Why the Padlock is your Friend

Posted At : July 29, 2005 10:10 AM | Posted By : Mark Kruger
Related Categories: Security

When submitting personal information, most users know enough to look for that little padlock in the status bar that indicates a "secure" site. Most of them believe their information to be safe. They do not know why, but it has something to do with encryption doesn't it? Actually (and surprisingly) most web developers are fairly uninformed on the topic as well.

t's certainly true that making a GET or a POST request (either one) using SSL causes the data to be encrypted. In fact, the entire request is encrypted. Before the "HTTP" part of the request is made, the server and the browser do some "handshaking" and create an encrypted session between them. Only *after the encryption is established* are any of the details of your request made. Consider this raw example an http request:

```
POST /homepage.cfm       HTTP/1.1
Content-Type: text/xml
Referer: /someotherpage.cfm
Content-Length: 912
User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows.NT.5.0)
Host: www.mydomain.com
Connection: Keep-Alive
Pragma: no-cache

....request content
      post params xml etc...
```

Huh? If the "Host" header is part of the request, how does the SSL session get created without part of the request being sent? Ah, but you forget that computers don't make requests to fun and friendly domain names like "heyyou.org" and "spankme.com". They make requests to IP addresses. When you hit go on your browser the first thing the browser considers is "Hmmmm... where do I send this request?" It looks up the IP address for the URL, makes contact and works with the server to create that encrypted session. So the server doesn't see the host header until after the session is established with the IP address.

And that, ladies and gentlemen, is why it is necessary to bind the Secure Certificate to an IP address, and why SSL only works correctly with sites that are set up with their own IP address (instead of host headers). The session is established on an IP basis *before the host header is delivered*. If you had 5 sites on an IP address and 5 different certificates - how would the server know which cert you needed? Remember - it has to establish SSL before seeing the request (before it examines the host header). Without a 1 to 1 IP to Cert relationship, the server will find it impossible to determine what to do next.

On a virtual site, the IP address receives the request and hands it off to the web server. The web server examines the request and looks at the "host header". It hands the request further down the line to the "site" responsible for that header. This is why a site bound to an IP with no other sites also listening to that IP will perform better than an IP with virutal sites on it. The web server has to play traffic cop and direct the request to the right spot - although the difference is miniscule until you get a very high volume of requests. By the way, to finish the process, the site receives the request it takes a look at the first part (verb and resource - "POST /homepage.cfm" in our example) and retrieves the resource in question. Or, in the case of an Application engine - hands the request off for further processing.

Posting from an HTTP to HTTPS

Now, if you grasped all the information above, you will understand why data posted from a plain old http page to a secured page (https) is encrypted. In my view it is still a bad idea to post from http to https, but the *data you send is indeed secure*. That doesn't protect you from cross-domain scripting attacks or other sneaky ways of getting data out of a non-ssl page. That's why Phishing is so effective. A malicious developer can actually create a page that does indeed log into your bank, but steal your username and password prior to it being submitted. If, however, you are on your bank's secure login page - where the browser matches the domain with the cert served by the server, and the page is not a mix of https and http requests - then your login is *reasonably* safe. Of course you still have to worry about keystroke loggers, physical security, shared hard drives, Packet Sniffers, DOS, viruses, worms, spyware and that late check from Bill Gates who promised you 500 dollars if you sent an email to 25 people. Now if I could just find that email from the Nigerian Oil Ministers Son....