[LAB'5]: Hook 函数

- 实验环境：Ubuntu 16.04 i386
- 实验工具：GCC、GDB
- 实验目的：熟悉Linux环境下的Hook函数

1. roshambo.c

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

#define NOTICE "[R]ock-[P]aper-[S]cissors!"
#define DRW 0
#define LOS 1
#define WIN 2
#define LONGGEST_ITEM "Scissors"

char RPS_result[3][3] = {DRW, LOS, WIN, WIN, DRW, LOS, LOS, WIN, DRW};
char RPS_item[3][sizeof(LONGGEST_ITEM)]= {"Rock", "Paper", "Scissors"};

char round(char player, char npc)
{
    printf("Your %s vs. npc's %s\n", RPS_item[player], RPS_item[npc]);
    return RPS_result[player][npc];
}

char play(char X)
{
    unsigned char choice = -1;
    switch(X){
        case 'R':
        choice = 0;
        break;
        case 'P':
        choice = 1;
        break;
        case 'S':
        choice = 2;
        break;
        default:
        return -1;
    }

    return round(choice, rand()%3);
}

int main(int argc, char *argv[])
{
    char player_input = -1;
    int count = 0;

    srand(time(0));

    puts(NOTICE);
    while((player_input = getchar())!=EOF){
        if(player_input == '\n')
            continue;

        char res = play(player_input);

        if(res < 0){
            puts("[-]Bad input\n");
            printf("%x\n", player_input);
            return 0;
        }
        else if(res == 1){
            puts("GM:u lose\n");
            count = 0;
        }
        else if(res == 0){
            puts("GM:draw game\n");
            count = 0;
        }
        else if(res == 2){
            puts("GM:u win\n");
            count += 1;
        }

        if(0x100 <= count){
            puts("GM: WINNER WINNER CHICKEN DINNER!\n");
            return 0;
        }

        puts(NOTICE);
    }


    return 0;
}
```

2. evil_libc.c，Hook函数srand使得随机数发生种子已知

```
#include <stdio.h>
#include <string.h>
#include <dlfcn.h>
#include <time.h>

typedef void(*SRAND)(unsigned int seed);

#define EVILSEED 0xdeadbeef
```

```
int srand(const char *s1, const char *s2)
{
    static void *handle = NULL;
    static SRAND old_srand = NULL;

    if( !handle )
    {
        handle = dlopen("libc.so.6", RTLD_LAZY);
        old_srand = (SRAND)dlsym(handle, "srand");
    }
    printf("hack function invoked. Seed going to be : 0x%X\n", EVILSEED);
    old_srand(EVILSEED);
}
```

3. 编译代码

```
make
```

4. 运行并观察实验结果

普通运行

```
./roshambo
```

有Hook的运行

```
LD_PRELOAD="./evil_libc.so" ./roshambo
```

Q1：在Hook的条件下，设法赢得游戏（生成一个可以赢得游戏的输入）

Q2：Linux下动态链接与静态链接的区别?

Q3：LD_PRELOAD的作用?

Q4：试分析延时绑定(Lazy Bind)的过程（Tips，_dl_runtime_resolve)

Q5：尝试Hook其它函数以达到赢得游戏的目的