

奇安信攻防社区 – 对 SEMCMS 再一次审计

奇安信攻防社区 – 对 SEMCMS 再一次审计

前言 最近在社区看文章的时候, 发现了这一篇文章:

<https://forum.butian.net/share/291> 感觉应该还有洞, 所以通读了下代码, 所以有了这篇文章。 # sql 注入 1 在 / Include/web_inc.php 中我们可以...

最近在社区看文章的时候, 发现了这一篇文

章: <https://forum.butian.net/share/291> (<https://forum.butian.net/share/291>) 感觉应该还有洞, 所以通读了下代码, 所以有了这篇文章。

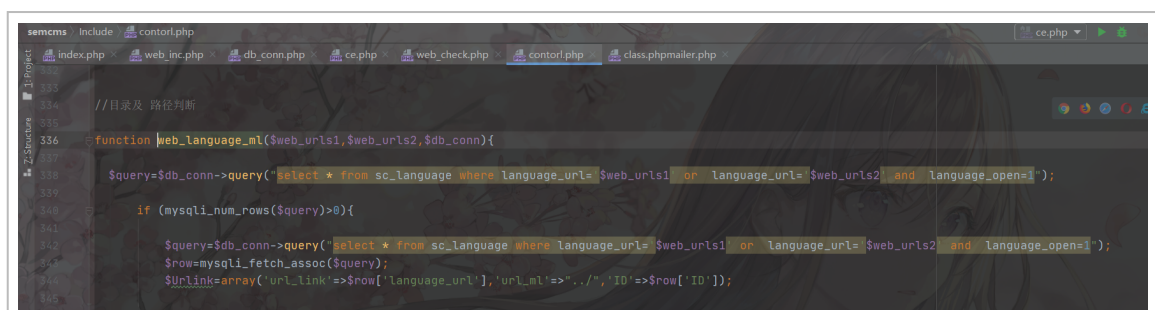
在 / Include/web_inc.php 中我们可以发现这一串代码:



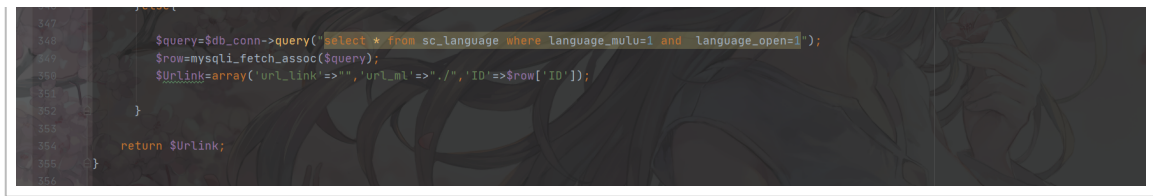
```
semcms / Include / web_inc.php
46 $weballmu=str_replace( search: "\\ ", replace: "/", getcwd()); //处理 windows下的路径
47 $webmu=explode( delimiter: "/", $webmu);
48 $weballmu=explode( delimiter: "/", $weballmu);
49 $webmu=str_replace( search: "/".$webmu[1]."/", replace: "/".$weballmu[1]."/", $webmu);
50 $weburldir=str_replace($webmu, replace: "", $weballmu);
51 $weburldir=str_replace( search: "/Template/" . $webTemplate . "/Include", replace: "", $weburldir);
52 if ($weburldir==""){ $weburldir="/"; }
53 $web_urlm=$http."/".$_SERVER_NAME.$weburldir;
54 $web_urls=$_SERVER["REQUEST_URI"]; //获取 url 路径
55 $web_urls=explode( delimiter: "/", $web_urls);
56 $urlml=web_language_ml(@$web_urls[1],@$web_urls[2],$db_conn); // 大写的问号。
```

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-d3729ad17b26ba16c2840d2533b840f953bb654a.png)

我们可以看到先让 `$web_urls` 接收 `$_SERVER["REQUEST_URI"]` 的值, 然后分割后, 传进 `web_language_ml` 函数中, 所以继续跟进该函数:

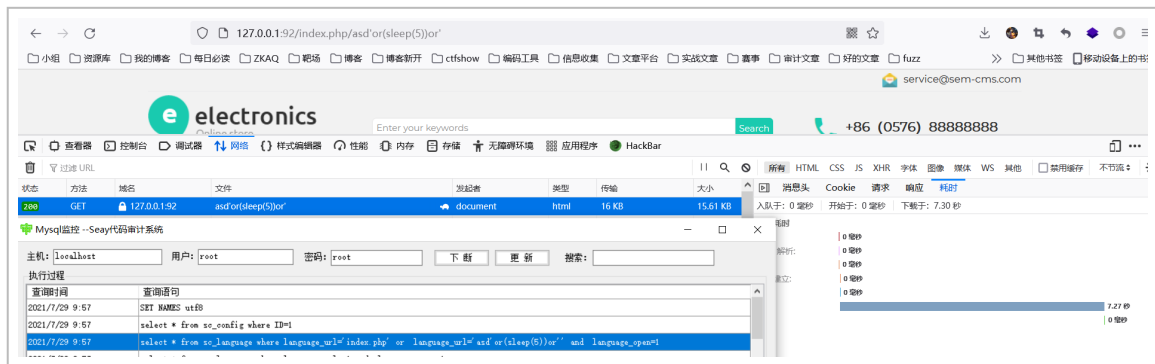


```
semcms / Include / contorl.php
335 //目录及 路径判断
336 function web_language_ml($web_urls1,$web_urls2,$db_conn){
337
338     $query=$db_conn->query("select * from sc_language where language_url= '$web_urls1' or language_url= '$web_urls2' and language_open=1");
339
340     if (mysqli_num_rows($query)>0){
341
342         $query=$db_conn->query("select * from sc_language where language_url= '$web_urls1' or language_url= '$web_urls2' and language_open=1");
343         $row=mysqli_fetch_assoc($query);
344         $urlink=array('url_link'=>$row['language_url'],'url_ml'=>"/".$row['ID']);
345     }
346 }
```



(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-cb3a717950ed93a132e5012107cb464e4a3a6eb0.png)

很明显的 sql 注入, 而且是用 `$_SERVER["REQUEST_URI"]` 接收值, 而代码只过滤了传入的参数, 对这个是没有对其过滤, 所以直接注入就 ok 了



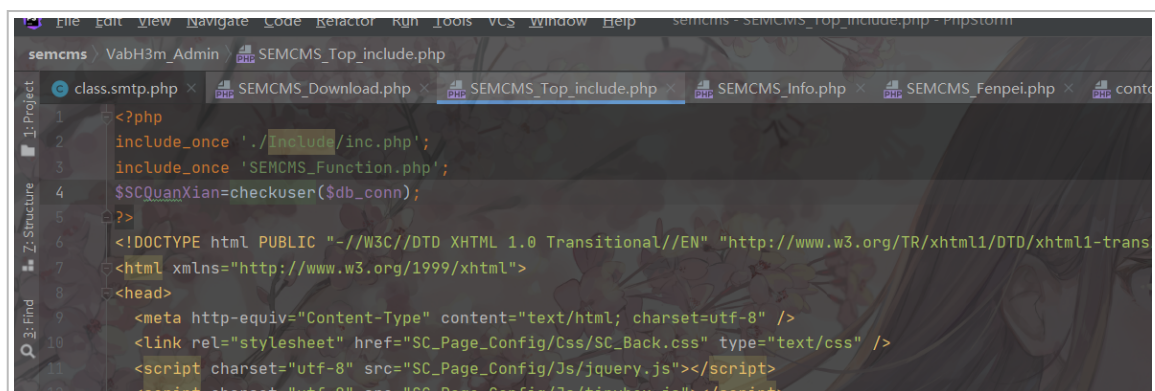
(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-33dcec87919320ed09bb7a13e76fa42d6099e04e.png)

这里用的是时间盲注, 但是 `$_SERVER["REQUEST_URI"]` 接收值不会解码, 所以空格要简单的绕过一下, 不然传入的就是 %20, 后面的注释直接 `or'` 就好了!

如果想要访问后台的文件, 其所有文件都存在于这段代码:

```
<?php include_once 'SEMCMS_Top_include.php'; ?>
```

所以跟进查看, 发现内容是:



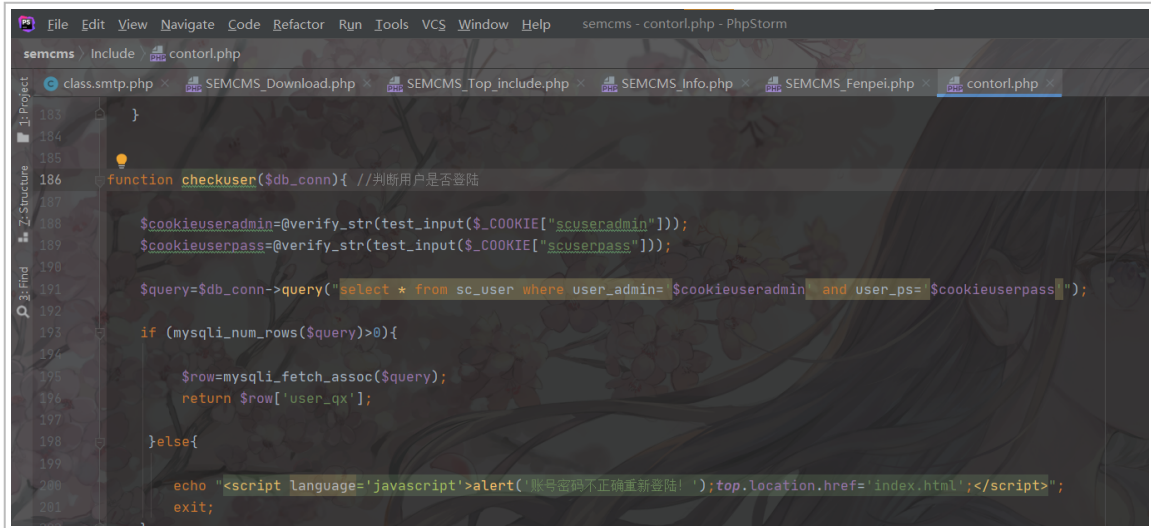
```

13 <link rel="stylesheet" href="../../Edit/themes/default/default.css" />
14 <link rel="stylesheet" href="../../Edit/plugins/code/prettify.css" />
15 <script charset="utf-8" src="../../Edit/kindeeditor.js"></script>
16 <script charset="utf-8" src="../../Edit/lang/zh_CN.js"></script>
17 <script charset="utf-8" src="../../Edit/plugins/code/prettify.js"></script>

```

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-f43efefd8357c218116fc5ae5b6762299ff421c1.png)

发现会使用 `checkuser()` 函数来进行判断, 所以继续跟进:



```

186 function checkuser($db_conn){ //判断用户是否登陆
187
188     $cookieuseradmin=@verify_str(test_input($_COOKIE["scuseradmin"]));
189     $cookieuserpass=@verify_str(test_input($_COOKIE["scuserpass"]));
190
191     $query=$db_conn->query("select * from sc_user where user_admin= '$cookieuseradmin' and user_ps= '$cookieuserpass'");
192
193     if (mysqli_num_rows($query)>0){
194
195         $row=mysqli_fetch_assoc($query);
196         return $row['user_qx'];
197     }else{
198
199         echo "<script language='javascript'>alert('账号密码不正确重新登陆!');top.location.href='index.html';</script>";
200         exit;
201     }
202 }

```

(https://shs3.b.qianxin.com/attack_forum/2021/07/attach-35d088a06988a9829a8fdcfbe5e33c9621d1ec9.png)

发现它是通过使用 cookie 中的值, 方入 sql 中进行执行, 这里先把

`verify_str` , `test_input` 两个函数的过滤内容奉上:

test_input:

```
function test_input($data) {
```

verify_str:

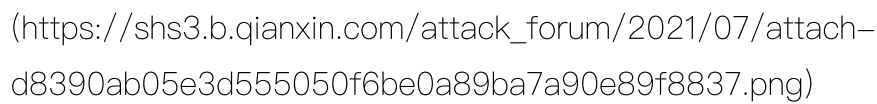
```
$data = str_replace("%", "percent", $data);
```

inject_check_sql:

```
$data = trim($data);
```

可以看到, 过滤了很多东西, 正常的 sql 注入似乎没办法搞, 但是, 这里是想要登录后台, 所以我们完全可以不用注入得到密码, 可以直接想办法构造使该 sql 语句正确就好了, 所以这里的想法是构造出 `'xxx' or 1=1` 这种形式, 所以直接上 payload:

```
$data = stripslashes($data);
```



这个 cms 感觉漏洞挺多的, 最近有些事, 没看完, 有兴趣的小伙伴可以全篇通读一下, 不出意外还能找到一些洞...