**Assignment No: 1**

**Problem Statement:** To study the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website (banking, e-commerce etc.)

**Objectives:**  To learn modern tools for network traffic analysis

**Outcome:** Use network security services and mechanisms

**Software & Hardware Requirments:**

Operating System: Ubuntu

Packet Analyzer Tool: Wireshark

Hardware: i3 Processor, 4GB RAM, 500GB HDD

**Theory:**

**1. What is SSL, TLS, and HTTPS?**

SSL (Secure Socket Layer) and TLS (Transport Layer Security) are popular cryptographic protocols that are used to imbue web communications with integrity, security, and resilience against unauthorized tampering. PKI uses the TLS protocol to establish secure connections between clients and servers over the internet, ensuring that the information relayed is encrypted and unable to be read by an external third party.

*Note:* SSL was the predecessor of TLS, and the world began moving away from SSL once TLS was introduced in 1999, thanks to the improved security features of the latter. TLS is currently in its third iteration, and is called TLS 1.3. However, SSL continues to be used as a metonym for both protocols in general (for example, the word ‘SSL certificate’ is widely used, but SSL has been completely deprecated and no modern systems support SSL anymore).

Connections that are secured by TLS will indicate their secure status by displaying HTTPS (Hypertext Transfer Protocol Secure) in the address bar of web browsers, as opposed to just HTTP.

While TLS is primarily used to secure client-server connection, it is also used to protect emails, VoIP calls, and other connections.

**1.1. Why are they necessary?**

In theory, web connections are completely possible without TLS to secure them. However, without a security protocol in place, the communication would be rendered completely open to external access. If a browser connected to the website of an online store, and a user had to enter their credentials to log in, those credentials could easily be lifted by an observing party.

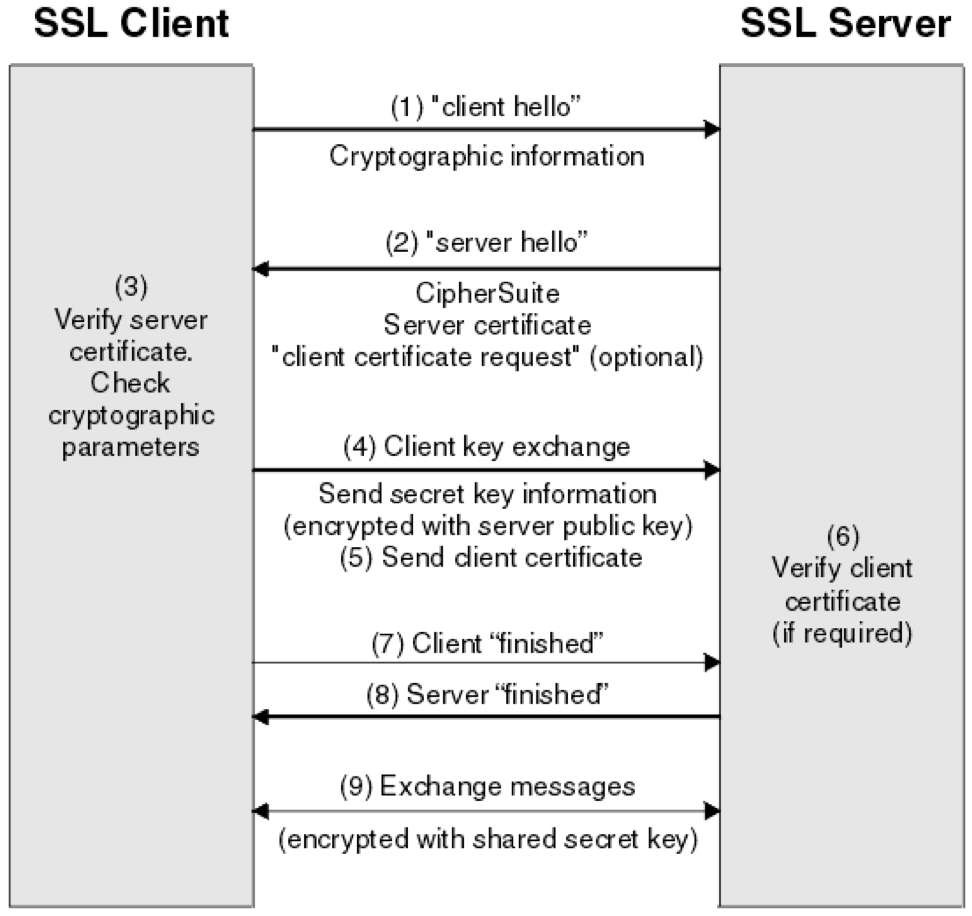
TLS, at its core, serves to provide end-to-end encryption for all data transmitted from one point to another, and uses cryptography to ensure that only the two transacting bodies are capable of reading this information. Every service in the world now mandates that connections are secure by TLS – leading browsers do not allow users to access websites without a valid TLS connection.

TLS has the following benefits:

* The contents of the connection remain encrypted, private, and fully secure – and cannot be easily deciphered by malicious actors.
* The connection is only made if it is reliable – this reliability check is a part of TLS communications, and is enforced by the exchange of a Message Authentication Code.
* The use of PKI and TLS certificates ensures that the identities of both communicating parties are verified.

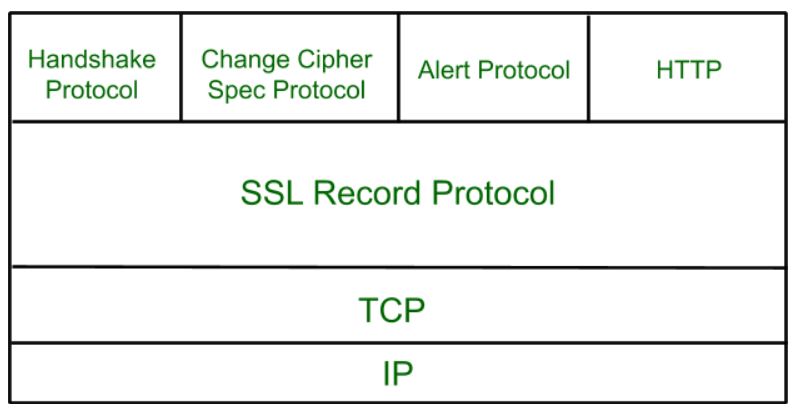
**2. SSL/TLS Handshake Process**

1. The “client hello” message: The client sends a “client hello” message that lists cryptographic information such as the TLS version and, the [cipher suites](https://www.venafi.com/blog/what-are-cipher-suites) supported by the client in an order of preference. The message also contains a string of random bytes, known as the “client random”, that is used in subsequent computations.
2. The “server hello” message: The server responds with a “server hello” message that contains the cipher suite chosen by the server from the list provided by the client, the session ID, and another string of random bytes, called the “server random”. The server also sends its TLS certificate. If the server requires a digital certificate for client authentication, the server sends a "client certificate request" that includes a list of the types of certificates supported and the Distinguished Names of acceptable Certification Authorities (CAs).
3. Server authentication: The client verifies the server's TLS certificate with the certificate authority that issued it. This confirms that the server is who it says it is, and that the client is interacting with the actual owner of the domain.
4. Premaster secret: The client sends one more random string of bytes, the “premaster secret.” The premaster secret is encrypted with the server’s public key (included in the server’s TLS certificate) and can only be decrypted with the private key by the server.
5. Client authentication: If the server has sent a “client certificate request”, the client sends its digital certificate. The server verifies the client's certificate.
6. Session keys creation: The server decrypts the premaster secret. Both client and server generate session keys from the client random, the server random, and the premaster secret.
7. Client is ready: The client sends the server a “finished” message, which is encrypted with the secret session key, indicating that the client part of the handshake is complete.
8. Server is ready: The server sends the client a “finished” message, which is encrypted with the secret session key, indicating that the server part of the handshake is complete.
9. Secure symmetric encryption achieved: For the duration of the TLS session, the server and client can now exchange messages that are symmetrically encrypted with the shared secret session key.



**Fig.** **SSL/TLS Handshake Process**

**3. Secure Socket Layer Protocols:**



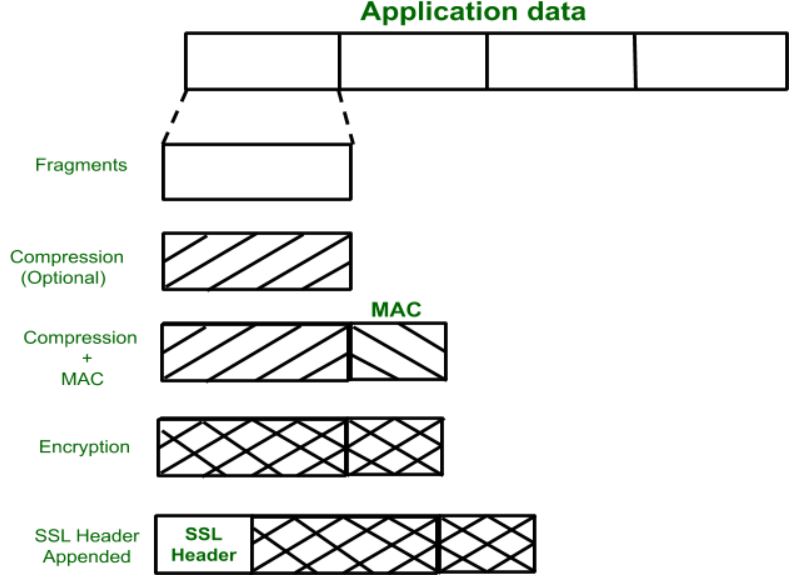
**Fig. Secure Socket Layer Protocols**

**3.1 SSL Record Protocol:**

SSL Record provides two services to SSL connection.

* Confidentiality
* Message Integrity

In the SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.



**3.2 Handshake Protocol:**

Handshake Protocol is used to establish sessions. This protocol allows the client and server to authenticate each other by sending a series of messages to each other. Handshake protocol uses four phases to complete its cycle.

* **Phase-1:** In Phase-1 both Client and Server send hello-packets to each other. In this IP session, cipher suite and protocol version are exchanged for security purposes.
* **Phase-2:** Server sends his certificate and Server-key-exchange. The server end phase-2 by sending the Server-hello-end packet.
* **Phase-3:** In this phase Client reply to the server by sending his certificate and Client-exchange-key.
* **Phase-4:** In Phase-4 Change-cipher suite occurred and after this Handshake Protocol ends.

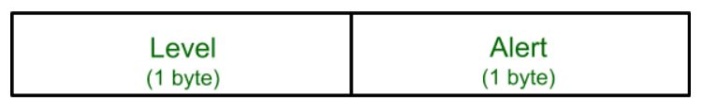
**3.3 Change-cipher Protocol:**

This protocol uses the SSL record protocol. Unless Handshake Protocol is completed, the SSL record Output will be in a pending state. After handshake protocol, the Pending state is converted into the current state.   
Change-cipher protocol consists of a single message which is 1 byte in length and can have only one value. This protocol’s purpose is to cause the pending state to be copied into the current state.



**Alert Protocol:**

This protocol is used to convey SSL-related alerts to the peer entity. Each message in this protocol contain 2 bytes.



The level is further classified into two parts:

* **Warning:**   
  This Alert has no impact on the connection between sender and receiver.
* **Fatal Error:**

This Alert breaks the connection between sender and receiver.

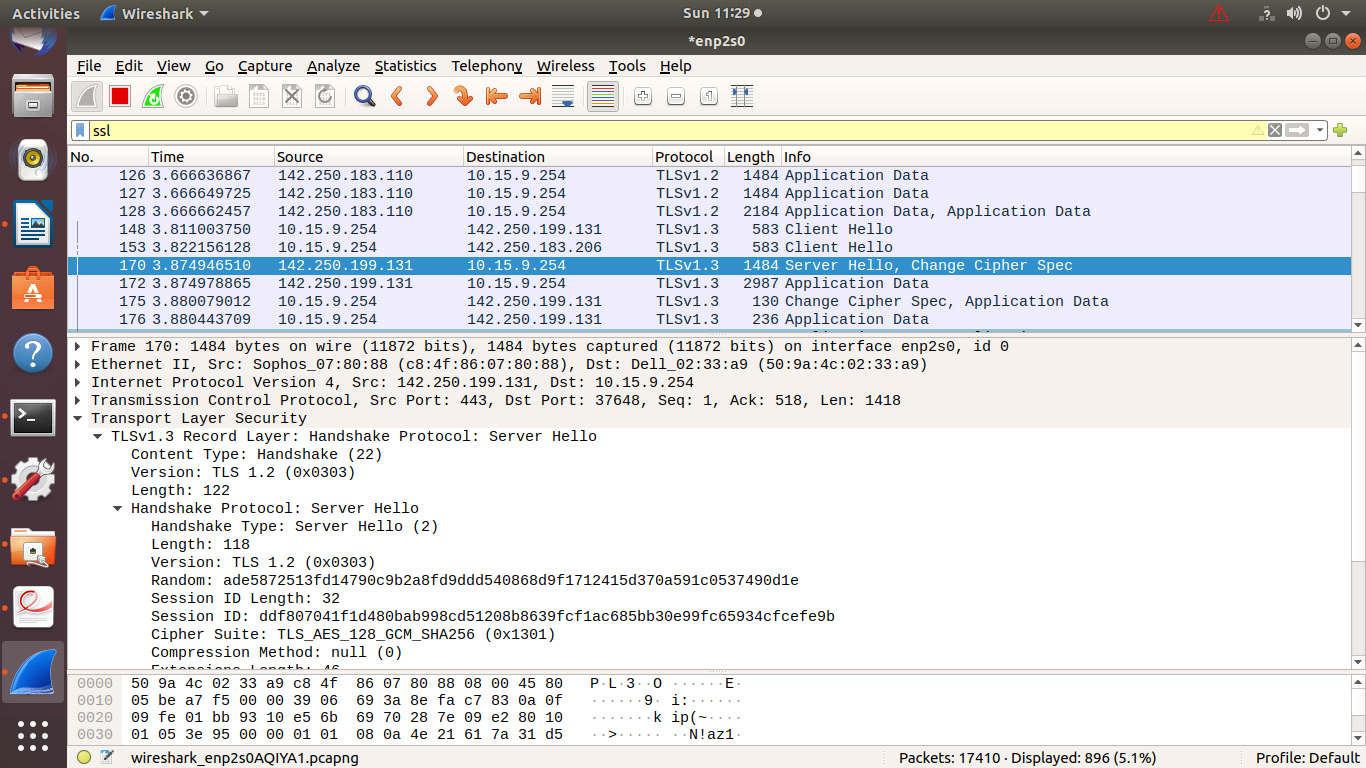
**Silent Features of Secure Socket Layer:**

* The advantage of this approach is that the service can be tailored to the specific needs of the given application.
* Secure Socket Layer was originated by Netscape.
* SSL is designed to make use of TCP to provide reliable end-to-end secure service.
* This is a two-layered protocol.

**Conclusion:**

Hence, we have successfully studied the SSL protocol by capturing the packets using Wireshark tool while visiting any SSL secured website.

**Wireshark Analysis**

**:**