Owen Jones

Professor Conley

Math 116

22 March 2024

TLS: Making a Safer Internet

Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS) are security protocols designed to secure an internet connection between a client and a server. The importance of these protocols can not be dismissed. With millions of people transmitting information online every second, what stops hackers from eavesdropping or tampering with your data? These protocols enable users to trust that their data and communications are secure. This essay will give a brief overview of SSL and its history. By the end, you will understand the origins, functionality, applications, securities, and vulnerabilities of SSL and related technology.

Netscape developed the first SSL protocol in the early 1990s, but SSL 1.0 was never released to the public due to various security vulnerabilities. Netscape released the first publicly available version, SSL 2.0, in 1995, and a year later, a subsequent version 3.0 was released to patch up some of the flaws of version 2.0. However, SSL 3.0 still had several flaws. In 1999, the Internet Engineering Task Force, an independent Internet standards organization, proposed the TLS protocol Version 1.0. The IETF changed the name from SSL to TLS to denote the separation from Netscape. The protocol was first mentioned in “Request For Comments 2246,” a memo published by the IETF. While TLS 1.0 was based on SSL 3.0, RFC 2246 states that the two protocols are different enough that they can’t interoperate. Some of the differences between

the two protocols include the key derivation functions, the message authentication codes (MAC), the finished messages, more alerts, and TLS required Digital Signature Standard (DSS)/Diffie-Helman (DH) support. Various versions of TLS have been released since. The current version, TLS 1.3, was released in 2018. While TLS improved upon its predecessor, SSL was so widespread that it took over a decade to transition to TSL completely. In 2011, the IETF announced that SSL 2.0 was no longer secure and should be abandoned. A few years later, in 2015, the IETF made a similar announcement about version 3.0. Nowadays, TLS is almost exclusively used and is compatible with all internet browsers.

We must define a few terms before discussing the SSL/TLS protocols. First, the transport layer is the fourth layer in the Open Systems Interconnection (OSI) model. It is responsible for delivering messages between computers (hosts). It is an end-to-end layer, i.e., it doesn't depend on any third party to operate. Second, a cipher suite is a package of cryptographic algorithms used together in authentication, key exchange, and encryption. Each TLS cipher suite has four components: the key exchange algorithm, authentication algorithm, encryption algorithm, and message authentication code (MAC) algorithm. A MAC is similar to a digital signature but for symmetric ciphers. Not every version of TLS supports identical cipher suites. Newer versions of TLS add support for more cryptographically secure and terminate support for less secure cipher suites. While TLS 1.3 is the newest, fastest, and most secure TLS protocol, many have yet to switch from version 1.2, so entities must be compatible with various cipher suites. Third, a TLS certificate is a type of digital certificate. TLS certificates use the widely accepted International Telecommunication Union X.509 standard as its certificate framework. Like all digital certificates, TLS certificates are used to establish the identity of a client or server. They can either be signed by a certifying authority or self-signed. TLS certificates contain information

about the owner of the certificate, version serial number, length of validity of the certificate, information about the public key, etc. X.509 certificates use RSA, Elliptic Curve Cryptography (ECC), and the Digital Signature Algorithm (DSA) for public key generation.

Now that you know a little about their history, how exactly do SSL and TLS protocols work? Symmetric key cryptosystems are computationally faster than public key cryptosystems, so it is far more efficient for servers to have some method for securely establishing secret keys. The goal of the TLS protocol is to enable a client to verify a server's authenticity and establish a secret key through a key exchange method for future communication. This process is what is known as the TLS handshake. The handshake varies slightly from version to version, but the TLS 1.2 handshake can be broken down into four main steps: the client hello, the server hello, the key exchange, and the finished message.

The first step is for the client/browser to initiate the handshake with the client hello. The client hello is a message to the server that states which TLS versions and cipher suites the client can utilize and the Client Random (a 28-byte random number and 4-byte timestamp).

Next, the server checks if the Client uses a valid version of TLS, saves the Client Random, and picks a preferred cipher suite for key exchange. Ephemeral Elliptic-curve Diffie–Hellman is typically used. Subsequently, the server sends a server hello message to the client. The message contains the Server Random (another 28-byte random number and 4-byte timestamp), the chosen cipher suite for key exchange, the server’s TLS certificate, and the key exchange algorithm parameters with the server’s digital signature attached.

Upon receipt of the server hello message, the client uses the certificate to authenticate the server's identity and the server’s public key to verify the digital signature attached to the list of

parameters. After the server's identity has been authenticated, the client and server use the chosen key exchange method and list of parameters to generate a shared 48-byte pre-master secret. Combining the pre-master secret with the Client and Server Random, the client and server generate the shared 48-byte master secret. To add additional security, the master secret is not used directly for encryption. Instead, the client and server use a pseudo-random number generator to create session keys for encryption.

Now that the master key is created, the client sends a Change CipherSpec message to the server, saying all future messages will be encrypted. The server responds with a similar message, the Finished message, and an encrypted list of all previous messages. The client verifies the server's Finished message using the encrypted list of messages. Now, both entities can securely transmit data using symmetric encryption.

Hypertext transfer protocol secure (HTTPS) is the primary application of TLS. It is the encrypted version of HTTP and uses the TLS protocol to encrypt all data between a browser and a web server. HTTPS aims to enable entities to securely connect with authenticated servers and prevent eavesdroppers from viewing or tampering with their data. HTTPS is supported by all major browsers, and web servers that use HTTPS are heavily favored over servers that don't. For example, Chrome warns users when connecting to servers using HTTP that their data is insecure. Moreover, people are more hesitant to connect to unencrypted websites. Using HTTPS can significantly influence a site's search engine optimization (SEO) ranking. Using HTTP instead of HTTPS reduces a website's visibility to users, thus limiting the amount of traffic a server receives. All in all, HTTPS is a direct upgrade to HTTP because of TLS.

Another critical application of TLS is in email. When an entity sends an email, the client checks whether the email server is secure. Like with data encryption over the internet, the client initiates a TLS handshake with the server, obtaining the server's certificate and establishing a secret key for email encryption. There are two methods in which the client and server attempt to create a secure connection. In the Opportunistic TLS method, the client will attempt to create a secure connection by running the STARTTLS command, but if the command fails, the email will be sent unencrypted. In the Forced TLS method, the email will be discarded if the STARTTLS command fails.

We've discussed how TLS prevents eavesdropping and tampering through the handshake, how it's widely implemented across every major browser, and how users feel safer about inputting personal or financial details on websites that use TLS. However, with every piece of technology comes its vulnerabilities. TLS has a high level of latency relative to alternative cryptographic encryption protocols, which negatively affects consumer experience. In addition, versions 1.0-1.2 of TLS are still weak to some Man in the Middle attacks. These versions have also been found to be weak to other forms of cyber attacks, such as POODLE, DROWN, and SLOTH. Still, version 1.3 has its issues. It is the newest version, and not every platform supports TLS 1.3, so while it fixes many of the problems of its predecessors, it suffers by requiring backward compatibility with older versions.

Despite its vulnerabilities, TLS remains a fixture of modern internet encryption, protecting data transmission across the digital landscape. Through its robust cipher suites and handshake protocol, TLS is a wellspring of trust for its users, providing authenticity, integrity, and confidentiality. Despite this, the world of digital security is constantly changing. As we

discussed previously, individuals are continually developing new ways to break internet encryptions. Surprisingly, only 62.1% of websites have implemented support for TLS 1.3. Even relatively new protocols like version 1.2 are subject to cyber-attacks. This is why it is so important to be on the cutting edge of cryptographic technology. Still, TLS is an incredible technology that has helped the internet become safer.

Works Cited

Admin. “Unveiling SSL: Building a Secure Web Connection - Global Sign.” *GlobalSign*, 16 Aug. 2023, www.globalsign.com/en/ssl-information-center/what-is-ssl.

Bellore, Arthur. “The TLS Handshake Explained.” *Auth0*, 7 Mar. 2023, auth0.com/blog/the-tls-handshake-explained/.

Chen, Lily. “The Cornerstone of Cybersecurity – Cryptographic Standards and a 50-Year Evolution.” *NIST*, Lily Chen, 26 May 2022, www.nist.gov/blogs/cybersecurity-insights/cornerstone-cybersecurity-cryptographic-standards-and-50-year-evolution.

Contributor, TechTarget, et al. “What Is Secure Sockets Layer?” *Security*, TechTarget, 20 July 2021, www.techtarget.com/searchsecurity/definition/Secure-Sockets-Layer-SSL.

Contributor, TechTarget. “What Is PKI (Public Key Infrastructure).” *Security*, TechTarget, 9 Apr. 2021, www.techtarget.com/searchsecurity/definition/PKI.

Cryptographic Security Protocols: TLS, www.ibm.com/docs/en/ibm-mq/9.2?topic=mechanisms-cryptographic-security-protocols-tls. Accessed 21 Mar. 2024.

Editor, CSRC Content. “Secure Sockets Layer (SSL) - Glossary: CSRC.” *CSRC Content Editor*, csrc.nist.gov/glossary/term/secure_sockets_layer. Accessed 21 Mar. 2024.

Fatima. “Understanding Cipher Suites: Algorithms, Security, & More.” *RedSwitches*, 31 Jan. 2024, www.redswitches.com/blog/cipher-suites/#:~:text=The%20Basic%20Elements%20of%20a%20SSL%2FTSL%20Cipher%20Suite,-A%20TLS%20cipher&text=The%20cipher%20is%20used%20to,in%2Dthe%2Dmiddle%20attacks.

GfG. “Blockchain - Elliptic Curve Cryptography.” *GeeksforGeeks*, GeeksforGeeks, 17 Nov. 2022, www.geeksforgeeks.org/blockchain-elliptic-curve-cryptography/.

GfG. “Client-Server Model.” *GeeksforGeeks*, GeeksforGeeks, 2 Dec. 2022, www.geeksforgeeks.org/client-server-model/.

GfG. “Secure Socket Layer (SSL).” *GeeksforGeeks*, GeeksforGeeks, 12 Mar. 2024, www.geeksforgeeks.org/secure-socket-layer-ssl/.

Heath Kath View Profile. “What Is SSL, TLS, and HTTPS?” *What’s the Difference Between SSL, TLS, and HTTPS? | GoAnywhere MFT*, www.goanywhere.com/blog/what-is-ssl-tls-and-https. Accessed 21 Mar. 2024.

HyperCube. “What Type of Encryption Does SSL Use?” *HyperCube*, 27 Feb. 2023, hypercube.co.nz/what-type-of-encryption-does-ssl-use/.

HyperCube. “What Type of Encryption Does SSL Use?” *HyperCube*, 27 Feb. 2023, hypercube.co.nz/what-type-of-encryption-does-ssl-use/.

“An Introduction to Cipher Suites.” *Keyfactor*, 12 Sept. 2023, www.keyfactor.com/blog/cipher-suites-explained/.

Kaspersky. “What Is an SSL Certificate – Definition and Explanation.” *Usa.Kaspersky.Com*, 19 Mar. 2024, usa.kaspersky.com/resource-center/definitions/what-is-a-ssl-certificate.

An Overview of X.509 Certificates, www.ibm.com/support/pages/system/files/inline-files/An_Overview_of_x.509_certificates.pdf. Accessed 22 Mar. 2024.

An Overview of X.509 Certificates, www.ibm.com/support/pages/system/files/inline-files/An_Overview_of_x.509_certificates.pdf. Accessed 22 Mar. 2024.

Shacklett, Mary E., and Peter Loshin. “What Is a Digital Certificate?” *Security*, TechTarget, 23 Sept. 2021, www.techtarget.com/searchsecurity/definition/digital-certificate.

SSL Encryption, www.ibm.com/docs/en/cics-ts/5.6?topic=protocols-ssl-encryption. Accessed 21 Mar. 2024.

Team, SSL Support. “What Is SSL/TLS: An in-Depth Guide.” *SSL.Com*, 8 Dec. 2023, www.ssl.com/article/what-is-ssl-tls-an-in-depth-guide/.

“Transport Layer Security.” *Wikipedia*, Wikimedia Foundation, 17 Mar. 2024, en.wikipedia.org/wiki/Transport_Layer_Security#SSL_1.0,_2.0,_and_3.0.

Trappe, Wade, and Lawrence C. Washington. *Introduction to Cryptography: With Coding Theory*. Pearson Education, 2006.

What Happens in a TLS Handshake? | SSL Handshake | Cloudflare, www.cloudflare.com/learning/ssl/what-happens-in-a-tls-handshake/. Accessed 22 Mar. 2024.

“What Is SSL, TLS & HTTPS?” *DigiCert*, www.digicert.com/what-is-ssl-tls-and-https. Accessed 21 Mar. 2024.

“What Is TLS & How Does It Work?” *Internet Society*, 22 Dec. 2023, www.internetsociety.org/deploy360/tls/basics/.

What Is Transport Layer Security? | TLS Protocol | Cloudflare, www.cloudflare.com/learning/ssl/transport-layer-security-tls/. Accessed 22 Mar. 2024.