

**实验报告**



**题目： 拆解二进制炸弹**

**班 级：** 2021211321

**学 号：** 2021212171

**姓 名： 杨晨**

**学 院： 计算机学院**

2022**年** 11**月** 13 **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. 服务器
   1. 服务器：10.120.11.12（X86版本）：2022**年**11**月**10**日**23**:**59**分关机（二周）；提交作业截止时间**2022**年**11**月**13**日**00**：**00**，延迟扣分（每晚一天扣**1**.**5**分）**
   2. 服务器：10.99.0.230（华为arm版本）：2022**年**11**月**25**日**23**:**59**分关机（三周）；提交作业截止时间**2022**年**11**月**27**日**00**：**00**，延迟扣分（每晚一天扣**1**.**5**分）**
   3. 服务器密码找回指南：<https://khbgo05wz2.feishu.cn/docx/doxcnioT927j2gDSTwHoAQLGVpg>
3. MobaXterm
4. Linux
5. Objdump命令反汇编
6. GDB调试工具
7. 积分榜（http://10.120.11.13:19240/scoreboard）

三、实验内容

四、实验步骤及实验分析

建议按照：准备工作、阶段1、阶段2、…等来组织内容

各阶段需要有操作步骤、运行截图、分析过程的内容

**准备工作**

解压文件（tar **-**xvfbomb262**.**tar），得到文件夹bomb262

进入文件夹bomb556（cdbomb262），文件夹中有3个文件，可执行文件，c代码文件，readme文件

查看README文件（catREADME）,确认炸弹属于bupt2021212171

文本

描述已自动生成

用vi编辑器，创建一个answer，用以存储答案（vianswer）

文本

描述已自动生成

1. **拆解**phase**\_**1

首先进入gdb调试阶段（gdbbomb）



在第一阶段设置断点（bphase**\_**1）

在爆炸处设置断点（bexplode**\_**bomb）

反汇编phase\_1函数（disasphase**\_**1）

电脑屏幕截图

描述已自动生成

观察函数strings\_not\_equal，猜测是比较字符串是否相等的函数

若函数返回值不等于0，则会引爆炸弹；若返回值是0，则成功拆除，因此接下来分析strings\_not\_equal函数

反汇编strings\_not\_equal函数（disasstrings**\_**not**\_**equal）

图形用户界面, 文本

描述已自动生成

可以确定，这个函数就是判断两个字符串是否相等的

注意到，寄存器x1中存储的是我们要匹配的字符串，而x1存储的是地址0x402000+0x6e0处的内存

将其打印出来（x**/**s0x402000**+**0x6e0）



第一阶段的答案为Joy shared with others are more enjoyed.

将答案保存到answer中（vianswer）

存入第一阶段的答案（cataIamjustarenegadehockeymom**.**）

退出，保存（Esc **:**wq）

文本

中度可信度描述已自动生成

gdb调试中，运行答案（ranswer）

文本

描述已自动生成

第一阶段成功拆除

* 1. **拆解**phase**\_**2

在爆炸处设置断点（bexplode**\_**bomb）

在第二阶段设置断点（bphase**\_**2）

文本

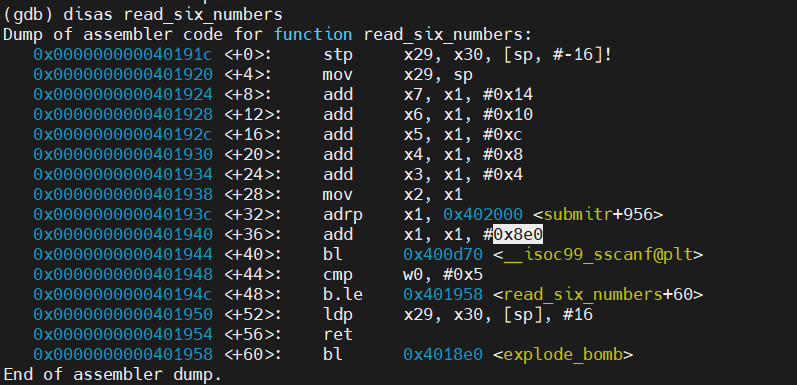
描述已自动生成

反汇编函数phase\_2（disasphase**\_**2）

图形用户界面, 文本

描述已自动生成

注意到函数read\_six\_numbers，反汇编这个函数（disasread**\_**six**\_**numbers）

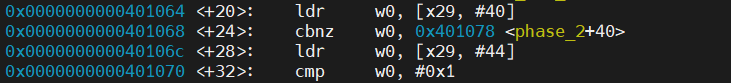


在read\_six\_numbers函数中调用了sscanf函数，查看第二个参数x1的值（x**/**s0x402000**+**0x8e0）



所以第二阶段应该输入6个int类型的整数，分别存在x29+40,x29+44,x29+48,x29+52,x29+56,x29+60

容易发现，前两个数字必须是0和1



第一次循环

x19是第一个数字的地址，要求[x19] + [x19 + 4] == [x19 + 8]，即前两个数字之和为第三个数字

第二次循环，x19 = x19 + 4

同理，[x19] + [x19 + 4] == [x19 + 8]，第二个数字加第三个数字要等于第四个数字

文本

低可信度描述已自动生成

即斐波那契数列，故第二阶段的答案为0 1 1 2 3 5

存入answer中

文本

中度可信度描述已自动生成

运行

第二阶段成功拆除

文本

描述已自动生成

* 1. **拆解**phase**\_**3

在爆炸处设置断点（bexplode**\_**bomb）

在第三阶段设置断点（bphase**\_**3）

文本

描述已自动生成

反汇编phase\_3函数（disasphase**\_**3）

文本

描述已自动生成

文本

描述已自动生成

图形用户界面, 文本

描述已自动生成

观察到phase\_3中同样调用了sscanf，第二个参数x1在调用前被赋值0x402000+0x710

gdb中查看0x402000+0x710的值（x**/**s0x402000**+**0x710）



因此阶段三应该输入1个int类型的整数,1个char类型的字符，1个int类型的整数

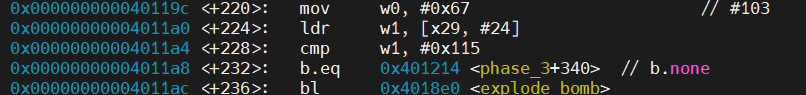
注意到cmpw0*, #*0x2指令，第一个数字如果小于等于2，则会爆炸



接下来是一个switch语句，取决于第一个数字是3,4,5,6,7，如果不是这些数字，则会爆炸

所以我们不妨让第一个数字为3

由接下来的语句，可以看出，若第一个数字是3，则第三个数字必须是0x115（277），否则会爆炸



此时，w0被赋值为103

接下来，比较第二个字符的ascii码值和w0，要求必须相等

查看ASCII码表，得到第二个字符为g

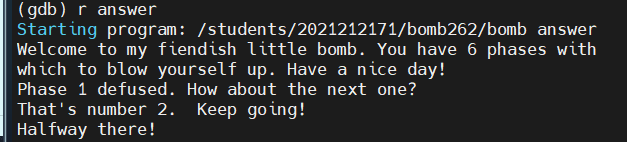
电脑萤幕画面

描述已自动生成

因此，第三阶段的答案是3 g 277

将答案存入answer中，运行

第三阶段成功拆解



1. **拆除**phase**\_**4

在爆炸处设置断点（bexplode**\_**bomb）

在第四阶段设置断点（bphase**\_**4）

文本

描述已自动生成

反汇编phase\_4函数（disasphase**\_**4）

图形用户界面, 文本

描述已自动生成

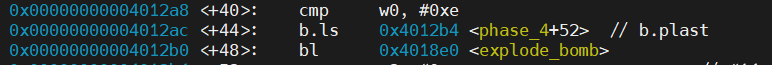
发现也调用了sscanf函数

查看0x402000+0x720，发现第四个阶段需要读入两个整数（x**/**s0x402000**+**0x720）

此处和其他地方有些不同，第一个数存储在x29+28里，第二个数存储在x29+24里



第一个数字要小于14,否则会爆炸



之后进入函数func4 观察后续代码可知，要让函数的返回值等于15

屏幕上有字

描述已自动生成

查看函数func4的汇编代码（disasfunc4）

图形用户界面, 文本

描述已自动生成

对应的递归c代码为

电脑的屏幕

描述已自动生成

容易看出，当第一个数字为5的时候，返回值是15

回到phase\_4函数，注意到下面这些指令

第二个数字必须是15

文本

描述已自动生成

第四阶段的答案是5 15

存入answer中，运行

第四阶段成功拆解

文本

描述已自动生成

1. **拆解**phase**\_**5

在爆炸处设置断点（bexplode**\_**bomb）

在第五阶段设置断点（bphase**\_**5）

文本

描述已自动生成

反汇编phase\_5函数（disasphase**\_**5）

图形用户界面, 文本

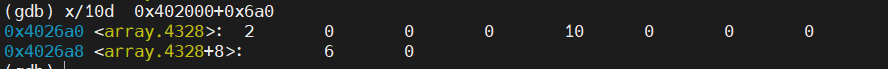
描述已自动生成

注意到下面这些指令，所以需要输入一个长度为6的字符串长度



注意到有一个数组0x40200+0x6a0，循环的时候要从中取值

打印这个数组内的值（x**/**10d0x402000**+**0x6a0）



观察循环，每次w1的值是数组内的下标为[x1 \* 4]，而x1的值为字符串的每个字符（即若字符串是111111，则x1的值都是1）

文本

描述已自动生成

在循环的过程中，w3 = w3 + w1

最后w3的值要求是0x24（即十进制数36）

电脑屏幕的照片上有文字

描述已自动生成

由于字符串的长度是6，所以需要循环6次

根据数组内的值，arry[8] = 6，所以将arry[8]的值取6次即可

那么x1的值依次为2,2,2,2,2,2

第五阶段的答案为 222222

将答案存入answer中，运行

第五阶段成功拆解

文本

描述已自动生成

1. **拆解**phase**\_**6

在爆炸处设置断点（bexplode**\_**bomb）

在第六阶段设置断点（bphase**\_**6）

图形用户界面, 文本

描述已自动生成

反汇编phase\_6函数（disasphase**\_**6）

文本

描述已自动生成

文本

描述已自动生成

观察到调用了函数read\_six\_numbers，这个函数在phase\_2中出现过，这里同理

输入6个int类型的整数，分别存在x29+68,x29+72,x29+76,x29+80,x29+84,x29+86

这里是一个双重循环，每次先将数字-1，然后判断是否大于5（无符号大于），若大于5则会爆炸

同时，内层循环要判断是否前一个数等于当前的数字，若相等则爆炸

因此，输入的六个数字是1~6且各不相同

等价的c代码为

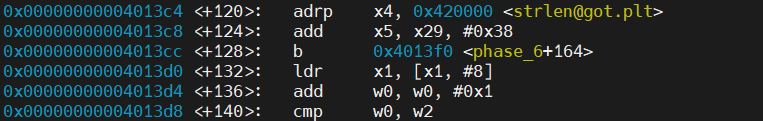
文本

描述已自动生成

文本

描述已自动生成

注意到ldr x1, [x1, #8]，每次移动8个字节，因此猜测这是个结构体，且0x8为该结构体指针偏移量



用gdb查看0x420000+0x110内的值（x**/**4dw0x42000**+**0x110）

容易看出看出，0~3字节是一个数字，4~7字节是节点编号，8~15字节一个是指针，打印可以发现，4326024指向node2，同样node2中的指针指向node3……可以推测出，这个结构体是一个链表

将整个node的节点都打印出来（0x20dw4326024）

图片包含 表格

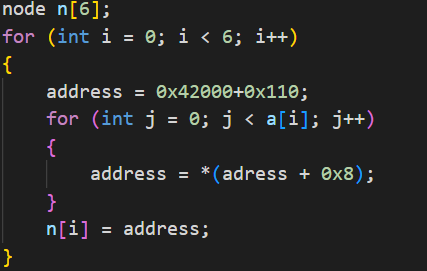
描述已自动生成

大致可以确定，这个结构体为

图形用户界面, 文本

描述已自动生成

接下来是一个循环，对应的c代码如下



接下来进入下一个循环，按照数组n的顺序去修改链表的next，使得链表按照数组的元素顺序，一一对应重新构成一个链表

最后一个循环，六个数字（node -> val）要从小到大排列

图形用户界面, 文本, 应用程序

描述已自动生成

所以，答案为5 1 3 4 6 2

将答案存入answer中，运行

第六阶段成功拆解

文本

描述已自动生成

1. **拆解**secret**\_**phase

在爆炸处设置断点（bexplode**\_**bomb）

在第隐藏阶段设置断点（bphase**\_**defused）

文本

描述已自动生成

反汇编phase\_dufused函数（disasphase**\_**defused）

文本

低可信度描述已自动生成

从这里可以看出，在输入数据有3个时候，才会检查是否输入了进入隐藏阶段的密码



所以需要在前面有两个输入的地方，后面再补上一个字符串，而之前六个阶段，只有第四个阶段是2个输入，所以要在第四组输入后面补上某个字符串才能进入隐藏阶段

进入隐藏阶段的密码为（x**/**s0x402000**+**0x968）

文本

描述已自动生成

反汇编secret\_phase函数（disassecret**\_**phase）

文本

描述已自动生成

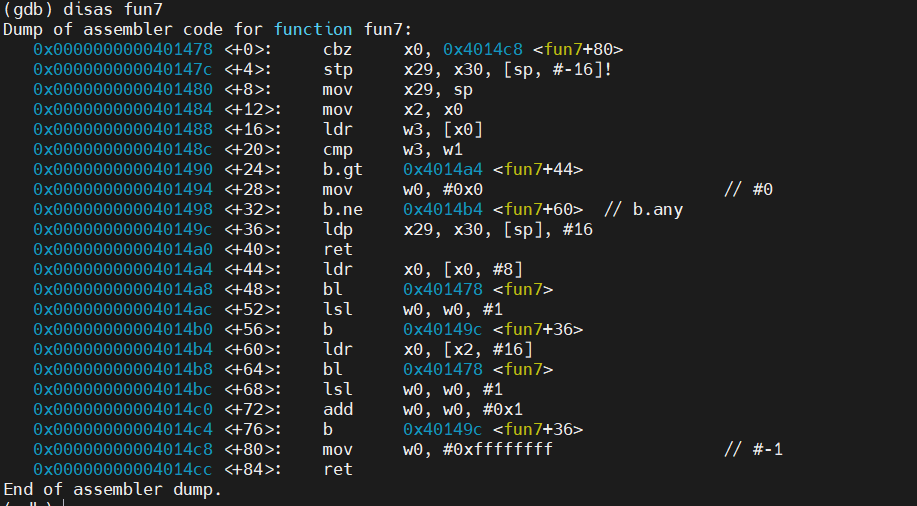
输入的数字，在减一之后，如果大于1000，则会爆炸



要求函数fun7的返回值必须是6，否则会爆炸



反汇编函数fun7（disasfun7）



fun7是一个递归函数，而且是一个二叉树的查找函数

还原的c代码如下

电脑的屏幕

描述已自动生成

根节点下标为0，若小于当前数值，则向左查找，下标\*2；若大于当前数值，则向右查找，下标\*2+1

注意到根节点的数值为36（x**/**d0x420000**+**0x110**+**0x10）

文本

描述已自动生成

6 = 2 \* 3

3 = 2 \* 1 + 1

1 = 2 \* 0 + 1

所以需要向左查找1次，向右查找2次

通过试探，可以画出二叉树的大致结构

图示

描述已自动生成

所以，隐藏阶段答案是35

将答案存入answer中，运行

隐藏阶段成功拆解

文本

描述已自动生成

五、总结体会

总结心得（包括实验过程中遇到的问题、如何解决的、过关或挫败的感受、实验投入的时间和精力、意见和建议等）

刚开始的时候，arm语句不够熟悉，会卡在某个地方很长时间，然后系统浏览arm指令，我有了一个清晰的理解，指令的作用也可以判断出来。

在分析过程中，可以根据跳转指令对程序进行分段，并确定循环部分和分支判断部分，这样划分可以快速确定汇编代码的功能。

在分析函数的时候，如果看到不明白的地方，先往下看，根据后面的逻辑，推测前面的逻辑，可以帮助我们更好的理解函数。

递归函数和循环比较复杂。这时，将相应的汇编代码转换成C语言的伪代码或代码，在C语言编译器的帮助下帮助我们找到答案

拆解二元炸弹的过程就是熟悉arm指令的过程，在这个实验中，我对跳转指令、赋值指令、比对指令等有了更深入的理解，初步具备了举一反三的能力。

六、诚信声明（不签扣10分）

需要填写如下声明，并在底部给出手写签名的电子版。

在完成本次实验过程中，我曾分别与以下各位同学就以下方面做过交流：

1、简单描述交流内容，例如：来自\*\*\*的建议，采用\*\*\*方式\*\*\*

2、

此外，我还参考了以下资料：

1. 网址等
2. [arm 指令集 - 菜鸟学院 (noobyard.com)](http://www.noobyard.com/article/p-mkgstiji-bw.html)

在我提交的程序中，还在对应的位置以注释形式记录了具体的参考内容。

我独立完成了本次实验除以上方面之外的所有工作，包括分析、设计、编码、调试与测试。

我清楚地知道，从以上方面获得的信息在一定程度上降低了实验的难度，可能影响起评分。

我从未使用他人代码，不管是原封不动地复制，还是经过某些等价转换。

我未曾也不会向同一课程（包括此后各届）的同学复制或公开我这份程序的代码，我有义务妥善保管好它们。

我编写这个程序无意于破坏或妨碍任何计算机系统的正常运行。

我清楚地知道，以上情况均为本课程纪律所禁止，若违反，对应的实验成绩将按照0分计。

（签名）卡通人物

低可信度描述已自动生成