The easiest way to get around SSL

ing. Roberto Larcher
http://webteca.altervista.org

robertolarcher@hotmail.com

rev. 2 - 03/09/2003

Abstract

This paper explains how it is often possible, with the simple substitution of a string, to get around a "secure" implementation based on an incorrect use of SSL.
Please note that this document does not contain any information about weaknesses of the SSL protocol; it simply shows the easiest way to get around the correct functioning of the SSL protocol.
In this document typical "weakly secure" implementation based on the SSL protocol are illustrated.
A simple test application is also proposed to check if existing implementations are indeed "weakly secure".
This document has an informative purpose.

The weak ring

When speaking about ICT security, we often make a comparison with a chain: just as the strength of a chain is equal to the strength of its weakest ring, the security of an ICT structure is equal to the security guaranteed by its less secure component.
The SSL – Secure Sockets Layer – protocol was created to provide a secure way to conduct commercial and financial transaction over Internet[1].
The problem is that this protocol is often used with other weaker "rings", so producing only apparently secure implementations, whereas actually they are subject to different kind of attacks.

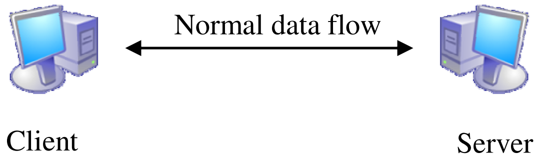
Some notes about the SSL protocol

"The SSL protocol is intended to provide a practical, application-layer, widely applicable, connection-oriented mechanism for Internet client/server communication security" [2].

In other words, it is an application layer that establishes a secure connection between a client and a server, based upon a set of technologies, such as public key cryptography.
For example, data coming from an *Internet browser* are encrypted (with strong ciphers) and then sent to the server. The server, upon receiving the data, deciphers them to clear text.
This operation is almost always transparent to the final user, who, at most, sees a request tagged with the https:// prefix, instead of the normal http:// prefix.

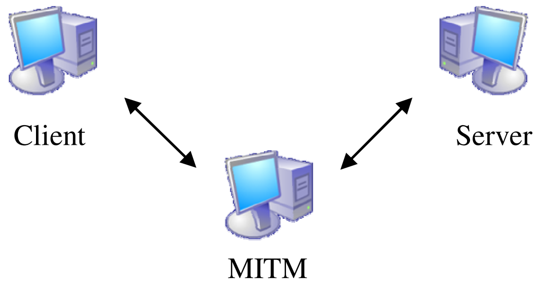
MITM attacks

The main objective of attacks to ICT systems, and particularly to communication systems, is access to information.
Amongst a variety of techniques that can be used, there is one that is particularly powerful. It is the MITM, Man In The Middle, attack. This kind of attack uses the chance to interpose itself between the sender and the receiver.



Technical details about how to set up this kind of attack are not included in this document. For the purpose of this document it is enough to know it's possible to have a node send data not to the intended receiver but to a third entity, known as the man in the middle, that impersonates the real receiver.

[1] The most recent SSL 3.0 specification [1] may be viewed here: http://wp.netscape.com/eng/ssl3/.



In other words “Client” is induced to believe that “MITM” is “Server”.
As a result, the man in the middle can sniff and even tamper data as he likes.

MITM attack and SSL

One of the requirements of the SSL protocol is that it must resist to a MITM attack [4].
As already said, this document doesn’t include new kind of attempt to violate or crack this protocol; it simply explains how it is often sufficient to induce a client not to use this protocol when trying to access sensitive information that would be protected.

Normal use of the SSL protocol

The SSL protocol can be used in several ways. One of the most popular is the one used by commercial and financial sites over the Internet to protect sensible data such as, for example, credit card numbers.
When sensible data are going to be transmitted, a secure connection is created from the client to the server. In other words a new https session is started from a normal http session.
That can be done in an explicit way – a user clicks on a link in a html page – or in an implicit way, managed by the browser and transparent to the user.
In technical terms, in the description of an http page there is a link to a secure page:

https://www.site.com/secure_page.html

A simple MITM attack

The easiest way to make SSL useless is to use it in an inappropriate way.
It is possible to imagine a MITM attack that intercepts the normal http flow and replaces every instance of the “https://” string with “http://”.
Thus, the Client will never have the chance to request a page from a secure Server because this chance simply does not exist at all for him.
The Client will make a normal clear-text http request.

The following part of this document expounds the results of experimental tests based on this type of attack, and proposes a classification of sites that are vulnerable to it.

The test application

To make the experimental test easier and to explain the principle of this attack, a simple Windows test application, called https2http[2], was created.

This application acts as a normal proxy[3] that replaces every instance of the **https:**// string with the **http:**// string in each clear-text page that is downloaded from a Server to the Client.

It is not necessary to setup a real MITM attack to run the test. To simulate its behavior you can:

- run https2http on a PC that will act as local proxy
- configure https2http to redirect traffic to a remote proxy
- configure the Internet browser to use the PC with https2http as proxy

Experimental results

A number of real Internet sites were tested with the abovementioned setup.
Some risky behaviours are quite common and can be classified as follow.

“weakly secure site of first type”

Sites of this category permit access to pages that have to be transmitted in a secure way not only via https:// requests but also via http:// requests.

In other words, servers of a “weakly secure site of first type” permit access to all the pages via port 443 and via port 80 too.

It is easy to check this behaviour, using a normal Internet browser: try to request a secure page

https://www.site.com/secure_page.html

and then try to remove the “s” from https

http://www.site.com/secure_page.html

[2] You can download this application here http://webteca.altervista.org/https2http.htm
[3] The current version of https2http needs a remote proxy to connect to the Internet.

If the site replies with the same data instead of with a denied access error, then the data were transmitted in clear-text and the site is “weakly secure of first type”.

In fact a MITM attacker can replace every instance of the https:// string with the http:// string with no troubles, because the site will continue to reply in a correct way, though transmitting data in clear-text mode.

It’d be clear that such an implementation is highly risky.

“weakly secure site of second type”

Sites of this second category permit to send sensible data – for example the user/password couple – to secure links from normal clear-text transmitted page.

In other words, a non-secure http page is used to jump inside a secure site.

Also in this case you can easily verify this behaviour with a normal Internet browser: connect to

http://www.site.com/page_that_sends_data_to_a_
secure_link.html

and verify that in the page does exist a link to a secure page (for example, using Internet Explorer, you can view the HTML source code with the menu View⬂HTML).

If the data sent to the server, for instance from a form, are sensible, then the site is a “weakly secure site of second type”.

Please note that it is not necessary that the https: request – transformed by the attacker to an http: request – is accepted by the server in order to consider the site “weakly secure of second type”, since sensible data are sent over the net in clear-text anyway, and they can be sniffed or altered.

Conclusions

This document has illustrated how is possible, under certain conditions, to make a Client use clear-text transmission instead of ciphered transmission via the SSL protocol.

The technique is based on the simple replacement of any instance of the https:// string with the http:// string included in clear-text http requested internet pages.

Tests were made on a set of real sites. Some sites behave in a correct, secure manner. Other sites behave in different ways; those sites can be grouped into two categories of “weakly secure” sites, because they may be subject to MITM attacks as above described.

An application has been presented to facilitate the setup of a test environment, in order to verify if a site is vulnerable to this kind of threats.

Acknowledgements

I am greatly thankful to Marco Favaretto who revised both the Italian and English versions of this paper. Thanks also to my sister Cinzia for additional corrections.

Of course any mistake in this paper is only my responsibility.

References

[1] The SSL Protocol Version 3.0
A. O. Freier, P. Karlon, P. C. Kocher

http://wp.netscape.com/eng/ssl3/draft302.txt

[2] Analysis of the SSL 3.0 Protocol
D. Wagner and B. Schneier

The Second USENIX Workshop on Electronic Commerce Proceedings, USENIX Press, November 1996, pp. 29-40.
http://www.counterpane.com/ssl.html

[3] Finite-state analysis of SSL 3.0.
J. C. MICThell, V. Shmatikov, and U. Stern.

In Seventh USENIX Security Symposium, pages 201--216, San Antonio, 199.
http://citeseer.nj.nec.com/
mICThell98finitestate.html

[4] Penetration Testing with dsniff
Christopher R. Russel

SANS' Information Security Reading Room
February 18, 2001
http://www.sans.org/rr/threats/dsniff.php