*TLS -*

*Transport Layer Security, or TLS, is a widely adopted security protocol designed to facilitate privacy and data security for communications over the Internet. A primary use case of TLS is encrypting the communication between web applications and servers, such as web browsers loading a website. TLS can also be used to encrypt other communications such as email, messaging, and voice over IP (VoIP).*

*It should be noted that TLS does not secure data on end systems. It simply ensures the secure delivery of data over the Internet, avoiding possible eavesdropping and/or alteration of the content.*

*TLS is normally implemented on top of TCP in order to encrypt Application Layer protocols such as HTTP, FTP, SMTP and IMAP, although it can also be implemented on UDP, DCCP and SCTP as well.*

***What is the difference between TLS and SSL?***

*TLS evolved from a previous encryption protocol called Secure Sockets Layer (SSL). TLS version 1.0 actually began development as SSL version 3.1, but the name of the protocol was changed before publication in order to indicate that it was no longer associated with Netscape. Because of this history, the terms TLS and SSL are sometimes used interchangeably.*

***What is the difference between TLS and HTTPS?***

*HTTPS is an implementation of TLS encryption on top of the HTTP protocol, which is used by all websites as well as some other web services. Any website that uses HTTPS is therefore employing TLS encryption.*

***Why should businesses and web applications use the TLS protocol?***

*TLS encryption can help protect web applications from data breaches and other attacks. Today, TLS-protected HTTPS is a standard practice for websites.*

***What does TLS do?***

*There are three main components to what the TLS protocol accomplishes: Encryption, Authentication, and Integrity.*

* *Encryption: hides the data being transferred from third parties.*
* *Authentication: ensures that the parties exchanging information are who they claim to be.*
* *Integrity: verifies that the data has not been forged or tampered with.*

*How does TLS work?*

*A TLS connection is initiated using a sequence known as the TLS handshake. When a user navigates to a website that uses TLS, the TLS handshake begins between the user's device (also known as the client device) and the web server.*

*During the TLS handshake, the user's device and the web server:*

* *Specify which version of TLS (TLS 1.0, 1.2, 1.3, etc.) they will use.*
* *Decide on which cipher suites (see below) they will use.*
* *Authenticate the identity of the server using the server's TLS certificate.*
* *Generate session keys for encrypting messages between them after the handshake is complete.*

*The TLS handshake establishes a cipher suite for each communication session. The cipher suite is a set of algorithms that specifies details such as which shared encryption keys, or session keys, will be used for that particular session. TLS is able to set the matching session keys over an unencrypted channel.*

*The handshake also handles authentication, which usually consists of the server proving its identity to the client. This is done using public keys. Public keys are encryption keys that use one-way encryption, meaning that anyone with the public key can unscramble the data encrypted with the server's private key to ensure its authenticity, but only the original sender can encrypt data with the private key. The server's public key is part of its TLS certificate. Once data is encrypted and authenticated, it is then signed with a message authentication code (MAC). The recipient can then verify the MAC to ensure the integrity of the data.*

***What is a TLS certificate****?*

*For a website or application to use TLS, it must have a TLS certificate installed on its origin server (the certificate is also known as an "SSL certificate" because of the naming confusion described above). A TLS certificate is issued by a certificate authority to the person or business that owns a domain. The certificate contains important information about who owns the domain, along with the server's public key, both of which are important for validating the server's identity.*

***How does TLS affect web application performance?***

*The latest versions of TLS hardly impact web application performance at all.*

*Because of the complex process involved in setting up a TLS connection, some load time and computational power must be expended. The client and server must communicate back and forth several times before any data is transmitted, and that eats up precious milliseconds of load times for web applications, as well as some memory for both the client and the server.*

*However, there are technologies in place that help to mitigate potential latency created by the TLS handshake. One is TLS False Start, which lets the server and client start transmitting data before the TLS handshake is complete. Another technology to speed up TLS is TLS Session Resumption, which allows clients and servers that have previously communicated to use an abbreviated handshake.*

*TLS handshakes in TLS 1.3 only require one round trip (or back-and-forth communication) instead of two, shortening the process by a few milliseconds. When the user has connected to a website before, the TLS handshake has zero round trips, speeding it up still further.*

***What is mutual TLS (mTLS)?***

*Mutual TLS, or mTLS for short, is a method for mutual authentication. mTLS ensures that the parties at each end of a network connection are who they claim to be by verifying that they both have the correct private key. The information within their respective TLS certificates provides additional verification.*

*mTLS is often used in a Zero Trust security framework\* to verify users, devices, and servers within an organization.* ***\*Zero Trust means that no user, device, or network traffic is trusted by default, an approach that helps eliminate many security vulnerabilities.***

***How mTLS works*** *–*

*In mTLS, both the client and server have a certificate, and both sides authenticate using their public/private key pair. Compared to regular TLS, there are additional steps in mTLS to verify both parties (additional steps in bold):*

* *Client connects to server.*
* *Server presents its TLS certificate.*
* *Client verifies the server's certificate.*
* *Client presents its TLS certificate.*
* *Server verifies the client's certificate.*
* *Server grants access.*
* *Client and server exchange information over encrypted TLS connection*

***Certificate authorities in mTLS -***

*The organization implementing mTLS acts as its own certificate authority. This contrasts with standard TLS, in which the certificate authority is an external organization that checks if the certificate owner legitimately owns the associated domain (learn about TLS certificate validation).*

*A "root" TLS certificate is necessary for mTLS; this enables an organization to be their own certificate authority. The certificates used by authorized clients and servers have to correspond to this root certificate. The root certificate is self-signed, meaning that the organization creates it themselves.*

***Why use mTLS?***

*mTLS helps ensure that traffic is secure and trusted in both directions between a client and server. This provides an additional layer of security for users who log in to an organization's network or applications. It also verifies connections with client devices that do not follow a login process, such as Internet of Things (IoT) devices.*

*mTLS prevents various kinds of attacks, including:*

* *On-path attacks: On-path attackers place themselves between a client and a server and intercept or modify communications between the two. When mTLS is used, on-path attackers cannot authenticate to either the client or the server.*
* *Spoofing attacks: Attackers can attempt to "spoof" (imitate) a web server to a user, or vice versa. Spoofing attacks are far more difficult when both sides have to authenticate with TLS certificates.*
* *Credential stuffing: Attackers use leaked sets of credentials from a data breach to try to log in as a legitimate user. Without a legitimately issued TLS certificate, credential stuffing attacks cannot be successful against organizations that use mTLS.*
* *Brute force attacks: Typically carried out with bots, a brute force attack is when an attacker uses rapid trial and error to guess a user's password. mTLS ensures that a password is not enough to gain access to an organization's network.*
* *Phishing attacks: The goal of a phishing attack is often to steal user credentials, then use those credentials to compromise. Even if a user falls for such an attack, the attacker still needs a TLS certificate and a corresponding private key in order to use those credentials.*
* *Malicious API requests: When used for API security, mTLS ensures that API requests come from legitimate, authenticated users only. This stops attackers from sending malicious API requests that aim to exploit a vulnerability or subvert the way the API is supposed to function.*