How SSL-encrypted Web connections are intercepted

Sherri Davidoff, Contributor

09.17.2009

Encrypted Web connections are routinely intercepted by enterprises for legitimate reasons. Unfortunately, attackers can use the same methods for tapping into "secure" connections, most often because of endpoint weaknesses.

In this tip, we'll examine how enterprises and attackers intercept Web connections that are encrypted using the Transport Layer Security (TLS) protocol or its predecessor, the Secure Sockets Layer (SSL) protocol.

A digital certificate, often used in conjunction with TLS/SSL, is just a little chunk of data describing an identity -- such as the name and URL of an organization -- signed with a digital signature. Signing is a complex mathematical operation based on the contents of the certificate and the signer's cryptographic key. If the values in the certificate are altered in transit, the digital signature will not match, and a browser will display an error message.

How do you know if a digital certificate is really owned by the person you think? It's all a chain of trust. When you go to Alice's website, for example, she presents you with her certificate. Alice's certificate has been verified and signed by her friend Bob. In turn, Bob's certificate has been verified and signed by his friend Charlie. Charlie is also a good friend of yours, and you trust him implicitly. Charlie in this case represents a root certificate authority (CA) for our public key infrastructure (PKI). When you see Charlie's signature on Alice's verifiable certificate chain, you trust that Alice is who she says she is.

In real life, your Web browser comes with pre-installed, trusted root CA certificates for network infrastructure companies such as VeriSign Inc. Your Web browser will automatically trust digital certificates issued by the pre-installed root CAs. Attackers, however, can exploit this trust.

How trustworthy are digital certificates?

Nobody's perfect -- not even trusted root certificate authorities. In 2001, VeriSign mistakenly issued "code-signing digital certificates to an individual who fraudulently claimed to be a Microsoft employee" (MS01-017). According to the Microsoft Security Bulletin, "the ability to sign executable content by using keys that purport to belong to Microsoft would clearly be advantageous to a malicious user who wanted to convince users to allow the content to run. The certificates could be used to sign programs, ActiveX controls, Microsoft Office macros, and other executable content."

Digital signatures can also be forged. Last year, at the Chaos Communication Congress in Berlin, a group of researchers leveraged weaknesses in the MD5 cryptographic algorithm to create a "rogue" certificate with a valid root CA signature (Sotirov et al). This certificate had never been signed by the trusted root CA, but since it had a valid signature, it was trusted by all common browsers.

SSL interception tools

More commonly, attackers bypass TLS/SSL connections using man-in-the-middle techniques along with certificates that are generated on the fly.

Enterprises routinely intercept TLS/SSL connections. Why? Imagine you are an employee checking your Web-based personal email at work. Your company has strong incentive to peek into your traffic, to make sure you aren't leaking proprietary data or mistakenly downloading viruses. Enterprises frequently want to inspect all traffic flowing into and out of their network to prevent malware infections and protect their proprietary data.

To break a TLS/SSL connection and sniff employee traffic, enterprises often use an SSL proxy, such as ProxySG from Blue Coat Systems Inc. The SSL proxy intercepts traffic between an individual's computer and the outside world. When a user surfs to a "secure" site, the SSL proxy fetches the real Web server certificate and establishes a legitimate TLS/SSL connection between the proxy and the Web server. Then, the proxy makes a fake digital certificate on the fly, which looks similar to the Web server's certificate. It presents this fake digital certificate to the user, and sets up a second TLS/SSL session between his or her browser and the Web proxy. The user may receive a pop-up error message (and probably click it away) because the fake digital certificate is not trusted. Of course, if the organization takes the time to import the proxy's certificate as a trusted root in user Web browsers, then users won't see an error message at all. The net result? There is a "secure" TLS/SSL session between the user's computer and the proxy, and a second "secure" TLS/SSL session between the proxy and the Web server. On the proxy itself, the individual's information can be viewed in plain text. The company can then automatically search the traffic for specific keywords, or screen it for malware.

Unfortunately, attackers can use the same techniques as enterprises to intercept SSL connections. One particular free, publicly available tool makes this trivially easy. As with enterprise TLS/SSL interceptors, the attacker can use such a tool to automatically connect to the real Web server, capture certificate information, and generate a new certificate on the fly with the same information. It then presents the user with the new certificate and sets up an SSL connection. From that point on, there is a "secure" SSL session between the user's computer and the attacker, and a second "secure" SSL session between the attacker and the Web server. Another similar tool exists that removes the client SSL connection entirely, and uses social engineering techniques (such as lock icons) to trick users into thinking the connection is encrypted.

What can users do to protect against SSL interception attacks? Here are four key strategies:

1. Always use a trusted computer when surfing to sites with valuable information. If your computer is untrusted or has been compromised, then someone could have installed an illegitimate trusted certificate authority in your Web browser.

2. Consider using integrity-checking or rollback software to detect and eliminate unauthorized changes to trusted certificate authority lists.

3. Do not accept untrusted certificates. If possible, configure users' browser to automatically reject untrusted certificates.

4. Think before you click. Remember, even trusted CAs make mistakes. Train employees and home users to think critically about visiting websites.

TLS/SSL is like a nice sturdy two-by-four. Can you use it to build a secure infrastructure? Yes. Is it a secure infrastructure all by itself? No.

An entire industry has grown around SSL interception. Enterprises and law enforcement want to be able to tap into encrypted traffic just as much as attackers, so the incentives for stronger protections at the endpoints are mixed. However, with careful attention to detail, businesses and home users can detect and avoid TLS/SSL interception and bypass attacks.

About the author:

Sherri Davidoff is the co-author of the new SANS class "Sec558: Network Forensics" and author of Philosecurity. She is a GIAC-certified forensic examiner and penetration tester. She provides security consulting for many types of organizations, including legal, financial, healthcare, manufacturing, academic and government institutions.