Bob and Mary - HTML Injection Wars

Posted At : October 5, 2007 10:48 AM | Posted By : Mark Kruger
Related Categories: Security

I promised some information from the seminar on application security by **Shlomy Gantz**. This post is the first of what I hope is 3 or 4 posts unveiling some little thought about security issues when you are doing application programming. Of course we all know about cross site scripting (XSS), SQL injection attack (SIA) and acronym overload seizure (AOS). If you don't, you can find examples of the first two in part IV of my series on the **Security Pyramid**. In this article I'd like to explore what Shlomy called "HTML Injection". Now I knew that this little gem existed but mostly I thought of it as a XSS attack - where a user is able to place JavaScript designed to steal information from other unsuspecting users into a page (as in my cookie example). What Shlomy did was much harder to detect.

Let's say you have a comment application. It's designed to let users add comments in a text box. Users can add whatever comments they like and they can edit their comments. The code might look something like this:

```
<table>
<tr>
<Td>User</TD>
<td>Comment</td>
<td> </td>
</tr>
<cfoutput query="fback">
<tr>
    <td>#user#</td>
    <td>#comment#</td>
    <td><a href="#editLink#">edit</a></td>
</tr>
</cfoutput>
</table>
```

Bob and Mary both ad comments to your site. The output might look like this:

User	Comment	
Bob	I have Blue eyes and I like walks on the beach	edit
Mary	Bob has brown eyes.	edit
Mary	Bob is a Freak.	edit
Mary	Bob wears Superman underwear.	edit
Bob	My nickname is Studly Doright.	edit

Obviously Bob lives above the garage drinking Red Bull and playing Halo 3. But what Mary doesn't know is that Bob is a clever and geeky fellow who knows a thing or two about web programming in spite of the autographed picture of a sheep named Bambi hanging on his closet door. Bob is able to edit his own entries so he does the following.

- On entry number 1 he adds " and<!--" to the end of his comment.
- On entry number 5 he prepends --> to the front of his comment.

The resulting HTML looks like this:

```
<tr>
    <td>Bob</td>
    <td>I have Blue eyes and I like walks on the beach and<!--</td>
    <td><a href="">edit</a></td>
</tr>
<tr>
    <td>Mary</td>
    <td>Bob has brown eyes.</td>
    <td><a href="">edit</a></td>
</tr>
<tr>
    <td>Mary</td>
    <td>Bob is a Freak.</td>
    <td><a href="">edit</a></td>
</tr>
<tr>
    <td>Mary</td>
    <td>Bob wears Superman underwear.</td>
    <td><a href="">edit</a></td>
</tr>
<tr>
    <td>Bob</td>
    <td>-->My nickname is Studly Doright.</td>
    <td><a href="">edit</a></td>
</tr>
```

The HTML renders like this:

User	Comment	
Bob	I have Blue eyes and I like walks on the beach and My nickname is Studly Doright.	edit

Check the source code. Mary's mean spirited (and probably true) comments about Bob are neatly commented out. In fact, if Mary complains to the web master who goes and looks in the database he's going to have a hard time figuring it out. Why? Because he's likely to look for Mary's comments and see that they exist. That could have him scratching his head until he looks at the rendered HTML. As an aside, you need to be constantly looking at the rendered HTML. Don't trust your eyes - view source is your friend.

Anyway, using this technique a clever (and internally wounded and suffering in spite of his new pocket protector) Bob could wreak havoc on the site and control how virtually everything displays.

The Fix

As always the fix is to control user input *on the server side* and validate everything that you plan on displaying later. Each input should submit to some form of rules to make sure that it *is* what you believe it is *supposed* to be - especially on a web site where anonymous users produce content for others to see (like forums for example). It is not enough to merely strip out any "<script>" tags. If you want to allow HTML then think about the rules carefully and make sure you have included some scrubbing code to handle Bob.