

课程实验报告

课 程 名 称： 计算机系统

实验项目名称： 课程实验三

专 业 班 级： 计科1907

姓 名： 杨杰

学 号： 201908010705

指 导 教 师： 黄丽达、赵欢

完 成 时 间： 2021 年 5 月 8 日

信息科学与工程学院

|  |  |
| --- | --- |
| **实验题目：**LAB3-bomblab | |
| **实验目的：**本次实验为熟悉汇编程序及其调试方法的实验。 | |
| **实验环境：**个人电脑、linux发行版本、dlc编译环境 | |
| **实验内容及操作步骤：**  **4.1 实验内容**  程序运行在linux环境中。程序运行中有6个关卡（6个phase），每个phase需要用户在终端上输入特定的字符或者数字才能通关，否则会引爆炸弹！那么如何才能知道输入什么内容呢？这需要你使用gdb工具反汇编出汇编代码，结合c语言文件找到每个关卡的入口函数。然后分析汇编代码，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通关。  **4.2 操作步骤**  **phase\_1**  (一)汇编代码分析    (二)具体分析  首先我们看到调用的函数为<strings\_not\_equal>，即比较两个字符串是否相等。可以想到该函数需要的两个参数为两个字符串的首地址。按照这个思路我们先查看一下地址0x804a184：    可以猜到0x20(%esp)中存的就是我们刚刚输入的字符串的首地址，当输入和从0x804a184开始存放的字符串一样的字符串时，<string\_not\_equal>函数返回0，跳转，不会bomb。  故输入Border relations with Canada have never been better.    经验证正确。  **phase\_2**  (一)汇编代码分析      (二)具体分析  函数<read\_six\_numbers>读入6个数字，分别是a[0],a[1],a[2],a[3],a[4],a[5].根据<phase\_2>的汇编代码，不难发现当a数组各元素满足下列关系时不会bomb：    所以答案可以有多种，在此举出两种。  0 1 3 6 10 15  3 4 6 9 13 18    经验证正确。  **phase\_3**   1. 汇编代码分析       (二)具体分析  执行<\_\_isoc99\_sscanf@plt>之前传入了4个参数。其中0xc(%esp)，0x8(%esp)为输入的两个整数的地址；(%esp)应该和前面几题一样，存的就是我们输入字符串的首地址。  0x804a3c3这个地址是干什么的呢?    也证实了确实要输入两个整数。  第一个参数不能大于7，否则bomb.  一个关键的语句：  8048bf7: ff 24 85 00 a2 04 08 jmp \*0x804a200(,%eax,4) //switch分支  这里%eax存的是第一个参数x的值。  接下来进入跳转表    只有输入的x计算出的结果和第二个参数相等时不bomb.  此题会有多个答案，分别是：  0 539  1 519  2 725  3 128  4 730  5 658  6 221  7 385      经验证正确。  **phase\_4**  （一）汇编代码分析        (二)具体分析  <func4>是一个递归函数，写成c语言如下：    再来看<phase\_4>，首先读入两个整数x和y,然后调用func4(x,0,14),只有func4(x,0,14)=7并且y=7时才不会bomb。如果x=(14-0)/2+0=14, func4(x,0,14)=0,如果x<7, func4(x,0,14)=2\*func4(x,0,6),是一个偶数，肯定不等于7，所以x一定满足7<x<=14。      运行程序，找到当x=14时，func4(x,0,14)=7，所以本关卡答案为14 7    经验证正确。  **phase\_5**  (一)汇编代码分析    (二)具体分析  这个函数需要输入一个长度为6的字符串。实现了一个循环。依次处理这六个字符，只取每个字符ascii码的最后4位放入ecx，作为偏移量。将（0x804a220+%ecx\*4）放入edx，最终edx的值要为74(0x4a)。  其中地址0x804a220是一个数组的首地址，数组各元素值如下：    在此给出两组答案：  0xa+0x10+0x9+0xb+0xf+0xd=0x4a  数组索引分别为1 5 6 12 14 15    根据ASCII码表找出一组对应字符为AEFLNO  0xa+0xc+0x10+0x9+0xe+0xd=0xa4  数组索引分别为1 4 5 6 10 15  根据ASCII码表找出一组对应字符为ADEFJO    经验证正确。  **phase\_6**   1. 汇编代码分析            1. 具体分析   很显然是要输入六个整数。  接下来这一部分的代码对输入进行了限制。首先输入数据的范围是[1,6]。然后是一个双重的for循环，要求每个数组元素不能相同。    然后有一长串不好分析的汇编代码，这其中有一个重要的地址需要发现：0x804c13c.  打印其中内容，发现是个链表。三个数一组，第一个数是一个权值，第二个数按照123456排序下来，第三个数是下一个节点的首地址。    我们将中间那段不好分析的汇编语言跳过，直接看最后一部分。发现节点的权值要递增排列。所以猜测中间部分的汇编代码是对节点按照权值递增排列。    自行按照权值递增排列结果：  0x0a6 1  0x10e 2  0x20e 5  0x2c6 4  0x308 3  0x31b 6  所以应该输入1 2 5 4 3 6    经验证正确。  **secret\_phase**  (一)隐藏关卡如何进去  由于phase\_defused函数一直没有用到过，所以可能与这个有关,分析其汇编代码：      分析汇编代码不难发现，只有0x804c3cc中的值等于6时，才满足进入秘密关卡的第一个条件，而0x804c3cc中的值是在<read\_line>中被修改的。    又看到<\_\_isoc99\_sscanf@plt>函数的第二个参数是”%d %d %s”，所以猜测秘密关卡是跟输入两个数字的关卡联系在一起的，分别测试关卡3和关卡4，发现当关卡4输入两个数字再输入"DrEvil"后，通过关卡6会出现如下提示，说明我们已经进入了秘密关卡。    回顾一下第四次的输入：当时是输入两个整数，而这里是要输入两个整数以及一个字符串，说明我们需要在第四次输入时，输入完两个整数后还需要输入一个字符串。可能这里会有疑问，这里再输入一个字符串不会导致bomb 吗？其实是不会的，因为当时传的参数为“%d %d”，所以在读入两个整数以后便不会再读入。  或者也可以通过查看0x804c4d0中存的是关卡几输入的字符串，来找到进入秘密关卡的钥匙。    （二）隐藏关卡如何破解  下面就可以看一下<fun7>和<secret\_phase>的汇编代码了。    对应的c代码：    这个函数会用来对一颗二叉树进行查询操作，最初传入函数的那个地址就是根节点的地址，具体的返回值就像代码中写的那样：当当前节点的值等于你查询的值时候，返回0，否则根据值的大小进行不同的递归查询。当查询值大于当前节点的值的时候，递归查询右子树，否则递归查询左子树。    fun7返回值要为0，那么输入根节点的值即可，根节点值为十进制36。      经验证正确。  **实验结果及分析：**  实验完整结果为    注：答案不唯一。  在bomb.c中有这样一段代码：    这里的main函数也与我们平常写的main函数有所不同，我们写的main函数是没有参数的，而这里的main函数有两个参数。经过查阅资料：    所以，也就是说我们可以把已经破解的关卡密码存到一个文件中，然后在命令行运行bomb时加上文件名作为参数，这样bomb就可以自动读取已经破解的关卡密码，不需要重复输入了。      **收获与体会：**  当我面对满屏黑漆漆的汇编代码的时候，内心是有些小崩溃的，当我开始做实验的时候，根本不知道从哪里入手，手忙脚乱的输入指令，却没有一丝思路，内心崩溃程度50%。当终于找到路子，开始拆弹时，看到和csapp里完全不同的汇编代码，它们混乱，复杂，你会有种盲人摸象的感觉，分析了好长时间的代码，发现根本不是重点，崩溃100%。  但是，还是坚强地拆完了炸弹，写下了这篇报告。经过这一次拆弹体验，我才真正学会了调试，千言万语，尽在不言中。  所以如果你遇到困难，坚持做下去。  我总结了几个调试经验：  1.不要试图分析所有代码，分析关键，不然会很浪费时间。  2.仔细分析跳转指令，是拆弹的关键。  3.注意从call指令入手，分析传递的参数和返回值。  4.不要放弃。 | |
| 实  验成绩 |  |

实验报告撰写说明

1．实验题目和目的

请从实验指导资料中获取。

2．实验步骤和内容

包括：

（1）本次实验的要求；

（2）源程序清单或者主要伪代码；

（3）预期结果；

（4）上机执行或调试结果：包括原始数据、相应的运行结果和必要的说明（截图）；

3．实验体会

调试中遇到的问题及解决办法；若最终未完成调试，要试着分析原因；调试程序的心得与体会；对课程及实验的建议等。