《漏洞利用及渗透测试基础》实验报告

姓名：姜志凯 学号：2011937 班级：信安班

**实验名称：**

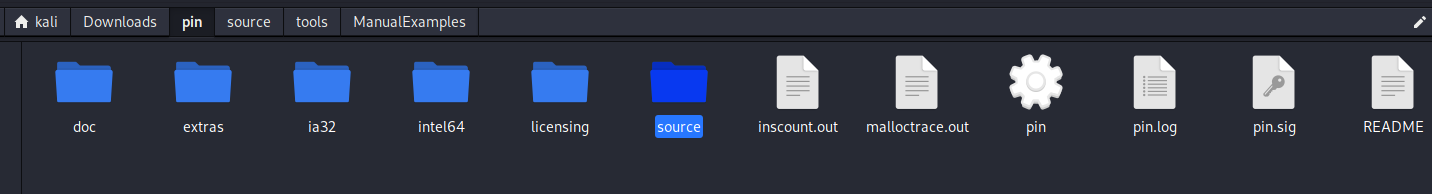
程序插桩及Hook实验

**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这个插桩工具的核心步骤及相关API，关注malloc及free函数的输入和输出信息。

**实验过程：**

下载pin压缩包，解压，复制到kali虚拟机上



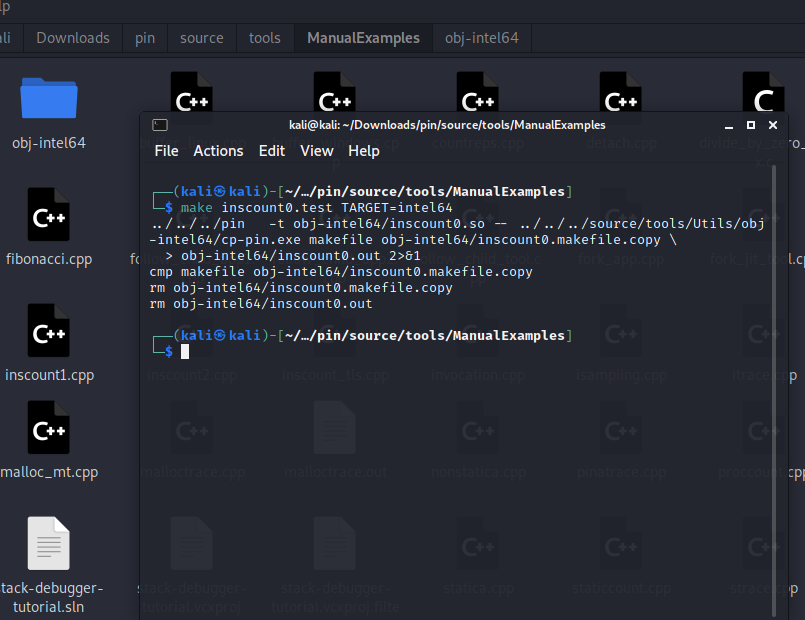
1. inscount0工具的复现：
2. 对PinTool进行编译：

·终端输入命令cd source/tools/ManualExamples进入工具源代码文件夹

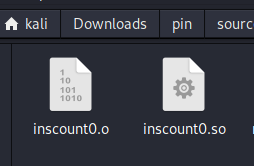
·或者直接打开ManualExamples文件，文件内打开终端

·可以输入命令make all TARGET=intel64将所有工具都编译

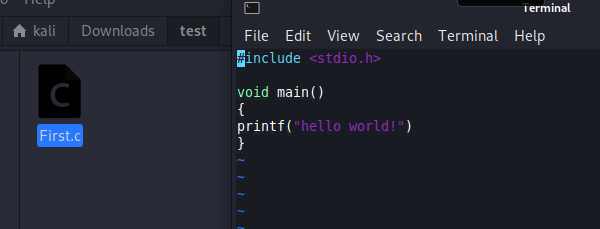
·也可以将all替换成需要的工具，只对需要的工具进行编译，如下：



可以在obj-intel64中看到编译好的动态链接库

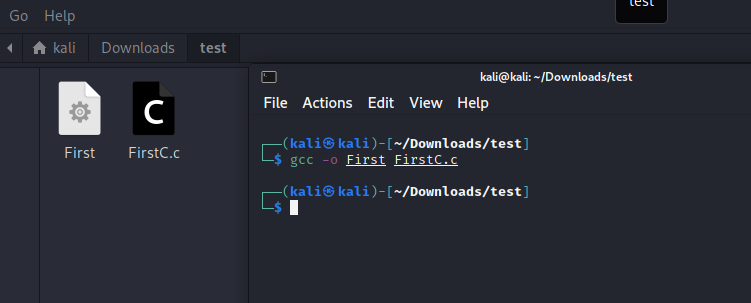


1. 新建一个简单输出hello world！的程序，用来进行插桩测试



1. 编译C程序：

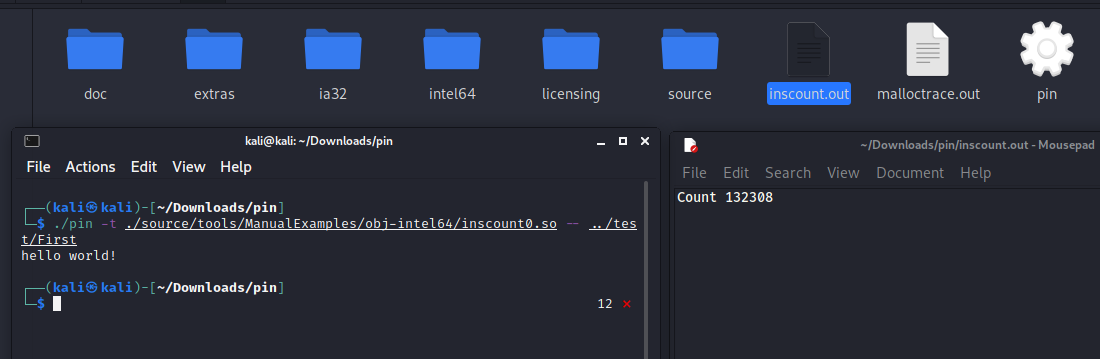
gcc –o First FirstC.c



1. 对First可执行程序进行程序插桩

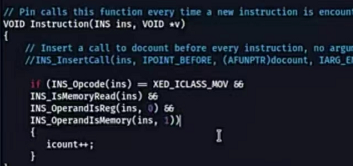
在Pin目录下的终端

Pin命令为：./pin -t ./source/tools/ManualExamples/obj-intel64/inscount0.so -- ../test/First

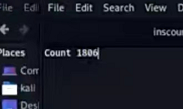


1. 也可以对工具进行修改，比如对一些特定形式的指令进行计数：

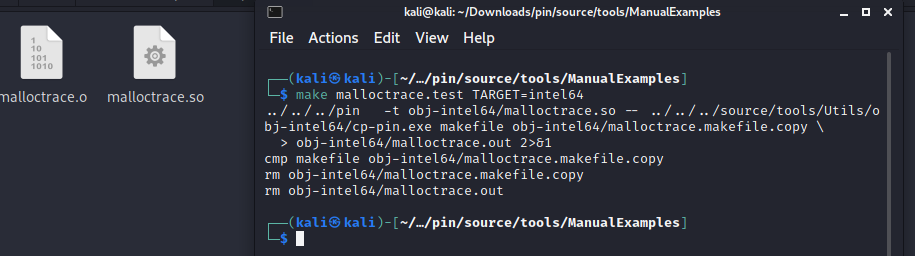
如果是mov指令、操作是读、操作数是寄存器和内存，则计数



再重新编译inscount0、对c程序进行插桩等

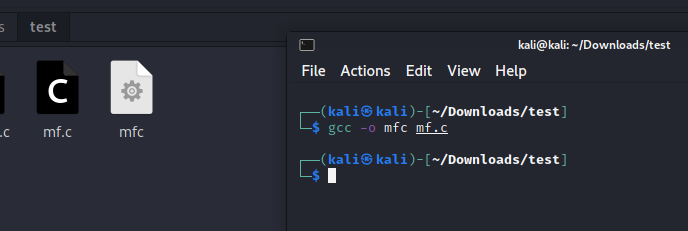


1. malloctrace工具的复现：
2. 对工具malloctrace进行编译：



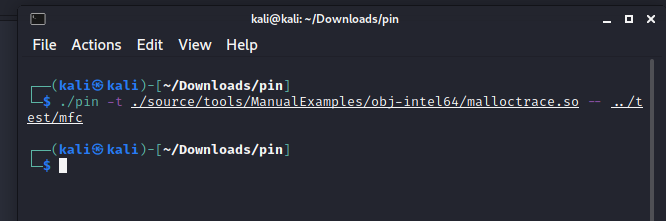
1. 新建测试程序，动态分配内存并释放，然后对程序进行编译：



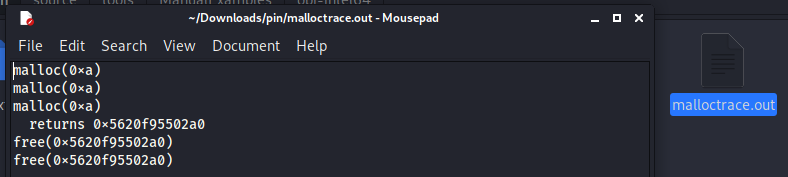


1. 对可执行程序mfc进行插桩，关注malloc和free的输入输出：

命令：./pin -t ./source/tools/ManualExamples/obj-intel64/malloctrace.so -- ../test/mfc



1. 查看输入输出信息：



**心得体会：**

Pin插桩工具核心步骤：

1. 首先根据需求选择适合的PinTool进行编译，命令为make xxx TARGET=intel64；也可以直接make all TARGET=intel64对目录下的所有工具进行编译；
2. 然后生成需要插桩的可执行程序；
3. 然后就可以对程序进行插桩了，命令格式为./pin -t ./source/tools/…/obj-intel64/xxxx.so -- TargetApp args 前边是工具的路径加上“ -- ”目标程序路径及相关参数；
4. 也可以根据需求对现成的PinTool工具进行修改，然后重新编译生成动态链接库，这样就可以实现自己想要的功能了。

此外，通过本实验，掌握了多个汇编语言的用法