ANNUAL

OF

KONSTANTIN PRESLAVSKI UNIVERSITY OF SHUMEN

VOL. VIII E

FACULTY OF TECHNICAL SCIENCES

Konstantin Preslavsky University Press 2018

ISSN 1311-834X

Content

USE IN INTERNET OF PROTOCOLS TRANSPORT LAYER SECURITY AND ITS NOW-DEPRECATED PREDECESSOR SECURE SOCKETS LAYER

Daniel R. Denev, Tsvetoslav St. Tsankov

ABSTRACT: *The subject of the study shows us how to use protocols on the Internet. The following protocols will be analyzed for the forthcoming report: Transport Security (TLS) And Secure Sockets Layer (SSL)*

KEYWORDS*: TLS, SSL.*

1. Introduction

Internet Protocol (IP) is the communication protocol that is the foundation of the Internet. The purpose of the protocol is to allow addressing of the information sent over the network. Each host on the network is given a unique address (called IP address). When sending information over the network, it is divided into small packages called IP packets. Each header is attached to the header, which contains the sender's and recipient's IP address and other service data. With the help of these addresses, the computers through which the package passes decide what to do with it. The protocol does not guarantee the secure arrival of the information and there is no error correction. IP is used by transport protocols like TCP and UDP [1], [2], [3].

2. Transport Layer Security (TLS) and Secure Sockets Layer (SSL)

2.1 Secure Sockets Layer (SSL)

Secure Sockets Layer (SSL) was the most widely deployed cryptographic protocol to provide security over internet communications before it was preceded by TLS (Transport Layer Security) in 1999. Despite the deprecation of the SSL protocol and the adoption of TLS in its place, most people still refer to this type of technology as ‘SSL’.

SSL provides a secure channel between two machines or devices operating over the internet or an internal network. One common example is when SSL is used to secure communication between a web browser and a web server. This turns a website's address from HTTP to HTTPS, the ‘S’ standing for ‘secure’. On Fig. 1 is shown the HTTPS websites [1], [2], [3].

Fig.1. HTTPS Web site

2.2 Transport Layer Security (TLS) and its current Secure Sockets Layer (SSL) predecessor are cryptographic protocols designed to provide communication security over a computer network. Several versions of the protocols are widely used in applications such as web browsing, email, instant messaging, and VoIP. Websites can use TLS to provide all communications between servers and Web browsers.

The TLS protocol is primarily intended to ensure privacy of data and data between two or more communications computer applications. When TLS is provided, links between a client (such as a web browser) and a server have one or more of the following properties [1], [2], [3]:

• The connection is private (or secure) because symmetric cryptography is used to encrypt the data transmitted. The keys for this symmetric encryption are generated uniquely for each connection and are based on a shared secret that was negotiated at the start of the session . The server and client negotiate the details of which encryption algorithm and cryptographic keys to use before the first byte of data is transmitted .The negotiation of a shared secret is both secure (the negotiated secret is unavailable to eavesdroppers and cannot be obtained, even by an attacker who places themselves in the middle of the connection) and reliable (no attacker can modify the communications during the negotiation without being detected).

• The identity of the communicating parties can be authenticated using public-key cryptography. This authentication can be made optional, but is generally required for at least one of the parties (typically the server).

• The connection is reliable because each message transmitted includes a message integrity check using a message authentication code to prevent undetected loss or alteration of the data during transmission.

TLS supports many different methods for exchanging keys, encrypting data, and authenticating message integrity. As a result, secure configuration of TLS involves many configurable parameters, and not all

choices provide all of the privacy-related properties. Attempts have been made to subvert aspects of the communications security that TLS seeks to provide, and the protocol has been revised several times to address these security threats. Developers of web browsers have also revised their products to defend against potential security weaknesses after these were discovered. The TLS protocol comprises two layers [1], [2], [3]:

- TLS record
- TLS handshake protocols.

TLS is a proposed Internet Engineering Task Force (IETF) standard, first defined in 1999 and updated in RFC 5246 (August 2008) and RFC 6176(March 2011). It builds on the earlier SSL specifications (1994, 1995, 1996) developed by Netscape Communications for adding the HTTPS protocol to their Navigator web browser. Figure 2 shows changes in protocols and standards over time. As a NOTE, it can be added that SSL 1.0 was formally never released due to multiple errors.

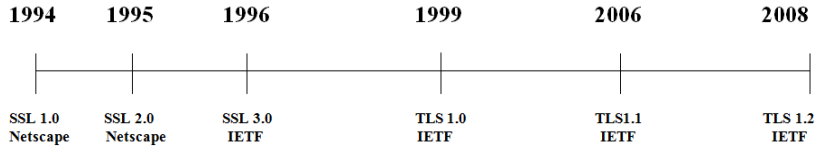

Fig.2. Changes in protocols and standards over time

3. TLS/SSL Architecture

The TLS/SSL security protocol is layered between the application protocol layer and the TCP/IP layer, where it can secure and send application data to the transport layer. Because it works between the application layer and the transport layer, TLS/SSL can support multiple application layer protocols.

TLS/SSL assumes that a connection-oriented transport, typically TCP, is in use. The protocol allows client/server applications to detect the following security risks [1], [2], [3]:

- Message tampering
- Message interception
- Message forgery

The TLS/SSL protocol can be divided into two layers. The first layer consists of the application protocol and the three Handshake sub-protocols: the Handshake Protocol, the Change Cipher Spec Protocol, and the Alert Protocol. The second layer is the Record Protocol. The following figure

illustrates the various layers and their components. Figure 3 illustrates the different layers and their components.

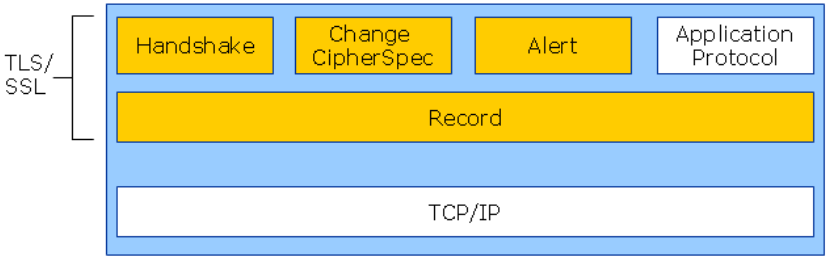

Fig.3. Preview of the layers and their components

3.1. The Handshake protocols of the TLS/SSL protocol are responsible for establishing or resuming secure sessions. The main goals of this layer are to:

- Negotiate cipher suites and compression algorithms.
- Authenticate the server to the client and, optionally, authenticate the client to the server through certificates and public or private keys.
- Exchange random numbers and a pre-master secret. Together with some further data, these values will be used to create the shared secret key that the Record Layer will use to hash and encrypt application data. The shared secret key is called the Master Secret

The Handshake protocol provides a number of very important security functions. It performs a set of exchanges that starts authentication and negotiates the encryption, hash, and compression algorithms [1], [2], [3].

3.2. The Change Cipher Spec Protocol

The Change Cipher Spec Protocol signals a transition of the cipher suite to be used on the connection between the client and server. This protocol is composed of a single message which is encrypted and compressed with the current cipher suite. This message consists of a single byte with the value 1. Message after this will be encrypted and compressed using the new cipher suite.

3.3. The Alert Protocol

The Alert Protocol includes event-driven alert messages that can be sent from either party. Following an alert message, the session is either ended or the recipient is given the choice of whether or not to end the session. The alerts are defined in the TLS specification in RFC 2246.

3.4. The Record Layer

The recording protocol receives data from the application layer and delivers them to the transport layer. Then he takes the data, fragments them to an appropriate size for the cryptographic algorithm, applies MAC or HMAC, and then encrypts (or decrypts) the data using the information agreed upon during the Handshake protocol. HMAC is only supported by TLS.

4. Operating principle

Once a client starts communication with the server, TCP connection gets established following these steps [1], [2], [3]:

- The Client first communicates with the server by sending a Hello message. The message includes number of options that will be used in the communication, such as version of the protocol to be used, Cipher Suite supported by the client, compression methods and a 32 byte random number.
- Server replies to the Hello message and makes choices about the option to be used, like version of protocol, Cipher Suite and compression methods. It also fills up the Session ID and replaces the 32 byte random number with date and timestamp.
- The server now sends Digital Certificates to the Client. The Digital Certificates contains the public key of the server.
- The Clients verifies the Digital Certificate with Certificate Authority.
- After the Digital Certificate is verified, the client starts to negotiate the symmetric key. There are a number of algorithms it can use. For an example of them is Diffie – Hellman key exchange algorithm shown on fig. 4.

Traditionally, secure encrypted communication between two parties required that they first exchange keys by some secure physical channel, such as paper key lists transported by a trusted courier. The Diffie–Hellman key exchange method allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure channel. This key can then be used to encrypt subsequent communications using a symmetric key cipher

Cryptographic explanation of the algorithm by the studied Example [1], [2], [3]:

In the most accessible and elementary way, the protocol uses the multiplicative group of integers mod (module) P, where P is base and G is the primitive root of P. These two values are chosen in such a way to ensure that the shared secret can accept any value from 1 to P-1. In the example of the protocol we will mark non-target values in blue and secret values in red.

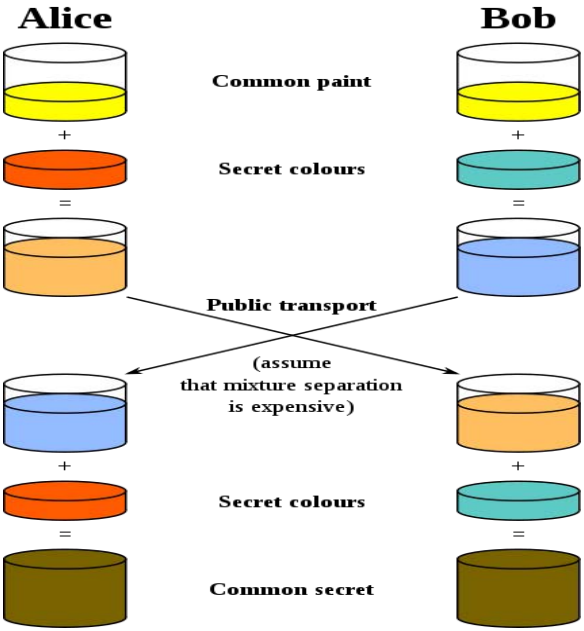

Fig.4. Diffie – Hellman key exchange algorithm

Silvi and Gosho agree to use a modulus $p = 23$ and base $g = 5$ (which is a primitive root modulo 23).

Silvi chooses a secret integer $a = 4$, then sends Gosho $A = g^a \bmod p$

- $A = 5^4 \bmod 23 = 4$

Gosho chooses a secret integer $b = 3$, then sends Silvi $B = g^b \bmod p$

- $B = 5^3 \bmod 23 = 10$

Silvi computes $s = B^a \bmod p$

- $s = 10^4 \bmod 23 = 18$

Gosho computes $s = A^b \bmod p$

- $s = 4^3 \bmod 23 = 18$

Silvi and Gosho now share a secret (the number 18).

Conclusion : Both Silvi and Gosho have arrived at the same value s, because, under mod p,

$A^b \bmod p = g^{ab} \bmod p = g^{ba} \bmod p = B^a \bmod p$ **and more precisely**

$(g^a \bmod p)^b \bmod p = (g^b \bmod p)^a \bmod p$

The server processes the key exchange parameters. It also checks the MAC or Message Authentication Code to the server.

If everything goes well, a secure TLS connection is established between the server and the client and secure communication starts to transfer sensitive application data. Fig. 5 shows a schematic view of the operating principle [1], [2], [3].

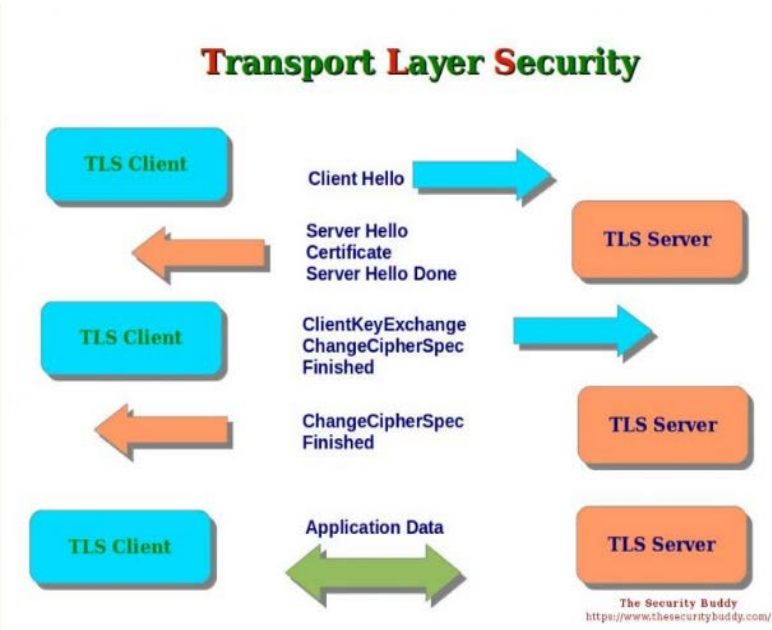

Fig.5. Principle of TLS protocol operation

5. Network ports used by TLS / SSL

Each standard TLS / SSL protocol is described by "service name" and its "network port"

Table 1. Description of TLS / SSL on Service Name and its network port

Service Name	smtp	https	nntps	ldaps	ftps-data	ftps	telnets	imaps	tftps
TCP Port	25	433	563	636	989	990	992	993	3713

3. Conclusion

The purpose of the study is to show the use of protocols on the Internet Transport Layer Security and Secure Sockets Layer. Our research is widely used in computer networks

REFERENCES:

1. Pease, S. (2009). Secure Sockets Layer. Cengage Learning, Boston.
2. Andreev A. The environment of security and influence national security and national interests. NMU- Shumen. Scientific Conference "Problems of national security", 2009, ISSN 1314-0647).
3. Andreev A. Any aspects of security what concept. NMU- Shumen, Scientific Conference "Problems of national security", 2009, ISSN 1314-0647.

Author's name: Daniel Rosenov Denev, student
Workplace: Konstantin Preslavsky University of Shumen, Faculty of Technical Sciences

Author's name: assoc. prof. Tsvetoslav St. Tsankov, PhD
Workplace: Konstantin Preslavsky University of Shumen, Faculty of Technical Sciences
E-mail: c.cankov@shu.bg