COMP 470 TME4
Stephanie Petrone

Overview

SSL stands for Secure Socket Layer. It has been succeeded by TLS - Transport Layer Security. These protocols sit in between the transport layer and application layer of the TCP/IP stack. Often the terms are used interchangeably, with modern certificates (explained below) still referred to as "SSL certificates". SSL/TLS provides security for the HTTP (Hypertext Transfer Protocol) application-layer protocol and other application-layer protocols. HTTPS is simply the HTTP protocol secured by TLS. The purpose, mechanisms and implementation of TLS are described below.

SSL vs. TLS

SSL was the security protocol that preceded TLS. It has since been deprecated by the IETF and has vulnerabilities that can be exploited. A webmaster should not currently be using SSL or older versions of TLS, and should disable these in web server configurations so that downgrade attacks cannot be done. TLS 1.3 is the currently-accepted Internet security protocol that should be used by webmasters to secure web traffic.

Purpose

SSL and TLS are used to secure and encrypt web traffic. In doing so they provide the following:

(1) Confidentiality: information sent over a secure TLS session will be encrypted, so no plain text information is sent over networks. This prevents eavesdropping and man-in-the middle attacks.
(2) Authentication: certificates from Central Authorities , which are trusted third parties, verify public keys for asymmetric cryptosystems that ensure authentication of web servers for clients, and can also be used to authenticate clients.
(3) Integrity: a hashing function (typically SHA-256 in TLS 1.3) and shared secret key set up during the handshake ensures the integrity of data.

Under the Hood of TLS

TLS works using a few specific protocols together (Stallings, 2016; Australian Government, n.d):

(1) Handshake Protocol: The Handshake Protocol is used to create a session between a client and a server. A session requires the handshake, using asymmetric cryptography, to negotiate and determine the cipher suite and to generate/exchange symmetric session keys that will be used for symmetric encryption during a session.

    (a) Authentication with an SSL/TLS certificate through a **CA (Central Authority)** also occurs during the handshake. CA's are trusted third parties that bind the ownership of the CA-verified certificate to public keys used for TLS handshakes, allowing web browsers to verify that websites are trusted (DigiCert, n.d.a). The public keys verified by the CA are used in an asymmetric cryptographic key exchange (e.g. RSA, Diffie-Hellman) to set up a secure session between a client and server.

(b) Change Cipher Spec Protocol: this is the protocol relating to a one-byte message set to 1 to notify the server that encrypted communications should start with the record protocol using the parameters that were defined during the handshake. All communication going forward is encrypted using the symmetric key generated/exchanged during the handshake.

(2) Alert Protocol: this protocol conveys TLS warnings and fatal errors between peers.

(3) Heartbeat Protocol: this protocol ensures peers are still alive and prevents firewalls from closing connections due to inactivity.

(4) Record Protocol: this protocol implements confidentiality and integrity by fragmenting, compressing, adding a MAC, encrypting with symmetric keys and adding headers. All shared symmetric keys and protocols used are first established during the handshake. Symmetric encryption is much faster with much less overhead than asymmetric encryption, so asymmetric encryption is only used during the handshake to exchange keys shared by the parties to be used for encrypted communication during the session.

Steps to Implementing SSL/TLS on a server as a webmaster (AWS, n.d.; DigiCert, n.d.b):

(1) **Create a public key/private key pair**. Keep the private key very secure (e.g. with permissions set to 600).

(2) **Generate a CSR (certificate signing request)** that includes information about the organization, website and the public key.

(3) **Request an SSL certificate from a trusted central authority** based on the CSR.

(a) There are a few types of certificates, for example DV (domain validation), the most common, and others with more in-depth validation of organizational information such as EV certificates (extended validation) (DigiCert, n.d.a).

(4) **Obtain the SSL certificate from the central authority** and download the certificate to the server.

(5) **Install the keys, the CSR, and the certificate in the appropriate files/locations on the web server** with appropriate file permissions.

(6) **Configure the web server for HTTPS** (dependent on web server).

(a) Update configuration files so that the server listens on port 443 and **redirects all HTTP connections on port 80 to HTTPS on port 443.** Also, ensure that deprecated versions of SSL/TLS are not allowed to be used to prevent downgrade attacks.

The result: The website secured by TLS with a CA certificate will run at https://domain.ext rather than http://domain.ext and will show a locked symbol in the browser's search bar that shows the website is using HTTPS. Communication between the client and web server will be encrypted with authenticity and integrity provided.

References

Australian Government. (n.d.). *Implementing certificates, TLS, HTTPS and opportunistic TLS*. Implementing Certificates, TLS, HTTPS and Opportunistic TLS | Cyber.gov.au. Retrieved from https://www.cyber.gov.au/acsc/view-all-content/publications/implementing-certificates-tls-https-and-opportunistic-tls

AWS. (n.d.). *Tutorial: Configure SSL/TLS on Amazon Linux 2*. AWS. Retrieved from https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/SSL-on-amazon-linux-2.html

DigiCert. (n.d.a). *What is a Ca? certificate authorities explained*. SSL Digital Certificate Authority. Retrieved from https://www.digicert.com/blog/what-is-a-certificate-authority

DigiCert. (n.d.b). *Apache: Create CSR & install SSL certificate (openssl)*. DigiCert. Retrieved from https://www.digicert.com/kb/csr-ssl-installation/apache-openssl.htm

Stallings, W. (2016). *Cryptography and Network Security: Principles and Practice* (7th ed.). Pearson.