

**课 程 实 验 报 告**

**课程名称： 计算机系统基础**

**实验名称： ELF文件与程序链接**

**院 系 ：计算机科学与技术**

**专业班级 ： 图灵2301班**

**学 号 ： U202311239**

**姓 名 ： 刘星佳**

**指导教师 ： 王多强**

**2024 年 10 月 29 日**

**一、实验目的与要求**

通过修改给定的可重定位的目标文件（链接炸弹），加深对可重定位目标文件格式、目标文件的生成、以及链接的理论知识的理解。

实验环境：Ubuntu

工具：GCC、GDB、readelf、hexdump、hexedit、od等。

**二、实验内容**

**任务** 链接炸弹的拆除

**在二进制层面，逐步修改构成目标程序“linkbomb”的多个二进制模块（“.o文件”），然后链接生成可执行程序，要求可执行程序运行能得到指定的效果。修改目标包括可重定位目标文件中的数据、机器指令、重定位记录等。**

**1、第1关 数据节的修改**

修改二进制可重定位目标文件 phase1.o 的数据节中的内容（不允许修改其他节），使其与main.o链接后，生成的执行程序，可以输出自己的学号。

**2、第2关 简单的机器指令修改**

修改二进制可重定位目标文件 phase2.o 的代码节中的内容（不允许修改其他节），使其与main.o链接后，生成的执行程序。在phase\_2.c 中，有一个静态函数 static void myfunc( ) ，要求在 do\_phase 函数中调用myfunc( )，显示信息myfunc is called. Good!。

**3、第3关 有参数的函数调用的机器指令修改**

修改二进制可重定位目标文件 phase3.o 的代码节中的内容（不允许修改其他节），使其与main.o链接后，生成的执行程序。在phase\_3.c 中，有一个静态函数 static void myfunc(int offset) ，要求在 do\_phase函数中调用myfunc(pos )，将do\_phase的参数pos直接传递myfunc，显示相应的信息。

**4、第4关 有局部变量的机器指令修改**

修改二进制可重定位目标文件 phase4.o 的代码节中的内容（不允许修改其他节），使其与main.o链接后，生成的执行程序。在phase\_4.c 中，有一个静态函数 static void myfunc(char \*s) ，要求在 do\_phase 函数中调用myfunc(s )，显示出自己的学号。

**5、第5关 重定位表的修改**

修改二进制可重定位目标文件 phase5.o 的重定位节中的内容（不允许修改代码节和数据节），使其与main.o链接后，生成的执行程序运行时，显示Class Name : Computer Foundation. Teacher Name : Xu Xiangyang。

**6、第6关 强弱符号**

不准修改 main.c 和phase6.o，通过增补一个文件，使得程序链接后，能够输出自己的学号。

#gcc -no-pie -o linkbomb6 main.o phase6.o phase6\_patch.o

**7、第7关 只读数据节的修改**

修改 phase7.o 中只读数据节（不准修改代码节），使其与main.o链接后，能够输出自己的学号。

**三、实验记录及问题回答**

**（1）实验结果及操作过程记录**

**1. 第1关**

在phase1.o中位置0x0000008e处起改为0x55,0x32,0x30,0x32,0x33,0x31,0x31,0x32,0x33,0x39,0x00。

1. **第2关**

在phase2.o中位置0x00000062处起改为0xe8,0xd9,0xff,0xff,0xff

1. **第3关**

在phase3.o中位置0x00000073处起改为0x8b,0x7d,0xfc,0xe8,0xc5,0xff,0xff,0xff

**（2） 描述修改各个文件的基本思想**

**1. 第1关**

不做任何调整时运行linkbomb1，发现程序输出了opqrstuvwxyz0123456789，然后对二进制模块phase1.o反汇编，发现.data数据段中存储了一个常量字符串abcdefghijklmnopqrstuvwxyz0123456789，因此程序输出的结果是从这个字符串中截取的，截取位置与学号有关。所以只需要把o之后的字符串修改为想要的结果U202311239即可（字符串之后一个字节设置为0x00表示字符串结尾）

1. **第2关**

反编译linkbomb2发现do\_phase函数中有一大段代码空白段，因此可以在这里实现一个call命令调用myfunc函数，即call 0x40131d，变成机器码为e8 d9 ff ff ff（e8后面跟着目标函数相对下一行指令的偏移量，为-0x27），直接在二进制文件中0x90（nop）对应的位置修改为以上机器指令即可调用myfunc。

1. **第3关**

这关与第2关很类似，传入参数只需要把参数传入edi寄存器即可，而先前参数存储在-0x4（%rbp)的位置，因此我们只要在do\_phase的空代码段处加入以下汇编：

mov -0x4(%rbp), %edi

call <myfunc>

编译成二进制文件后转化成机器码，再插入到phase3.o空代码段对应的位置即可。

1. **第4关**

**四、体会**