

How secure are your Web-based forms?

by Brian J. Moloney

Filling out forms on Web sites is commonplace. Whether it is an online job application, an online purchase, an information request - they all require Web site visitors to enter information into a form and press the submit button. Some of these - perhaps many - offer the opportunity to share sensitive information. Certainly in the case of a job application, purchase or appointment request, the form is likely to require some sensitive information.

So, how secure is that sensitive information?

Most people are familiar with the front line of form security - the secure Web page. This is the Web page that houses the form and is characterized by a lock icon appearing on the browser (also indicated in the address bar with https:// preceding the Web page address). This “secure” Web page is protected using a [Secure Sockets Layer \(SSL\) Certificate](#). An SSL Certificate costs less than \$200 per year and is one of the easiest and least expensive forms of security on the Web.

The SSL Certificate has two main functions. First, it verifies that the visitor is on the expected Web page. Second, it encrypts (scrambles) the form information when it is submitted to the Web server computer. This is referred to as “in transit” encryption because it scrambles the information while it is in transit between the user’s browser and the Web server. However, the “in transit” portion happens in the wink of an eye and the information is immediately decrypted once it is received. But what happens to the information after it is received and decrypted by the Web server?

Too often, no additional measures are taken. Security is most likely to break down after the form information arrives at the Web server.

A good indication of the security in place can be determined by the notification method when a form is received. Do you or someone in your organization receive an email when a form is received? Does the email contain the form information submitted? This is insecure.

In essence, it is taking the same information that was encrypted by the SSL Certificate when initially submitted and re-transmitting it unscrambled in an open text email message. This email may pass through any number of mail servers on its way to your inbox – each possibly saving a copy of it.

In all fairness, this is a convenient way to receive form information, especially if the information is not likely to contain sensitive information. Many general “Contact Us” forms operate in this fashion. If you do have a form that operates in this fashion and does not contain sensitive information, there are two precautions you should take.

First, make sure the form information is at least logged on the Web server as a backup. Sometimes, email can be fickle and you don’t want to lose a form submission because your email crashed. Second, the email should be sent to a role email

account– like forms@yourorganization.com – and not an individual’s email account. This role account should be monitored by or forwarded to at least two individuals. This reduces the chance that someone submits a form and receives the following email:

Re: Out of office

I will be on vacation free climbing the Alps for the next month. If your request is important please resend it to billy@yourorganization.com.

If, however, your Web site form has the possibility of containing sensitive information, a better alternative would be to store the form information within a database on the Web server.

It will still send an email notification when a form is submitted, but this time it doesn’t contain any of the form information. Instead, it simply states, "Someone just submitted a form, click here to view it," and links back to a password-protected page on the Web server. Once the proper password is entered, the form information is displayed on a Web page that is secured using the same SSL certificate.

Now we have security in place when the form is submitted and when it is viewed, but what about while it is stored in the Web server database? By default, information stored in a database is in clear text.

Granted, just because the database is in clear text doesn’t mean that the information is there for the taking. The Web server itself is most likely secured, requiring usernames and passwords to gain access. However, there may be a number of individuals with access to that server - staff at the company that hosts your site, staff at your Web development firm and other individuals with Web sites on the same server. Anyone with access to the server might be able to gain access to the database.

Of equal concern is the possibility that one or more of these individuals have a weak password - one that is easily guessed by hackers or hacker software. This makes storing the accumulated form information in a clear text database a long-term and growing security risk.

To counteract this risk, form data can be encrypted within the database. For example, certain fields can be designated as sensitive (social security number, driver's license number, credit card number, etc.) and encrypted. Now if someone gains access to the database, all of the sensitive information will be gibberish.

This database encryption requires one additional step before it can be accessed. To view information in an encrypted database, a password is required to decrypt it. This is in addition to the password required to access the Web server.

As a final measure, you should establish some rules about how long form information will be stored. For example, if you are storing appointment request information, keep it only as long as you need it – a week, a month, whatever. You can always keep the statistics, but retaining the details indefinitely is an unnecessary risk.

These days, hardly a week goes by without a news story about some database being

*Imaginary Landscape, LLC
5121 N. Ravenswood Ave.
Chicago, Illinois 60640
(773) 275-9144

<http://imagescape.com>
research@imagescape.com*

compromised. The potential public relations cost alone - let alone the cost of offering free credit monitoring services - is enough to warrant serious examination of your processes.

Make it a point to understand what happens to form information once it is received by the Web server. If you are receiving form information in email, be especially aware. Scrutinize each Web-based form to see if sensitive information might be entered. Even an innocuous "Contact Us" form can contain sensitive information if its context is a hospital Web site (sometimes people can describe specific health conditions in the comments field – even if cautioned not too).

In the case of forms, an ounce of prevention really does equal a ton of cure.

+++++

Brian Moloney is managing partner of Chicago-based Imaginary Landscape, a Web development firm specializing in the heavy-lifting technology of Web sites. He can be reached at 773.275.9144 or brian@imagescape.com.