Packet Capture Analysis (HTTP Protocol)

In this task, Wireshark was used to capture and analyze HTTP traffic between a virtual machine and an external server.

Objective

The goal was to observe and document unencrypted HTTP traffic as part of understanding how data packets are transmitted and viewed over a network.

Procedure

Using Wireshark:

- Interface eth0 was selected for capturing packets.

- A filter "http" was applied to isolate HTTP traffic.

- The browser was directed to http://neverssl.com, a site that runs entirely on HTTP (not HTTPS).

- Relevant packets were captured and saved.

Observations

HTTP GET Request:

- Source IP: 10.0.2.15 (the local machine).

- Destination IP: 34.223.124.45 (neverssl.com server).

- Request URI: /onLine/

This is a typical HTTP GET request which asks the server to send a specific resource (a webpage).

HTTP 200 OK Response:

- The server responded with status code 200 OK, meaning the request was successfully received and processed.

- Response Type: text/html

This indicates a basic HTML page was sent back in response.

Additional Requests:

• The browser also requested favicon.ico and a .png file, which are standard for site icons or embedded images.

Security Insight:

• All content was visible in plain text — confirming that HTTP lacks encryption.

• This highlights the vulnerability of HTTP: sensitive data like passwords or session tokens can be intercepted and read by attackers using tools like Wireshark.

Encryption Analysis (HTTPS vs HTTP)

If your assignment includes examining encryption, here's how to present it:

Why HTTPS Matters

HTTPS encrypts data using SSL/TLS protocols. This means:

• Packets are not human-readable in Wireshark.

• Instead of GET /onLine/, you'd see Encrypted Application Data.

HTTP vs HTTPS

Feature	HTTP	HTTPS
Encryption	No	Yes (SSL/TLS)
Port Used	80	443
Packet Visibility	Full content visible	Encrypted, unreadable content
Vulnerability Level	High	Much lower