**Simon Parris**

**Wireshark Lab 1**

**HTTP Traffic Analysis Report**

**Capture File:** wireshark\_lab\_http\_traffic analysis.pcapng

**Target Website:** neverssl.com

**Tool Used:** Wireshark

**1. Overview**

This packet capture was carried out while visiting the website neverssl.com, which is intentionally served over plain HTTP rather than HTTPS. The purpose of this exercise was to examine how unencrypted web traffic behaves when transmitted over a network. The captured traffic contained request and response exchanges between the client and the server, with no encryption applied. This allowed for the full visibility of the communication, making it possible to analyze the details of the HTTP messages.

**Lab Steps:**

1. Open **Wireshark**
2. Select your active network interface (usually **Wi-Fi**)
3. Click the **blue shark fin** to start capture
4. In browser, visit [**http://neverssl.com**](http://neverssl.com) **(Test browser)**
5. Let the page fully load
6. Go back to Wireshark → click **red square** to stop capture
7. In filter bar, type **http** to view only HTTP packets
8. Look for **GET requests**, **200 OK responses**, and **Host: neverssl.com**
9. Save capture: **File → Save As → .pcapng**

**2. HTTP Requests**

The capture revealed several HTTP requests made by the browser. The primary request was a **GET** request to the root page (/) of neverssl.com. Additional GET requests were issued for supporting resources such as images, stylesheets, or scripts required to render the page. These requests included standard HTTP headers such as the **Host** header, which specified the domain neverssl.com, the **User-Agent** header to identify the browser, and the **Accept** header indicating the types of content the browser was willing to receive.

**3. HTTP Responses**

In response to these requests, the server returned both successful and cached responses. For example, the main page returned a **200 OK** response, confirming that the page had been successfully retrieved. Other responses included **304 Not Modified**, which indicated that the browser could reuse cached content instead of downloading it again. Response headers such as **Server** and **Content-Type** (e.g., text/html) provided additional information about the content and the server delivering it.

**4. Unique Hosts**

The primary host identified in the traffic was neverssl.com, which served all the requests during the capture. In some cases, websites may also rely on supporting content delivery networks (CDNs) or third-party servers, but this particular capture was simple and focused mainly on traffic to the neverssl.com domain.

**5. Observations**

A key observation from this analysis is that all the communication between the client and the server was transmitted in plaintext. Because no TLS (Transport Layer Security) was used, every part of the communication—including URLs, headers, and page content—was visible in the capture. This means that anyone monitoring the network, such as an Internet Service Provider, a malicious attacker on the same Wi-Fi, or a proxy, could read, intercept, or even alter the transmitted data.

**6. Security Implications**

The absence of encryption exposes the connection to several security risks. Most notably, it is vulnerable to **man-in-the-middle (MITM) attacks**, where an attacker could intercept and manipulate the communication. In a real-world scenario, this could result in credential theft, sensitive data exposure, or malicious content injection into web pages. For this reason, modern websites universally adopt HTTPS to secure communications, and HTTP-only sites are considered unsafe for sensitive browsing.

**Conclusion**

This capture demonstrates the weaknesses of using HTTP without encryption. The website neverssl.com exists specifically to illustrate these risks and is commonly used in network analysis labs and security training. The analysis shows how easily web traffic can be inspected when no TLS encryption is present and emphasizes the importance of enforcing HTTPS in any real-world deployment to maintain confidentiality, integrity, and security of user data.