《软件安全》实验报告

姓名：曹瑜　　学号：2212794　　班级：密码科学与技术

实验名称：

程序插桩及 hook 实验

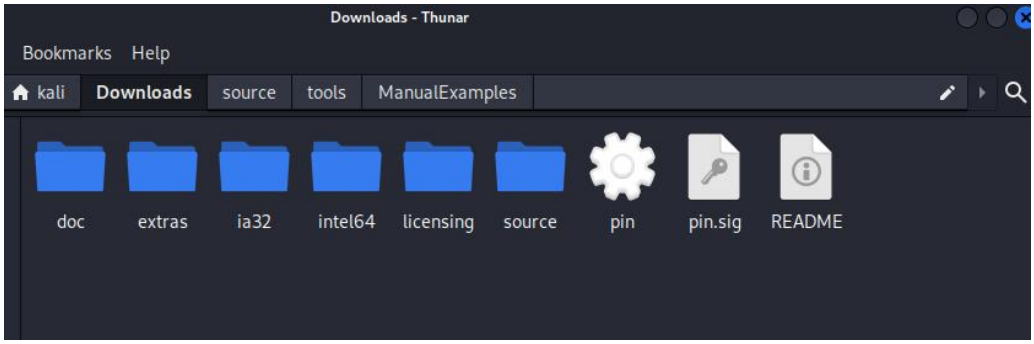
实验要求：

复现实验一，基于 Windows MyPinTool 或在 Kali 中复现 malloctrace 这个 PinTool，理解 Pin 插桩工具的核心步骤和相关 API，关注 maloc 和 free 函数的输入输出信息。

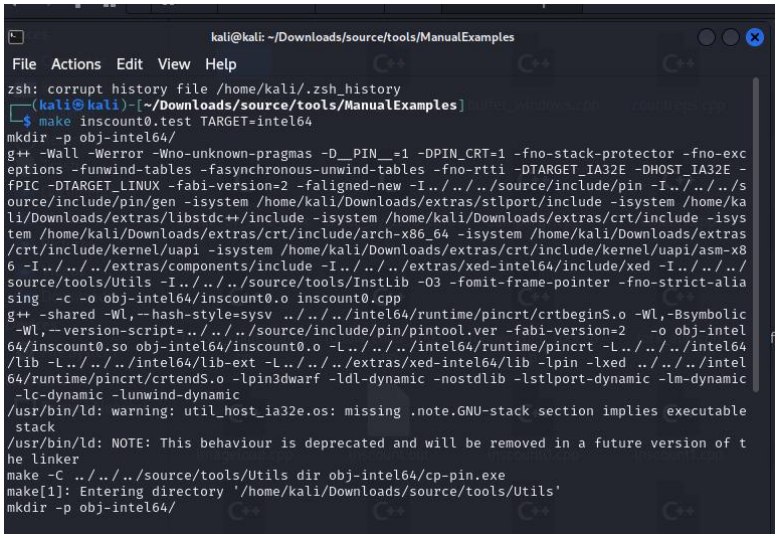
实验过程：

1、pintool 插桩调试

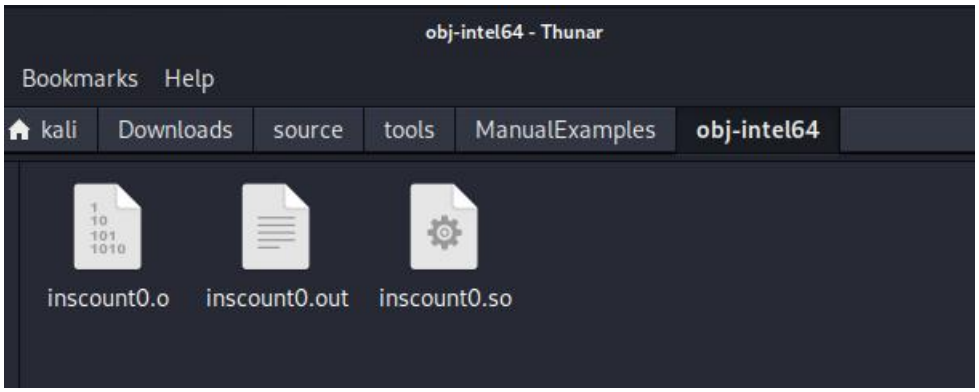
进入 kali 虚拟机完成 pintool 安装：



编译运行，产生动态链接库

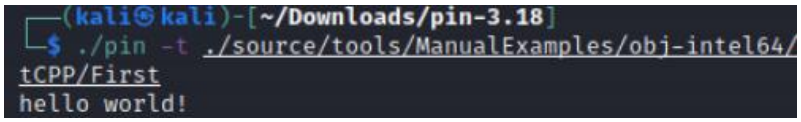


产生 inscount0.so 动态链接库

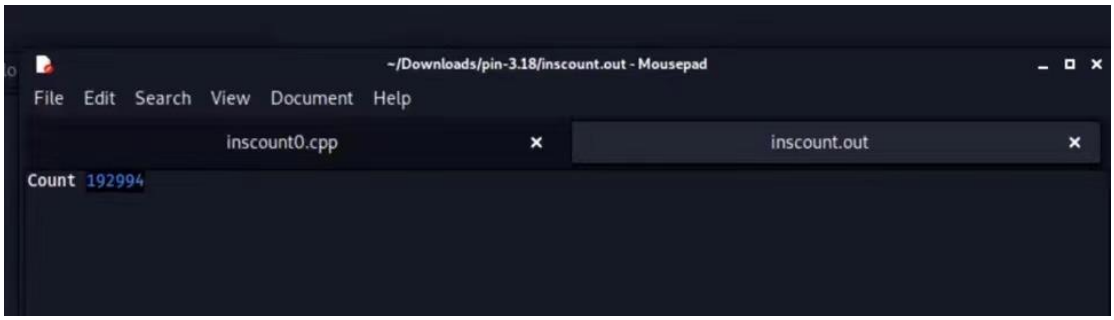


新建文件夹 testcpp，编写一个简单的“hello world！”程序并进行编译

然后对 Pin 进行插桩，对 First 执行插桩命令为：
./pin -t ./source/tools/ManualExamples/obj-intel64/inscount0.so -- ../testcpp/First
打印 hello world！ 执行成功

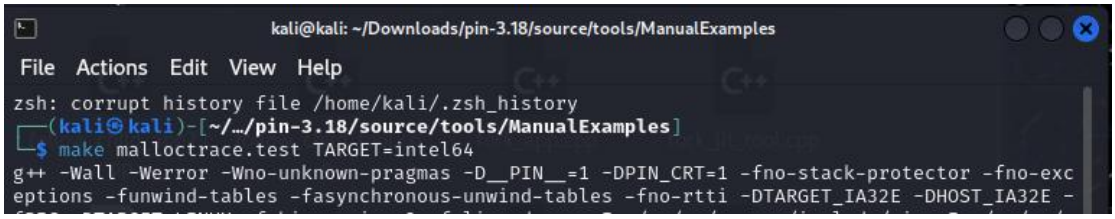


同时产生一个输出文件，文件内容为：
表示对指令数进行了插桩

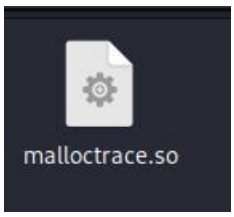


2、malloctrace 复现

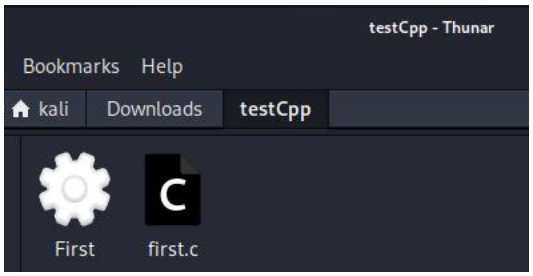
编译运行 malloctrace，产生动态链接库



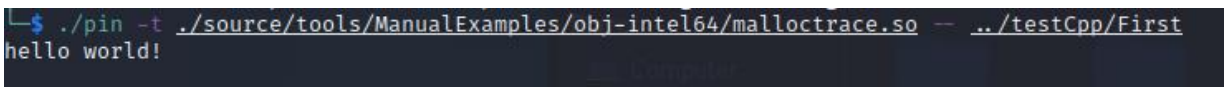
产生 malloctrace.so 动态链接库



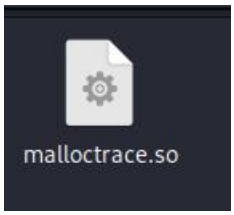
新建文件夹 testCpp，编写一个简单的“hello world！”程序并进行编译一个新的 First

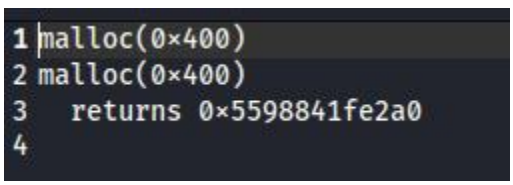


对 Pin 进行插桩，对 First 执行插桩命令为：
./pin -t ./source/tools/ManualExamples/obj-intel64/mallotrace -- ../testCpp/First
打印 hello world！ 执行成功



此时产生了 malloctrace 的输出文件，打开可见内容，可知已经进行了 Hook 插桩





心得体会：

通过本次实验，成功在 kali 虚拟机上安装了 pintool，在熟悉了 pintool 基本调试操作后，复现了 malloctrace 的代码，了解了 pin 插桩工具的核心步骤和相关 API，深入理解了 Pin 插桩工具的工作原理，掌握了如何在程序执行时动态插入代码来追踪 malloc 和 free 函数的调用。