|  |  |
| --- | --- |
|  | **Experiment No: 12** |
| **Name** | Vaibhav Sharma |
| **PRN** | 22070126125 |
| **Date of Performance** | 16th October 2024 |
| **Title** | Implement and analyze TCP and ICMP protocols using Wireshark |
| **Objective** | The purpose of this experiment is to utilize Wireshark to capture and analyse **TCP** and **ICMP** packets. The experiment aims to understand the structure and behaviour of these protocols during network communication. |
| **Setup** | * Wireshark was installed and configured to monitor the Ethernet interface.      * The focus was on capturing **TCP** and **ICMP** traffic, applying respective filters in Wireshark. |
| **Procedure** | **Step 1: Capturing TCP Packets**   1. A packet capture session was initiated on the Ethernet interface. 2. **TCP** filters were applied to narrow the capture to TCP traffic. 3. A TCP connection to a web server (port 443) was established, capturing the packets exchanged during the session.   **Step 2: Capturing ICMP Packets**   1. A new capture session was started, and the **ICMP** filter was applied. 2. **Ping** commands were issued to a remote server   (www.google.com) to generate ICMP Echo Requests and Responses. |

|  |  |
| --- | --- |
| **Analysis** | TCP Packet Analysis   * **Frame 1555** represents a **TCP segment** associated with encrypted application data over **TLSv1.2** * Source IP: 72.25.64.2 * Destination IP: 192.168.1.8 * Source Port: 443 (HTTPS) * Destination Port: 57014 (Client) * TCP Flags: PSH, ACK * Window Size: 130816 * Sequence Number: 3727216553 * Acknowledgment Number: 2348424014 * Payload Length: 1440 bytes   ICMP Packet Analysis   * **Frame 52** represents an **ICMP Echo Request** packet sent during a ping * Source IP: 2404:6800:4009:824::2004 (Google Server) * Destination IP: 2409:40e2:102d:c3d3:49e2:37cf:2c0b (Local Machine) * Protocol: ICMPv6 * Ping Request ID: 0x0001 * Sequence Number: 7 * ICMP Data: 32 bytes * The response (Frame 53) was received, confirming successful communication with a round-trip time of 39ms. |
| **Screenshots** | 1. TCP    Vaibhav Sharma |
|  | 2. ICMP    Vaibhav Sharma |
| **Observation** | The TCP packets captured were part of an encrypted session between a client and server over HTTPS. The PSH, ACK flags were used to ensure data was sent and acknowledged efficiently.  The ICMP packets were generated from ping requests, and responses confirmed connectivity and the average round-trip time (RTT) to the server was 39ms. |
| **Self-assessment**  **Q&A** | Q: What is the purpose of analyzing TCP and ICMP protocols using Wireshark?  Ans: To understand how TCP manages reliable data transmission and how ICMP handles error reporting and diagnostic messages during network communication.  Q: How does Wireshark capture TCP packets?  Ans: Wireshark captures TCP packets by monitoring communication on specific ports, such as 80 for HTTP or 443 for HTTPS, and displays sequence numbers, acknowledgments, and flags.  Q: What information can be gathered from ICMP packet analysis?  Ans: ICMP packet analysis provides details on network diagnostics, such as ping requests and replies, as well as error messages like destination unreachable or time exceeded. |
| **Conclusion** | This experiment allowed the exploration of both **TCP** and **ICMP** protocols using Wireshark. It provided insight into how TCP handles reliable transmission and how ICMP helps in diagnosing network connectivity through ping. The ability to capture and analyse the packet structure, including flags, sequence numbers, and headers, furthered the understanding of these crucial network protocols. |