**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： 计算机与软件学院所有专业**

**指导教师： 罗胜**

**报告人： 张欣杰 学号：2020151091 班级： 软工二班**

**实验时间： 2022年5月06日至5月31日**

**实验报告提交时间： 2022年5月31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：**  先反汇编得到bomb\_64的汇编代码，执行命令：objdump -d bomb\_64>asm.txt    退出控制台，进入到home目录下可以看到已生成的储存汇编代码的文本文件asm.txt。    得到的文本文件asm.txt中包含了反汇编得到的代码，其中包含主函数和6个关卡的具体代码。   1. 主函数：   前部分代码为初始化变量，初始化变量之后是输入和函数调用代码。注意到在每个关卡函数调用前都有一个callq 40165f，即调用real\_line，即输入函数，将调用后的函数返回值拷贝至寄存器rdi，调用关卡函数后会调用phase\_defused判断输入是否正确，若正确则继续游戏。     1. 关卡1，函数名为phrase1:   首先对寄存器rsp的值减去8，rsp为栈顶指针，rsp-8之后，表示rsp指向栈顶元素的下一位元素，然后使用mov操作，将0x401af8这个地址传入esi寄存器，即第二个参数对应的寄存器，然后调用string\_not\_equal检查字符串是否相同    跳转查看string\_not\_equal函数，可以看到，函数首先调用string\_lenngth计算两个字符串的长度，不相等便跳转到return前几行的操作，长度相等则继续比较，若出现不相同的字符，则也会跳转到return的前几行。在主函数中，输入的返回值mov到rai寄存器中，在phase1中，0x401af8通过mov存到esi寄存器中，在本函数中，rdi的值又被存到rbx中，经过多次操作后最终用于字符串比较，因此可以推测0x401af8为答案字符串的首地址。    通过GDB调试查看地址0x401af8中存放的字符串为 ”Science isn’t about why, it’s about why not?” 因此输入这句话尝试，成功通关。     1. 关卡2，函数为phase2：   首先将4个寄存器的值入栈，保持原始数据，本函数的主题为一个循环，从400eba到400ed4，因此正确答案可以在循环中找到。    通过观察可以看到，炸弹爆炸应该在地址400ec5处，前一行为跳转指令，根据比较eax寄存器和rbp寄存器的值，相等即可跳过爆炸。除此之外，在函数开头，rsp寄存器自减0x48，后续将其内容拷贝至rpb寄存器，之后eax的值被赋值为rbp+0xC指向地址的值，然后eax和rbp的值进行比较；其中+0xC为+12，通过read\_six\_number可推断，这里应该输入6个数字，并且前三个数字应与后三个数字相等，除此之外，还有一个限制条件为6个数的和应当不等于0。因此输入1 2 3 1 2 3进行尝试，如下图，成功通关。     1. 关卡3，函数为phase3   在输入前，分别将第三个和第四个参数的地址分别传给rdx和rcx，在第二个参数经过mov操作后，内容为0x401ebe，然后将eax置为0，再进行输入操作，将输入的数的返回值与1比较，若返回值不为1则输入非法，炸弹爆炸，否则跳转到0x400f20继续操作；在这之后判断输入的第一个数是否比7小，若不是则炸弹爆炸，若是跳转到下一步判断。再跳转后，对eax赋值，然后对eax进行判断，此时，调用指针一次查看401b60-401b98的值。    根据地址，再源代码查找对应的值，输入的第一个整数与其对应的正确值如下表   |  |  | | --- | --- | | 0 | 535 | | 1 | 926 | | 2 | 214 | | 3 | 339 | | 4 | 119 | | 5 | 352 | | 6 | 919 | | 7 | 412 |   输入任意一对数即可，这里我输入的是1和926，如下图成功通关     1. 关卡4，函数为phase4   首先将第一个参数传给rdx，将0x401ec1拷贝给esi，eax赋值为0，调用输入函数，若输入的返回值不为1，说明输入不合法，则炸弹爆炸。再合法性判断中，首先判断是否大于0，若是，调用fun4函数，pa那段返回值是否为0x37，即55，若不是，则引爆炸弹，若是，则通关。  Fun4函数：    通过观察fun4函数，可以看到，先将rbx和rbp函数的值保存下来，然后再给ebx和eax赋值，输入的数值被储存在edi中，使用edi的值对ebx进行初始化，eax设为1，然后将edi与1进行比较，相等则返回，若大于1，则将edi的值减一，然后再次调用函数fun4函数，不难看出这是一个递归函数。同时注意到edi的值是设置成原值-2再递归，调用函数的目的是将edi-1和edi-2的递归调用返回值存在eax中。  通过底层回溯，当edi为1的时候，返回1，当edi为2的时候，返回1，当edi为0的回收返回0，由此可知，fun4求的是给定下标的斐波那契数列的对应值，结合关卡4的返回值是否为55可知，只需要输入斐波那契数列中55对应的下标值即可过关，通过列出斐波那契数列  1 1 2 3 5 8 13 21 34 55可得55的下标为9，因此输入9进行测试，如下图所示，成功过关     1. 关卡5，函数为phase5   首先将都三个参数和第四个参数分别存入rdx和rcx，将0x401ebe装入esi，eax初始化为0，调用输入函数，检查输入值的返回值是否为1，若不是则引爆炸弹，若输入合法则跳转到后续操作。  将输入参数保存到eax，然后对eax和15做比较，结果存放在eax，然后该结果存回原来在内存中的位置。对eax与15比较，若等于15则直接跳转到调用炸弹函数的位置，说明我们的输入值不能是15的倍数。之后ecx和edx清零，注意之后一行位置是401043，到401054时为跳转指令，满足条件会跳回401043，说明这里是一个循环：    再循环结束之后有一行cmp指令，将0xc，与edx比较，比较结果不相等则引爆炸弹，若相等则往下操作，结合循环前edx清0，以及循环中对edx仅有的修改操作是加1，说明该循环应执行12次，才不会触发炸弹。注意到循环前rax的内容为输入数和15与操作，在循环中，mov 0x401ba0(,%rax,4),%eax这一条指令，将rax内容作为索引值，访问到0x401ba0+rax\*4这一内存地址指向的值后，存给eax，并将eax加给ecx，再比较eax是否等于15，不等的话回跳。可以看出，程序预留了一个数组，每个数组内容都是不大于15的正整数，故可以把数组元素又作为下标去跳转，直到碰到数组值为15就退出循环，结合上面的要求循环12次，说明应当跳转12次才碰到15。循环12次后还需比较我们输入的第二个参数和寄存器ecx的值是否相同，ecx在前面的循环中累加，每次跳转的值都加到ecx中，故除了控制我们一开始输入的索引值能保证跳转12次到15外，还要计算出这12次跳转的值的和的大小。  使用指令查看数组的内容：    因此根据汇编代码可以得到总跳转次数为12，，而11的索引值为7，前12个数的和为93，因此输入索引和和即为正确答案，即7 93。     1. 关卡6，函数为phase6   分析代码，可以得知，当程序执行401112时会引爆炸弹，当edx等于rax时可以跳转，避免引爆炸弹；首先将edx赋值为0xa，而esi的值为0。调用strtol函数将string类型数据转换为long型数据，然后将转换后的值存储到0x602780中，因此推测本函数的功能时将字符串转换为数字后，经过fun6函数处理后，使之与某个值相等，但是再fun6函数中，数据变换十分复杂，因此可以使用gdb调试测试rax和edx的关系。    根据上图可以得知，rax中存储的值为rax，因此尝试输入673。发现成功通关。 |
| **五、实验结论：**  经过6轮拆弹，成功通过所有关卡，成功拆弹。 |
| **六、心得体会：**  通过本次实验，大大增强了对指令集以及汇编代码的理解，促进了对gdb调试的熟悉程度，掌握了断点设置，寄存器内容查看等等。本次实验对掌握汇编语言阅读能力的要求较高，对给定的程序代码需认真推演，清晰地掌握各个命令的操作条件和效果，经过耐心求解后本次实验最后得到了正确结果。 |

|  |
| --- |
| 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： |