计算机科学与技术学院计算机系统原理课程实验报告

|  |  |  |
| --- | --- | --- |
| 实验题目：二进制炸弹拆除 | | 学号：202000130143 |
| 班级：2020级1班 | 姓名：郑凯饶 | |
| Email：1076802156@qq.com | | |
| 实验目的：   1. 熟悉mips指令集； 2. 根据反汇编程序分析程序功能和执行流程； 3. 熟悉GDB工具，帮助程序理解。 | | |
| 实验软件和硬件环境：  DellLatitude5411  Intel(R)Core(TM)i5-10400HCPU@2.60GHz(8GPUs),~2.6GHz  Windows10家庭中文版64位（10.0，版本18363） | | |
| 实验原理和方法：   1. mips32个通用寄存器：   C:\Users\DELL\AppData\Local\Temp\WeChat Files\4213ebff5e79681e224344a84744f0f.png   1. mips常用指令：   <https://www.cnblogs.com/grandyang/p/4010469.html>   1. GDB调试：   <https://blog.csdn.net/qq_38722334/article/details/109842394?spm=1001.2014.3001.5501> | | |
| 实验步骤：  （和队友分工，我拆除的是第4,5个炸弹）  *第1个炸弹：*  C:\Users\DELL\AppData\Local\Temp\WeChat Files\7f57c1f3b8da816141927d189fa4d27.png  主要是在+32时跳转至<strings\_not\_equal>比较a0与a1寄存器中存储的字符串，查看a1发现其存储着目标字符串“Let’s begin now!”，进入phase\_1时，a0应储存着input\_strings.  第一关卡输入：Let’s begin now!  通过！  *第4个炸弹：*  C:\Users\DELL\AppData\Local\Temp\WeChat Files\da125a98df00b71e0cceff85c0818c9.png  +0-+12：是常规的开辟栈空间，将返回地址存在栈空间36(sp)，s8原先的存储内容存在32(sp)  +28：<input\_strings+240>存储在40(s8)  +72：jalr寄存器跳转指令，跳转至+60存入的-32636(gp)，应该是系统的标准输入函数  C:\Users\DELL\AppData\Local\Temp\WeChat Files\2f042283b5c70c38767866e539b2b56.png  +92：phase\_2输入应该只能是一个数，v0存储的是类似于scanf()的返回值，输入数的个数  +100：24(s8)存储输入的整数  +132：熟悉的-32660(gp)，存储的是ID\_NUMS  +140:44(v0)对应的应该是ID的最后一位  C:\Users\DELL\AppData\Local\Temp\WeChat Files\53d5939d2c1c5d2fcc4b8ef101191ea.png  +148-+152：v0&0x1以及v0&0xff实际是奇偶判断  +156：偶数时跳转，我们不选择跳转选择奇分支，ID最后一位输入1  +176：将输入的整数取出，执行func4(n)  我们后面再探讨func4。  +188：返回值存在v0，后面和8进行比较，两者必须相等，否则+204处引爆！  C:\Users\DELL\AppData\Local\Temp\WeChat Files\9a5058a6eefdbe1847702f86d39a4c2.png  跳转至+276，关卡通过！  +280-+292：复原ra以及s8寄存器，回收栈空间，返回。  Func4:  C:\Users\DELL\AppData\Local\Temp\WeChat Files\112dd9277a61b7f328034f6a71cd694.png  Func4会递归调用自身！  +20：40(s8)处存储传入参数，我们暂记为n  +32：n<2时，跳转至+108，返回1  +44-+96：计算f(n)=f(n-1)+f(n-2)，斐波那契数列！  至此，我们判断f(k)=8，k=5.  第四关卡输入：5  通过！  *第5个炸弹：*  C:\Users\DELL\AppData\Local\Temp\WeChat Files\f152a2ccbf71047a91ff107c9d7bbb4.png  +0-+20：开栈操作，a0应存储着输入，lw 设置<string\_length>输入参数  +24：跳转至<string\_length>，函数如其名，计算输入串的长度  +40：输入串的长度要求为6  C:\Users\DELL\AppData\Local\Temp\WeChat Files\02ebde9f12c775faaa1015726a8dc81.png  +192：24(s8)存储着循环变量int i  +68-+204：为循环体  +200：v0应该从0递增至5，执行6次！  C:\Users\DELL\AppData\Local\Temp\WeChat Files\872110231e424374921f945757ce309.png  +72：72(s8)存储着input\_strings+320，长度为6,可以使用GDB的x命令的/s参数进行查看  +88：lb为按字节读取操作，因为char类型的空间为1Byte，看来要进行逐字符的操作了！  +96-+116：按位与操作，根据字符的ASCII进行，提取后4位，即是0-15,结果存在12(v0)  C:\Users\DELL\AppData\Local\Temp\WeChat Files\eaefb3095536234700b190ba479aaa3.png  +144-+156：可见它将之前提取的结果取出作为偏移量，以0x41\*2^16+12524为首地址进行偏移计算，可以预见此处存储了长度为16的密码表。查看该地址，果然密码表为“isrveawhobpnutfg”！  C:\Users\DELL\AppData\Local\Temp\WeChat Files\5361b40d0777e137fab3d9213f6fe85.png  循环体结束之后，我们来到+208  +216：s8+28处存储着我们方才根据密码表及输入字符计算出的“加密串”  +288：在0x40\*2^16+10160处取出串，查看得“gaints”  +232-+248：我们输入的串经过加密后应为“gaints”  根据ASCII表，我们可以反推出输入串相应字符应和“gaints”的在同一行！  第五关卡输入：OPekma  通过！ | | |
| 结论分析与体会：  这次拆炸弹实验，我们是两个人完成的，一开始mips调试运行环境的配置队友一人搞定了，他是在他的华为云服务器上配置的，极大地方便后面汇编实验的进行。我们主要是通过看汇编代码，看指定的phase，分析其代码逻辑，这个过程开始是极其痛苦的！但是在慢慢熟悉mips32个通用寄存器以及常规的汇编指令之后，我们就轻车熟路地阅读代码，进行代码功能上的分析、猜想，解出通过的输入！congratulation！ | | |