

**课 程 实 验 报 告**

**课程名称： 计算机系统基础**

**实验名称： 二进制程序分析**

**院 系 ：计算机科学与技术**

**专业班级 ： 本硕博202001**

**学 号 ： U202115674**

**姓 名 ： 姚晨炫**

**指导教师 ： 许向阳**

**2022 年 10 月 28 日**

**一、实验目的与要求**

通过逆向分析一个二进制程序（称为“二进制炸弹”）的构成和运行逻辑，加深对理论课中关于程序的机器级表示各方面知识点的理解，增强反汇编、跟踪、分析、调试等能力。

实验环境：Ubuntu，GCC，GDB等

**二、实验内容**

**任务5.1** 二进制炸弹拆除

作为实验目标的二进制炸弹（binary bombs）可执行程序由多个“关”组成。每一个“关”（阶段）要求输入一个特定字符串，如果输入满足程序代码的要求，该阶段即通过，否则程序输出失败。实验的目标是设法得到得出解除尽可能多阶段的字符串。

为了完成二进制炸弹的拆除任务，需要通过反汇编和分析跟踪程序每一阶段的机器代码，从中定位和理解程序的主要执行逻辑，包括关键指令、控制结构和相关数据变量等等，进而推断拆除炸弹所需要的目标字符串。

实验源程序及相关文件 bomb.rar

bomb.c 主程序

phases.o 各个阶段的目标程序

support.c 完成辅助功能的目标程序

phases.h support.h 公共头文件

**阶段1： 串比较 phase\_1(char \*input);**

要求输出的字符串(input) 与程序中内置的某一特定字符串相同。提示：找到与input串相比较的特定串的地址，查看相应单元中的内容，从而确定input 应输入的串。

**阶段2：循环 phase\_2(char \*input);**

要求在一行上输入 6个整数数据，与程序自动产生的 6个数据进行比较，若一致，则过关。提示：将输入串input拆分成 6个数据由函数 read\_six\_numbers(input, numbers) 完成。之后是各个数据与自动产生的数据的比较，在比较中使用了循环语句。

**阶段3：条件分支 phase\_3(char \*input);**

要求输入一个整数数据，该数据与程序自动生成的 一个数据比较，相等则过关。提示：在自动生成数据时，使用了 switch … case 语句。

**阶段4：递归调用和栈 phase\_4(char \*input);**

要求在一行中输入两个数，第一个数表示在一个有序的数组（或者binary search tree）中需要搜索到的数，该数是在一定范围之内的；第二个数表示找到搜索数的路径（在树的左边搜索编码为二进制位0，在树的有边搜索编码为二进制位1）。

**阶段5：指针和数组访问 phase\_5(char \*input);**

要求在一行中输入一个串，该串与程序自动生成的串相同。在生成串和比较串时，使用了数组和指针。

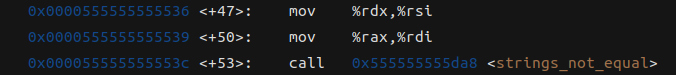
**阶段6：链表、结构、指针的访问 phase\_6(char \*input);**

要求在一行中输入6个数，这6个数是一个链表中结点的顺序号（从 1 到6）。按照输入的顺序号，将对应链表结点中的值形成一个数组。若该数组是按照降序排列的，则过关。

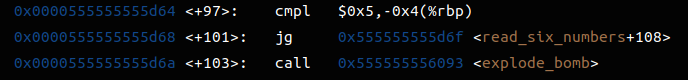
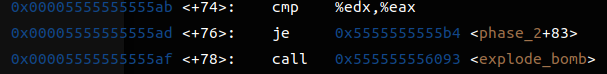
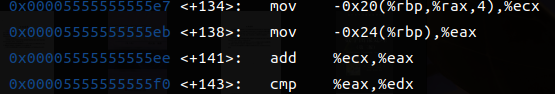
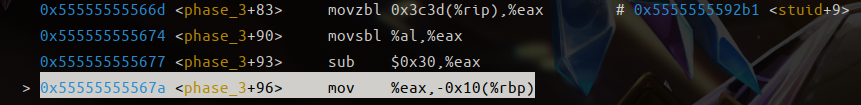
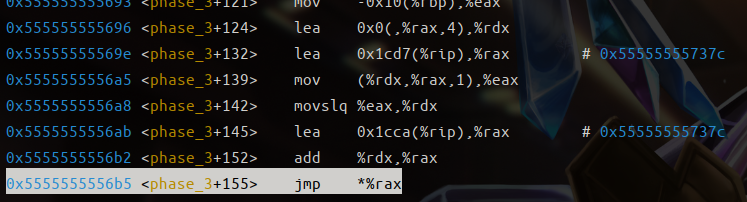
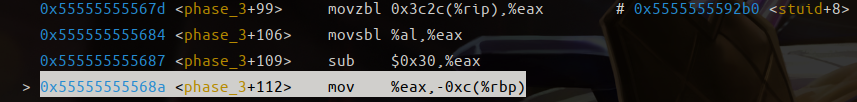
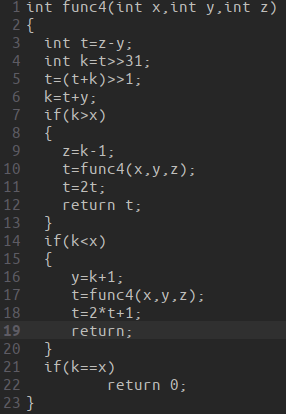
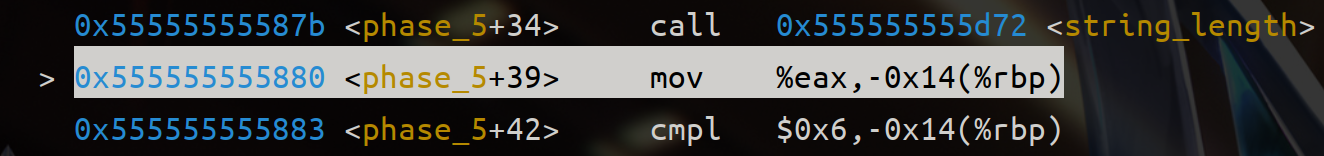
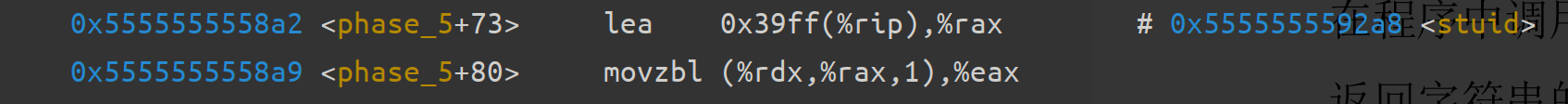
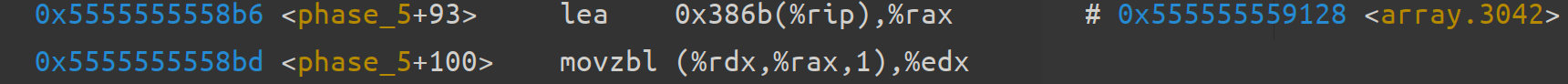
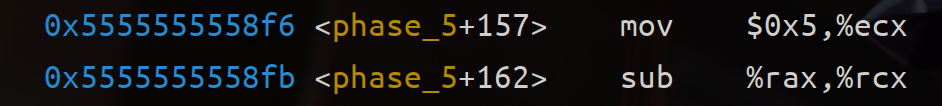
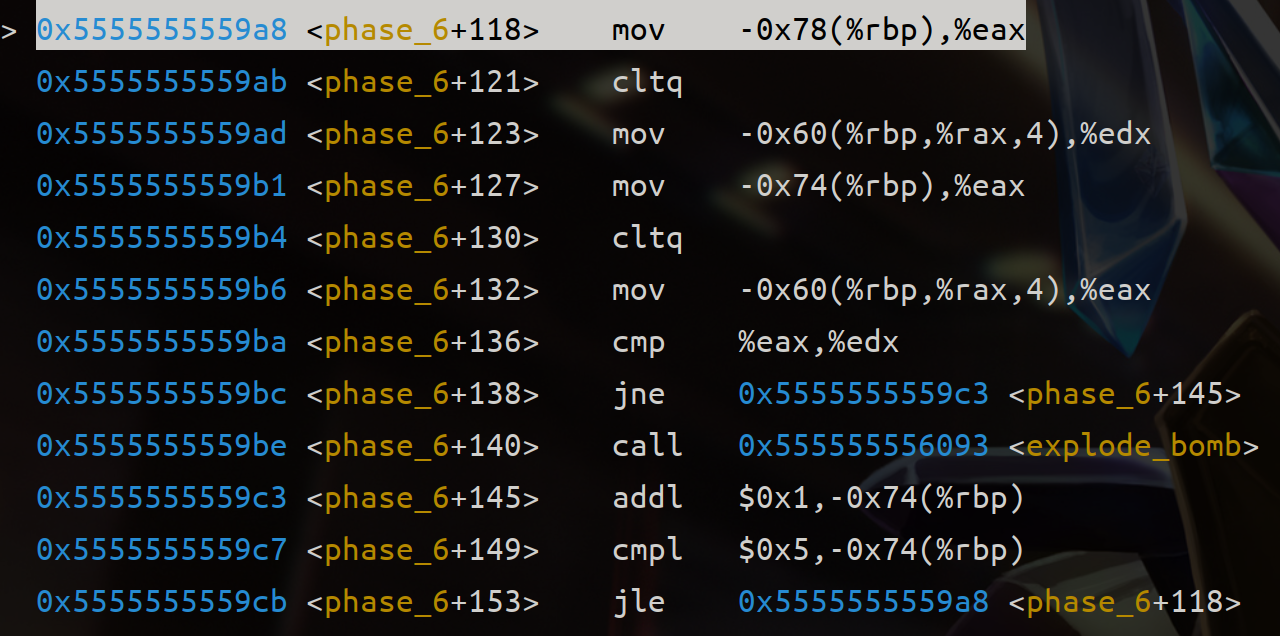
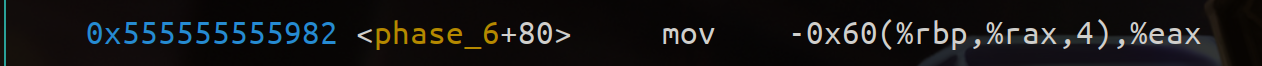
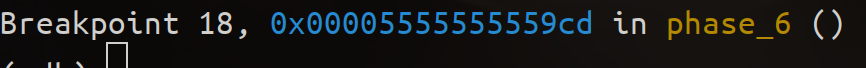
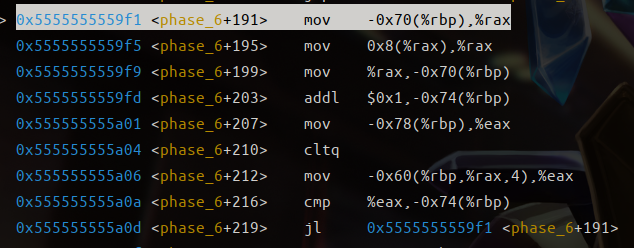
**三、实验记录及问题回答**

**（1）实验任务 5.1 的实验记录**

（1）在phase\_1函数处设断点 观察phase\_1函数

使用disass查看反汇编代码时发现phase\_1函数调用了strings\_not\_equal函数，给函数传入参数的两个寄存器是

其中rsi寄存器存的是相对应的密码，rdi寄存器存的是用户输入的数据，使用i register rsi来查看rsi下存放的地址地址为0x7fffffffda8d，使用x/s 0x7fffffffda8d将对应地址下的数据按照字符串的格式打印出来，得到第一个数字炸弹的密码

1. 在phase\_2函数中有一个函数名为read\_six\_num 这是读入用户输入的数字并判断的代码，在read\_six\_num函数，函数会判断操作者有没有输入5个以上的数据，否则就会引爆炸弹函数的第一次cmp操作在比较时候设置断点查看edx寄存器中存储的数据为4,得知密码的第一位是4，使用相同的方法可以得知密码的第二位是7，此时紧接着的语句是jmp 目标位置是phase\_2+156句，，此处为循环判断的位置，来判断-0x24（%rbp）下的数字是否小于等于5 ，同理跳到下一个cmp语句可以找到密码的第三位是9 ，其中 - 0x24(%rbp)中存放的是循环的次数，然后这段机器指令每次在前一次的密码上加入本次循环的次数来构成下一次的密码，所以7+2=9,因此不难得出接下来的密码是9+3=12,12+4=16,16+6=21,这样就得出了第二关的答案。
2. 首先程序读取处学生学号的第九位，存入栈-0x10处然后读取学生学号的第八位置存放在栈-0xc位置处在这里决定跳转的位置，将对应位置的数取出来，加上学号的第8位，43+7=50在调试的过程中，此处的cmp语句中eax的值为我们输入的值，在rdx中储存着用来比较的值，为50。
3. 在调试的过程中，rbp-0x18和rbp-0x1c下存放着用户输入的密码，在接下来的phase\_4+78处,程序讲第一个数字与立即数0xe进行了比较，说明第一个输入的数字是十六进制的e，也就是14, 然后程序调用了func4,传入参数(14,0)，将输出与7比较,然后再与用户输入的第二个参数比较,来观察是否于第二个参数相等。关于func4,用c语言写出其代码为输入（14,0,14）输出结果为7。
4. 在程序中调用了string\_length汇编函数，该函数在运行过程中不断比较地址下的内存是否为0,然后循环来返回字符串的长度。然后将返回的长度与6进行比较，如果长度相同就继续，长度不相同就引爆炸弹。可以看出炸弹的长度为6。接着程序利用来读取数字学生学号中的数字，然后利用这个数字在已经定义好的数组中取数，，利用(%rdx,%rax,1)来取到寄存在rax寄存器中的rdx位置的元素。最后将其存放起来。位置在栈中的位置。接下来函数利用者一段代码将5减去循环的次数，讲用户输入的代码与倒置之后的存放在栈中的数组进行比较。所以密码为存放在栈中的密码的倒置。cckmoi
5. 程序调用read\_six\_num来读入六个用户输入的数字，在0x60(%rbp)处存放。判断输入之间有没有相等的数字（循环好几次），如果有就引爆炸弹。这里表示已经比较好了。通过比较用户输入的顺序来在链表中查找元素，将查找到的node放在栈的8\*rax-40号的位置。如此循环5次。 通过观察rdx寄存器可以看到node 6 的数据为528，node 5的大小为 824, node 4为121，node 3为322 ，node 2为907,node 1为 281，然后使用降序讲他们排列一下过关。



我的答案为

U202115674

cacbbfghi

4 7 9 12 16 21

50

14 7

cckmoi

2 5 6 3 1 4

**四、体会**

本次实验的难度相对较大，但是我觉得十分有效的锻炼了我对于汇编语言的理解，看懂大断的汇编语言真的是一件很过瘾的事情，在本次实验里包含了函数调用，栈，循环，数据结构等汇编比较核心的内容，在做实验的过程中能明显感觉到自己的理解变得越来越深刻了。