《漏洞利用及渗透测试基础》实验报告

姓名：付文轩 学号：1911410 班级：19级信息安全单学位班

**实验名称：**

复现ROP CTF题目

**实验要求：**

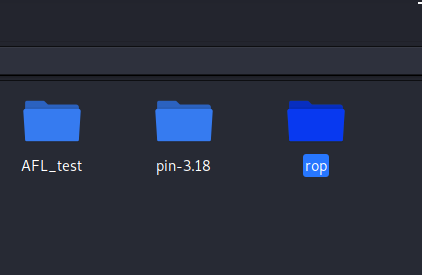
复现13.2.2中的ROP漏洞利用方式。

**实验过程：**

1. 实验准备

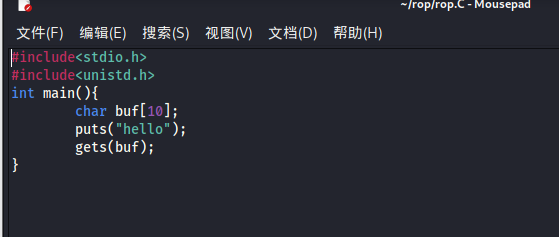
本次的ROP实验主要是在Linux的平台下进行，故可以考虑在kali的环境下进行实验

首先准备好kali，创建rop使用的目录和对应文件



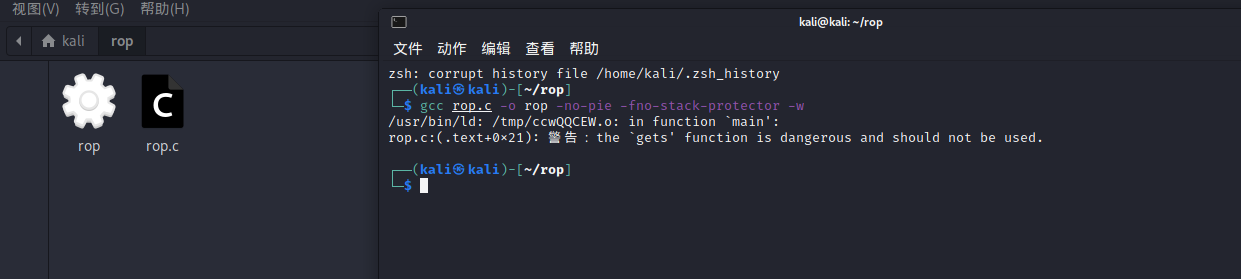


对应源码为：



之后使用gcc命令进行编译，具体命令为：gcc rop.c -o rop -no-pie -fno-stack-protector -w

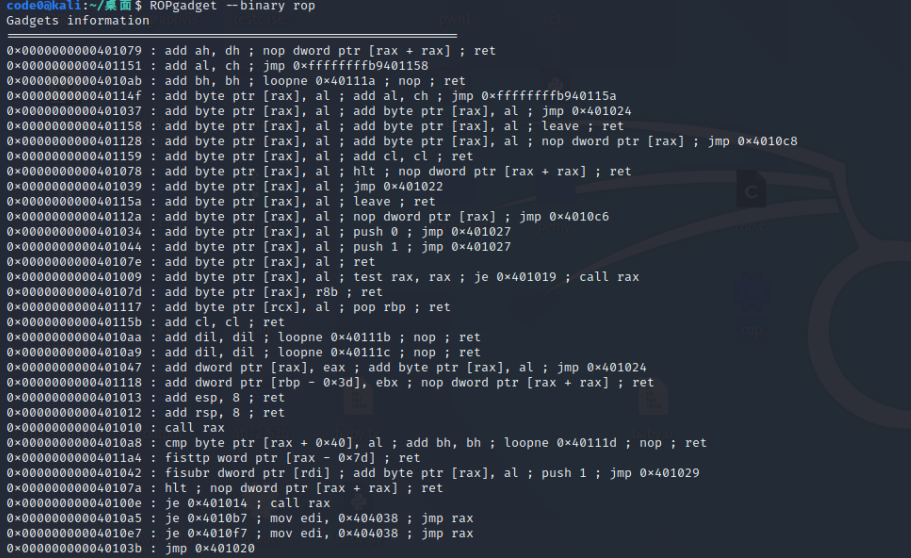
运行编译命令后，得到编译文件

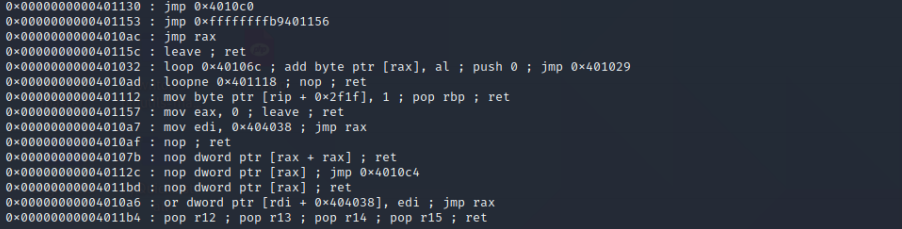


甚至从编译的反馈里就能看出来gets函数是一个有危险的、不安全的函数，不应该使用

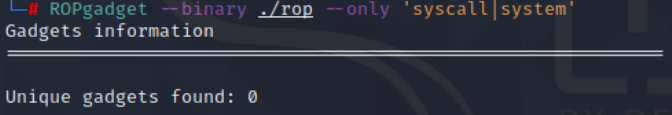
1. 寻找可以利用的gadget

使用ROPgadget工具查看是否存在有可以利用的gadget，在Terminal中输入指令：ROPgadget --binary ./rop，得到如下图反馈



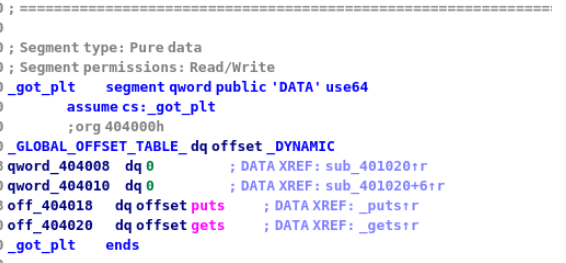


根据上图的反馈可以看出，rop里能使用的gadget太少了，好像是没有看到有syscall等类似的执行系统调用的gadget，再尝试运行指令：ROPgadget --binary ./rop --only ‘syscall|system’



根据这个found：0就更加确信了在这里面是没有可以利用的系统调用的gadget了。

直接找系统调用这个方式不行，那么可以换一种思路：寻找他调用的动态链接库里的gadget，而只要有动态链接库里函数的调用，就会在GOT表中留下相应的记录，因此可以利用ida工具查看，得到ida内截图如下：



可以看见使用了gets和puts函数，同时经过计算偏移地址不难找到这两个函数都是来自libc库中（同时这里也映证了在实验准备的时候，编译rop.c出现的对gets函数使用的警告）

puts函数的功能是打印指定位置的数据，那么这个函数也就能被利用：利用这个函数打印出我们想要知道的地址上的内容。根据前面的got表可以看出来puts实际存放的地址是0x404018



再之后就是用puts函数实际存放的地址减去在libc里的存放的偏移值，就能够得到libc是实际地址，然后就能对libc里的gadget进行利用。

那么接下来的思路就相对来说比较简单了：找到libc的基址->找到libc中system函数的入口地址->执行system函数获得权限

同时注意到puts函数是不能直接通过修改rdi中的值进行利用的，通常需要修改这类寄存器里的值，可以将pop指令结合起来使用。故再次使用ROPgadget得到如下指令：

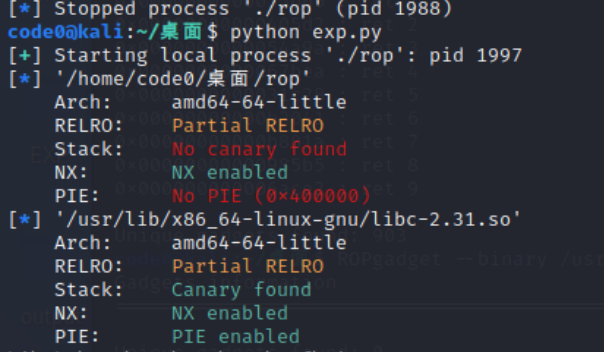


可以看出这个指令的地址是0x4011bb

根据上面的思路、结合书中信息可以得到如下代码：



运行脚本后得到：



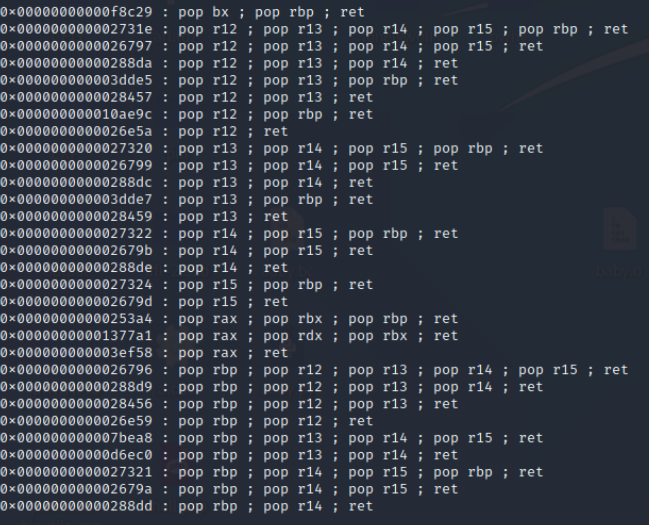
至此成功找到了获得libc基址的方法

1. 实现获取shell

根据上面的方法得到了libc基址后，可以调用execve(“/bin/sh”,0,0)来执行任意指令。查询系统调用表可以发现execve函数系统调用号为59。

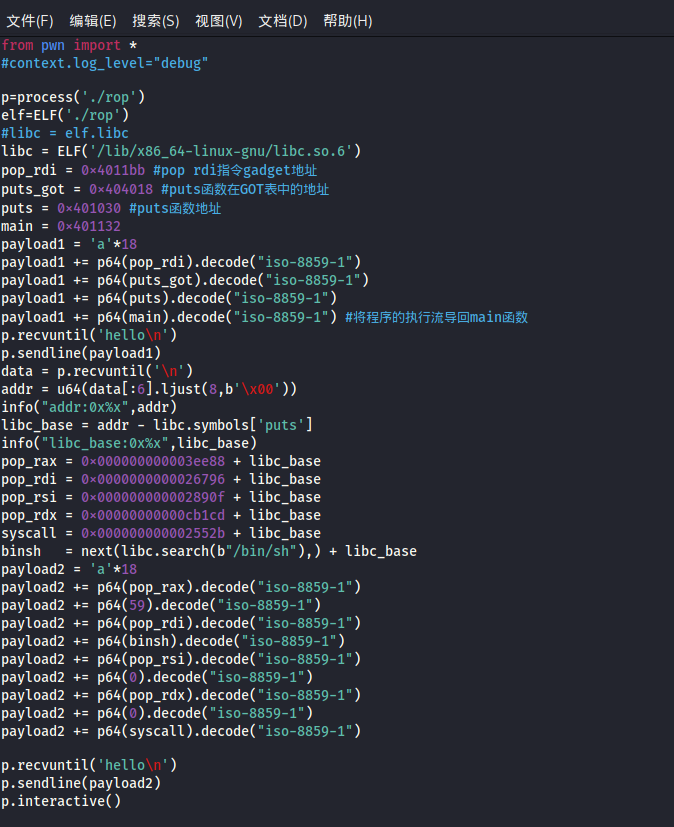


在执行execve()之前还需传入参数59，也就是需要把rax设置成59。rdi设置为“/bin/sh”，rsi和rdx设置为0。“/bin/sh”在libc中可以找到，不需另外构造，其他赋值寄存器的gadget也可从libc中找到：

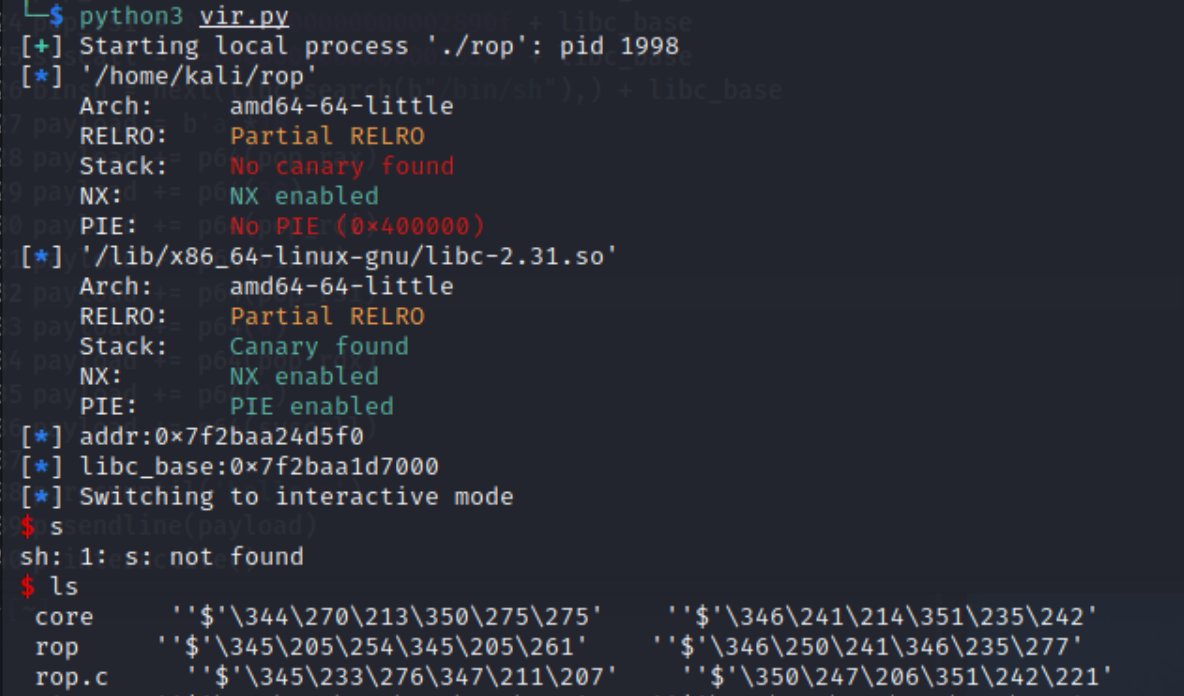


同时还需要注意，运行一次gets函数就只能有一次payload的机会，如果需要运行两次（第一次得到libc，第二次运行execve），一种方法是直接修改代码，让他有两个gets，每个执行我们想要的目标，但是这种实在是太麻烦；还有另一种办法就是在执行payload的时候，在payload的末尾加上main函数的地址，让他重新回来再执行一次main函数就行了，也就是再执行了一次gets函数

根据上述思路、结合课本，得到如下代码：



将上述脚本运行后，并在Terminal中执行ls指令后可以看见：



成功获取了shell权限，实验成功

**心得体会：**

这就是CTF的难度吗，真的和之前的完全不一样，之前的实验自己是有思路并且很多都能自己写出来代码的，比较难的就是shellcode编写、API自搜索、插桩、反序列化里的payload代码编写，但是至少是有思路的，结合书上的一些提示也能写出来一些自己的代码，但是这一次的，一开始是一点点攻击的思路都没有，后来根据书上的提示有了一些思路，但是写攻击脚本的时候又基本上可以说是卡住了，不知道该怎么写。

不过通过这次的实验，我对ROP链的构造也是熟悉了一些，也熟悉了一下ROP在CTF中具体是怎么作用的，学会简单的应用ROPgedget工具，对安全这个方向也更加有兴趣了