

Lee Brintle, Leepfrog Technologies

(319) 337-3877, lbrintle@leepfrog.com

Tumblr Data Protection

March 28, 2005

Abstract
SSL and https
Blind Store
Browser-based Encryption
Tumblr Downloading
Summary

Abstract

When collecting sensitive data online, such as credit card numbers, social security numbers, and other personal information, it is important to handle the information carefully and thoughtfully. Because credit card numbers have financial value and social security numbers can lead to identity theft, sites that collect this information are often targets for attacks. When successful, publicized site break-ins can cause hard financial losses for site visitors and loss of reputation for the operators.

In addition to a brief overview of common methods for protecting sensitive online information, this paper describes Leepfrog Technologies' Tumbler application. Tumbler is a highly effective mechanism for securely storing sensitive information from a website visitor without the usual expense associated with deploying a hardened environment.

SSL and https

The first layer of privacy protection is the Secure Sockets Layer (SSL), which is implemented at the “https” protocol level. Internet users are instructed to only provide sensitive information to websites that use an https connection. The browser usually provides a visual cue, such as a locked padlock, to help people know they are using such a connection.

SSL provides verification and encryption. It verifies to a visitor the identity of the site to which they are connected. Then it encrypts the data that is sent back and forth between the visitor and the site. Verification is important in stopping “phishing” attacks, where a visitor is tricked into providing sensitive information to a fraudulent site under the guise of a legitimate site. Encryption is important to prevent other people with access to the network connection – other users at the local wireless-enabled coffee shop or on the same cable modem cell – from viewing the transactions.

However, SSL, by itself, does nothing to protect the information once it reaches the web server. If the web server itself is compromised then the information collected may be compromised. Information collected prior to the compromise may be vulnerable, as well as collection of further data submissions after the point of compromise. In fact, many sites rely on the twin protections of SSL and securing the web server in order to protect privacy.

Using only SSL to secure information is a mistake. Due to the inherent nature of web sites, hostile access is always possible against a web server, and there may be new compromises and attacks which can end up compromising the web server itself. For example, when cross-site scripting attacks were first widely deployed, many existing web sites were vulnerable to the attack – the applications themselves were compromised, rather than the web server which was running the application.

Therefore, it is critical that a web site take steps beyond just purchasing an SSL certificate in order to protect their visitors’ personal information.

Blind Store

When dealing with sensitive information, one of the best practices is to immediately lock up information in a blind store – a data repository where the web server cannot access the data once submitted. For example, upon submission the application may immediately write sensitive information to a write-only database, where the application is permitted to insert information but cannot query or update information. The web server lacks the permissions – enforced by the database server – to read sensitive information from the database. Even if the web server were compromised, the attacker would be unable to access the storage database because of a lack of permissions. The attacker may be able to use the web server itself to launch attacks against the database machine, but the database machine is usually protected behind its own firewall which makes that attack avenue difficult.

This protects data that has already been collected at the time of a successful compromise from being stolen, since the data is not in an environment where an attacker can use the resources of a compromised machine to download the data. Although this is the most common type of attack, it does not protect against the more rare, but much more difficult to detect, attacker who compromises a machine and modifies the application or host environment to intentionally compromise further submissions.

This solution is also somewhat expensive, requiring at least four additional servers to implement. Because the blind store in itself does nothing to stop the compromise of a web server and the misappropriation of future submissions, it is usually further supplemented by additional pieces of hardware to provide further protections to protect the web server from compromise – and even then, the site may still be vulnerable to attacks, particularly at the application level.

Browser-based Encryption

Of course, the best way to make sure that a compromised web server cannot access personal data is to make sure the web server never sees the data in the first place. This can be accomplished by using JavaScript to ask the visitor's browser to protect the information. When the form is submitted, only the encrypted version is submitted – the original version is not submitted back to the web server.

To accomplish this, a non-symmetrical encryption method is used, where the key used to encrypt the information is different than the key used to decrypt the information. Also called public-key encryption, this technique means that although the web server knows the encryption key – after all, it was the one that provided it to the visitor's browser – it does not necessarily know the decryption key. In this manner, when the data is submitted to a potentially compromised web server, the information is encrypted and protected from the attacker – although they can view the information, it is not useful to the attacker.

From the visitor's point of view, this is all hidden and behind the scenes – it happens without the visitor's participation or understanding. In addition to performing the encryption, the Tumblr software also provides the user with visual feedback that items are being processed, such as replacing sensitive fields with "*" characters after the visitor leaves the field. Coupled with included features such as automatic credit-card number verification and credit card type detection, the visitor is reassured that the data is being handled in a safe manner. This can increase the comfort level with the visitor registration procedure.

In order for this browser-based solution to be effective, the starting page must be periodically downloaded and compared against a "known to be good" version that should be there. Otherwise, an attacker could modify the page to remove the encryption protection or to replace the encryption key with a different key, or to weaken the encryption software by replacing it with a version with known flaws. All of these attacks would defeat the purpose of protection and allow a compromised web server to begin recording sensitive information. The monitoring should be done, of course, such that the potentially compromised web server cannot detect whether a form download request is from a legitimate user or from a monitoring service.

Tumbler Downloading

In order to be useful, the encrypted data must at some point be decrypted into a processing environment, which is presumably in a more trusted environment than a web server. The Tumbler solution comes with a download and a physical decryption device which allows a legitimate user to download the information from the website on a periodic basis. The device itself contains the decryption key and is all that is necessary to decrypt the desired information. Connecting the device to a Windows-based computer's USB port begins the download process. Only the person in possession of this physical device can begin the download process.

Once the process is begun, all data, new data, or previously-downloaded "batches" of data may be selected. The information is then decrypted and written to a local disk. After downloading, the user is prompted to remove the device from the computer in order to safeguard the decryption key.

While the physical hardware device is the most common solution – it is easy to manage access and enforce download policies because access to the hardware is easily controlled – there are also download libraries available for integration into automated environments which can periodically poll the web server for new information.

Because the downloading is initiated by a trusted agent (the person in possession of the download device) and because the data is highly structured (structured data is easier to secure), the ability for an attacker to hijack the connection to covertly access the decryption key is remote. The sensitive information from site visitors is protected from the browser all the way through to the processing environment, regardless of whether or not the servers in between were compromised.

Summary

The combination of the Tumbler browser encryption libraries and the Tumbler download device provide an effective solution for protecting sensitive information on a potentially compromised web server. Compared to traditionally deployed solutions, not only is the Tumbler solution significantly less complex and less expensive, but it also protects data submitted after the server has been compromised. Together, this provides an aggressive solution to ensure the safety of personal information submitted to your website.