Global Information Assurance Certification Paper

SANS Security Essentials GSEC Practical Assignment
Version 1.2f (amended August 13, 2001)

Secure Socket Layer – Malicious Payload Delivery System

Patrick McGuire
December 26, 2001

Introduction

Many organizations eagerly embrace the principles of "defense-in-depth" by placing numerous layers of protection within their domain of responsibility. These layers include, but are not limited to: firewalls, host-based virus protection, network and host-based intrusion detection systems, proxy servers, and other software and/or hardware solutions. Generally, these protection measures rely on the inspection or identification of message content. For instance, if a virus arrives at the organization, hopefully the anti-virus signature file has been updated so the virus will be identified and isolated. Should the virus protection vendors not have time to publish a new version, the organization's active content filtering system may recognize the malicious behavior of the virus and still protect the company's valuable data assets.

The protection measures listed above rely on attacks in a form ready and able to be inspected (i.e., clear text). But what happens if the attack is encrypted? Encryption is the process of transforming information so it can't be decrypted or read by anyone except the intended recipient. Will the corporation's costly defense-in-depth measures effectively recognize encrypted harmful content and protect the organization? This research paper will answer this question after a brief introduction to the most popular method of encryption.

The most widespread method of data encryption to protect commerce on the Internet is the Secure Socket Layer (SSL) protocol. SSL relies on encryption to perform its mission of content delivery across the public Internet in a manner such that there is no unintended reading of the message. The strength of SSL hiding messages from prying eyes may also be a potential weakness.

Abridged SSL Primer

A connection between one person and any other person on the public Internet will be routed through dozens of independent computer systems. Without scrambling the message, neither the sender nor receiver can be confident that their information is secure and not inspected by unauthorized individuals. If the information is sensitive, such as credit card numbers, the transmission of clear text is generally not acceptable to most people.

In order to securely transmit sensitive data across public networks, in 1994 Netscape® developed the Secure Socket Layer (SSL) protocol. The SSL protocol has matured to

version 3.0 and is now generally accepted as the de facto standard for encrypted and authenticated communication between clients and servers on the Internet.

Encryption is a transformation process; taking a message and mixing it up so it can no longer be read or understood. This "mixed up" message is called cipher text and can only be restored to its original form by using the appropriate key. When the key is applied to the cipher text, the message is decrypted and is again able to be read. The key(s) to this encryption and/or decryption process only reside on the sender and receiver systems.

SSL is commonly bundled as part of web servers and client-side browsers. Due to the ubiquitous nature of SSL and its general public acceptability, most Internet users consider their information to be safe when the lock appears in the bottom right corner of their browser. Clearly, this feeling of safety is well founded during data transmission. So long as the SSL handshake applies key lengths of 128-bit or more, the chances of someone intercepting the transmission and decrypting the message is effectively zero. Even with today's advanced computer processing power, to decrypt a cipher text message with a 128-bit encryption key length will take many life times.

In spite of this effective level of SSL transmission security, vulnerabilities still exist before the message is encrypted and after the message is decrypted. The next section of this paper explores some of these possible exposures, followed by a section titled *Encrypted Attack* that highlights the vulnerability most associated to the purpose of this research paper.

Potential SSL Security Threats

1) *Physical Theft of a Server* – If a server is stolen, the information stored on that server (too often in clear text) is subject to inappropriate disclosure. Also, the private key used to encrypt SSL transmissions is generally stored on the SSL server.

2) *Theft of an Encryption Key* – If an SSL server is not properly hardened, the chance of compromise exists that may result in the unauthorized disclosure of the SSL server's private key. Both the hard drive and the processor are potential security risks should anyone take control of the web server.

3) *Use of SSL Short Key Length* – Short keys can be "brute forced". By attempting all possible key combinations, a 40-bit key length can be guessed in less than a day. It is recommended that servers use SSL key lengths of 128-bit or better.

4) *Acceptance of Invalid Certificates* – Although rare, Certification Authorities, such as VeriSign® or Thawate®, have issued server certificates to unauthorized entities. The most famous was when VeriSign® issued Microsoft® certificates to a person unrelated to Microsoft®. This error was quickly discovered but Microsoft® was forced to issue a patch to their Internet Explorer® software. Also, the SSL protocol relies on web browsers (Internet Explorer® and Netscape Navigator®) that have a pre-installed list of top level Certificate Authorities. If the certificate owner losses control of the private key that is

associated with these certificates, the certificate will be revoked but the browser does not have the ability of certification revocation.

5) *Man-in-the-Middle Attack* – When properly implemented, the chances of SSL session hijacking is dramatically reduced, but not eliminated (due to many people using self-signed certificates). Although technically not part of the SSL protocol, the client should match the actual domain name to the domain name in the server certificate. This step will assist with server authentication and reduce the chance of a Man-in-the-Middle attack.

6) *Encrypted Attack* – The balance of this research paper will focus on the risk of an attack launched against the enterprise network by encrypting the message content. Consider the risk of allowing active content to enter your secure enterprise network without any manner of inspection. That's what happens prior to decryption in a SSL session.

Encrypted Attack

The vulnerability and exploit potential of malicious payloads during an SSL transmission is not new; however, it is still far too available to those who wish to use it.

The following excerpt from the CERT® Coordination Center (CERT® Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests) was released on February 2, 2000[3]:

> *The malicious script tags are introduced before the Secure Socket Layer (SSL) encrypted connection is established between the client and the legitimate server. SSL encrypts data sent over this connection, including the malicious code, which is passed in both directions. While ensuring that the client and server are communicating without snooping, SSL makes no attempt to validate the legitimacy of data transmitted.*
>
> *Because there really is a legitimate dialog between the client and the server, SSL reports no problems. Malicious code that attempts to connect to a non-SSL URL may generate warning messages about the insecure connection, but the attacker can circumvent this warning simply by running an SSL-capable web server.*

This CERT® advisory goes to the heart of the problem but, in my opinion, does not properly express the severity of the vulnerability. Therefore, organizations frantically patch and fix less risky problems and ignore the criticality of code inspection after SSL decryption.

In this paper's introduction, the concept of "defense-in-depth" is referenced. Central to this concept is the added security based on layers of inspection and discovery. Certainly, there are no guarantees in data security; therefore, early discovery of an attack is equal in importance to measures that attempt to isolate, eliminate, or minimize the impact. When the attack is wrapped in an encrypted transmission as it passes

through border routers, firewalls, network intrusion detection systems (IDS), and server-based anti-virus (AV) systems, it does so without challenge.

Given a choice between port 80 (HTTP) and port 443 (HTTPS), hackers will choose 443 every time. The evil doers know that in most cases, the traffic through port 443 will not be inspected by the network IDS and it is not likely the organization is running a host-based IDS.

When a network client launches the browser of choice and visits a website, the risk potential risk event begins. To help illustrate this point, consider two corporations: Corporation A and Corporation B. Both companies have identical infrastructures - same firewalls, same IDS systems, same routers, same DMZ structures, etc. At the desktop client both corporations keep their AV software up-to-date with the most recent virus signature files, but Corporation B goes the extra measure by installing a client-based IDS on each desktop.

A trusted employee of Corporation A receives an email from a known sender. The message contains a link to https://www.accf.net/malware and a request to "click here for additional research material for the project." Thinking nothing is unusual, Employee A follows the instructions and visits the site. Perplexed, Employee A sees nothing related to any of her projects, clicks on a couple links, still sees nothing, and closes her browser. Unknown to Employee A, during the website visit a Java applet was downloaded to her local hard drive. This malicious piece of mobile code begins recording all keystrokes (including userids and passwords), stores them in a hidden encrypted file, and during the next system startup, the file is sent to the author of the Java applet (a.k.a., hacker).

Now, for Employee B, the same scenario and actions take place. However, this time the employee receives a call from a network administrator indicating he was paged by the IDS system. It seems the Java applet was blocked from system resources and not allowed to intercept keystrokes. Further forensics cleaned Employee B's system, Corporation B's network, and the incident was reported according to policy.

Both employees are senior engineers and worked on major development projects with external consultants. Corporation B (the one that caught the hack) completed their project on time, market penetration exceeded projections, and profits surpass estimates two-fold. Corporation A, however, cancelled their project when a competitor introduced a similar product. Corporation A's development project spent $2.3 million before it was summarily cancelled.

The average Internet user will respond to many inappropriate requests so long as the session is "secured" by SSL. Considering the capabilities of Java®, ActiveX®, and other mobile codes, proper user behavior should not be part of the organization's security plan. It is best to only permit users permission based on business need and inspect and scan all traffic. This requires an acknowledgment that SSL has inherent shortcomings and is only truly effective for data in transit.

Conclusion and Recommendations

The SSL experience for the corporate customer is one of assurance. Security assurance is the knowledge that their information and identity is safe. Customers are worried their credit card number will be stolen as it travels across the public Internet. Little do they know that thousands of credit cards are stolen from server storage devices and millions are stolen by physical theft, yet there are no documented cases of credit card numbers stolen by sniffing the public Internet. Nevertheless, websites will continue to exercise due diligence and masquerade as secure by declaring they are running SSL and displaying the closed lock in the corner.

This paper does not suggest SSL is unnecessary (although it gets close to that conclusion), but rather that SSL must be put in perspective. The dangers of the public Internet are not as data travels but rather when it's at rest. Any self-respecting hacker will not waste his time on sniffing out packets and possibly discovering a key piece of information when he can easily break into a server that has megabytes of data for the taking. When encrypted SSL data is decrypted, on the SSL web server or within the desktop browser, the data packets need to be immediately inspected for malicious content and/or potential suspicious behavior.

Just as most organizations maintain AV software on desktops, they should also install IDS software on clients. Corporations and governmental agencies should spend less staff time, organizational resources, and limited computing cycles on the encryption and decryption of data in transit and spend more time and resources on protecting the data at rest. Network packets need to be inspected and certified safe where ever they are decrypted. This may be at the network border or on individual desktops.

There is an emerging model of combining AV and IDS services into a single integrated product with centralized management. This will go a long way to providing the cost-based justification and hopefully the momentum necessary for general adoption by the corporate and governmental communities.

The SSL mobile code exploits and recommendations described above are supported by reputable sources. The Department of Defense (DoD) recommends the use of IDS software on the client. The DoD recognizes the limitations of attempting to recognize or block mobile code at enclave boundaries. Due to the stealth nature of encrypted connections, the DoD declares "protection against malicious mobile code must be done at the workstations."

In conclusion, effective security measures require trade-offs based on ease of use, business need, costs, and risk assessments. Certainly, SSL is effective during transmission, it's when data is at rest when most vulnerable. Know your system, know your enemy, know your data, and know when you're at risk. Information security is the sensible application of this profound knowledge.

Bibliography

1) Netscape Communication Corporation – *Introduction to SSL*
URL: http://developer.netscape.com/docs/manuals/security/sslin/contents.htm

2) Secure Sockets Layer Is Not A Magic Bullet by Rik Farrow
URL: http://www.networkmagazine.com/article/NMG20001219S0006/

3) The CERT® Coordination Center (CERT/CC), Software Engineering Institute, Carnegie Mellon University. CERT® Advisory CA-2000-02 Malicious HTML Tags Embedded in Client Web Requests
URL: http://www.cert.org/advisories/CA-2000-02.html

4) Stuart McClure & Joel Scambray
URL: http://www.inquiry.com/pubs/infoworld/vol22/issue50/001211opswatch.asp

5) Netscape Communication Corporation - Using encryption and SSL
URL: http://developer.netscape.com/docs/manuals/enterprise/admnunix/encrypt.htm

6) webreview.com - Steve Franklin
URL: http://www.webreview.com/2001/07_13/developers/index02.shtml

7) Finjan Software SurfinShield Corporate: Overview
URL: http://www.finjan.com/product_detail2.cfm?product_id=3&type=description

8) Beyond-Security's SecuriTeam.com
URL: http://www.securiteam.com/exploits/5IP000K0LI.html

9) Cross Site Scripting Info
URL: http://httpd.apache.org/info/css-security/

10) Microsoft Solutions Framework - Security Planning: Best Practices for Enterprise Security - Christopher Benson
URL: http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/security/bestprac/secplan.asp