A Network Security for E-Commerce

Abdalhakim A.M. Damegi  
Republic Of Libya  
University Of ALJABIL AI – GARBY  
kimo_acqoua@yahoo.com

Abstract

Computer networks have a layered structure and we need to protect these layers against attacks that may compromise their security. The Transport layer uses TLS as the security protocol while the IP layer uses IPSec as the security protocol. Transport Layer Security (TLS) is a connection-oriented protocol that provides a secure channel between a client and a server at the transport layer of the network. It has been developed by the Internet Engineering Task Force (IETF) as the standard protocol for providing security services in the context of E-commerce over the Internet. Thus TLS enabled web servers form the portals through which E-commerce client server interaction takes place. This paper provides an overview of the design and workings of the TLS protocol and how it enables network security for e-commerce.

Keywords: Network security, E-Commerce

I. INTRODUCTION

Internet commerce requires secure communications. To order goods, a customer typically sends credit card details. The customer might have to supply confidential personal data. Internet users would like to know that such information is safe from eavesdropping or alteration. Many Web browsers protect transmissions using the protocol Secure Sockets Layer (SSL). The client and server machines exchange nonces and compute session keys from them. Version 3.0 of SSL has been designed to correct a flaw of previous versions, the cipher-suite rollback attack, whereby an intruder could get the parties to adopt a weak crypto system. The latest version of the protocol is called TLS (Transport Layer Security); it closely resembles SSL 3.0.x.

History of SSL and TLS

Fortunately, engineers were thinking about these security issues from the Web’s beginning. Netscape communication began considering Web security while developing its very first Web browser. To address the concerns of the previous section, Netscape designed the secure Sockets Layer Protocol.

Figure 1 shows the evolution of SSL in the context of general Web development. The timeline begins in November 1993, with the release of Mosaic 1.0 by the National Center for Supercomputing Applications (NCSA), was the first popular Web browser. Only eight months later, Netscape Communication completed the design for SSL version 1.0 five months after that, Netscape shipped the first product with support for SSL version 2.0 - Netscape Navigator.
TLS, but check the sidebars for more information. Support for SSL is now built in to nearly all browsers and web servers. For users of Netscape Navigator or Microsoft's Internet Explorer, SSL operates nearly transparently. Observant users might notice the "http" prefix for an SSL-secured URL, or they may see a small icon that each browser displays when SSL is in use. For the most part, however, SSL simply works, safely providing confidentiality, authentication, and message integrity to its Web users.

Today's popular web servers also include support for SSL. It's usually as simple to enable SSL in the server. As we'll see, though, to support secure web browsing, a web server must do more than simply enable the SSL protocol. The server must also obtain a public key certificate from an organization that Web browser trust. For users on the public Internet, those organization are generally public certificate authorities. Popular certificate authorities include AT&T certificate server, CTE Cyber Trust, Key Witness International, Microsoft, Thawte Consulting, and VeriSign.

Security Requirements of an E-commerce Information Exchange

We assume that the function of a system that is the target of an attack is to provide information. In general, there is a flow of data from a source to a destination over a communication channel. The task of the security system is to restrict access to this information to only those parties (persons or processes) that are authorized to have access according to the security policy in use. In the case of an automation system which is remotely connected to the Internet, the information flow is from a control application that manages sensors and actuators via communication lines of the public Internet and the network of the automation system.

The normal information flow and several categories of attacks that target it are shown in Figure(2) explained below.

This clarification of these attacks

1. **Interruption**: An asset of the system gets destroyed or becomes unavailable. This attack targets the source or the communication channel and prevents information from reaching its intended target. Attacks in this category attempt to perform a kind of denial-of-service (DOS).
2. **Interception**: An unauthorized party gets access to the information by eavesdropping into the communication channel (e.g. wiretapping).
3. **Modification**: The information is not only intercepted, but modified by an unauthorized party while in transit from the source to the destination. By tampering with the information, it is actively altered (e.g. modifying message content).
4. **Fabrication**: An attacker inserts counterfeit objects into the system without having the sender doing anything. When a previously intercepted object is inserted, this process is called replaying. When the attacker pretends to be the legitimate source and inserts his desired information, the attack is called masquerading (e.g. replay an authentication message, add records to a file).

The four classes of attacks listed above violate different security properties of the computer system. A security property describes a desired feature of a system with regards to a certain type of attack. A common classification following is listed follows:

**Confidentiality**
This property covers the protection of transmitted data against its release to unauthorized parties. In addition to the protection of the content itself, the information flow should also be resistant against traffic analysis. Traffic analysis is used to gather other information than the transmitted values themselves from the data (e.g. timing data, frequency of messages).

**Authentication**
Authentication is concerned with making sure that the information is authentic. A system implementing the authentication property assures the recipient that the data is from the source that it claims to be. The system must make sure that no third party can masquerade successfully as another source.

**Non-repudiation**
This property describes the feature that prevents either sender or receiver from denying a transmitted message. When a message has been transferred, the sender actually been sent.

**Availability**
Availability characterizes a system whose resources are always ready to be used. Whenever information needs to be transmitted, the communication channel is available and the receiver can cope with the incoming data. This property makes sure that attacks cannot prevent resources from being used for their intended purpose.

**Integrity**
Integrity protects transmitted information against modifications. This property assures that a single
message reaches the receiver as it has left the sender, but integrity also extends to a stream of messages. It means that no messages are lost, duplicated or reordered and it makes sure that messages cannot be replayed. As destruction is also covered under this property, all data must arrive at the receiver. Integrity is not only important as a security property, but also as a property for network protocols. Message integrity must also be ensured in case of random faults, not only in case of malicious modifications.

**TLS Protocol Data Unit (PDU)**
The TLS protocol actually consists of two groups of sub protocols:
1. The Record Protocol
2. The TLS Control Protocols

**The Record Protocol**
The TLS Record Protocol is a layered protocol. At each layer, messages may include fields for length, description, and content. The Record Protocol takes messages to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, and transmits the result. Received data is decrypted, verified, decompressed, and reassembled, then delivered to higher level clients.

The Record Protocol provides a standard packaging for TLS messages where as the control protocols of TLS consist of the following three protocols:

<table>
<thead>
<tr>
<th>Handshake Protocol</th>
<th>Change Cipher Spec Protocol</th>
<th>Alert Protocol</th>
<th>HTTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS Record Protocol</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>IP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Handshake Protocol**
The Handshake protocol is responsible for the cipher suite negotiation, the initial key exchange, and the authentication of the two peers. This is fully controlled by the application layer, thus your program has to set up the required parameters. Available functions to control the handshake protocol include

**Change Cipher Spec Protocol**
The change cipher spec protocol exists to signal transitions in ciphering strategies. The protocol consists of a single message, which is encrypted and compressed under the current (not the pending) connection state. The message consists of a single byte of value 1

**Alert Protocol**
One of the content types supported by the TLS Record layer is the alert type. Alert messages convey the severity of the message and a description of the alert. Alert messages with a level of fatal result in the immediate termination of the connection. In this case, other connections corresponding to the session may continue, but the session identifier must be invalidated, preventing the failed session from being used to establish new connections. Like other messages, alert messages are encrypted and compressed, as specified by the current connection state.

The goals of TLS Protocol, in order of their priority, are:
1. Cryptographic security: TLS should be used to establish a secure connection between two parties.
2. Interoperability: Independent programmers should be able to develop applications utilizing TLS that will then be able to successfully exchange cryptographic parameters without knowledge of one another's code.
3. Extensibility: TLS seeks to provide a framework into which new public key and bulk encryption methods can be incorporated as necessary. This will also accomplish two sub goals: to prevent the need to create a new protocol (and risking the introduction of possible new weaknesses) and to avoid the need to implement an entire new security library.
4. Relative efficiency: Cryptographic operations tend to be highly CPU intensive, particularly public key operations. For this reason, the TLS protocol has incorporated an optional session caching scheme to reduce the number of connections that need to be established from scratch. Additionally, care has been taken to reduce network activity.

**TLS Protocol Operation**
A TLS client and server negotiate a stateful connection by using a handshaking procedure. During this handshake, the client and server agree on various parameters used to establish the connection's security.

1. The handshake begins when a client connects to a TLS-enabled server requesting a secure connection, and presents a list of supported ciphers and hash functions.
2. From this list, the server picks the strongest cipher and hash function that it also supports and notifies the client of the decision.
3. The server sends back its identification in the form of a digital certificate. The certificate usually contains the server name, the trusted certificate authority (CA), and the server's public encryption key. The client may contact the server that issued the certificate (the trusted CA as above) and confirm that the certificate is authentic before proceeding.
4. In order to generate the session keys used for the secure connection, the client encrypts a random
number (RN) with the server's public key (PbK), and sends the result to the server. Only the server can decrypt it (with its private key (PvK)): this is the one fact that makes the keys hidden from third parties, since only the server and the client have access to this data. The client knows PbK and RN, and the server knows PvK and (after decryption of the client's message) RN. A third party may only know PbK, unless PvK has been compromised.

5. From the random number, both parties generate key material for encryption and decryption. This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the key material until the connection closes.

Performance of TLS
The impact of TLS on web server performance has been extensively studied in the literature. As an enabling technology for E-commerce, it is highly desirable that TLS signature should be minimal on Ecommerce servers. However, due to the nature of the cryptographic algorithms that need to be executed in a TLS session, there is degradation in the TLS-enabled web server’s performance. Good studies of TLS Performance have been undertaken. A great deal of performance improvement can be achieved using
1. caching of server certificate by the client
2. optimizing the TLS handshake.

II. CONCLUSION
In this paper I provided an overview of the historical development for TLS, and some security requirements for the exchange of information across the Internet, also analyzed the TLS protocol and how it works and some of it performances and goals, which is designed for which this TLS Protocol

Biography
Abdalhakim A.M Damegi was born in Yefren Libya on 12a of January 1980. He graduated from University of Aljabil Algarby in 2002. He master degree from University of Brawijaya Malang-Indonesia, His major is Communication and Information System.