

**课 程 实 验 报 告**

**课程名称： 计算机系统基础**

**实验名称： 机器级语言理解**

**院 系 ：计算机科学与技术**

**专业班级 ： 计卓202201**

**学 号 ： U202215322**

**姓 名 ： 濮澍**

**指导教师 ： 许向阳**

**2024年 4月 7 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

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);**

要求输入两个整数数据，与程序中给定的数据比较，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和一个串，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1 的实验记录**

实验记录分为六个阶段分别拆弹

1. 串比较

在0x0000000000401853 <+83>: call 0x40154d <strings\_not\_equal> 前打断点 观察rsi中字符串（rdi中是自己输入的字符串）

1. 循环

在反汇编代码显示的几个je机器代码处打断点，每次都观察寄存器rax和rdx的关系，让输入与rdx（有几个是rax）中值相同

1. 条件分支

观察反汇编代码，第一个必须跳过的于输入有关的地址上打断点，观察rax和rdx。可得第一个数3

0x0000000000401997 <+109>: je 0x40199e <phase\_3+116>

函数后半段有一堆跳转到phase\_3+254地址的指令，因此直接看phase\_3+254附近代码 发现第二个弹坑

0x0000000000401a28 <+254>: mov eax,DWORD PTR [rbp-0x14]

0x0000000000401a2b <+257>: cmp DWORD PTR [rbp-0x10],eax

0x0000000000401a2e <+260>: je 0x401a35 <phase\_3+267>

这时需要观察eax和rbp-0x10的值，获得答案

1. 递归调用

0x0000000000401b15 <+79>: mov eax,DWORD PTR [rbp-0x1c]

0x0000000000401b18 <+82>: cmp eax,0xe

0x0000000000401b1b <+85>: jle 0x401b22 <phase\_4+92>

第一个 数与0xE = 14 有关

0x0000000000401b43 <+125>: cmp eax,DWORD PTR [rbp-0x10]

0x0000000000401b46 <+128>: jne 0x401b50 <phase\_4+138>

对比了eax和[rbp-0x10]。因此观察rax和栈中内容，获得第二个答案7

1. 指针和数组

本题中对输入数据做了处理and eax 0xf --> 即输入的字符的ASCII码只有后四位有效。 而根据代码中的循环，发现是通过低四位代表的数字作为索引，向一个特定数组中取值获得一个需要的数组。需要的数组根据观察代码末尾call stringnotequal前的rdi和rsi获得bruins

再观察代码，发现特定数组地址

0x0000000000401bc6 <+90>: lea rdx,[rip+0x27e3] # 0x4043b0 <array.3096>

取出数组内容

(gdb) x/s 0x4043b0

0x4043b0 <array.3096>: "maduiersnfotvbyl8\003"

因此要输入的字符与0xf后应该获得索引 13 6 3 4 8 7

所以查ASCII表后得到solution “MFCDHG”

1. 链表

此题具体代码没分析明白，但取巧了一下，因为反汇编代码给出了node1的地址

0x0000000000401c40 <+37>: lea rax,[rip+0x27b9] # 0x404400 <node1>

发现只需打印3个字 就可获得该结点的 数据域 编号 指针域

记录数据 根据指针域获得下一个node地址 再反复进行此操作五次 进行数据排序

获得结果。

**（2）拆除炸弹的过程中关键操作**

主要的操作就是

1. 反汇编phase\_x
2. 根据反汇编代码的地址打断点(b \*0x...)
3. 在反汇编关键地址打断点（对比，跳转）以观察寄存器和栈(i r ; x/<nxw> address或x/s观察字符串)
4. 根据寄存器和栈值获得常量答案

**四、体会**

了解了gdb的使用，了解了如何对地址打断点，比较浅显的了解了机器代码怎么阅读，函数调用方法。掌握了一定从大段机器代码中阅读重要信息的能力，但对整体代码的理解还是不够。