The Minstrel's Articles

SSL: Whom Do You Trust?

Version 0.1 Draft

20 April 2005

Table of Contents

1 INTRODUCTION 3

1.1 Sources and Accuracy 3
1.2 Audience 3
1.3 Contact Details 3

2 SSL BACKGROUND 4

2.1 What is SSL? 4
2.2 How Does SSL Work? 4
2.3 The Chain of Trust 5

3 BREAKS IN THE CHAIN 7

3.1 TCP/IP, Encryption and other Standards 7
3.2 Your Client Machine 8
3.3 The Destination Server & Organisation 8
3.4 Background: Man in the Middle 9
3.5 Your Application Software 10
3.6 Your ISP 11
3.7 The Internet 11
3.8 The Certification Authority 11
3.9 Summary – Whom Do You Trust? 12

4 WHAT CAN I DO? 13

4.1 Client Machine 13
4.2 Destination Server 13
4.3 Connection (Man in the Middle) 13

5 SUMMARY 14

6 FURTHER READING AND REFERENCES 15

6.1 Other Analyses and Documentation 15
6.2 Organisations 15
6.3 Products and Services 15
6.4 Legislation 15

1 Introduction

We are all used to the little padlock at the bottom of our browser when we're connecting to a 'secure' Web site, and we are urged to accept that when we see it, we're completely safe. Is this true? Can we believe the assurances we hear when we view our bank accounts, make a purchase on a credit card, or even submit our Tax Return? And what of the future, when we may start electing our leadership over SSL?

In this article, I look at SSL greater depth, and the general sequence of events is as follows:

- Starting with an introduction to SSL, the reader is presented with sufficient analysis to frame the remainder of the discussion;
- Next, the concept of a 'Chain of Trust' is introduced – such a model is inherent in SSL, but here I extend it beyond the commonly accepted boundaries;
- With these concepts in mind, the article moves on to examine where 'breaches of trust' can be introduced;
- The article finishes with some ideas on how we can strengthen our confidence in the ability of SSL to protect us.

1.1 Sources and Accuracy

This article is based purely on my current knowledge of SSL, gained through personal and professional experience, combined with some parallel/paranoid thinking and a little research on the finer points.

As with all my articles, I reserve the right to be wrong, but where you do spot an inaccuracy, I would very much appreciate hearing about it! Feel free also to disagree with any of my opinions – it should make for a lively discussion!

1.2 Audience

This series of articles is written primarily with the general Internet user in mind, and this one is no exception. Given the subject matter, however, it will also be of interest to any individual or organisation providing services over SSL. It is written, as far as possible, in terms the layman will understand, and no previous specialist knowledge is required.

My experience extends only to the use of commercial encryption in the UK, and so this article should be viewed in that context. The legality and/or feasibility of some of the 'breaches of trust' mentioned herein will differ in other countries.

1.3 Contact Details

Please come and discuss this or any of my articles on the 'Front Line' discussion list:

http://lists.internetgremlin.com/mailman/listinfo/front-line

Or come and join the forums at Privacysense:

http://www.privacysense.com/

Alternatively, you can contact me privately at:

http://www.minstrel.org.uk/contact/

I look forward to hearing from you!

2 SSL Background

***Note**: SSL is a flexible encryption method, and as such can be used for a wide variety of different purposes (for example the creation of Virtual Private Networks) but this article focuses solely on the use of HTTP over SSL, i.e. creating an encrypted connection to a Web server. The concepts, however, may apply in other scenarios. Additionally, no specific attention is paid to TLS, the successor of SSL, but again the concepts will equally apply.*

2.1 What is SSL?

SSL stands for **S**ecure **S**ockets **L**ayer, and provides an end-to-end encrypted connection between two hosts (usually a client and a server). This connection is created at the IP level, and any TCP protocol (e.g. HTTP, SMTP, FTP, etc.) can then be run over the top. In this article, we focus on HTTP over SSL, the most common implementation, used worldwide to encrypt Web sessions for transfer of sensitive or important information.

SSL was originally created by Netscape and implemented in their browser, but has become a de facto standard across the Internet, adopted by all browser vendors for encrypted communications between desktop and server.

2.2 How Does SSL Work?

Without going into too much technical detail, the encryption is achieved through 'Public Key' cryptography using X.509 certificates. There will always be a certificate involved in creating an SSL connection – most commonly, there is only one certificate (at the server end), but occasionally another certificate will be used on the client for authentication. Usually, certificates are formally issued by a 'Certificating Authority', but they can be created by anyone with access to the relevant tools. As will be explained, though, a 'self-signed' certificate defeats some of the purposes of SSL connections.

The SSL certificate a server presents in this context (you can view the details in your browser by double-clicking the padlock icon, or using the relevant menu options) serves two primary purposes:

1. To create an encrypted connection between your browser and the Web server
2. To validate the identity of the server and (usually) its owner(s)

The first purpose is pretty much achieved whatever the status of the certificate a server presents – it could be expired, self-signed (i.e. untrusted), or relate to a completely different host, but the encrypted connection will still be created. With modern browsers, the strength of this encryption is generally sufficient to prevent the communications content being observed if intercepted. The primary focus of the remainder of this article, then, is on the second purpose – that of identity validation and trust.

On making a connection to an SSL-enabled Web site, your browser will check a number of things to ensure the certificate being presented is valid:

1. That the certificate is being used for its intended purpose (i.e. 'Ensure the identity of a remote computer');
2. That the 'Common Name' of the certificate exactly matches the domain of the server you are connecting to (e.g. `onlinebanking.mybank.com`);
3. That the certificate is current (i.e. today's date is somewhere between the Valid From and Valid To dates of the certificate);

4. That the certificate has been signed by a trusted party (e.g. VeriSign, Entrust.net, Thawte, etc.);
5. Depending on your browser settings and the type of the certificate, an online check may be performed to determine whether the certificate has been 'revoked' by the issuing authority – this is generally not a reliable test, however, and is switched off by default in most browsers.

If one or more of these three tests are not satisfied, the browser will present you with a warning, and ask if you wish to proceed. If you receive such a warning from your browser, you will usually be told which of the tests has failed. Some examples of why the tests may fail include:

1. It is rare for a certificate that is not intended for server identification to be presented in this context, since most server software will prevent misuse of (for example) an e-mail encryption certificate;
2. You are connecting to `onlinebanking.mybank.com`, but the certificate 'Common Name' is secure.mybank.com – this is most likely to be an administrative error at the bank, but if the Common Name is `onlinebanking.mubank.com`, the difference is more serious, but may not be as obvious;
3. The certificate expired last week – this is almost always an administrative problem – it is common for organisations, even large ones, to miss certificate renewal until far too late;
4. The certificate has not been signed by a trusted party – if this is the warning given to you by your browser, it is almost always worth looking into further (please see the following sections), as it indicates a complete failure in the server validation process;
5. The certificate has been revoked – since there is not yet a reliable means of ensuring Certificate Revocation Lists (CRLs) are kept up-to-date and accessible to browsers, this is more likely to be a problem with your browser settings than anything else.

The degree to which you see each of these as a problem is down to you, and the following sections should enable you to make an informed decision. As mentioned, the encrypted connection will still be created, but there is obviously a problems elsewhere. If the problem is with the 'Chain of Trust', you may decide not to continue your session.

2.3 The Chain of Trust

At this point, I will introduce the concept of a 'Chain of Trust'. When you open an SSL connection to a server, there are quite a number of things in which you must place your trust to provide the appropriate level of security, including:

- TCP/IP in general – the underlying protocols you use to make the connection
- Your client machine – your PC/Mac/WebTV/etc., its Operating System and its connection to the Internet
- Application software – primarily your browser, but in some cases perhaps a proprietary component running an SSL session for you (e.g. PC-based banking software)
- Your ISP – the organisation responsible for carrying your traffic across various devices to their outer boundary
- The Internet itself – from your ISP's outer boundaries, your SSL session will hop across a variety of further devices, managed by a variety of organisations, until it reaches the destination server

- The destination server – the machine providing the far end of your encrypted session
- The Certificating Authority – the organisation that is asserting the destination server's identity
- The destination organisation – the people and processes behind the server you are connecting to

Note that even this list is not exhaustive! The extent to which you trust each of these will vary considerably, and may be entirely subjective. In addition, the extent to which is able to abuse your trust and risk your security varies, and the next section focuses on where this extensive Chain may be broken. If you know where the weakest link is during any session, you can make a clear judgement on the security of your financial or personal information.

3 Breaks in the Chain

As can be seen, the security of an SSL session is dependent on our trust in a series of different things, and even the most trusting user may think twice about the idea of placing implicit trust in one or two of the items in the Chain. What is not immediately obvious, however, is the extent to which each link in the chain could be responsible for a breach of security.

This section focuses on the potential means by which each of the links in the Chain of Trust could be sufficiently weak for us to consider whether our trust is well-placed.

***Note**: this section will ignore 'application-level' threats, such as viruses and other exploits, which may bypass content filtering mechanisms by virtue of a connection being encrypted. Protection against such things is the subject of other articles in this series.*

3.1 TCP/IP, Encryption and other Standards

We all place a great deal of trust in the technology we use, without even beginning to understand the technical detail. This is simply a fact of life, as we can't possibly hope to understand everything from microwave protection and distribution mechanisms in the ovens we use to the fuel transmission processes in the aircraft we fly in. We **trust** that specialists in the relevant areas have addressed any potential problems and fixed the worst problems over time.

The same is true of our time online. We trust that there are no fundamental flaws in TCP/IP, SSL or the other underlying mechanisms that could be responsible for a serious breach of our privacy or security. In general, this is the case; such fundamental protocols are generally the subject of detailed public scrutiny before they become the basis of the software we use every day.

There have been a few concerns in recent times about the security of TCP/IP and deeper underlying protocols. Fortunately, they have generally been concerns about *availability*, rather than *confidentiality*, and the standards we rely on are under constant review and analysis to ensure we are as safe as we wish to be.

Encryption, however, whatever its strength, is not a 'silver bullet', and the following sections discuss problems even the strongest encryption cannot address.

Whatever your level of understanding of SSL or encryption in general, there is a fundamental point to bear in mind. Whatever the strength of the encryption, the size of keys, hashes, etc., if the algorithm is public, the encryption can be broken. Given sufficient time and resources, a brute force approach can (in theory) decrypt anything encrypted using a known algorithm. Therefore, it is important that SSL is used only to protect information that would be of less value to an attacker than the cost of breaking the encryption. For example:

- The encryption strength of a standard SSL connection is likely to be sufficient to encrypt your credit card details, or to protect online access to your bank accounts (unless you have a vast amount of money!);
- A commercial-grade SSL connection is unlikely to be sufficient to protect submission of your multi-billion patent applications, or to protect the lives of a community in exile, unless the information will be worthless after a short time.

Extreme examples, perhaps, but it is sometimes surprising to see the kinds of information SSL is trusted to protect.

3.2 Your Client Machine

Remember that the encrypted SSL session is created between your browser (or other local software) and the server. Anything you do with the data (or your machine does for you) outside this environment will not be encrypted. For example, if your browser caches pages retrieved over SSL, they will be saved to your hard disk unencrypted. Your keystrokes are obviously sent to the browser as you type them, and if you copy/paste text from the browser, it will be transferred as plain text, unencrypted.

If you are sitting on your own personal machine at home, and you are the only user (or you completely trust any other users), and you follow all best practice guidelines to protect your machine (regularly-updated antivirus software, spyware protection, a well-configured personal firewall, etc.) you may be reasonably safe on this front, bearing in mind the points mentioned in the following sections. However, would you access your online bank details from an Internet café? I would recommend you don't, unless you do the following (or trust the café owner to do so):

- Check for any keystroke logging devices between the keyboard and the machine
- Check inside the keyboard for any keystroke logging devices
- Check inside the machine for any keystroke logging devices
- Check the machine for any other monitoring device
- Ensure the antivirus data files are fully up-to-date
- Perform a full antivirus sweep of the machine
- Examine the running processes on the machine, and ensure you know what every task is doing
- Ensure you are not overlooked while you work
- Ensure the browser does not store any information you enter (i.e. save passwords, etc.)
- While you work, ensure you are aware of all outbound traffic, and can deny any that you do not originate
- Ensure the browser cache is fully cleared before you leave the premises

As you can imagine, a rather difficult list to complete in a public environment, some of which may not even be possible on a machine you do not fully control. If the above list seems overly paranoid, consider the value of capturing the personal details of everyone visiting a single Internet café in one day, or even the details of all the users on a single machine.

For the remainder of this article, I will assume you are using your own personal machine at home (or at work, in which case replace 'ISP' with 'Employer').

3.3 The Destination Server & Organisation

We have established that the information concerned, whatever it is, is accessible in unencrypted form on your local machine. This is also the case on the destination server and, therefore, the organisation that owns and manages it. How well do you understand and trust their internal processes?

An example of how the destination organisation can break the chain without the user being aware is frighteningly common (from recent experience). Consider the following:

- A user connects to an SSL-enabled Web site and places an order
- The server processes the order and saves the user's credit card details in a database on the server
- The server e-mails the full order details across the Internet to a distribution list for the order to be processed

There are two immediate problems here – the fact that the credit card details have been stored in a database that may be unprotected in a variety of ways, and the fact that those details have been sent unencrypted across the Internet. The user, meanwhile, is unaware of both of these points.

Server security plays a very large part in this area – the SSL connection between you and the server may be strong, but if the server is riddled with security holes, there is no way of knowing what will happen to the information you submit, whether any information you receive is accurate, or even if the server certificate has already been stolen and you are actually accessing a different server!

Of course, trust comes into play to very large extent on this point. It is likely that your bank, for example, will take reasonable care of your financial details online. Similarly, a large and reputable online bookseller is likely to safeguard your privacy, and to provide assurances to that effect. However, what of the .biz domain from which you just bought an MP3 file using your credit card?

3.4 Background: Man in the Middle

Once we have determined that we trust the local end-point and the remote one, both of which are outside the encrypted channel, and putting aside the possibility of brute-force decryption, there is one common means by which SSL security can be breached. It is known as the 'Man in the Middle' attack.

Consider a normal SSL session – there are three main aspects:

- The client
  - The connection, encrypted
- The destination server

We have established our trust in the client and the server, so what of the connection? Consider the following scenario:

- The client
  - The connection, encrypted
  - A proxy server
  - Another connection, encrypted
- The destination server

As far as our browser is concerned, as long as the proxy server has a current certificate signed by a trusted CA, everything seems fine, and no warning is generated. However, do **you** trust all the CAs that your browser does? More on this later, but the concept of a Man in the Middle attack is demonstrated – the proxy (or other) server in the middle of the second scenario has full access to unencrypted information.

There are commercial products available that do this by design (for content checking purposes), so the technique is well understood, and even commonly used.

The next few potential breaks in the chain are all related to the Man in the Middle idea.

3.5 Your Application Software

As discussed earlier, you should ensure you protect your machine to the highest degree when connected to the Internet, especially when considering submitting or receiving sensitive information, even over SSL. Other articles in this series focus on the software you need on your machine to enable this.

Let us consider, though, the software you are using to create your SSL connection. In this context, we are primarily looking at your Internet browser software (e.g. IE, Mozilla, Opera, Safari, etc.). Whatever you are using, your browser is responsible for checking several of the links in the Chain of Trust.

The various checks your browser performs in setting up an SSL connection have already been discussed. If a flaw in the browser means that any of these tests are not performed, or are performed inadequately, you may be given a false sense of assurance. Indeed, one of these tests (for intended purpose) has been fixed very recently in some popular browsers.

The tests your browser performs, however, are not exhaustive, and cannot provide 100% assurance. Here they are again (in summary):

- Certificate is intended for server identification
- Certificate is valid (dates)
- Certificate is signed by a trusted CA
- Certificate Common Name matches domain

Consider the following typical scenario:

1. User types `https://onlinebanking.mybank.com/` in their browser
2. The browser accesses the ISP's DNS service or proxy server to determine the IP address of `onlinebanking.mybank.com`
3. Having found the IP address, a connection is initiated
4. The relevant certificate checks are made for the SSL session
5. The session commences

Thinking laterally, there are several ways in which the security of this session could be subverted without the browser complaining, including:

- Step 1 – if the user followed a link (from an HTML-formatted e-mail, perhaps), they may not be accessing the correct server
- Step 2 – if the DNS or proxy lookup is subverted, a malicious destination server could be contacted

Of course, any malicious server would have to have a current certificate signed by a trusted Certification Authority, so the question then comes down to the integrity of the CA and its processes, on which more in a later section.

If the certificate were not signed by a trusted CA, the browser would generate an error and urge the user to check the certificate details. How many users, though, would recognise the following as an invalid certificate:

- Certificate is intended for Server Identification

- Certificate is valid
- Certificate is self-signed
- Certificate Common Name matches domain

I suspect that to the average user, if they even check the certificate, the above would appear perfectly OK, and they would allow the SSL session to continue. This would especially be the case if the results received from the server were as expected. Bear in mind, though, that the malicious server in question could be acting as a 'Man in the Middle', relaying requests between the browser and the real destination server, and logging the information in between.

3.6 Your ISP

There is little you can do but trust your Internet Service Provider to carry your traffic from your machine to the intended destination, but it is worth considering the level of control they could have over all your communications. Many ISPs employ filtering and proxying technology designed to be invisible to the user – at the moment, UK legislation does not require or authorise ISPs to monitor or intercept SSL-encrypted communications, but this could change.

It is already possible for organisations to transparently intercept and monitor their employees' SSL-encrypted traffic using commercial filtering software. Although I am not aware of any test case being brought to court, it would be interesting to see the outcome of a challenge under Human Rights or Data Protection legislation, and I suspect much would revolve around statements in the organisation's Internet Usage Policy.

There are several pieces of existing legislation in the UK that would not have to change a great deal before monitoring of SSL sessions could become legal or even required in certain circumstances.

3.7 The Internet

Of course, none of us trusts the Internet – it's a Bad Place. But then don't we trust it to route our traffic from our machine to the correct destination server? Indeed, we do, and generally, we can be sure this works, or we simply wouldn't get to the place we wanted to. However, when considering sensitive communications, there is some value in subversion of normal Internet communications, and we see regular security alerts warning us of this.

The recent spate of 'Phishing' e-mails, attempting to obtain users' login details for banking and auction sites is one example. Similarly, malformed hyperlinks that fool the user into believing they are visiting a different site to the one to which they are actually connected. And what of Web server security? If a small business selling cork table mats online had their Web site subverted, it would be simple for an attacker to insert a process to e-mail credit card details to themselves, or squirt them down an IRC channel.

Overall, we must be constantly aware of one fact – we have no way of knowing what is being done with information we provide to any online service once it reaches the destination server. Neither do we have any control over the security of that server.

3.8 The Certification Authority

Quite literally, the CA is the root of all trust on the Internet. It is the CA that verifies an organisation legally owns a domain name, and that they control the server an SSL certificate will be issued to. We trust their validation processes,

their worldwide industry knowledge, their integrity and, above all, that they jealously protect their signing certificates.

However, look in your browser's options (how you get there will vary from browser to browser), and the list of 'Root CAs' that your browser's vendor believes you should trust. Do you recognise all of those organisation names? And do you fully trust that each of those organisations can protect their signing certificates until the expiry date (some of which may be in 30 years' time)? Even if you recognise some of the organisations, do you trust them implicitly?

In my view, this is one of the fundamental flaws of SSL as a protective measure. Whilst we can be assured of the strength of an encrypted channel, we can never be completely assured of the means by which a certificate has been obtained. Bear in mind that there have been some high-profile cases of valid certificates being obtained through fraudulent means – but what of the lower-profile ones that may not have been discovered? Combine this with the lack of a reliable mechanism for certificate revocation, and the Chain of Trust appears quite weak indeed.

3.9 Summary – Whom Do You Trust?

Overall, if you must use electronic means to transmit or receive sensitive information (personal, financial, etc.), it is far better to use SSL than **not** to use it. But do consider whether you can place your trust in **all** the links in the Chain of Trust. If not, consider where you feel the weak link may be, and whether it can be strengthened or avoided by taking the steps described in the next section. If not, perhaps it is worth considering another mechanism. Are there fewer links in the telephone banking Chain of Trust? Or in a visit to a branch?

4 What Can I Do?

I have presented a frightening picture so far, and hopefully made the reader aware of the fact that the padlock icon in a browser is by no means the end of the story. There are ways, however, in which you can increase your confidence in the Chain of Trust. As before, there are three main areas to look at.

4.1 Client Machine

Be assured in the security of your own machine. Other articles discuss the protective measures you can employ in more detail, but the following are essential in ensuring your personal data is secure:

- Up-to-date antivirus software, preferably monitoring Web content, e-mail and local processes
- Anti-spyware software, again preferably monitoring running processes as well as anything being downloaded
- A personal firewall, and the means to ensure it is well-configured (including monitoring outbound connections as well as inbound)
- Regular monitoring for Operating System patches and Service Packs – some systems will provide you with an automatic update mechanism

Beyond this, simple best practice dictates that you should be cautious about the software you download and/or install on your machine. If you don't trust it, don't use it!

4.2 Destination Server

There is little a user can do to ensure the security of the other end of an SSL connection, or the processes in place to protect any submitted data, and a great deal more must depend on trust. Questions you may want to ask yourself, however, include:

- Does the organisation have a good reputation?
- Is there a published privacy/security policy that applies?
- Are you confident in the organisation's security awareness?
- Does the organisation have any history of security problems?

Your decision will be judgement call, but again – if you don't trust the organisation you are providing sensitive information to, don't hand it over!

4.3 Connection (Man in the Middle)

There is one means by which you can be sure you are connecting with the intended destination server, and that your SSL session is not being intercepted, and that is to check the 'Thumbprint' (Unique ID) of the server certificate. You can view this by viewing the certificate details. Unfortunately, not all organisations will publish the Thumbprint of their certificate, and so you may not have any basis for comparison.

If you are at all suspicious about the authenticity of a certificate, do not trust the connection.

5 Summary

It is not my intention to imply that SSL is an unreliable mechanism, but to point out that the existence of an SSL connection does not automatically ensure the security of the information it carries. It has its limits, and it is my aim to make the reader aware of them.

Of course, in life we must impart trust in those that handle our personal and financial information. It is important, though, that the reader is content that this trust is not misplaced.

One thing I would very much like to see change is the fact that few SSL-enabled servers publish the Thumbprint of their certificate. Without this crucial piece of information, we can never be completely sure a Man in the Middle attack isn't taking place.

Never ignore a browser warning about an SSL certificate, unless you know for certain what the cause of that warning is. Also, if possible, make a habit of looking at the certificate every time you open an SSL session.

6 Further Reading and References

[To be completed]

6.1 Other Analyses and Documentation

6.2 Organisations

6.3 Products and Services

6.4 Legislation