



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Security Essentials: Network, Endpoint, and Cloud (Security 401)"
at <http://www.giac.org/registration/gsec>

An Overview of SSL and its implementation on IIS

What is SSL and how does it work?

Secure Sockets Layer (SSL) is a protocol that functions between the transport layer and the application layer of the TCP/IP protocol suite. SSL itself is composed of two layers. Directly above the transport layer is the SSL Record Protocol, which is responsible for encapsulation of the higher layer protocols. The SSL Handshake Protocol allows the client to authenticate the server and (potentially) allow the server to authenticate the client. It is also responsible for negotiating an encryption algorithm before application data can be sent.

Example Hello Session:

Client

Web Server

Client sends "Hello" >>>

<<<

Server Hello

(Certificate, server key, possible client certificate request)

SSL was designed to be available for use in a variety of tcp/ip applications, such as telnet, http and future applications. Most noticeable of its use is on the Internet, for securing http data. If you purchase an item from a web merchant, your web browser will inform you, using a message popup and/or a lock icon on the browser window. That message lets you know that https (http over ssl) is in use.

So what happens actually when you type https://www.xxxyyzzz.com? The client sends out an "Hello" message to the web server's tcp port 443 (default). The server responds with a response to the client Hello, including the web server's certificate. The client's browser, depending on its settings, will either accept the certificate or prompt the user to either accept/deny the certificate. This is a one way server authentication. Web servers using SSL (such as IIS) can also require that the client authenticate with the server, allowing for a two-way authentication in the "Hello" handshake. This is more popular for web servers that have a strict amount of client access.

Why use SSL?

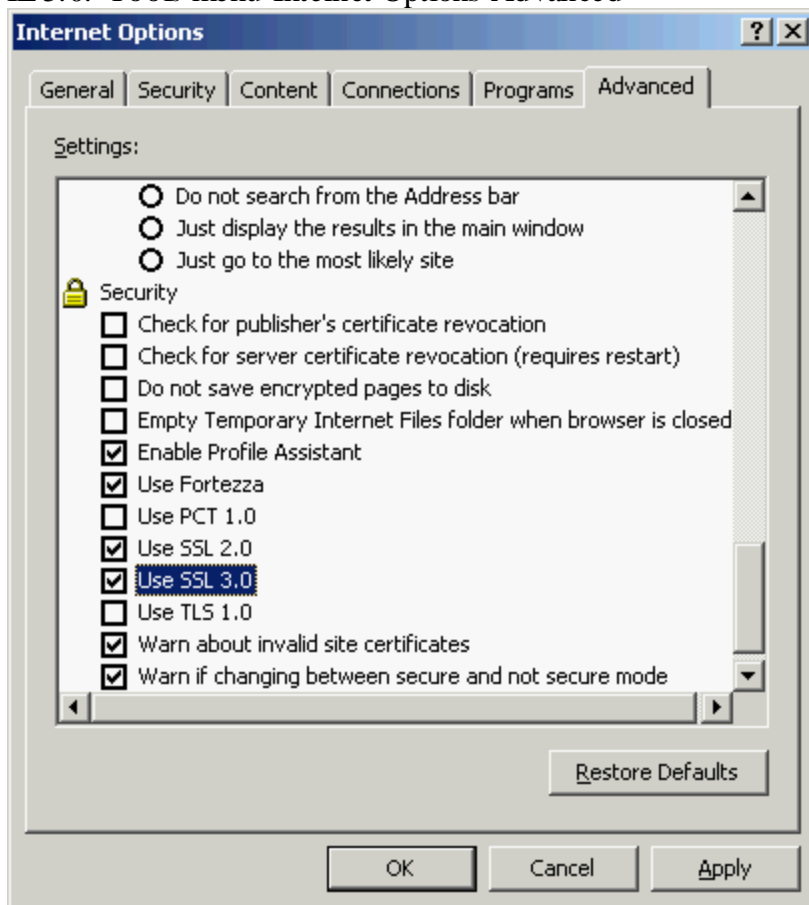
The primary reason for using SSL on any web service, not just IIS, is to encrypt the data stream from the web server to the client. SSL also provides for server authentication and (less common) client authentication. Typical uses include web sites that allow for secure payment transactions to occur. By secure, I mean that the data is sent in ciphertext, not cleartext. Anyone with a network "sniffer" would only see unreadable garble.

As with any security measure, SSL should be used only in conjunction with other security measures and policies. What good is encrypted http data if your important database is lying unprotected in cleartext on a server?

The client (web browser) configuration of SSL

Luckily for web sites that want to use this technology, the client portion (the web browser) is ready for SSL. Current versions of IE and netscape provide for support and configuration options for SSL beyond the default options of the installation.

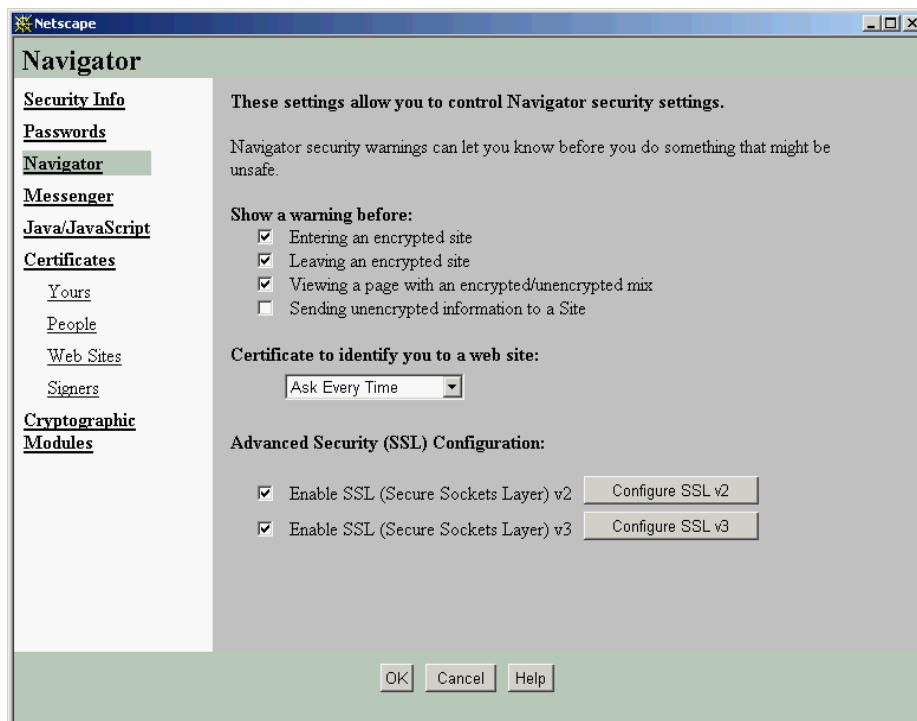
IE 5.0. Tools menu-Internet Options-Advanced



Notice that if you click “restore defaults”, SSL 2.0 and 3.0 are enabled.

In Netscape 4.75, there are similar options.

Communicator menu-tools-security info-navigator



The web browser is more important than you may think. Keeping up to date with the latest web browser versions and patches is just as important as the maintenance of the server itself. Web Browsers, like IE and Netscape Navigator come ready to work with Certificate Authorities like Verisign.

The server side installation of SSL.

OK. Now you need to do a little more work. First, determine who is going to be your CA. The CA (Certificate authority) is used to provide verification of a web site's identity. You can choose to be your own CA (for example, use Microsoft Certificate Server) or use the services of a third party CA, such as Verisign. A CA such as Verisign will go through a verification process before issuing you a certificate. Once that process is finished, you can continue install a certificate on your web server.

In IIS, the process of creating and installing an SSL certificate is fairly easy. This is covered by MS knowledge base article 228991 (IIS 4.0) and 228821 (IIS 5.0). The key portions to be aware of are the "common name" and the "bit length" of the request. The common name is the fully qualified domain name of the web site you wish to have SSL enabled. For example, if the web site is `www.youmamehere.com`, then that is the common name you want entered in the "common name" field. The certificate is created specifically for that FQDN. Web browsers will flag an alert if the certificate's common name does not match the FQDN you types in the browser window. If you have a corporate web server using SSL, type in `https://<IP address>` instead of `https://<FQDN>` to see what I mean. It is not an error per say, but just the web browser saying, "hey take a look at this before you continue." Verisign has to make sure that the company requesting a certificate owns any FQDN you place in a certificate request. Finally,

the bit length refers to the strength of the key pair that is generated. Don't get this confused with the 40bit or 128 bit "session key" used for communications between the client's browser and web server.

Here are some words of caution when filling out the request for a Certificate from a CA like Verisign. Be very careful and certain when choosing the FQDN that you submit to the CA! Once you receive the certificate for that particular FQDN, you cannot change your mind. For example, if you submitted a request for `www.xxxyyyzzz.com` and then later to decide to make your site `www2.xxxyyyzzz.com`, you cannot change the current certificate to reflect this. You can "revoke" the certificate if you no longer wish to use it, but you are out the cost of a certificate! Certificates can cost several hundred dollars or more, depending on the strength (40 or 128 bit) and length of time you have the certificate for (typically 1-2 years). The certificate request is for the FQDN only, not for any IP address or specific URL for a website. Discuss the FQDN carefully with the webmaster and determine how this SSL site will be accessed. Most sites have an initial HTTP web site that lead to an SSL enabled web site via some hyperlink. Since most web surfers don't start off with https in their browser and they usually type `www.xxxyyyzzz.com` or just `xxxyyyzzz.com`, you may want to have the SSL certificate be registered with something like `www2.xxxyyyzzz.com`. A link from `www.xxxyyyzzz.com` would lead to the SSL web site. Another option is to just have certain pages of the SSL web site not "require secure channel when accessing this device". Those pages could be the gateway to the SSL enabled pages.

Depending on who deals with what tasks in your company, you will also list several types of contacts in the certificate request process (at least in Verisign's case). These relate to technical and billing departments within your company. They may or may not be the same person, depending on how your business is structured. Make sure you confirm the contacts and let the contacts know that they are listed as contacts! Nothing is worse than not being informed of a role in project that you did not even know existed!

The server side configuration of SSL.

Just like any other aspect of IIS, there are settings that can be fine-tuned to make the SSL work better for you. One of the most important check boxes is to "require secure channel when accessing this device". This forces the user to use SSL (`https://`) instead of `http` when accessing any web page that is covered by this IIS setting. Of course, you can have specific pages not use SSL (most likely a home page) by setting this property for the specific html files in the web site. The pages you want secure could have links from the non-secure pages specifying `https://<FQDN>`. This is a fairly seamless move to make for the web browser. Typically, the browser will just inform you that you are entering a secure site.

Another important checkbox is "require 128 bit encryption". If you are planning to have web visitors from outside of the US and Canada, leave this unchecked. Due to export restrictions, 128 bit capable browsers are not available outside the US and Canada.

Just like any other application, IIS needs to be regularly updated and maintained with the latest patches and bug fixes. This is ongoing maintenance that, if left undone, could lead to more

vulnerabilities and potential security breaches on your system. Check out Microsoft's web site for the latest updates for NT and IIS. A "default" configuration will be "your fault" when things go bad!

A network trace.

Here is a very simple network trace using Microsoft's network monitor. The first 7 frames (29 frames total captured to load the test html page) show a connection to the home page of a very basic web site. Notice the initial HTTP GET requests and corresponding responses immediately following the initial TCP 3-way handshake:

```
1 2.463542 LOCAL LITE-O61D5C0 TCP ....S., len: 0, seq:2262466794-2262466794, ack
LOCAL_PC 192.168.100.2

2 2.463542 LITE-O61D5C0 LOCAL TCP .A..S., len: 0, seq: 170525658-170525658, ack:
192.168.100.2 LOCAL_PC IP

3 2.463542 LOCAL LITE-O61D5C0 TCP .A....., len: 0, seq:2262466795-2262466795, ack
LOCAL_PC 192.168.100.2

4 2.463542 LOCAL LITE-O61D5C0 HTTP GET Request (from client using port 1108)
LOCAL_PC 192.168.100.2

5 2.463542 LITE-O61D5C0 LOCAL HTTP Response (to client using port 1108) 192.168.100.2
LOCAL_PC IP

6 2.583715 LOCAL LITE-O61D5C0 HTTP GET Request (from client using port 1108)
LOCAL_PC 192.168.100.2

7 2.583715 LITE-O61D5C0 LOCAL HTTP Response (to client using port 1108) 192.168.100.2
LOCAL_PC IP
```

Now consider the same connection, displaying the same home page, using the same web browser (cache previously cleared). There is no mention of HTTP GET requests throughout the entire 59 frames (only first 8 shown here to save space). In fact, frame 6 actually contains the certificate being sent from the server to the client. I would guess that the client SSL "Hello" starts at frame 4, after the initial TCP handshake.

Frame	Time	Src MAC	Addr	Dst MAC	Addr	Protoco	Description	Src Other	Addr	Dst Other	Addr
Type											

```
1 4.977156 LOCAL LITE-O61D5C0 TCP ....S., len: 0, seq:2281988521-2281988521, ack
LOCAL_PC 192.168.100.2
```

2 4.977156 LITE-O61D5C0 LOCAL TCP .A..S., len: 0, seq: 190029642-190029642, ack: 192.168.100.2 LOCAL_PC

3 4.977156 LOCAL LITE-O61D5C0 TCP .A...., len: 0, seq:2281988522-2281988522, ack LOCAL_PC 192.168.100.2

4 5.748265 LOCAL LITE-O61D5C0 TCP .AP..., len: 48, seq:2281988522-2281988570, ack LOCAL_PC 192.168.100.2

5 5.918510 LITE-O61D5C0 LOCAL TCP .A...., len: 0, seq: 190029643-190029643, ack: 192.168.100.2 LOCAL_PC

6 7.110224 LITE-O61D5C0 LOCAL TCP .AP..., len: 721, seq: 190029643-190030364, ack: 192.168.100.2 LOCAL_PC

7 7.210368 LOCAL LITE-O61D5C0 TCP .AP..., len: 208, seq:2281988570-2281988778, ack LOCAL_PC 192.168.100.2

8 7.240411 LITE-O61D5C0 LOCAL TCP .AP..., len: 71, seq: 190030364-190030435, ack: 192.168.100.2 LOCAL_PC

The points to be made from the trace are these:

- a). Network monitor (and other sniffers I've tested) do not show the application layer protocol and corresponding data. This means SSL is doing what it states it does. It is hiding the details of the higher layer protocols.
- b). Capturing the same exact data took twice as many network frames with SSL enabled than w/o SSL. Now, this is only one example, so don't read too much into it! Adding many complicated graphics to a web page can also impact the performance of SSL. Only thorough testing with specific web server "load testing" software can help determine what your network and web servers can handle! See <http://www.softwareqatest.com/qatweb1.html> for an overview of several products that may help with this task. I have run into many products that do not support SSL! As of this writing, Mercury Interactive's Astra LoadTest product (<http://www-heva.mercuryinteractive.com/products/testing/>) is one that does not require modification to the browser (as far as proxy settings) nor to the server to test SSL. Others require modifications to both server and client. Load testing is a whole separate paper by itself!

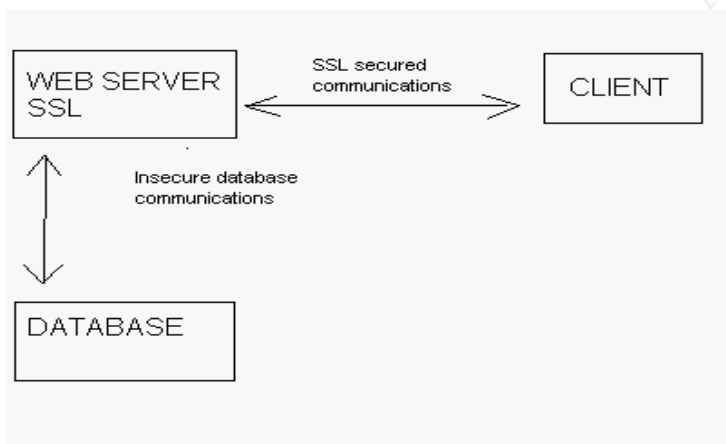
Pitfalls and shortcomings.

SSL only covers a tiny aspect of data security. Without the implementation of other layers of security, this feature really does not do too much good at all. In fact, just the process of going through a CA for a certificate is insecure. The reason is that people ultimately make the decisions that govern the rest of the process. Verisign, a well known and built in CA to the popular web browsers, has recently made a blunder with code signing certificates. The result: two code signing certificates, issued in Microsoft's name from Verisign, are floating around the

Internet. Damage could have been more serious if the Internet community was not informed of the two bogus code-signing certificates that now exist. Microsoft has released a patch for this blunder (see MS Q 293818 for details).

IIS allows for the backup and restoration of certificates. Use these options! Don't save the all-important certificate on the server. It is small and can be easily backed up to a floppy disk. Label it (so you can identify it), write protect it and lock it away. If the IIS server crashes (this never happens...right?) then it can be restored (there is password protection for this process). Be careful with when saving and restoring to your brand new shiny windows 2000 IIS 5.0 server. Microsoft has released a patch (also included in SP2 for windows 2000) to fix an issue when trying to restore a previously saved certificate. See MS Q261655 for details.

The next example shows a situation where the client has a secure communications link with the web server. All of the web pages are sent encrypted to the client. The problem displayed in the diagram goes unseen by the web client. The web server, in order to display the correct information to the client, has to retrieve data from a database on another server. That connection is not secured by SSL in this scenario. So, if there was a network sniffer between the web server and database server, that information is compromised even before the web server sends the completed https request back to the client.



The same type of scenario can occur when using a proxy server (such as MS proxy server 2.0 or Netscape proxy server). The connection from the proxy server to the web client is secured by SSL, but the connection between the web server and proxy server is not. Using "server proxy" (the recommended method from microsoft) helps resolve the issue. The resolution involves installing the Winsock Proxy Client on the web server and configuring it for the proper SSL port (usually 443). Microsoft discusses the differences between their "reverse proxy" and "server proxy" methods in MS Q article Q184030.

Summary

Any web site that does not secure transactions using SSL or something that else that encrypts data is behind the times (not to mention probably out of business). The fact is that SSL helps to add another layer of security to a networked environment, but its purpose is specific, just like

other network security products and solutions. SSL serves a purpose and seems to do it well. However, it does have problems. There are still possibilities of the so-called “man in the middle” attack, where an SSL session is hijacked. More often than not, a lot of the problems that occur are from the so-called “layer zero” of the security structure (the human layer). If a person makes an incorrect and uninformed decision (such as in the verisign case) then all efforts up through the successive layers are for nothing. It seems everything related to securing networks comes back to this human layer, one way or another. I don’t think anyone has a patch to cure that!

© SANS Institute 2000 - 2002, Author retains full rights.

Bibliography

Treuhart, Jeff Netscape Corporation “Overview of SSL 3.0”
(URL) <http://developer.netscape.com/misc/developer/conference/proceedings/cs2/sld005.html>

Netscape – “Secure Sockets Layer”
(URL) <http://home.netscape.com/security/techbriefs/ssl.html>

Hickman, Kipp E.B. AOL/Netscape Communications Corp “SSL 2.0 Protocol Specification”
(URL) http://home.netscape.com/eng/security/SSL_2.html

Freier, Alan Netscape Communications “The SSL protocol version 3.0”
(URL) <http://home.netscape.com/eng/ssl3/draft302.txt> November 18, 1996

Microsoft corporation “How to create and install an SSL certificate in IIS 4.0”
(URL) <http://support.microsoft.com/support/kb/articles/Q228/9/91.ASP> last revised May 11, 1999

Microsoft corporation “Generating a certificate request file using the certificate wizard in IIS 5.0”
(URL) <http://support.microsoft.com/support/kb/articles/Q228/8/21.ASP> last Reviewed: April 18, 2001

Morey, James Microsoft Corporation “Untangling Web Security: Getting the Most from IIS security”
(URL) <http://www.microsoft.com/TechNet/iis/technote/websec2.asp> January 5, 1999

Vijayan, Jaikumar Computerworld, Inc “Verisign Certificate snafu highlights the threat of human errors”
(URL) http://www.computerworld.com/cwi/stories/0,1199,NAV47_STO59099,00.html March 30, 2001

Microsoft Corporation “Erroneous Verisign-Issued Digital Certificates Pose Spoofing Hazard”
(URL) <http://support.microsoft.com/support/kb/articles/Q293/8/18.ASP> last reviewed March 29, 2001

Microsoft Corporation “Cannot Make an SSL Connection After Exporting and Importing an SSL Certificate”
(URL) <http://support.microsoft.com/support/kb/articles/Q261/6/55.ASP> last reviewed May 15, 2001

Microsoft Corporation “Using Server Proxy with SSL in Proxy Server 2.0”

(URL) <http://support.microsoft.com/support/kb/articles/Q184/0/30.ASP> last reviewed December 5, 2000

© SANS Institute 2000 - 2002, Author retains full rights.