SiteRecon

Article

Publication Date: 1/21/2007

Submitting Secure Information from Unsecured Pages

Author
Lew Newlin created the email round-robin monitoring process and is CTO of SiteRecon, a provider of internet email monitoring and web site monitoring services for business.

Abstract
Review of the practice of submitting secure information such as user names and passwords from unsecured pages.

Article
Using SSL encryption to secure information is server and client processor intensive, not to mention that the process can significantly slow the presentation of pages to your visitors. Not surprisingly, some webmasters have instituted an underhanded method to avoid the entire problem by placing sensitive information such as login/password inputs on home pages that are not SSL encrypted. The general programming concept seems to be that since the login/password information is being submitted to a HTTPS encrypted page, the data secure. Well…not so fast.

Using my sector, web site monitoring, I decided to first check and see how prevalent this practice actually is. Out of 12 sites checked, 10 (or 83%) provided login/password inputs on the home page. Clearly this practice is widely used within our sector.

The next step was to determine if the login/password information of the 10 sites using this practice actually submitted the information to an SSL enabled page. Shockingly, nine of the 10 did not. A sniffer (HTTPLook by BinaryAge Software) was used to confirm this as shown below. The results were confirmed and indeed nine companies employing this practice transmitted information in clear text across the internet.

```
POST /User/clients-login.aspx HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg …..
Referer: (blanked out to protect the guilty)
Accept-Language: en-us
Content-Type: application/x-www-form-urlencoded
UA-CPU: x86
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; .NET CLR ……
Host: (blanked out to protect the guilty)
Content-Length: 54
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: Dana-Net=CookieEnabled=YES; ASP.NET_SessionId=123
Action=Login&Name=test&Pwd=test&Submit.x=23&Submit.y=5
```

Why would a business put themselves and their customers at risk by employing a practice that clearly makes sensitive data vulnerable to a man in the middle (MITM) attack? Were the companies attempting to save a few dollars by not installing SSL server certificates? Was this just a "convenience" so customers could save a mouse click, or was this just implemented incorrectly?

Attempting to answer these questions, I first appended https://www to the 9 company's domain name to see if their home page would display using SSL encryption. Two out of the 9 returned errors indicating no server SSL certificate was installed. Two others returned errors indicating the certificates did not match the domain name. So 44% did not have SSL certificates installed or had certificate validation warnings displayed to the user. GoDaddy offers SSL certificates for $19.99 per year so it's hard to imagine this practice is driven by cost. Not a comforting thought.

Having a site visitor input his/her login/password from the home page for example, is clearly more convenient and does save a mouse click. The question becomes, how is a visitor to know if his/her information is actually being transmitted securely? Some sites reviewed actually used graphics and verbiage to indicate customer data was being transmitted securely, when in fact it is not. Short of reading code, or testing with invalid information, a site visitor would not know. This is a large blow to user confidence to save a mouse-click in my opinion.

So what about the company that actually uses this practice, and does indeed submit to a HTTPS page? Based on HTTPLook, the process is secure and the information in encrypted. If you desire to submit secure information from unsecured pages, it appears it can be done securely if implemented correctly. However in doing so, you place visitors in the unenviable position of trying to determine if your site correctly implements security. For that reason, I would strongly suggest avoiding this practice. If you're still not convinced this is a bad practice, repeat my steps with your bank, credit card companies, brokerage firm, or favorite online website. You may find yourself shocked, outraged, and an evangelist against this practice. I know I was!

Webmasters/Ezime Publishers
You are granted permission to re-publish this article on your website or publication. The only requirement is that you include the complete article, links, and byline.