**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： 软件工程**

**指导教师： 罗胜**

**报告人：谢弘烨 学号： 2020151036 班级： 软工02**

**实验时间： 2018年4月18日至5月15日**

**实验报告提交时间： 2018年5月X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：**   1. 对bomb\_64文件进行反汇编   依照材料中方法，将bomb\_64文件反汇编结果输出至1.txt文件。由于实验周期比较长，操作过程忘记截图。       1. Phase\_1   跳转到phase\_1函数处，查看phase\_1汇编代码    可以看见函数将地址0x401af8处数据作为strings\_not\_equal函数的第一个参数传入，猜测此处数据为phase\_1的正确答案。  利用gdb调试工具查看此处数据，结果为：    可见phase\_1的正确答案为“Science isn't about why, it's about why not? ”  对答案进行验证，答案正确。     1. Phase\_2   观察phase\_2函数，发现其调用名为read\_six\_numbers的函数，猜测其正确答案为六个数字    跳转到read\_six\_numbers函数，发现其调用了scanf函数，传入的格式为地址0x401eb2处的数据。    利用gdb调试器观察此处数据，发现其为“%d %d %d %d %d %d”，可以验证上述正确答案为六个数字的猜测。    返回到phase\_2函数中，发现函数在400eac处将rbp寄存器设为六个数字中第一个数字的地址，r13寄存器设置为第三个数字的地址，r12d寄存器设置为0.    接着  在400eba，拷贝rbp的值至rbx中，即复制第一个数字的值  在400ebd，将第四个数字赋值给eax寄存器  在400ec3，处比较rbp指向的值与eax寄存器中的值，不相等就爆炸  在400eca，r12d加上rbx指向的数字  在400ecd，将rbp加4，即指向下一个数字  在400ed4判断rbp是否指向第三个数字，若是则跳出循环    该过程实现了将1、4，2、5，3、6，三组数字两两比较，不相等就爆炸的操作。  之后还判断了r12d寄存器的值是否为0，即六个数字中的前三个数字和是否为0。    综上，可以判断出正确答案为：六个数字和不为0的，每隔两个相同的数字。  多组数据可以验证判断正确。         1. Phase\_3   跳转到phase\_3函数处，发现scanf的参数的0x401ebe处。    用gdb调试器可以看到此处数据为“%d %d”，即正确答案为两个数字    回到汇编代码，可以发现在400f20处，函数对于传入的第一个参数进行了是否不大于7的判断，若为否则直接爆炸    此外，观察到多处代码都进行了给eax寄存器赋值，后跳转到400f74的操作，猜测其可能为switch语句    观察400f74处代码，发现其将输入的第二个数字与eax的值进行比较，不相同就爆炸    猜测正确答案第一个数字为2至7其中一个，第二个数字与第一个数字在函数中一一对应  在400f74处设置断点，发现第一个数字为2时，eax的值为214        输入2 214，结果正确。    重复上述过程，可以得到答案3 339，4 119，5 352，6 919，7 412。   1. Phase\_4   观察phase\_4函数代码，发现scanf参数在0x401ec1处，gdb观察其值为“%d”      即正确答案仅为一个数字  在400fde处，函数对输入数字进行判断，小于0则爆炸    观察后续代码，发现在调用func4函数后，若eax的值不为0x37（55）则会发生爆炸    观察发现func4函数主体含有递归调用操作，较为复杂    此处进行尝试利用断点解题。在400ff3处设置断点，观察不同输入下eax的值。    输入为1时，eax为1    输入为2时，eax为2    输入为3时，eax为3    输入为4时，eax为5    输入为5时，eax为8  观察发现，从输入为3开始，每一次eax的值为前两次eax值之和，猜测eax的值满足数列。计算得出当n=9时，即输入9，eax的值为0x37。  验证猜测正确，答案为9。     1. Phase\_5   观察phase\_5函数代码    gdb检查地址0x401ebe，结果为“%d %d”    返回代码，发现在401029处，函数检查输入的第一个数字是否大于15，若是则发生爆炸    接着，将ecx和edx均设置为0    观察发现，在401048处出现一个地址    用gdb调试器可以看到其中存储的数据为整型数组    同时，  在401043，edx自加  在401048，会以当前eax的值为下标访问上述数组中的元素，并将eax的值设为新的元素  在40104f，ecx会加上当前eax的值  在401051，判断eax是否等于15，是则跳出循环  即edx为循环计数器，ecx记录下eax所有值之和，同时以eax的值为下标，在数组中取出元素  最后判断循环次数是否为12次，不是则爆炸；判断求得eax数据和是否与输入的第二个数字相同，不是则爆炸    即输入第一个数字为eax第一次取元素的下标，确保在12次取元素后eax的值为15，之和为输入的第二个数字。  通过gdb调试器观察数组中的元素，猜测正确答案应为7 93  验证猜测正确     1. Phase\_6   观察代码    发现其中由对其他函数的调用，以及仅有一次的比较。在比较处（40110e）设置断点，输出3进行观察。    发现此时rdx的值为输入的值，rax的值为地址，其指向的值为600。因此猜测正确答案为600。  验证猜测正确 |
| **五、实验结论：**  **拆弹成功** |

|  |
| --- |
| 指导教师批阅意见：  成绩评定：  指导教师签字：刘刚  2018年 5月15 日 |
| 备注： |