Web Security

Session 5

LEARNING OBJECTIVES

On completion of this session you should:

- Outline the security threats of communicating over the Internet
- Discuss how security of web server can be compromised
- Describe how SSL can be used to make a communication channel secure
- Outline the services provided by SSL
- Understand SSL record protocol
- Understand SSL handshake protocol
- Understand HTTPS communication
- Understand SSH protocol for secure network communications

Contents

Reading

Prescribed readings

Reading 1: Read through the article "Introduction to SSL" to understand how server and client authentication is done by SSL.

Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 154-156

Reading 2: Read W. Stallings, Network Security Essentials – Application and Standard, 4th edition, Prentice Hall, 2011, pp. 157-168.

Reading 3: Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 174-176.

Reading 4: Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 176-186

Reading 5: Read the online material "Securing Public Web Servers" at CERT co-ordination Center, Carnegie Mellon University.

5.0 Introduction

Now-a-days almost all the business organizations, government agencies, big retailers have web sites. The web definitely presents huge opportunities to disseminate information rapidly and to communicate easily at a much cheaper price. But it also opens an easy way for the hackers to misuse and abuse. In this study guide, we discuss the various security threats to web, and the approaches to reduce the associate risks.

5.1 Web Security Threats

Security concern over the web connections falls mainly in two categories. The first is the risk of compromising the security of the web server. For example, databases that contain sensitive financial information of the company or clients' credit cards may be exposed; or the attackers gain ability to execute malicious code on the server. The second category of concern relates to attacks perpetrated on the external network, e.g., collection of credit card numbers by eavesdropping, or diversion of client-originated traffic to a malicious web site. Once the web server's security is compromised, the attacker may gain access to data and system which are connected to that server. Stalling has presented four categories of threats related to web server and their countermeasures [1]:

- Integrity: countermeasure- cryptographic checksum
- Confidentiality: countermeasure- encryption, web proxies
- Denial of Service: difficult to prevent
- Authentication: countermeasure-cryptographic techniques

These problems can be addressed through standard security protocols supported by the web server and browser. Two most widely used such web protocols are Secure Socket Layer (SSL) and Transport Layer Security (TLS). HTTPS refers to combination of HTTP and SSL to implement secure communication between Web browser and a Web server. Another protocol that provides secure remote logon and other secure client/server facilities is SSH.

5.2 Secure Socket Layer (SSL) Protocol

SSL/TLS is a new layer of protocol inserted in the protocol stack above the Transport Layer (TCP) in the TCP/IP protocol stack. Therefore, it is capable of protecting communication from any application protocol that operates above TCP. One common use of SSL/TLS in protecting HTTP communication is a URL with "https://" indicating the use of SSL/TLS. In order to use SSL, both the web server and the client browser must be SSL enabled. The services provided by the SSL are the following [2]:

- Server authentication: allows a user to confirm a server's identity. SSL-enabled client software can use standard techniques of public-key cryptography to check that a server's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the client's list of trusted CAs. This confirmation might be

important if the user, for example, is sending a credit card number over the Internet and wants to check the receiving server's identity.

- Confidentiality: data items transferred in the session are encrypted to protect against eavesdropping.

- Integrity: data item transferred in the session are protected with a MAC (Message Authentication Code) to ensure that data have not been modified in transit.

- Client Authentication: allows a server to confirm a user's identity. Using the same techniques used for server authentication, SSL-enabled server software can check that a client's certificate and public ID are valid and have been issued by a certificate authority (CA) listed in the server's list of trusted CAs. This confirmation might be important if the server, for example, is a bank sending confidential financial information to a customer and wants to check the recipient's identity.

Every SSL/TLS communication is protected with server authentication, confidentiality and integrity while client authentication is optional. TLS can be viewed essentially as SSLv3.1 and is very close to and backward compatible with SSLv3.

SSL operates on the following principles:

1. The server presents its digital certificate signed by a CA to the client and the client gets the public key of the server from the certificate. Certification plays an important role here. Without certificate how does the client know that the server is actually what it claims to be, and not some hacker sending own set of public keys? The client can now have the confidence that it has received the intended server's public key.

2. The client generates a random master secret key and the sends it to the server, encrypted with server's public key. Since only the server can decrypt the master secret key, now only the server and the client can securely share that particular master secret.

3. The master secret key is then used to generate symmetric encryption keys for confidentiality and MAC keys for integrity purpose.

Reading 1:
Read through the article "Introduction to SSL" to understand how server and client authentication is done by SSL.

Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 154-156

The SSL protocol includes two sub-protocols: the SSL record protocol and the SSL handshake protocol. The SSL record protocol defines the format used to transmit data. The SSL handshake protocol involves using the SSL record protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when they first establish an SSL connection.

5.2.1 SSL Record Protocol

The SSL Record Protocol provides confidentiality and integrity services for SSL connection. The purpose of the SSL record protocol is to take an application message to be transmitted, fragment the data which needs to be sent, encapsulate the fragment with appropriate headers and create an object called a 'record'. The 'record' is then encrypted and forwarded for sending under the TCP protocol. The operation of SSL Record Protocol involves the following steps [1]:

- **Fragmentation:** breaks up the data stream to be transmitted into 16Kb (or smaller) data fragments.
- **Compression:** these data fragments may be further compressed (optional). Compression must be lossless. SSL 3.0 protocol specification includes no particular compression algorithm, so the default compression algorithm is null.
- **Message Authentication Code (MAC):** a MAC is calculated over the compressed data using SHA-1 or MD5 using a shared secret key.
- **Encryption:** MAC plus compressed message is encrypted using symmetric encryption (DES, 3DES, Fortezza etc).
- **SSL Record Header:** prepends header consisting of four fields: a) content type, b) major version, c) minor version, d) compressed length. The first three fields are of 8-bit and the last one is of 16-bit.

5.2.2 SSL Handshake Protocol

The handshake protocol constitutes the most complex part of the SSL protocol. It is used to initiate a session between the server and the client. Using this protocol, various components such as algorithms and keys to be used for data encryption are negotiated. It allows the server and the client to authenticate each other and negotiate appropriate parameters for the session between them. Consult Fig. 5.6 of the textbook that illustrates the process of negotiation between the client and server. Mainly it consists of the following phases [1]:

Phase 1- Establishing security capabilities: The client sends a *client_hello* message which includes client's SSL version number, cipher settings, randomly generated data, session ID and compression method to be used. The client then waits for the *server_hello* message which contains the same parameters. Random field generated by the server is different and independent of the client. If session ID of the client is nonzero, the same value is used by the server, otherwise the server generates a new session ID.

Phase 2- Server Authentication and Key exchange: The server sends *certificate message* to the client for authentication. The message sent to the client contains one or a chain of X.509 certificates. Depending on the negotiated method of key exchange, the server may send an additional *server_key_exchange* message, which is however not required in the case when the fixed Diffie-Hellman parameters or RSA key exchange is negotiated. Moreover, the server can request a certificate from the client by sending *certificate_request* message. The final step of Phase 2 is the *server_done* message, which has no parameters, and is sent by the server merely to indicate the end of the server messages.

Phase 3- Client Authentication and Key Exchange: Upon receipt of *server_done* message, the client should verify the server's certificate, the certificate validation data, as well as any other parameters sent by the server in the *server_hello* message. If the server has requested a certificate, the client send a certificate message. If no suitable certificate is available, the client sends a *no_certificate* alert message. Next the client sends a *client_key_exchange* message, which must be sent to deliver the keys. The content of this message depends on the negotiated method of key exchange. Moreover, at the server's request, the client's certificate is sent along with the message so that the server can verify client's certificate. Finally, the client may send *certificate_verify* message to provide explicit verification of the client certificate.

Phase 4- Finish: The client sends a *change_cipher_spec* message (in accordance with the pending SSL ChangeCipher Spec), and then sets up the pending set of algorithm parameters and keys into the current set of the same. The client then sends the *finished* message under the new algorithms, keys and secrets. If the *finished* message is correctly read by either party, this confirms that the transmitted data, negotiated algorithms and the session key are correct, i.e., key exchange and authentication processes are successful.

Phase-2 and 3 are used by both parties to verify the authenticity of the server's certificate and possibly the client's certificate during the handshake step. If the server cannot be successfully authenticated by the client on the basis of the delivered certificate, the handshake **terminates** and the client generates an error message. The same will occur at the server end if the client's certificate authenticity cannot be confirmed.

Reading 2:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 157-168.

5.3 HTTPS

HTTPS (HTTP over SSL) refers to the combination of HTTP and SSL to implement secure communication between a Web browser and a Web server. All modern Web browsers support HTTPS communications.

When HTTPS is used the following elements of the communication are encrypted:

- URL of the requested document.
- Contents of the document and browser forms (filled in by browser user).
- Cookies sent both ways.
- Contents of HTTP header.

HTTPS consists of:

- Connection Initiation

- Connection Closure

Reading 3:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 174-176.

5.4 Secure Shell (SSH)

Secure Shell (SSH) is a protocol for secure network communications designed to be relatively simple and inexpensive to implement. SSH provides a general client/server capability and can be used for network functions like file transfer.

SSH consists of the following protocols:

- Transport Layer Protocol
- User Authentication Protocol
- Connection Protocol

Reading 4:
Read W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011, pp. 176-186.

5.5 Web Server Security Policy

An organization must implement a concrete security policy for its web server. As a general practice, the following steps should be followed [3]:

1. Formulate a security policy.
2. Isolate the web server.
3. Configure the web server with appropriate object, device and file access controls in line with the security policy.

4. Identify and enable web server specific logging mechanism.
5. Consider security implications for programs, scripts, and plug-ins.
6. Configure the web server to minimize the functionality of programs, scripts and plug-ins
7. Configure the web server to use authentication and encryption technologies
8. Maintain the authoritative copy of the web site content on a secure host

Reading 5:
Read the online material "Securing Public Web Servers" at CERT co-ordination Centre, Carnegie Mellon University.

5.5 Conclusion

Almost all the business and government organizations maintain web presence as a vehicle to reach wider audience. Research shows that increasingly more consumers are visiting web sites to search the products they want to buy, if not buying things over the Internet. Organizations can remain open virtually 24 hours and small companies can complete big companies by doing business over the Internet. All these make establishing a secure communication channel between the business and the customer very crucial. The Internet Protocol is inherently insecure. The general guidelines for securing web servers should be followed. SSL/TLS establishes a secure communication channel over the Internet. Most of business activities over the Internet uses SSL.

5.6 References

[1] W. Stallings, Network Security Essentials - Application and Standard, 4th edition, Prentice Hall, 2011.

[2] W. Ford and M.S. Baum, Secure Electronic Commerce, 2nd edition, Prentice Hall, 2001.

[3] http://www.sei.cmu.edu/pub/documents/sims/pdf/sim011.pdf