**深 圳 大 学 实 验 报 告**

**课程名称： 计算机系统(2)**

**实验项目名称： 逆向工程实验**

**学院： 计算机与软件学院**

**专业： 计算机科学与技术**

**指导教师： 王 毅**

**报告人： 沈晨玙 学号： 2019092121 班级： 19计科国际**

**实验时间： 2021年6月10日**

**实验报告提交时间： 2021年6月10日**

**教务处制**

|  |
| --- |
| **一、 实验目标与要求：**   1. 理解程序（控制语句、函数、返回值、堆栈结构）是如何运行的 2. 掌握GDB调试工具和objdump反汇编工具 |
| **二、实验环境：**   1. 计算机（Intel CPU） 2. Linux64位操作系统（Ubuntu 17） 3. GDB调试工具 4. objdump反汇编工具 |
| **三、实验方法与步骤：**  本实验设计为一个黑客拆解二进制炸弹的游戏。我们仅给黑客（同学）提供一个二进制可执行文件bomb\_64和主函数所在的源程序bomb\_64.c，不提供每个关卡的源代码。程序运行中有6个关卡（6个phase），每个关卡需要用户输入正确的字符串或数字才能通关，否则会引爆炸弹（打印出一条错误信息，并导致评分下降）！  要求同学运用**GDB调试工具和objdump反汇编工具**，通过分析汇编代码**，**找到在每个phase程序段中，引导程序跳转到“explode\_bomb”程序段的地方，并分析其成功跳转的条件，以此为突破口寻找应该在命令行输入何种字符串来通关。  本实验需解决Phase\_1(**15分**)、Phase\_2(**15分**)、Phase\_3(**15分**)、Phase\_4(**15分**)、Phase\_5(**15分**)、Phase\_6(**10分**)。通过**截图+文字**的形式把实验过程写在实验报告上，最后并撰写**实验结论与心得(15分**)。 |
| **四、实验过程及内容：**  0. 反汇编    获得汇编代码     1. 第一关    * 1. 定位到phase\_1处，以下为个人注释      * + 1. 整体思路：通过阅读汇编代码及注释（如图 3），可以看到，有je指令能在字符串相等的情况下跳过爆炸，所以通关则必须保证字符串相等。而函数string\_not\_equal从名字上可以看出，这是个判断字符串是否相等的函数。而在调用函数前，有将0x401af8赋值给寄存器esi的操作，所以猜测解题的关键应该是这个常量的值。     2. 使用gdb，查看内存地址字符串      * + 1. 答案：Science isn't about why, it's about why not?      1. 第二关    * 1. 定位到phase\_2，以下为个人注释      * + 1. 整体思路   首先读取6个元素，然后进入循环判断  循环跳出条件为r13与rbp是否相等，其中r13=rsp+12，rbp每轮+4，因此循环次数为3次。  每次循环判断M[rbp+12]与M[rbp]是否相等，如果不相等则爆炸，因此a[i]=a[i+1]。同时讲a[i]存入r12（累加器）  退出循环之后，如果r12（累加器）为0，则爆炸。   * + 1. 答案：a[0]=a[3], a[1]=a[4], a[2]=a[5], a[0]+a[1]+a[2] != 0   不妨设数组为1 3 5 1 3 5进行测试     1. 第三关    * 1. 定位到phase\_3，以下为个人注释      * + 1. 整体思路：   首先读入0x401ebe，根据gdb可知为“%d %d”，因此先读入两个数字。    判断scanf的返回值，如果小于1则爆炸。  判断了第一个数是否大于7，如果是则bomb，那么可以确定第一个数取值为0-7。  然后有一个地址跳转（根据rax的值），利用gdb查询目标地址    根据rax的值进行跳转，随后将输入的值与预设值进行比较，如果不相等则爆炸。   * + 1. 答案：共有六组解   0——535  1——926  2——215  3——339  4——119  5——352  6——919  7——535     1. 第四关    * 1. 定位到phase\_4，以下为个人注释        * + 1. 整体思路：   在主函数中由gdb可知，读入一个数字。如果读入的数字小于1，爆炸。    然后将读入的数字作为参数，调用func4()  在func4中，递归结束条件为参数小于等于1，否则返回func4(n-1)+func4(n-2)。显然是一个斐波那契数列。  主函数的含义是输入一个数字n，判断斐波那契数列第n项是否为55，否则爆炸。   * + 1. 答案：   斐波那契数列为1，1，2，3，5，8，13，21，34，55，...  所以55为第9项（0开始），答案为9     1. 第五关    * 1. 定位到phase\_5，以下为个人注释      * + 1. 整体思路：   同前几题一样读入两个数字，如果第一个数字为15，爆炸。  将ecx，edx置零，ecx记录循环次数，edx记录累加值。  循环的跳出条件是eax=15，而eax的值等于M[0X401BA0+rax\*4]是变址寻址。用p \*0X401BA0@16，可以查看数组所有的值。    需要循环12次，并且最终结果值为15。其中每一步执行k = a[k]，并记录累加值。  因此由15向前反推：   |  |  |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | | k | 6 | 14 | 2 | 1 | 10 | 0 | 8 | 4 | 9 | 13 | 11 | 7 | | A[k] | 15 | 6 | 14 | 2 | 1 | 10 | 0 | 8 | 4 | 9 | 13 | 11 |  * + 1. 答案：累加值为93 初始输入为7      1. 第六关    * 1. 定位到phase\_6，以下为个人注释        * + 1. 整体思路：   调用strtol函数，将字符串转换为long类型，同时将转换的long存储在地址0x602780，并传入edi作为参数传入func6,用gdb查询地址0x602780。  然后对rax进行了三次加偏移，应该是和链表有关。  避免爆炸的条件是rax=edx，而edx=[0x20168e+rip]。  调用函数func6后，返回值rax已经确定，可以通过gdb设置断点查看返回值rax。   * + 1. 答案：   在rax偏移24字节后，设置断点在0x401108处。    运行到断点处时，查看rax内存值，为600.所以答案是600. |
| **五、实验结论：**    第一关答案：Science isn't about why, it's about why not?  第二关答案：1 3 5 1 3 5 （不唯一，详解见上）  第三关答案：0 535（不唯一，8组任选其一，详解见上）  第四关答案：9  第五关答案：7 93  第六关答案：600 |
| **六、心得体会：**  终于艰难的完成了拆炸弹的实验。通过实验慢慢体会到了x86汇编代码的精髓，一开始进度特别慢，到后面逐渐了解汇编代码，最终完成实验。  其中难点在于跳转部分与循环部分。对于循环部分的解决办法是利用纸记录下每一个寄存器的变化过程，并理解思路。跳转部分需要更细心的阅读代码，理解每一块跳转语句的含义。 |
| 指导教师批阅意见：  成绩评定：  指导教师签字：王毅  2021年 6月1日 |
| 备注： |