SearchSecurity.com E-GUIDE

Protect Highly Sensitive Data with SSL and TLS

Volumes of traffic flow freely across the internet everyday, and unless it's protected, it is in the open for anyone to access. E-commerce demands an increasing exchange of highly sensitive information, from credit card numbers to financial data, safely across the public network. Secure Sockets Layer (SSL) and Transport Layer Security (TLS) secure Internet traffic in an encrypted tunnel, ensuring that it is seen only when it arrives at its destination, using digital certificates to guarantee that a Web site is what it purports to be.

This E-Guide explains the differences between SSL and TLS and how they work, and offer some important points to keep in mind when implementing them in your organization.

TechTarget
The IT Media ROI Experts

Transit Safety

by Eric Cole

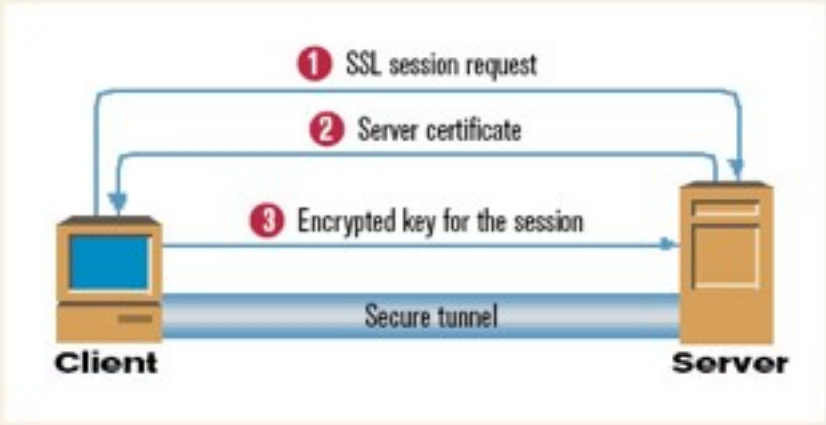
SSL-encrypted tunnels protect sensitive data traveling the Information Superhighway.

It's no accident the Internet has been called the Information Superhighway. Huge volumes of traffic flow freely and, unless protected, in the open. E-commerce demands an increasing exchange of highly sensitive information, from credit card numbers to financial data, safely across the public network.

Secure Sockets Layer (SSL) and Transport Layer Security (TLS) secure Internet traffic in an encrypted tunnel, ensuring that it is seen only when it arrives at its destination, using digital certificates to guarantee that a Web site is what it purports to be.

Most Internet users have performed some type of online transaction involving SSL or TLS. The familiar closed pad-lock icon that appears in the Web browser window indicates that SSL or TLS is being used to secure the connection. In this article, we'll explain the differences between SSL and TLS and how they work, and offer some important points to keep in mind when implementing them in your organization.

How SSL works



(1) Client sends a request to the server for a secure SSL/TLS session.

(2) Server sends its certificate from a recognized certificate authority, such as VeriSign or Entrust, to the client for authentication along with its public key.

(3) Client receives the server's certificate, verifies it, and creates a one-time session key using the server's public key, and sends it to the server. Server decrypts the session key using its private key and establishes a secure tunnel.

Why TLS?

SSL and TLS both use cryptography to provide authentication and privacy to Internet communications. TLS was designed to replace SSL, and identifies itself in the protocol version field as SSL 3.1. There are a handful of minor differences.

So, why create a new protocol? Because SSL, created by Netscape about a decade ago, is a closed proprietary protocol. The community cannot make changes or validate its security. The Internet Engineering Task Force (IETF) created TLS, an open version of the protocol, so everyone would be free to use and comment on it.

In practice, it does not matter which you select. But since more organizations are migrating to TLS, it will give you a wider range of support.

Nevertheless, though very similar, SSL and TLS are not interoperable. This means that if your server is set up to utilize TLS, it isn't downward compatible with clients only using SSL. Newer browsers and other Web applications support both SSL and TLS, so this is generally not much of an issue.

How SSL and TLS Work

At a high level, it's simple: A key is established between the sending and receiving computers, the information is encrypted with the key, and the encrypted information is transmitted (see "How SSL Works"). However, there are important details to understand.

First, the encryption is done by the application, not the operating system. The application programmer doesn't have to implement the protocol, but must specify a secure socket when establishing a connection. A socket is simply a special type of file descriptor. Instead of specifying the name of the file to be opened, the IP address and port of the destination computer are specified. The operating system packages this data into packets and sends them to the appropriate spot. Low-level work, like calculating checksums and tracking sequence numbers, is done by the operating system.

SSL and TLS protocols work in three basic steps:

1. Negotiation occurs between the client and server on the use of TLS or the version of SSL (2.0 or 3.0). This step also decides the cipher that is to be used for the rest of the protocol exchange. There are a number of public and symmetric key encryption algorithms that can be used.

2. After ciphers have been negotiated, the server is authenticated and a symmetric key is created to be used throughout the rest of the communication. This is all done using public key algorithms and X.509 certificates. This certificate is issued by a certificate authority (CA), a trusted third party that verifies the identity of the server.

   This one-sided authentication is all that is required; users must know they are talking to the proper server, not an impostor—such as a bogus bank site used in a phishing scam. The user then provides his user name and password, or multifactor authentication.

3. The symmetric key is sent to the server using public key encryption. The public key for the server is included in the certificate validated by the CA. After the symmetric keys have been established and exchanged, communications are encrypted using symmetric key algorithms instead of the public key one used before. This is done simply because symmetric key algorithms are faster and computationally easier to use. All client-server traffic is now encrypted using this key until the connection is dropped or the key expires. This provides a secure tunnel of communication.

Security Considerations

SSL 2.0, the first publicly released version, has a number of security flaws, so SSL 3.0 and TLS are the clear choices. Current browsers, including Inter-net Explorer, Firefox and Netscape, allow users to enable support for SSL 2.0, SSL 3.0 and TLS. IE 7.0, now in beta, goes a step further by requiring either SSL 3.0 or TLS.

The primary consideration is picking a strong cipher. Though a number of both public and symmetric ciphers can be used during an SSL/TLS connection, TripleDES and, even better, the Advanced Encryption Standard (AES) are the most secure and strongly recommended. DES as a cipher and MD5 as a hash function for authentication are no longer considered secure.

DES is vulnerable to brute-force attacks because it's limited to a 56-bit key, which can be cracked with a little time and some hefty computing power. It's not secure against well-funded attackers or someone with access to a cluster of computers. TripleDES applies DES three times, with three different keys.

AES, a new block cipher standard, supplants DES. It uses a 128-bit block and supports 128, 192 and 256-bit keys, making it a much more secure alternative. As a variable-length algorithm, as opposed to DES/TripleDES's fixed length, its key size can be increased to meet the challenge of faster computers' applied brute force. AES is typically used in TLS implementations.

As for hashing functions, both MD5 and SHA-1 have come under attack recently—especially MD5, which is all but obsolete. While both are based on the older MD4 protocol, SHA-1 was developed later and avoids some of MD5's vulnerabilities. It's still heavily used even though its security is debated. The National Institute for Standards and Technology is expected to hold a competition, much like the one that established AES, to create stronger cryptographic hash functions.

Among public key ciphers, the venerable RSA still stands as the best choice. The only requirement to ensure security is picking a large enough key, so any brute-force attack is bound to fail. RSA has been the most closely reviewed public key encryption algorithm and still stands strong. At this point, the chances of finding a flaw appear slim.

SSL and TLS are secure if they are implemented correctly and robust crypto algorithms are used. If the user is checking digital certificates, then a man-in-the-middle attack—spoofing the legitimate server—will be detected and defeated. However, a careless user can still be fooled by failing to check messages indicating that a certificate is suspect.

Versatile Protocol

There are many applications of SSL besides just securing connections to Web servers. SSL/TLS can be used by any application designer to create secure connections to a server. The protocol can be used just as easily on local networks, and theoretically on any type of network connection that is reliable. For example, SSL and TLS are often used on intranets and private networks to protect from insider threats. SSL can also be used as a wrapper for unsecured protocols like SMTP and FTP.

Since SSL and TLS treat all data as binaries, any type of data can be sent securely, such as HTML, Word documents and images.

However, anticipate some extra work on the programming side for internal implementations. Beyond the socket call described earlier to enable an application to use SSL, there is a whole suite of system/library calls that are used to establish a secure connection to a server. All of this work must be done by the application programmer on both the client and the server.

Above all, SSL makes e-commerce possible. It assures the customer that the Web site is genuine, and that he can do business on the Web with peace of mind.

Resources from thawte

Securing your Online Data Transfer with SSL
This white paper provides an introduction to SSL security covering the basics of how it operates and how to deploy appropriate SSL certificates.

Securing your Apache Web Server with a *thawte* Digital Certificate
Read this white paper and learn more about securing your Apache Web Server with thawte digital certificates.

Extended Validation (EV) SSL Certificates
This white paper details the benefits of extended validation (EV) SSL certificates and how they can help your company.

Securing your Microsoft IIS Web Server with a *thawte* Digital Certificate
In this guide you will find out how to test, purchase, install and use a *thawte* Digital Certificate on your Microsoft Internet Information Services (MS IIS) web server.

The *thawte* Starter PKI Program
Read this white paper and learn about the advantages and benefits of the *thawte* Starter PKI Program.

About thawte

Thawte is a leading global Certification Authority which offers a complete range of digital certificate products adds the elements of trust, integrity and privacy to all forms of information in transit over the internet, ensuring protection and peace of mind. Thawte digital certificates interoperate smoothly with the most common web servers and browsers, so you can rest assured that you purchase of a thawte Digital Certificate will give your customers confidence in the integrity of your systems.

www.thawte.com