Apache 2.0 with SSL / TLS protocol: Step by step instructions

For more than ten years, the SSL protocol has been widely used to ensure the safety of web transactions over the internet. You can imagine millions of dollars every day, billions of dollars of transactions on the network using SSL. However, the simple truth is us

PART I : **General**

For more than ten years, the SSL protocol has been widely used to ensure the safety of web transactions over the internet. You can imagine millions of dollars every day, billions of dollars of transactions on the network using SSL. However, the simple fact is that we have used SSL in a way that is not really necessary. The information sent via this protocol remains secure. Weak encryption, no verification of web servers (on the server) certificates, security holes, and many other types of attacks can allow intruders to access sensitive information. , despite the fact that it is being sent over SSL.

This article begins with an introduction to Apache 2.0 configuration series that supports SSL / TLS protocol, to ensure maximum security and optimal performance in SSL transmission.

- **Part I** : Introduction to the aspects of key in SSL / TLS and instructions on how to install and configure the 2.0 configuration that support these protocols.

- **Part II** : Discuss the configuration of *mod_ssl* and the sources that provide the address with the web server validation set.

- **Part III** (also the last part): Discuss issues of client identification (client) and some typical configuration errors of administrators. These errors can reduce the security level of any SSL-using communication network.

Introduction to SSL

Secure Sockets Layer (SSL) is the best known protocol for security and reliability in client-server (client-server) on the Internet. SSL itself is based on fairly simple concepts. It arranges encryption algorithms and locks between two send-and-receive transactions. Then set up a virtual encrypted path through other protocols (like HTTP). SSL can also authenticate both sides of the transaction through the use of 'certificates'.

SSL is a layered protocol, consisting of the following four sub-protocols:

1. SSL Handshake protocol
2. SSL Change Cipher Spec protocol
3. SSL Alert protocol
4. SSL Record Layer

The location of the above protocols, corresponding to the TCP / IP model, is illustrated in the following chart:

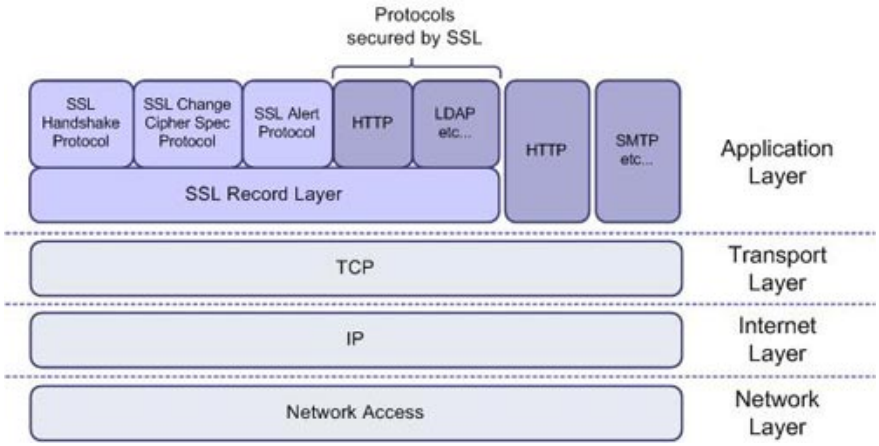

Figure 1. Sub-protocols of SSL in the TCP / IP model

According to the chart above, SSL is in the application layer of TCP / IP protocol. Due to this feature, SSL can be used in almost every operating system that supports TCP / IP without having to modify the system kernel or TCP / IP stack. This gives SSL a strong improvement over other protocols such as IPSec (IP Security Protocol). Because this protocol requires the operating system kernel to support and modify the TCP / IP stack. SSL can also easily bypass firewalls and proxies, as well as NAT (*Network Address Translation*) without the need for a source.

How does SSL work? The diagram below shows simply the step by step process of establishing an SSL connection between the client (client - using a web browser path) and the server (server - using an SSL web server)

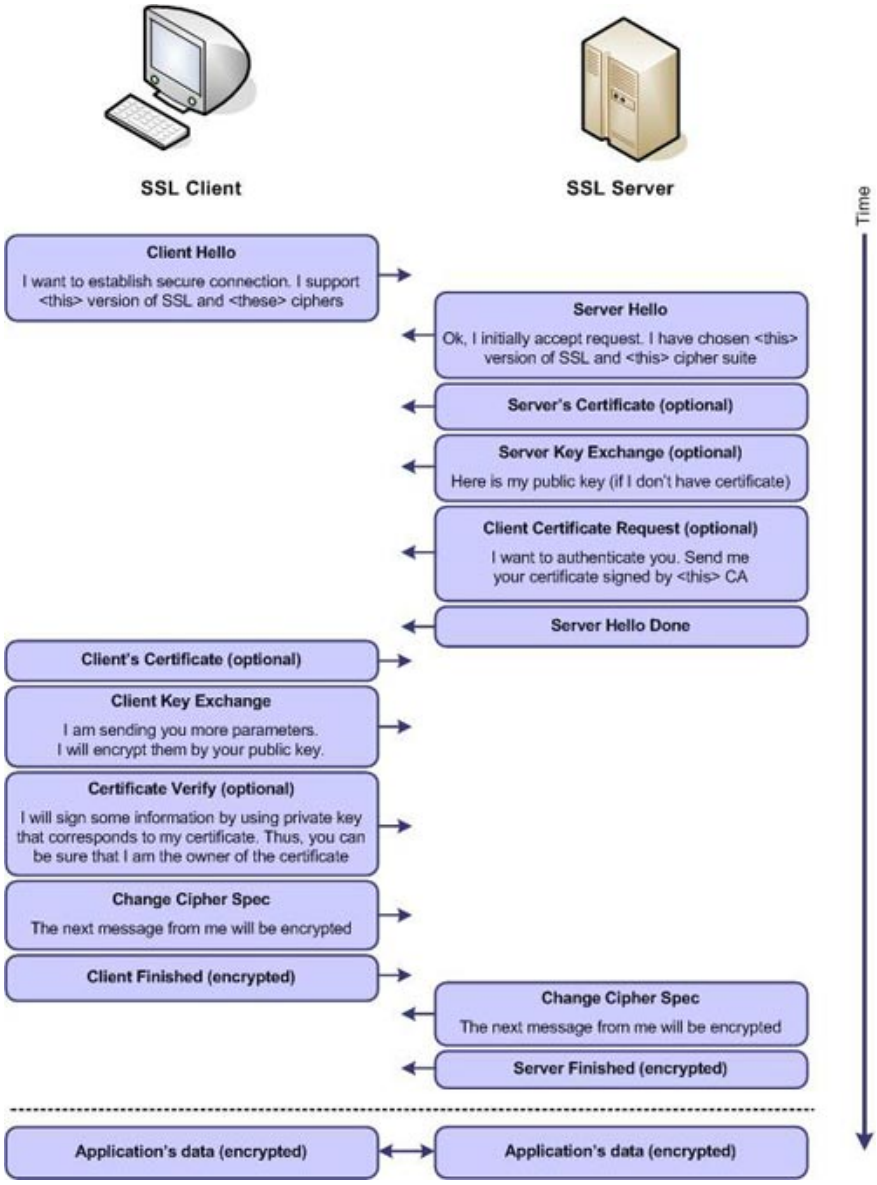

Chart 2. Step by step establish an SSL connection

As you can see in the figure, the process of establishing an SSL connection starts with exchanging the encryption parameters and then confirming the servers arbitrarily (using *SSL Handshake* protocol). If 'handshake' is successful, both sides accept the common encoder and encryption keys, the data is in the application layer (usually using HTTP, but it can also be another protocol) can be sent via encrypted tunnel (using *SSL Record Layer*).

In fact, the process is a bit more complicated. To avoid unnecessary 'handshakes', some coded parameters are retained. Notifications are sent. Encoders can also be changed. However, despite these technical characteristics, the most common way of doing this process really works.

SSL, PCT, TLS and WTLS (but no SSH)

Although SSL is the most popular and most popular protocol, it is not the only protocol used for web safety and

transport purposes. It is also important to know that, since the SSL v1.0 invention was born, there are at least five other or less or more protocols that play an important role in access to the World Wide Web. Specifically:

SSL v2.0

This version was created by Netscape Communications in 1994. The main purpose of this protocol is to provide security for transactions on the World Wide Web. Unfortunately, quickly after that people saw a weak number of safety in the early version of this SSL protocol. Thus making it less reliable with commercial use.

1. The structure of MAC is weak.
2. It is possible for groups to use weak encryption
3. Does not protect the process of 'shaking hands'
4. It is possible that attackers use truncation attack.

PCT v1.0

Developed by Microsoft in 1995. PCT (Privacy Communication Technology) v1.0 addresses some of the weaknesses of SSL 2.0 and aims to replace SSL. However this protocol has never obtained popular results such as SSL v3.0.

SSL v3.0

Released in 1996 by Netscape Communications. SSL v3.0 solves most of the problems of SSL v2.0 and incorporates many components of PCT. Soon it became the most popular protocol for media safety on the World Wide Web.

TLS v1.0 (*known as SSL v3.1*)

Created by IETF in 1999 (RFC 2246). This protocol is based on SSL v3.0 and PCT. It balances both ways of Netscape and Microsoft. It should be noted that, although TLS is based on SSL, it is not a 100% compatible version with previous versions. IETF has implemented a number of safety improvements. For example, using HMAC instead of MAC, use other calculations in the security of servers and key documents, add editors, do not support Fortezza encoders, etc. Results of upgrades This is the protocol does not work fully. Finally TLS also falls into oblivion compared to SSL v3.0.

WTLS

The 'mobile and wireless' version of the TLS protocol, uses the UDP protocol as a communications carrier. WTLS is designed and optimized for lower bandwidths, smaller processes with mobile devices that allow WAP. WTLS offers the same WAP 1.1 protocol by WAP Forum. However, after the WAP 2.0 protocol was introduced, WTLS was replaced by an original version of TLS with a higher level of security. It does not need to decrypt or re-encrypt the traffic at WAP gateway.

Why is the SSH protocol not used for security purposes when accessing WWW?

There are several reasons! First, right from the start, TLS and SSL were designed for network security sessions (HTTP), while SSH was backed up to replace Telnet and FTP. SSL does nothing more than 'shake hands' and set up 'encrypted tunnels'. And at the same time, SSH offers a way to login between people - computers, transmit

secure files, support for multiple steps of checking permissions (including passwords, public keys, Kerberos .). On the other hand, SSL / TLS is based on X.509v3 and PKT certificates, which make it much easier to distribute and manage authority authentication. For these reasons and a number of other reasons that make SSL / TLS more suitable than secure access to WWW and other similar types of communication, including SMTP, LDAP . while SSH is more and more convenient. for managing remote systems.

In summary, although there are many 'safe' protocols in practice, we should only use two web transaction protocols (at least at this time): TLS v1.0 and SSL v3.0. Both are highlighted in this series with the simple name SSL / TLS. Because of the weaknesses of SSL v2.0 and the well-known WTLS 'WAP vulnerability', we should avoid using these protocols, or at least to minimize.

Software requirements

In the next part of this series, we will see how to configure Apache 2.0 with SSL / TLS support, using the *mod_ssl* model. Before going further, readers should download the source code of the latest version of Apache 2.0 from the Apache website. Most examples also do in Apache 1.3.x. In this case *mod_ssl* needs to be downloaded separately from Apache's source code on the *mod_ssl* website.

The specific examples in this article mostly work on Linux operating systems, like Linux and BSD - the basic operating system. The only requirement for the operating system is to have both *OpenSSL* and *GCC libraries* .

As a default web browser, MS Internet Explorer has been selected for our test mainly because of its popularity. However, any other modern web browser can be used like FireFox, Mozilla, Netscape, Safari, Opera .

Installing Apache supports SSL / TLS

The first step to installing Apache software that supports SSL / TLS is to configure and install Apache 2 web server, create a user and a group named 'apache'. A safe installation has been published on SecurityFocus, in the Security Apache 2.0 article: Step-by-Step. The only difference is the process for setting mod_ssl and mod_setenvif. It requires compatibility with some versions of Internet Explorer as follows (changes indicated in bold):

./configure
--prefix = / usr / local / apache2
--with-mpm = prefork
- **enable-ssl**

--disable-charset-lite
--disable-include
--disable-env
- **enable-setenvif**
--disable-status
--disable-autoindex
--disable-asis
--disable-cgi
--disable-negotiation
--disable-imap
--disable-actions
--disable-userdir
--disable-alias
--disable-so
After configuration is complete, we can install Apache under the root directory:

make
su
umask 022
make install
chown -R root: sys / usr / local / apache2

See more Part I

You finished reading the article "**Apache 2.0 with SSL / TLS protocol: Step by step instructions**" edited by the **TipsMake** team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.