Research Article Impact Factor: 0.621 ISSN: 2319-507X
Priyanka Tikekar, IJPRET, 2014; Volume 2 (9): 500-509 IJPRET

INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

AN OVERVIEW OF SSL ARCHITECTURE

MS PRIYANKA C. TIKEKAR

Bhartiya Mahavidyalaya, Amravati, Ms, India.

Accepted Date: 27/02/2014 ; Published Date: 01/05/2014

Abstract: SSL is one of the most common protocols used for secure communication over the internet. In 1995, Netscape Communications Corporation, the then-dominant browser vendor, responded by introducing a security package called SSL (Secure Sockets Layer). Secure socket layer (SSL) is the most popular protocol used in the Internet for facilitating secure communications through authentication, encryption, and decryption. Although the use of SSL provides adequate security, the performance degradation has been drastic compared to non-secured data retrieval. This paper is based on overall layers of the secure socket layer architecture. The Secure Sockets Layer (SSL) provides security for web based applications. It uses TCP to provide end–to-end secure services. SSL is not a single protocol but rather two layers of protocols. It can be seen that one layer makes use of TCP directly. This layer is known as the SSL Record Protocol and it provides basic security services to various higher layer protocols. An independent protocol that makes use of the record protocol is the Hypertext Markup Language (HTTP) protocol. Another three higher level protocols that also make use of. this layer are part of the SSL stack. They are used in the management of SSL exchanges

Corresponding Author: MS PRIYANKA C. TIKEKAR

Access Online On:

www.ijpret.com

How to Cite This Article:

Priyanka Tikekar, IJPRET, 2014; Volume 2 (9): 500-509

PAPER-QR CODE

INTRODUCTION

SSL is designed to make use of TCP to provide a reliable end-to-end secure service. Secure Sockets Layer (SSL) is designed to provide end point authentication and communication encryption over the internet. The most recent version of SSL is Transport Layer Security. [15] Secure socket layer (SSL) is the most popular protocol used in the Internet for facilitating secure communications through authentication, encryption, and decryption. SSL is not a single protocol but rather two layers of protocols. The Secure Sockets Layer (SSL) protocol was developed by Netscape Communications to provide application-independent secure communication over the Internet for protocols such as the Hypertext Transfer Protocol (HTTP). The SSL protocol is intended to provide a practical, application-layer, widely applicable connection oriented mechanism for Internet client/server communications security. [5]

The SSL Record Protocol provides basic security services to various higher layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher layer protocols are defined as part of SSL: the Handshake Protocol, may be multiple secure connections. In theory, there may also be multiple simultaneous sessions between parties, but this feature is not used in practice. There are a number of states associated with each session. Once a session is established, there is a current operating state for both read and write (i.e., receive and send). In addition, during the Handshake Protocol, pending read and write states are created. Upon successful conclusion of the Handshake Protocol, the pending states become the current states. A session state is defined by the following parameters [14].

• Session identifier: An arbitrary byte sequence chosen by the server to identify an active or resemble session state.

• Peer certificate: An X509.v3 certificate of the peer. This element of the state may be null.

• Compression method: The algorithm used to compress data prior to encryption.

• Cipher spec: Specifies the bulk data encryption algorithm (such as null,AES, etc.) and a hash algorithm (such as MD5 or SHA-1) used for MAC calculation. It also defines cryptographic attributes such as the hash_size.

• Master secret: 48-byte secret shared between the client and server.

• Is resumable: A flag indicating whether the session can be used to initiate new connections. A connection state is defined by the following parameters.

• Server and client random: Byte sequences that are chosen by the server and client for each connection.

• Server write MAC secret: The secret key used in MAC operations on data sent by the server.

• Client write MAC secret: The secret key used in MAC operations on data sent by the client.

• Server write key: The secret encryption key for data encrypted by the server and decrypted by the client.

• Client write key: The symmetric encryption key for data encrypted by the client and decrypted by the server.

• Initialization vectors: When a block cipher in CBC mode is used, an initialization vector is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter, the final cipher text block from each record is preserved for use as the IV with the following record[1].

• Sequence numbers: Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed 264 – 1.

II. SSL RECORD PROTOCOL

The SSL Record Protocol provides two services for SSL connections:

• Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads.

• Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC). The Record Protocol takes an application message to be transmitted, fragments the data into manageable blocks, optionally compresses the data, applies a MAC, encrypts, adds a header, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified, decompressed, and reassembled before being delivered to higher-level users[1].

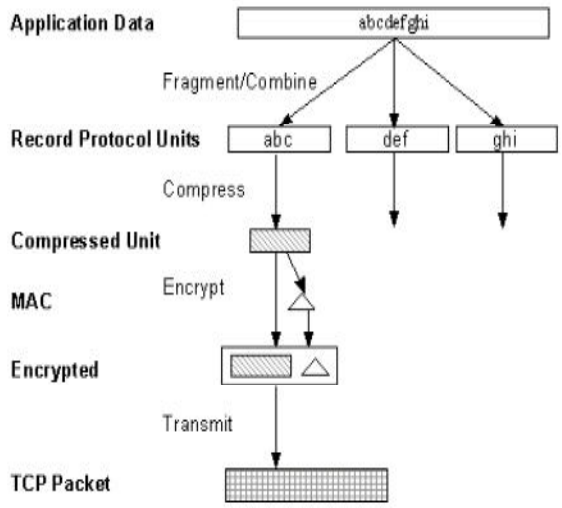

Fig: SSL Record Protocol

The first step is fragmentation. Each upper-layer message is fragmented into blocks of 214 bytes (16384 bytes) or less. Next, compression is optionally applied. Compression must be lossless and may not increase the content length by more than 1024 bytes.1In SSLv3 (as well as the current version of TLS), no compression algorithm is specified, so the default compression algorithm is null. The next step in processing is to compute a message authentication code over

the compressed data. For this purpose, a shared secret key is used[7]. The SSL Record Layer receives uninterpreted data from higher layers in non empty blocks of arbitrary size. Then the information blocks are fragmented into plain-text records of 214 bytes or less. All records are compressed using the compression algorithm defined in the current session state and protected using the encryption and MAC (Message Authentication Code) algorithms defined in the current Cipher Spec. Finally encryption and MAC functions translate compressed units to encrypted data, ready to be send into TCP packet [14].

III. CHANGE CIPHER SPEC PROTOCOL

The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest.

```
Struct {
        Enum { change_cipher_spec(1), (255) } type;
    } ChangeCipherSpec;
```

This protocol consists of a single message of a single byte with the value 1.The sole purpose of this message is to cause the pending state to be copied into the current state, which updates the cipher suite to be used on this connection. The Change Cipher Spec message is sent by both the client and the server to notify the receiving party that subsequent records will be protected under the newly negotiated Cipher Spec and keys. Reception of this message causes the receiver to instruct the record layer to immediately copy the read pending state into the read current state. Immediately after sending this message, the sender must instruct the record layer to make the write pending state the write active state. The Change Cipher Spec message is sent during the handshake after the security parameters have been agreed upon, but before the verifying Finished message is sent.

IV. ALERT PROTOCOL

SSL includes a small provision for sending event driven alert messages. Many of these indicate fatal error conditions and instruct the recipient to immediately tear down the session. For instance, the close-notify alert message indicates that the sender is finished sending application data on the connection; since alert messages are normally authenticated, this prevents a truncation attack[10]. The Alert Protocol is used to convey SSL-related alerts to the peer entity. As with other applications that use SSL, alert messages are compressed and encrypted, as specified by the current state. Each message in this protocol consists of two bytes. The first byte takes the value warning (1) or fatal (2) to convey the severity of the message. If the level is fatal, SSL immediately terminates the connection. Other connections on the same session may continue, but no new connections on this session may be established. The second byte contains a code that indicates the specific alert. First, we list those alerts that are always fatal[8].

•unexpected_message: An inappropriate message was received.

•bad_record_mac: An incorrect MAC was received.

•decompression_failure: The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).

• Handshake failure: Sender was unable to negotiate an acceptable set of security parameters given the options available.

• illegal_parameter: A field in a handshake message was out of range or inconsistent with other fields. The remaining alerts are the following.

• close_notify: Notifies the recipient that the sender will not send any more messages on this connection. Each party is required to send a close_notify alert before closing the write side of a connection.

• no_certificate: May be sent in response to a certificate request if no appropriate certificate is available.

• bad_certificate: A received certificate was corrupt (e.g., contained a signature that did not verify).

• unsupported_certificate: The type of the received certificate is not supported.

• certificate_revoked: A certificate has been revoked by its signer.

• certificate_expired: A certificate has expired.

• certificate_unknown: Some other unspecified issue arose in processing the certificate, rendering it unacceptable[1].

V. HANDSHAKE PROTOCOL

The most complex part of SSL is the Handshake Protocol. This protocol allows the server and client to authenticate each other and to negotiate an encryption and MAC algorithm and cryptographic keys to be used to protect data sent in an SSL record. The Handshake Protocol is used before any application data is transmitted. The Handshake Protocol consists of a series of messages exchanged by client and server. The handshake allows the server to authenticate itself to the client using public-key techniques like RSA, then allows the client and the server to cooperate in the creation of symmetric keys used for rapid encryption, decryption, and tamper detection during the session that follows. Optionally, the handshake also allows the client to authenticate itself to the server. This process is detailed in Figure Two different handshake types can be distinguished: The initial handshake and the resumed handshake. The initial handshake is negotiated when a client establishes a new SSL connection with the server, and requires the negotiation of the full SSL handshake. The resumed handshake is negotiated when a client establishes a new HTTP connection with the server but using an existing SSL connection. As the SSL session ID is reused, the part of the SSL handshake negotiation can be avoided[7][10].

• Type (1 byte): Indicates one of 10 messages.

• Length (3 bytes): The length of the message in bytes.

• Content (bytes): The parameters associated with this message.

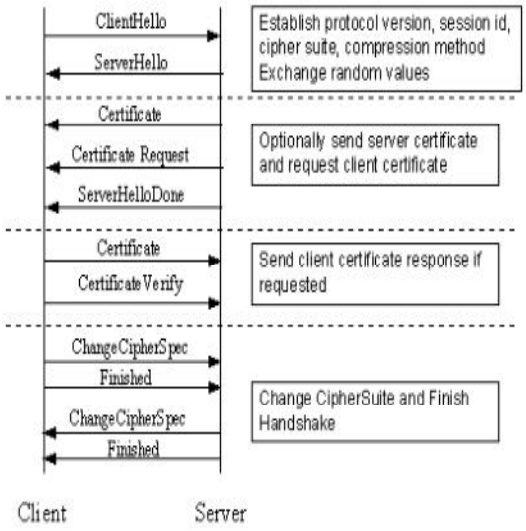

Fig. 1. SSL Handshake protocol

The SSL protocol sits between the application and transport layers. An application, such as HTTP, would call the SSL API instead of the TCP interface. The SSL session is established with a handshake between the server and the client. The client initiates the session by sending a Client Hello message with the cipher suites supported by the client. The server responds with a Server Hello identifying the strongest cipher suite supported by both parties, and the server's certificate. The client application authenticates the certificate and generates a random number called the pre-master key. The client encrypts the pre-master key with the server's public key and sends it to the server. The server decrypts the pre-master key with its private key. Both parties use the pre-master to generate the session key. At this point, the client and server exchange the Change Cipher Suite message to indicate that all future communication will be encrypted with the session key. Finally, both parties send a finished message with a message authentication code (MAC) of previous messages[1][10].

VI. CONCLUSION

The SSL protocol is intended to provide a practical, application-layer, widely applicable connection oriented mechanism for Internet client/server communications security. This paper analyzed the architectural impact of SSL. Secure Socket layer has four protocols such as SSL Record Protocol, Change Cipher Spec Protocol, Alert Protocol, Handshake Protocol. The SSL Record Protocol provides two services for SSL connections: Confidentiality, Message Integrity. The Change Cipher Spec message is sent by both the client and the server to notify the receiving party that subsequent records will be protected under the newly negotiated Cipher Spec and keys. alert messages are normally authenticated and prevents a truncation attack. Handshake Protocol consists of a series of messages exchanged by client and server.

REFERENCES

1. Stallings, "Cryptography and Network Security", 5th ed., 2005, ISBN 13: 978-0-13-609704-4

2. Atul Kahate "CRYPTOGRAPHY AND NETWORK SECURITY" ,Tata McGraw-Hill Publication ISBN 0-07-049483-5

3. Tanenbaum and Wetherall "Computer Network" 5th ed, ISBN-13: 978-0-13-212695-3

4. Josef Migga Kizza "Computer Network security"; ISBN-13: 978-0-13-212695-3

5. Prakash Kuppuswamy, 2 Prof. Osama Amer, 3 Peer Mohamed Appa, "IMPLEMENTINNG SECURE SOCKET LAYER BYUSING LBC-PUBLIC KEY ALGORITHM"; International Journal of Latest Research in Science and Technology volum.1,Issue 3 :Page No.225-228 ,September-October (2012), ISSN (Online):2278-5299

6. Emilia Kasper," Fast Elliptic Curve Cryptography in OpenSSL"

7. Vicen¸c Beltran, Jordi Guitart, David Carrera, Jordi Torres, Eduard Ayguad´e and JesusLabarta "Performance Impact of Using SSL on Dynamic Web Applications"; XV JORNADAS DE PARALELISMO—ALMERIA, SEPTEMBER 2004

8. David Wagner ,Bruce Schneier;" Analysis of the SSL 3.0 protocol",RevisedApril 1995.

9. Krishna Kant and Ravishankar Iyer;" Architectural Impact of Secure Socket Layer on Internet Servers"; November 18, 1995

10. Mr. Pradeep Kumar Panwar * Mr. Devendra Kumar; "International Journal of Advanced Research in Computer Science and Software Engineering", Volume 2, Issue 12, December 2012 ISSN: 2277 128X.

11. Norman Lim, Shikharesh Majumdar, Vineet Srivastava;" Engineering SSL-Based Systems for Enhancing SystemPerformance.

12. Krishna Kant, Ravishankar Iyer Prasant Mohapatra;" Architectural Impact of Secure Socket Layer onInternet Servers: A Retrospect".

13. Homin K. Lee, Tal Malkin, Erich Nahum;" Cryptographic Strength of SSL/TLS Servers: Current and Recent Practices".

14. Mr. Pradeep Kumar Panwar * Mr. Devendra Kumar;" Security through SSL"; International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 12, December 2012 ISSN: 2277 128X.

15. Nancy Smith;" Security Threats Against Secure Sockets Layer (SSL)", April 2010.

16. Hong lei Zhang;" Three attacks in SSL protocol and their solutions".