Basic understanding of SSL\TLS Certificates

Overview

SSL (Secure Sockets Layer) and TLS (Transport Layer Security) are cryptographic protocols that enable secure communication over a network. SSL was the earlier protocol, and TLS is its more advanced successor that addresses some of SSL's limitations. It will act as a trust mechanism for secure communication between a web server and a web browser. When you bind a certificate, typically a public SSL certificate, it encrypts data transmitted between them, ensuring privacy and preventing man-in-the-middle attacks.

Types of Certificates

By Validation Level:

- **Self-Signed Certificates:** These certificates are created by yourself or your organization. They are relatively easy to generate but come with a major drawback: distrust. Because they aren't verified by a trusted third-party, browsers will display warnings when users try to access your website outside the organization network. This can discourage users from proceeding and harm your site's credibility.
- **CA-Signed Certificates:** These are the most common and recommended certificates for Internet websites. CAs are trusted entities that verify the identity of the website owner before issuing a certificate. This verification process ensures users connect to the intended website and not malicious imposters. There are different validation levels offered by CAs, resulting in various types of CA-signed certificates:

By Domain Coverage:

- **Single Domain Certificates:** Secure one specific domain.
- **Wildcard Certificates:** Secure a domain and all its subdomains under a single certificate.
- **Multi-Domain Certificates (MDC):** Secure multiple distinct domains (not necessarily subdomains) under one certificate.

Certificate Authorities (CAs)

Certificate Authorities (CAs) are entities that issue digital certificates, which are used to verify the identity of individuals, organizations, or servers in online communications. These certificates are crucial for establishing secure connections over the internet.

- **Public Certificate Authorities (Public CAs):** These are third-party organizations that are trusted to verify the identity of entities requesting digital certificates. Public

CAs issue certificates for public-facing websites, ensuring that users can trust the authenticity of the sites they visit.

- **Internal Certificate Authorities (Internal CAs):** These are CAs that are operated within an organization's internal network. They issue digital certificates for internal use only, such as securing internal websites, servers, and communication channels. Internal CAs are particularly useful for ensuring secure communications within an organization's network without relying on public CAs.

How it works?

Imagine you're logging in to your bank account on " https://www.hdfc.com/" to pay your bills. Here's how the SSL/TLS certificate ensures a secure connection:

Handshake Initiation: When you enter " https://www.hdfc.com/" in your browser, a behind-the-scenes communication called a handshake begins between your browser and the bank's web server.

Server Sends Certificate: The bank server sends its SSL/TLS certificate to your browser. This certificate contains information like:

- The website's domain name (www.hdfc.com)
- The public key of the server
- The digital signature of a trusted Certificate Authority (CA) that verified the bank's identity

Browser Verification: browser performs several checks on the certificate:

- It verifies that the CA that issued the certificate is trustworthy (recognized by your browser).
- It checks that the certificate's validity period is current (not expired).
- It confirms that the domain name on the certificate matches the website you're trying to access.

Secure Connection Established: If all the checks pass, your browser establishes a secure connection with the bank's server. This means any data exchanged between your browser and bank site (like your login credentials and financial information) gets encrypted using the public key from the certificate.

Encryption with Public and Private Keys:

Public Key: The bank's server included its public key in the certificate. This key is like a public lock anyone can use to scramble a message.

Private Key: The server also has a corresponding private key (like a private key that only unlocks the scrambled message). This private key is kept confidential on the server.

Encryption Process:

Scrambling with Public Key: When you enter your username and password, your browser scrambles them using the public key from the certificate. This scrambled data is like a locked box – only someone with the right key can open it.

Server Decrypts with Private Key: The scrambled data is sent to bank's server. Only the server's private key can unscramble the data using the public key's "lock". This allows the server to decrypt and securely process your login information.

Trust Store

Client Trust Store:

The client trust store, sometimes referred to as the "trusted CA store" or "root CA store," is a collection of digital certificates of trusted Certificate Authorities (CAs) maintained by the client.

When a client initiates a connection to a server (such as when accessing a website), the server presents its digital certificate, which is verified by the client against the certificates stored in the client's trust store.

Server Trust Store:

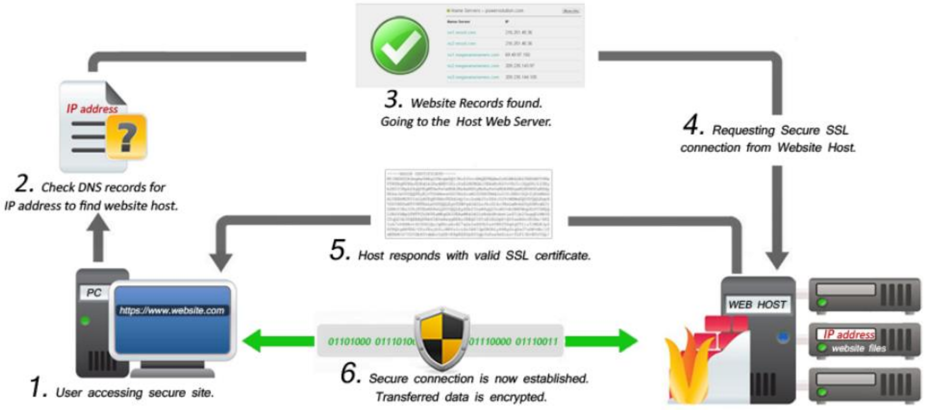
The server trust store, also known as the "server certificate store," contains the digital certificate and private key of the server.

When a client connects to the server, the server presents its digital certificate, which includes its public key and other information.

The server's digital certificate is validated by the client using its trust store. If the client's trust store contains the root CA certificate that issued the server's certificate, and if the server's certificate is valid and trusted, the client proceeds with the SSL/TLS handshake.

How does HTTPS work: SSL explained

This presumes that SSL has already been issued by SSL issuing authority.



1. User accessing secure site.

2. Check DNS records for IP address to find website host.

3. Website Records found. Going to the Host Web Server.

4. Requesting Secure SSL connection from Website Host.

5. Host responds with valid SSL certificate.

6. Secure connection is now established. Transferred data is encrypted.