malicious Activity Detection

- The server script includes a mechanism to detect and respond to messages containing the word "malicious."

Observations

- The chat application successfully establishes a TCP connection between the client and server.
- Messages are transmitted between the client and server.
- The server detects and responds to messages containing the word "malicious."

Conclusions

The packet capture analysis, conducted on January 5, 2024, at 15:30:00, by Rejen Thompson, demonstrates a solid understanding of Wireshark capabilities in capturing and analyzing network traffic. The scripts successfully handle message transmission, and the server includes a mechanism to detect and respond to potentially malicious activity.