The Hidden Costs of Self-Signed SSL Certificates

Why self-signed certificates are much costlier—and riskier—than working with a trusted security vendor

Introduction

Even when business is booming, smart companies always have an eye on the bottom line and look for ways to reduce costs. Security is not usually one of the first places companies look to trim expenses, but some IT professionals believe that they can easily lower cash outlays by eliminating third-party Secure Sockets Layer (SSL) Certificate Authorities (CAs) from the budget equation.

While spending money on SSL security for external facing sites—such as the company home page or e-commerce pages—still seems necessary, some IT professionals think that self-signed SSL certificates are an acceptable alternative for internal sites. They believe that, since only internal employees have access to servers that host internal-facing sites such as intranet portals and wikis, self-signed certificates provide adequate protection at practically no cost.

However, this kind of reasoning can backfire—badly.

The total cost of ownership (TCO) of an SSL certificate is far more than just the price of the certificate. From security hardware, to management software, to data center space and more, the costs of establishing a secure self-signing architecture can quickly add up. Not only that, but a do-it-yourself approach to SSL security may put an organization at risk—from both a technical and business perspective—in a variety of ways.

This paper explores the true TCO for self-signed SSL certificates, including a side-by-side comparison of a self-signed architecture versus working with a third-party SSL vendor. Before a company decides to use self-signed certificates, these issues deserve careful consideration.

Third-Party Verified Versus Self-Signed Certificates

When the SSL protocol debuted in 1995, the world finally had a foundation for a safe and secure way to transact business over the web. Since then, SSL has evolved to be the single most important authentication protocol used in web-based transactions.

Why is SSL necessary? Most web traffic goes over the Internet in an unencrypted form. This means that anyone with sufficient technical expertise and tools can easily “eavesdrop” on the conversations between two parties. SSL security encrypts the data moving between a web server and a browser, making it extremely difficult to intercept and decode the information.

However, SSL security goes beyond mere encryption. From a purely technical perspective, Public Key Infrastructure (PKI) does an excellent job of safeguarding data transfers, but it leaves a gaping hole in the security of a transaction. How can parties to the transaction be sure they are communicating with the proper participants? For example, if a customer is trying to purchase an expensive camera at the web site of an online retailer, the business must be able to confirm its identity to the customer. Otherwise, the customer’s credit card information is encrypted when in transit, but if the retailer’s web

site has been spoofed, all of that well-encrypted data may be sent to a cybercriminal who can easily decrypt it.

This is where the importance of third-party validation is most apparent. A certificate signed by a trusted, independent CA helps ensure the organization that owns the certificate is indeed what it claims to be.

From a technical standpoint, however, third-party validation is not required for SSL security. Organizations can "self-sign" certificates. When companies use self-signed certificates, in effect they are saying, "I verify that I am myself. Trust me."

However, to many standard web browsers such as Internet Explorer and Firefox, this guarantee is meaningless. Users who try to access a site "protected" with a self-signed certificate will usually get an error message that says the signing entity is unknown and not trusted. Not surprisingly, this kind of message scares off potential customers, partners, and other stakeholders. For this reason, few businesses will self-sign external-facing web sites. Retaining user trust is simply too important.

Internal-facing sites and servers, on the other hand, present a different use case scenario for SSL certificates. Corporate email servers, human resource (HR) portals, wikis for individual project management, software development sandboxes—these are just a few of the internal sites and servers that are often protected by SSL security. Do organizations really need third-party signed certificates when only employees access these areas? Once again, when a business uses a self-signed certificate, it asks its employees to trust that its systems are secure. Even if they will—should they?

Without the proper data center architecture and management framework—and the skilled personnel who make sure certificates are issued and signed properly—an organization simply has the ability to encrypt data between two points. Given that internal web sites are now the target of malicious attacks as much as external-facing ones, encryption is not enough.

The High Cost of Infrastructure for SSL Security

Data Centers and Physical Security

On their face, self-signed certificates are not inherently less trustworthy than those signed by leading CAs. However, reputable third-party CAs have robust processes in place to help ensure that their encryption keys, especially their highly sensitive private "root" keys, are kept safe. For these CAs, security is always a top priority: Personnel are rigorously vetted and highly trained, and these CAs have strict policies concerning where private keys are stored. In fact, if a CA wants to be approved by mainstream web browsers, these keys must be kept on non-extractible storage on smart cards.

To offer strong SSL security, a CA must also provide high-availability and failover mechanisms to prevent system failure. This helps to ensure that it can provide the proper authentication on demand whenever users need it.

Replicating this infrastructure to match the high security standards in place at leading CAs requires a number of costly components. First, an organization must have high-availability (HA) replication of the SSL system and data. A second, related requirement is that this replication must be achieved using two different secure rooms in two different data centers in two separate locations. This helps to ensure that

if one data center goes down, due to power loss or other unforeseen factors, the other will be there to provide backup authentication. Without replication across data centers, servers and browsers would not be able to complete the authentication process and vital SSL-protected transactions—such as credit card purchases at an e-commerce site or uploading new employee information to an HR portal—would stop.

Moreover, the data centers housing the SSL systems and data themselves also need to be secure, which means establishing strict physical security measures. In addition to screening employees who would have physical access to data rooms, these extra precautions would include installing key card readers to grant entry to locked areas, mounting video surveillance cameras, and even hiring security guards to do regular walk-bys. If an unauthorized person gained access to these restricted rooms, he or she could obtain the key to crack encrypted data, once again putting transactions at risk.

The basic cost for a secure, one-rack colocation data center room—with all connectivity and utilities included—can range from $1,000 to more than $10,000 *per month*.[1] Adding more racks, increasing bandwidth, or utilizing technical support can raise costs even more, often by hundreds of dollars. Not only that, but all of these expenses will double to replicate data in two data centers. Clearly, the costs of maintaining the physical infrastructure and security needed to protect SSL encryption and authentication processes are more than many businesses can afford.

Hardware Components

Although you can easily acquire free or very low-cost software that will allow you to generate self-signed SSL certificates, you will still need a hardware security module (HSM) for *each* data center to manage encryption.

An SSL HSM is a secure crypto-processor—a physical piece of hardware—dedicated to managing digital keys and for authenticating private keys in a PKI SSL protocol system. An HSM has four purposes. First, it securely generates public and private keys for encrypting transactions over the web. Second, it securely stores keys in a way that prevents them from being extracted. Third, it allows companies to manage sensitive cryptographic data.

Finally, companies use HSMs to offload application servers for both asymmetric and symmetric cryptography. This last point is increasingly relevant since the National Institute of Standards and Technology (NIST) recommends today that companies use 2048-bit RSA keys. Using these larger keys can slow down system performance, so HSMs are critical to help organizations issue keys and authenticate certificates rapidly.

An HSM is a highly specialized piece of hardware that is usually quite expensive, ranging from $13,000 on the low end to upwards of $30,000 each. Once again, for purposes of replication and achieving high availability, any SSL infrastructure needs at least two HSMs, one for each data center.

[1] Multiple sources: http://www.hostventures.com/colocationprices.html, http://www.creativedata.net/index.cfm?webid=168, http://www.datacenterknowledge.com/archives/2011/02/11/analysis-colocation-pricing-trends/

Management and Personnel

Beyond pure hardware costs, the time and expense associated with finding and training skilled professionals to manage self-signed SSL security—as well as to create policies to govern the use of SSL certificates—are also a major consideration.

Tools that allow you to self-sign certificates—such as Microsoft Certificate Authority—do not include certificate management functionality. Given that, organizations will need to plan and implement robust processes to help ensure that SSL protocols are being strictly followed. Without such safeguards, anyone could ask for an SSL certificate and receive it, which in turn would allow anyone to spoof a supposedly "secure" site at will.

First, an organization needs to carefully control who has the authority to create and sign certificates for its domains, and establish processes for ensuring that this is done according to established policies. This authority should not be given lightly, and a clear audit trail is needed in case an investigation is ever required.

Leading third-party CAs typically offer web-based applications with easy-to-use management interfaces that automate and accelerate many processes, including delegating authority for creating certificates and approving certificates for signing by the CA. Certificate Signing Requests (CSRs) must eventually be approved by someone vested with authority for a particular domain. Trusted CAs have robust automated procedures in place to help ensure that all of this occurs as prescribed.

If an organization decides to use self-signed certificates, it will need processes similar to those described above. Some businesses attempt to automate the SSL security workflow by writing custom software, but many simply attempt to manually manage the processes. This takes a considerable amount of time and effort from highly skilled and trusted staff—which may mean more highly paid senior employees.

Moreover, without the management tools and alerts that often come with certificates from a trusted CA, organizations will not be notified when certificates expire. The expiration of self-signed certificates—as well as their renewal—will need to be tracked manually, an extremely time consuming task that can take skilled personnel away from other mission-critical work. The cost of expired SSL Certificates is unacceptably high; "rogue" certificates can create an uneven patchwork of security, leading to warning messages that may negatively impact customers and internal stakeholders alike.

Retaining personnel who possess the right talent and expertise to perform all of these management tasks is expensive. According to *ComputerWorld's* IT Salary Survey 2011, mid-level security professionals earn approximately $100,000 a year. Depending on the size of an organization, the expense of hiring even one experienced worker could raise the cost of self-signed SSL security above a reasonable threshold, particularly when compared to the cost of using a trusted third-party SSL vendor.

A company could always choose to outsource infrastructure management, but this tactic not only adds additional cost, it also raises other key questions: Who is going to manage the outsourcer? What happens if the outsourcer makes costly mistakes? Adding to these concerns, if an organization's IT strategy changes, infrastructure outsourcers are notoriously difficult to replace given the dependencies that such relationships create.

Technical and Business Risks of a Do-It-Yourself SSL Security Strategy

In addition to all the "hard" costs an organization may accrue with self-signed SSL certificates, it also faces increased operational risks. Although difficult to quantify, these dangers can add up to substantial expenses if not mitigated.

Some of these risks are technical, including the potential for security breaches that can happen at both ends of the encryption/decryption process if the environment is not secured properly. In addition, it is extremely difficult to revoke certificates in unmanaged, self-signed certificate schemes.

Business risks are arguably even more serious than technical ones. Most of these perils involve building trust with customers and end users. Trust is critical for any web-based transaction, whether it's online banking or uploading a Social Security Number to an internal employee portal.

Although the true value of trust is difficult to quantify, *not* winning the trust of potential customers could be disastrous to revenues. For an internal site, like an HR portal, a lack of trust among employees—who might wonder if their salary histories and other personal data are truly secure—could impact worker morale and productivity.

Another factor to consider is the warranty protection that a third-party SSL vendor can provide. These warranties can range anywhere from $10,000 to $250,000 (or more) and are meant to compensate a business if a data breach occurs. Self-signed certificates do not provide warranty guarantees.

Finally, one of the very real risks of using self-signed certificates internally is that, over time, employees may start to ignore security warnings given by their browsers. They may even begin to add untrusted certificates to their browsers' store of trusted certificates. Not only can this potentially compromise internal networks and systems, but it can also create a lax attitude toward security across the organization and undermine general policies meant to safeguard internal systems.

Conclusion

Although many IT professionals believe that using self-signed SSL certificates can help their organizations lower security costs, the real numbers tell a different story. From data center infrastructure and physical security, to the hardware and software required for the PKI SSL system, to the personnel needed to manage the certificate lifecycle, the true costs of self-signed SSL security can become very expensive, very fast.

Both external- and internal-facing sites need strong SSL protection, and working with a reputable third-party provider is the easiest, most cost-effective way to protect customers and other stakeholders with best-in-class SSL security.