

**实验报告**



**题目： 拆解二进制炸弹**

**班 级：**

**学 号：**

**姓 名：**

**学 院： 计算机学院**

**2020年 11 月 15 日**

一、实验目的  
1.理解C语言程序的机器级表示。  
2.初步掌握GDB调试器的用法。  
3.阅读C编译器生成的x86-64机器代码，理解不同控制结构生成的基本指令模式，过程的实现。

1. 实验环境
2. SecureCRT（10.120.11.12）
3. Linux
4. Objdump命令反汇编
5. GDB调试工具
6. 积分榜（http://10.120.11.34:19240/scoreboard）

**报告邮寄（最迟时间：2020年11月22日晚23：59）：**

**大二班（1班，16-19班）：**[**clavicle@bupt.edu.cn**](mailto:clavicle@bupt.edu.cn)

三、实验内容

登录bupt1服务器，在home目录下可以找到Evil博士专门为你量身定制的一个bomb，当运行时，它会要求你输入一个字符串，如果正确，则进入下一关，继续要求你输入下一个字符串；否则，炸弹就会爆炸，输出一行提示信息并向计分服务器提交扣分信息。因此，本实验要求你必须通过反汇编和逆向工程对bomb执行文件进行分析，找到正确的字符串来解除这个的炸弹。

本实验通过要求使用课程所学知识拆除一个“binary bombs”来增强对程序的机器级表示、汇编语言、调试器和逆向工程等方面原理与技能的掌握。 “binary bombs”是一个Linux可执行程序，包含了5个阶段（或关卡）。炸弹运行的每个阶段要求你输入一个特定字符串，你的输入符合程序预期的输入，该阶段的炸弹就被拆除引信；否则炸弹“爆炸”，打印输出 “BOOM!!!”。炸弹的每个阶段考察了机器级程序语言的一个不同方面，难度逐级递增。

为完成二进制炸弹拆除任务，需要使用gdb调试器和objdump来反汇编bomb文件，可以单步跟踪调试每一阶段的机器代码，也可以阅读反汇编代码，从中理解每一汇编语言代码的行为或作用，进而设法推断拆除炸弹所需的目标字符串。实验2的具体内容见实验2说明。

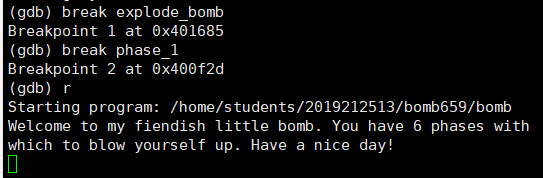
四、实验步骤及实验分析

一.准备阶段

1. 用tar xvf bomb659.tar解压炸弹压缩包

2.另开一个服务器连接，使用objdump指令反汇编，并储存至bomb.txt文件，使用vi编辑器打开该文件待查。

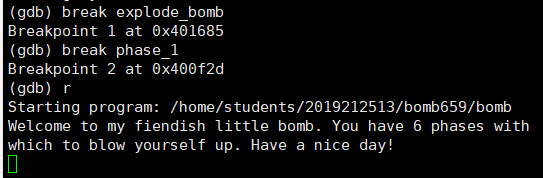
3.在答题用的服务器上进入bomb659文件夹，使用gdb指令进入bomb进行调试，由于bomb本身就是可执行文件，因此不需要再进行gcc编译

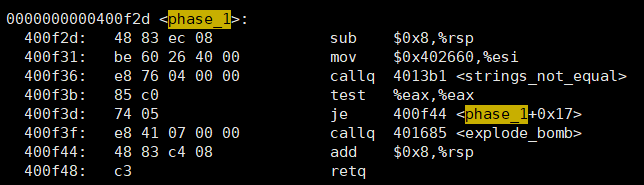
4.预先在explode\_bomb位置设置断点，防止爆炸

5.观察反汇编文件的结构，发现每一步都是输入一个字符串，程序判断是否匹配。由此开始答题。

二.phase\_1

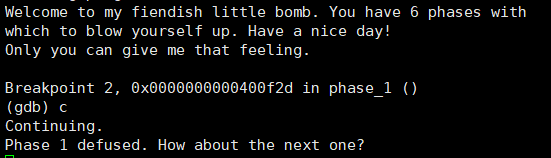
1.在phase\_1入口处设置断点并执行至断点处

2.观察phase\_1的反汇编文件，看到如下代码

C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\}MVJ5DO~IK(7_I@YY}54ENG.png由mov $0x402660,%esi一行得知，输入的字符串需要与$0x402660地址中储存的字符串相等，因此在gdb输入如下指令

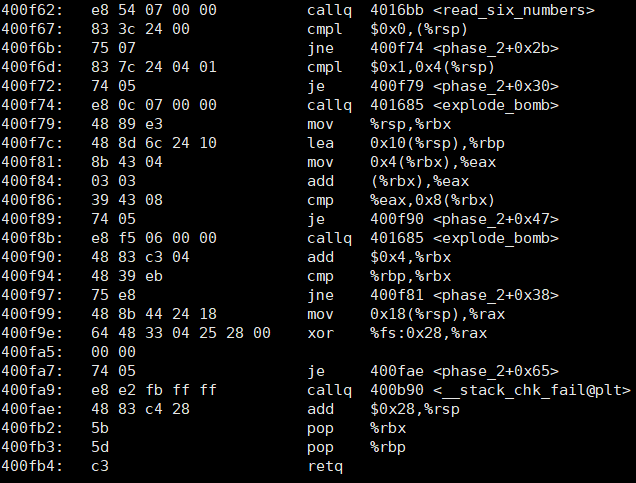
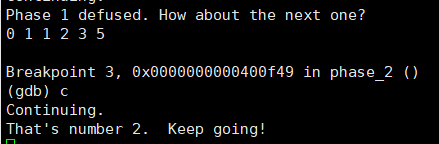
由此得知，第一关的答案是Only you can give me that feeling.

3.运行文件，输入这句话，再点击单步运行，发现过关成功。

三.phase\_2

1.观察反汇编文件，phase\_2首先调用了一个read\_six\_number函数，可以推测阶段二让我们输入的字符串是6个数。并且可以看出(%rsp)即为第一个输入数，每个输入数占4字节。

然后逐行观察：400f67和400f6d可以看出，第一，二个数为0，1，随后则是每个数都等于前两个数之和的循环，即斐波那契数列，推得6个数是0 1 1 2 3 5。

2.输入0 1 1 2 3 5，通过第二关。

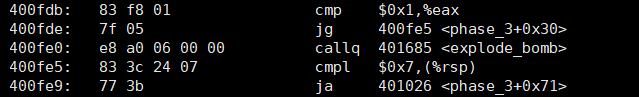
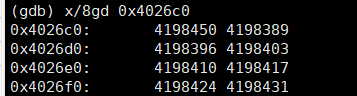
四.phase\_3

C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\JQ`$8TT9W8U6[T6Y%$MF213.png1.首先观察到此关调用了scanf函数，并且将0x40295d送入该函数

此打印这个字符串，得到：

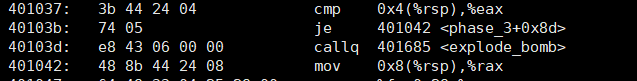
由此可知本关需要输入C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\0H6`}V)MEJ0HL~CO)0S_RBR.png2个数字。

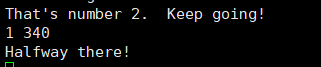
2.接下来将eax中的返回值，也就是第一个输入数与1，7比较，若不在1到7之内则引爆炸弹。

3.发现接下来是switch语句的跳转表，使用x/8gd查看跳转表的内容

C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\XO[_1F$LYXCRW2Y42W[S)`U.png4.观察代码，4198389即为16进制的0x400fee，查看内容，发现是

将0x154与第二个输入数进行比较，相等则过关。

输入C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\8C5Y)JEPG_ZDHWI$AITYOF3.png1 340 发现过关。

2-7也有类似的解，本关共有6个输入均可过关，为

1 340

2 133

3 591

4 355

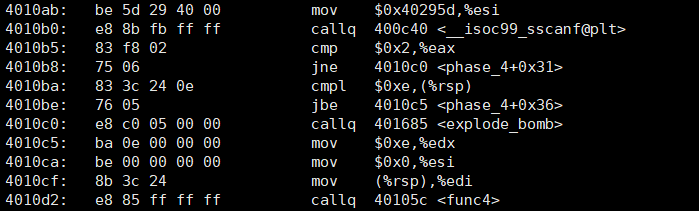
5 209

6 589

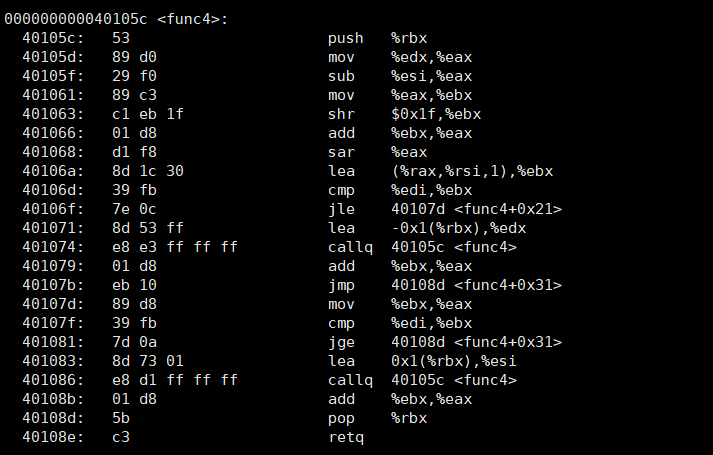
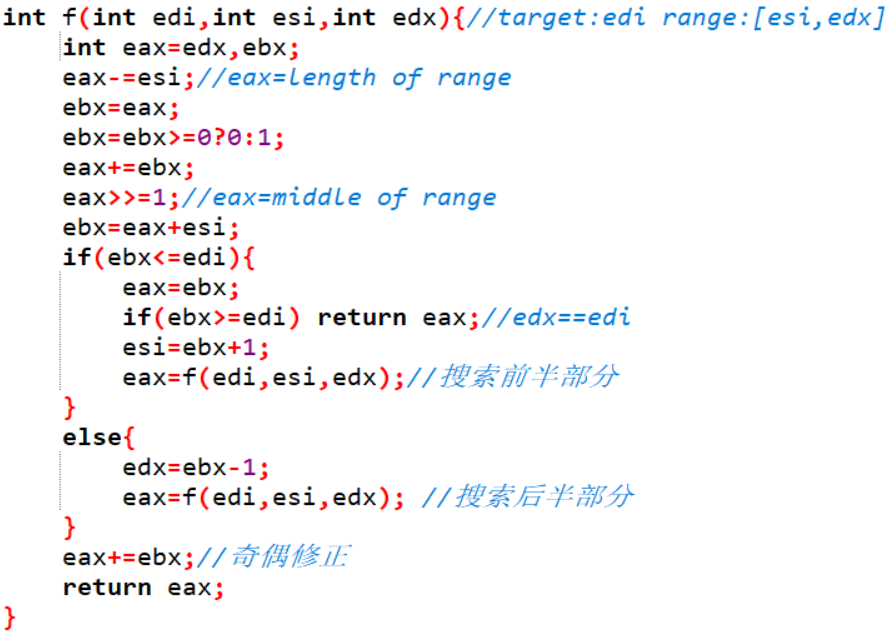
7 997

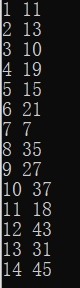
五.phase\_4

1.同上，发现此关需要输入两个整数。先将第一个数与0xe比较，若大于则爆炸，否则进入函数func4

且传入func4的3个参数是第一个整数，0x0，0xe。

2.查看func4函数

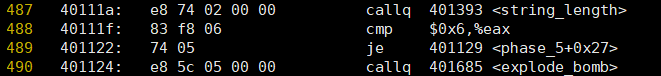
我把func4翻译为了c语言程序，并且加上了简单的注释，如下所示：

依次输入1-14，得到结果为：

在phase\_4中，发现过关要求是返回值=19，第二个数等于19，因此对照得到输入为4 19，输入，通过。

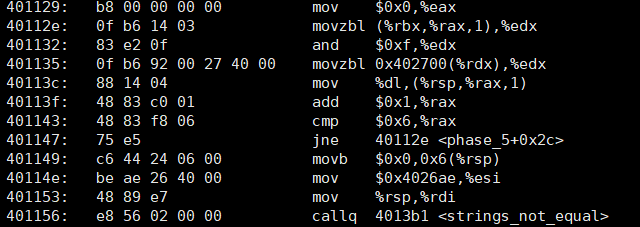
六C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\TT%CMV7VUVP8MPP0~~@}MAN.png.phase\_5

1.读入一个6长度的字符串，否则引爆炸弹：

2.可以看出，经过操作后的字符串将与C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\F38X@DXL@$1(5L~[3DLQ(UT.png0x4026ae中的字符串进行比较，因此先打印出该地址中的值

同时，发现每次需要取0x402700位置的地址，也打印出该字符串

C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\OEO4WLNU`_(()80V8FHGZ$1.png

3.观察本段代码，发现操作为对输入的字符串对做&0xf运算，即取低四位，然后取0x402700位置的相应字母，使之最后与目标oilers匹配。

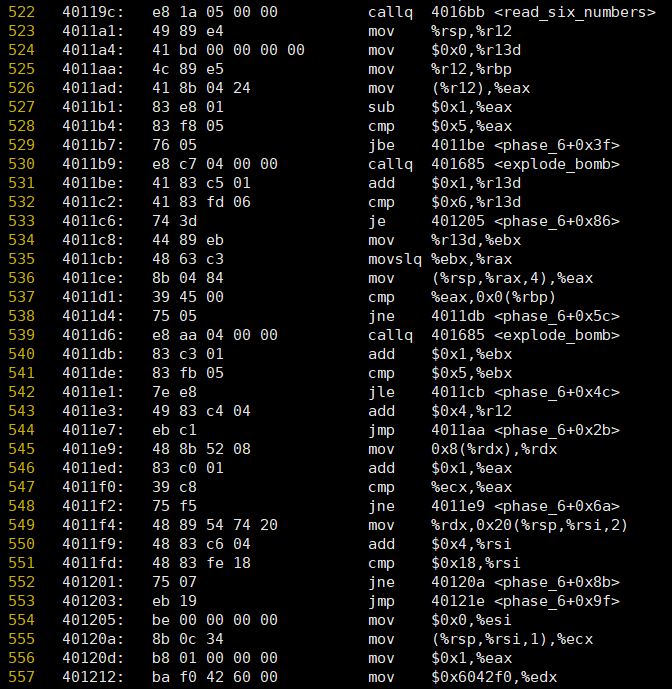
对比发现，应该取位于10，4，15，5，6，7位置的字母，在ascii码表中查到低四位为这6个数字二进制表示的字符，其中一组为jdoefg（结果不唯一）。

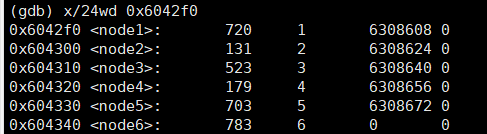
4.输入jdoefg，通过。

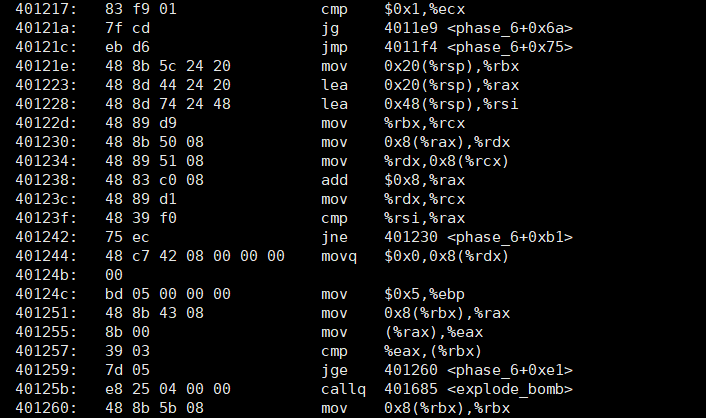
七C:\Users\1\Documents\Tencent Files\1193351983\FileRecv\MobileFile\Image\IYU`~`@N7SXO_[$TIAWH[VY.png.phase\_6

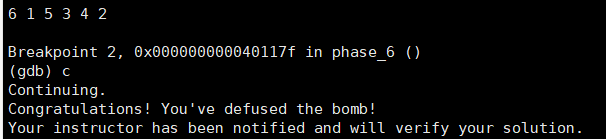
1.首先查看反汇编程序，和以上几题一样，本题也是读入6个数字。

随后紧接着的程序的含义是判断6个数字是否有重复，若没有重复则通过，反则炸弹爆炸。

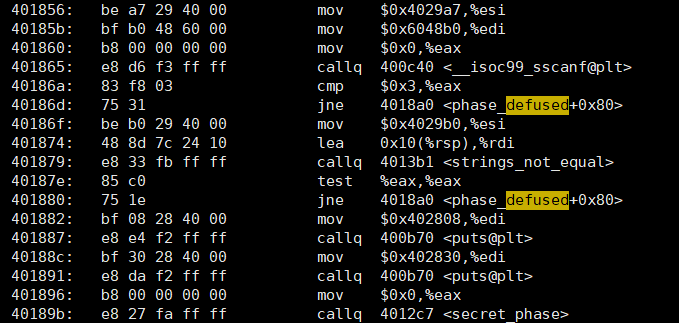
2.随后，我发现此处移入了一个地址0x6042f0，经过几次尝试，利用x/24wd可以输出完整内容如下，显然这是一个链表。

3.继续读程序，发现接下来的代码块的意思是循环取出链表第x个元素，x为输入数字，若后一个数字比前一个数字大则爆炸。观察上图打印的链表，发现从大到小的顺序是6 1 5 3 4 2

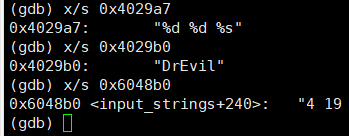
4.输入6 1 5 3 4 2 通过。

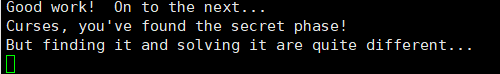
八.secret\_phase 隐藏关

1.进入隐藏关

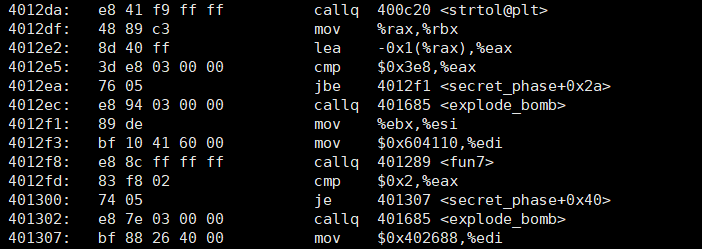
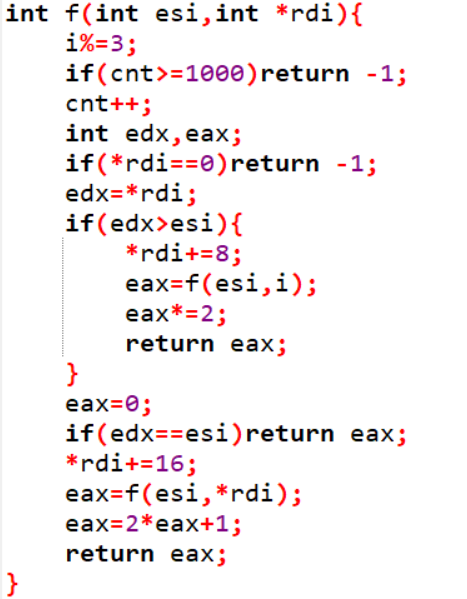
找到过关完成时调用的defused函数的反汇编代码，分析代码，前一段是判断是否通过了全部6关，然后发现一段代码，为对两个地址指向的代码进行比较，若相等则进入隐藏关。

设置断点，依次打印这三个位置的值，发现：

此处应该是在某一个连续输入2个int值后再输入一个字符串DrEvil。打印被比较处的字符，发现是第四关的答案，因此在第四关后再输入DrEvil，成功进入隐藏关。

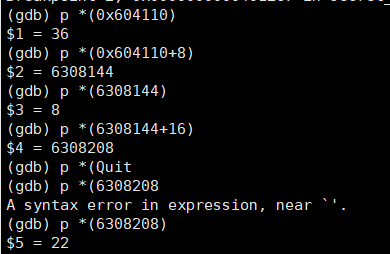


2.先看隐藏关的主函数，读取一个2字节的值作为一个参数，并且这个参数要小于0x3e8，否则爆炸，然后传入了一个地址存在rsi中作为第二个参数。而fun7的返回值应当为2.

3.然后看fun7函数。我将这个函数转化为了等价的c语言代码便于理解，如下所示。但是注意，这个代码并不能，也没必要运行，因为rsi中的后续地址是已经存在内存里的。

用倒推的方法推测输入的参数值。输出为2说明最终输出是在if（edx>esi）中return的，往上倒退知道上一个eax值为1，那显然是在最后一行return的，此时eax=0，再往上倒推可知eax等于0只可能再if（edx==esi）处返回，而esi即为输入参数，edx为此时的rsi指向的值。rsi在此过程中经历了一次+8和一次+16.

打印出rsi+8得到的地址再+16的值，如下

得到输入的值应该为22.

输入22，通过此关。

