Tribhuvan University

Prithivi Narayan Campus

Pokhara,Kaski

Lab-report of Computer Network

Subject code: CSC

Submitted to:
Dev Timilsina
Department of CSIT
Prithivi Narayan Campus

Submitted by:
Sabin Paudel
Roll no: 56
4th Semester

INDEX

S.N	Name of experiment	Date of submission	Remarks
1.	TO FIND THE DIFFERENCE BETWEEN HTTP AND HTTPS.	2081/03/28	
2.	TO ANALYZE THE TCP/IP MODEL USING WIRESHARK	2081/03/28	
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			

EXPERIMENT NO :1

TO FIND THE DIFFERENCE BETWEEN HTTP AND HTTPS.

OBJECTIVE:

- To understand and analyze the differences between HTTP and HTTPS by capturing and inspecting network traffic using Wireshark.

THEORY:

HTTP:

HTTP is the foundational protocol used for transmitting data on the World Wide Web. It operates at the application layer of the Internet protocol suite and is stateless, meaning each request from a client to server is independent.

- **Port**: Default port is 80.
- **Security**: Data is transmitted in plain text, making it vulnerable to interception and attacks like eavesdropping and man-in-the-middle.
- **Use Case**: Suitable for non-sensitive data transfer like browsing non-secure websites.

HTTPS:

HTTPS is an extension of HTTP and provides secure communication over a computer network. It uses SSL/TLS to encrypt the data transmitted between the client and server.

- **Port**: Default port is 443.
- **Security**: Data is encrypted, making it secure against interception and attacks. It ensures data integrity, confidentiality, and authenticity.
- **Use Case**: Used for sensitive data transfer like online banking, email, and secure transactions.

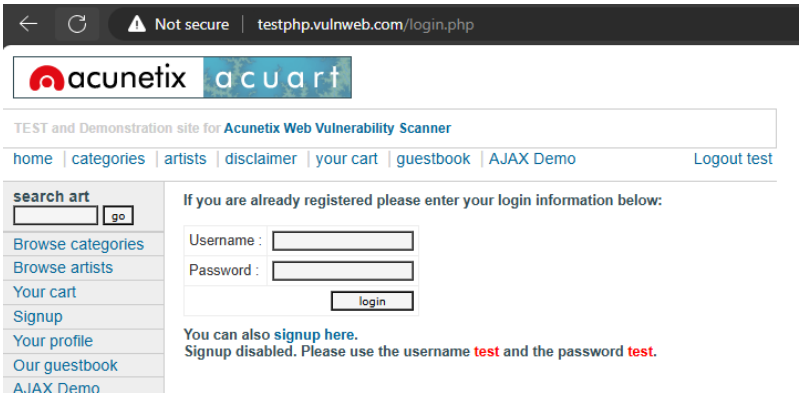
Steps in Wireshark

1. **Setup Environment**:
   - Install Wireshark from Wireshark.org.
   - Ensure you have administrator privileges to capture packets on your network interface.
2. **Capture HTTP Traffic**:
   - Open Wireshark.
   - Select the network interface used for internet connection.
   - Start capturing packets by clicking the blue shark fin button.
   - Open a web browser and navigate to a non-secure website (e.g., http://example.com).
   - Stop the capture after the page loads.
3. **Capture HTTPS Traffic**:
   - Clear the previous capture in Wireshark.
   - Start a new capture.
   - Open a web browser and navigate to a secure website (e.g., https://example.com).
   - Stop the capture after the page loads.
4. **Analyze Packets**:
   - Use the filter bar to isolate HTTP and HTTPS packets.
   - For HTTP: Use filter http.
   - For HTTPS: Use filter tls (TLS is used for HTTPS).

DEMONSTRATION:
1)Packet Capture from HTTP websites

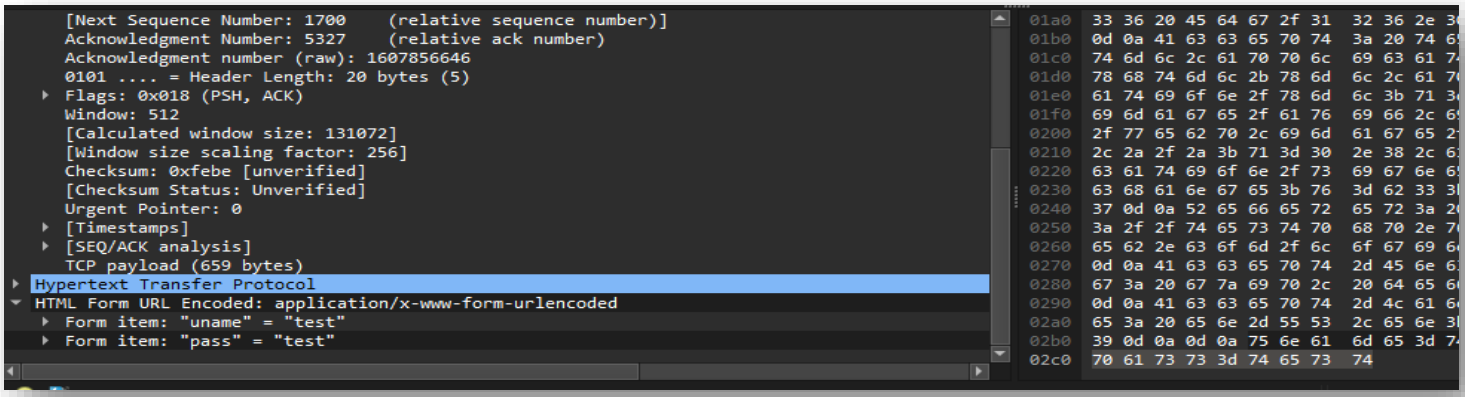
1. HTTP

For HTTP we are going to use website http://testphp.vulnweb.com/login.php as it uses http protocol to communicate with its server.



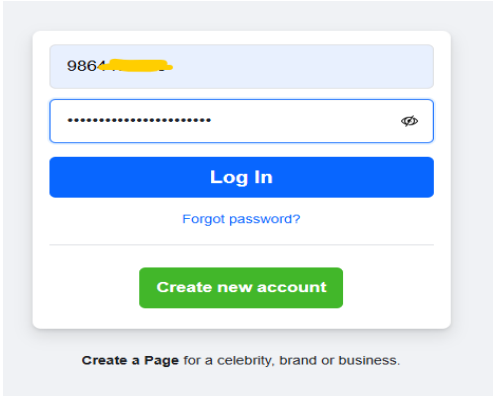
Username: test and password: Test is used.

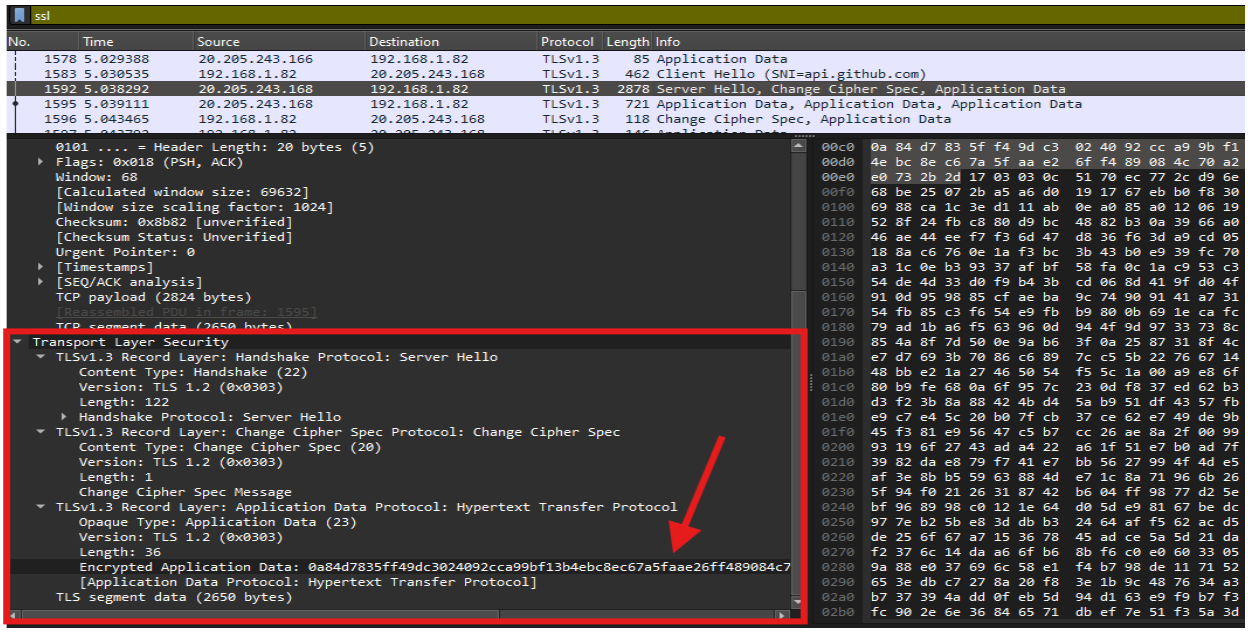
Http filter is used to capture only http packets.



2)Packet Capture from HTTPS websites

For https we are going to use https://www.facebook.com/login as it uses https protocol





DISCUSSION:

Observations in HTTP Packets

When examining HTTP packets, we notice that data is transmitted in plain text. This includes URLs, headers, and any data sent in GET and POST requests. This lack of encryption means that any sensitive information can be easily intercepted and read by attackers, posing significant security risks.

Observations in HTTPS Packets

In contrast, HTTPS packets exhibit encrypted data. The actual content of the communication, such as HTML and form data, is not readable in the captured packets. Additionally, HTTPS traffic includes a TLS handshake process, where the client and server agree on encryption methods and exchange keys to secure the session. This ensures that the data remains confidential and intact during transmission.

CONCLUSION:

The key difference between HTTP and HTTPS lies in security. While HTTP transmits data in plain text, making it susceptible to various attacks, HTTPS secures data through encryption using SSL/TLS. This makes HTTPS the preferred choice for transmitting sensitive information over the internet. The analysis using Wireshark clearly demonstrates the visible differences in packet content and structure between the two protocols, emphasizing the importance of using HTTPS for secure communication

EXPERIMENT NO :2

TO ANALYZE THE TCP/IP MODEL USING WIRESHARK.

OBJECTIVE:

- To analyze the TCP/IP model by capturing and inspecting network packets using Wireshark.

THEORY:

The TCP/IP model, also known as the Internet protocol suite, is a set of communication protocols used for the Internet and similar networks. It is commonly represented in four abstraction layers:

1. **Link Layer**: Handles the physical connection between the devices, including hardware addressing.
2. **Internet Layer**: Responsible for packet forwarding, including routing through different routers.
3. **Transport Layer**: Provides end-to-end communication services for applications.
4. **Application Layer**: Contains all protocols for specific data communication services on a process-to-process level.

Wireshark is a network protocol analyzer that captures network packets and displays detailed information about them. It allows users to analyze the structure of different network protocols and troubleshoot network issues.

Steps in Wireshark

1) Install Wireshark:

- Download and install Wireshark from www.wireshark.org.

2) Capture Packets:

- Open Wireshark.
- Select the network interface to capture traffic from (e.g., Ethernet, Wi-Fi).
- Click on the 'Start' button to begin capturing packets.

3) Generate Network Traffic:

- Perform activities that generate network traffic, such as browsing the internet, sending emails, or transferring files.

4) Stop Capture:

- Click on the 'Stop' button to end the packet capture.

5) Analyze Captured Packets:

- Use the 'Filter' bar to narrow down specific packets (e.g., tcp, udp, http, dns).
- Inspect the different layers of captured packets:
  - **Frame**: General information about the captured frame.
  - **Ethernet II**: Link layer information, including source and destination MAC addresses.
  - **Internet Protocol (IP)**: Internet layer information, such as source and destination IP addresses.
  - **Transmission Control Protocol (TCP)**: Transport layer information, including port numbers and sequence numbers.
  - **Hypertext Transfer Protocol (HTTP)**: Application layer information for web traffic.

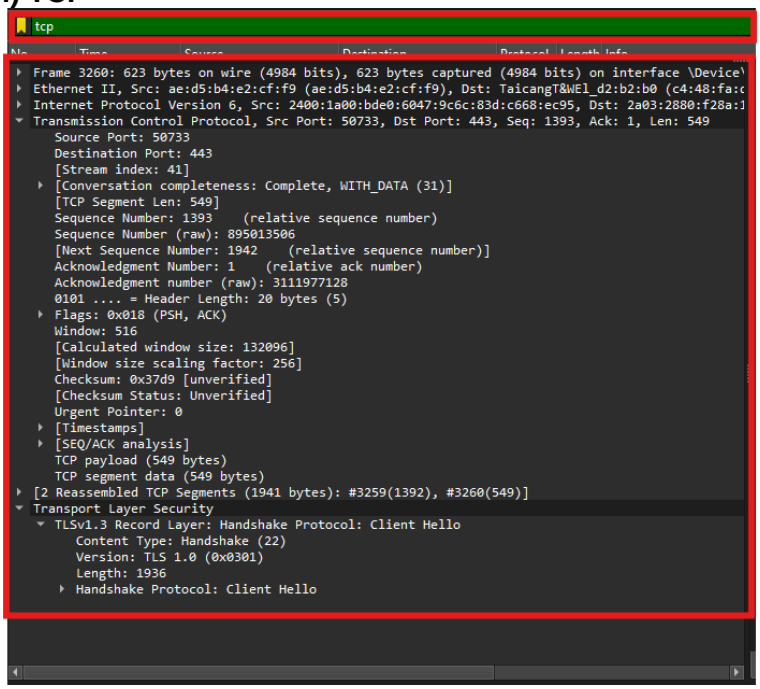
DEMONSTRATION:

1) Demonstrate the TCP/IP Model using Wireshark

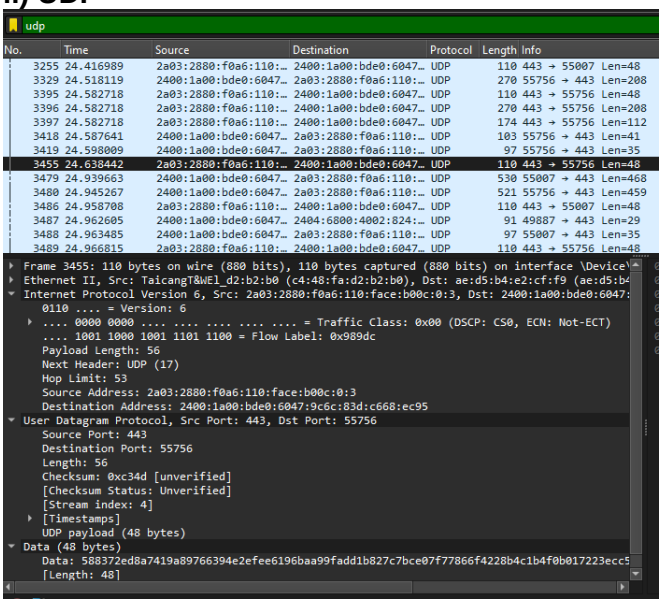
I. Transport layer (Layer 4)

Capturing packets related to TCP or UDP. To capture the TCP and UDP packets we need to filter out the packets. For this filters “tcp” and “udp” was used.

i) TCP



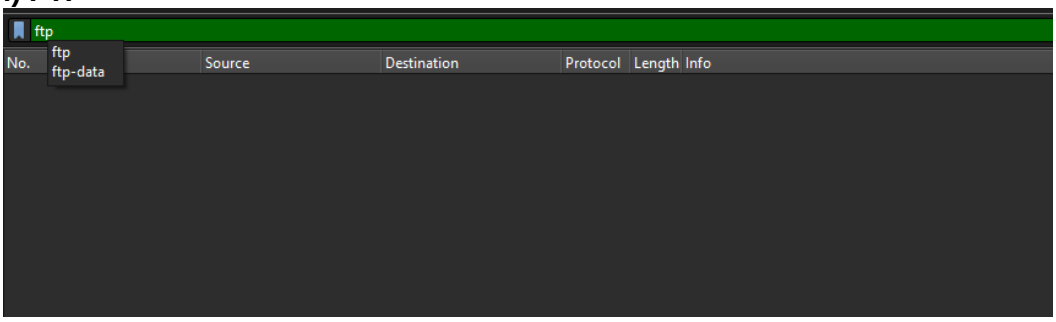
ii) UDP



II. Application layer (Layer 5-7)

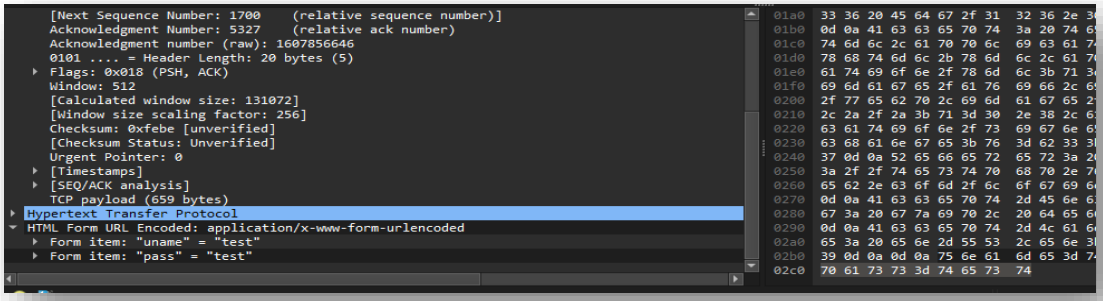
Capturing packets associated with specific application. In this layer HTTP , DNS or FTP is captured. To filter these packets filters “http”, “dns” or “ftp” is used.

i) FTP



No FTP were capture during the session as no files were transferred.

ii) HTTP



DISCUSSION:

Using Wireshark, we captured and analyzed packets from different layers of the TCP/IP model. The analysis provided insight into the following:

- **Link Layer**: Ethernet frames showed the MAC addresses and basic frame structure.
- **Internet Layer**: IP packets revealed the source and destination IP addresses, allowing us to understand how packets are routed across the network.
- **Transport Layer**: TCP and UDP packets highlighted the differences between connection-oriented (TCP) and connectionless (UDP) protocols. We observed sequence numbers, acknowledgments, and retransmissions in TCP packets, demonstrating reliable data transfer.
- **Application Layer**: HTTP packets displayed the requests and responses between web clients and servers, including headers and payload data.

CONCLUSION:

Wireshark is a powerful tool for analyzing network traffic and understanding the TCP/IP model. By capturing and examining packets at various layers, we can diagnose network issues, monitor performance, and gain insights into the underlying protocols that enable internet communication. This practical experience reinforces the theoretical concepts of the TCP/IP model and demonstrates its real-world application in network analysis and troubleshooting.