

# 《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

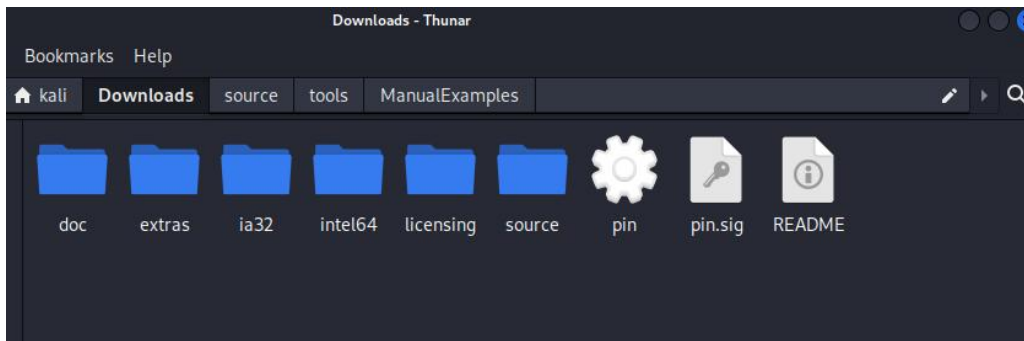
实验要求：

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 malloc 和 free 函数的输入输出信息。

实验过程：

## 1、pintool 插桩调试

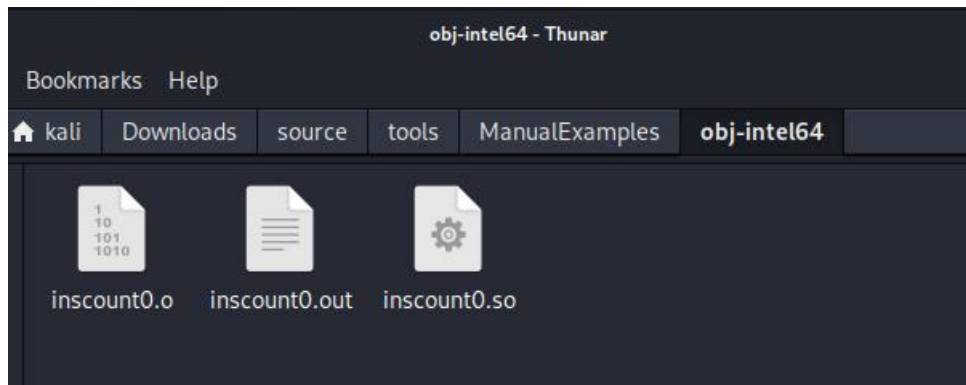
进入 kali 虚拟机完成 pintool 安装：



编译运行，产生动态链接库

```
kali@kali: ~/Downloads/source/tools/ManualExamples
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)~/Downloads/source/tools/ManualExamples
$ make inscount0.test TARGET=intel64
mkdir -p obj-intel64/
g++ -Wall -Werror -Wno-unknown-pragmas -D _PIN_ =1 -DPIN_CRT=1 -fno-stack-protector -fno-exceptions -funwind-tables -fasynchronous-unwind-tables -fno-rtti -DTARGET_IA32E -DHOST_IA32E -fPIC -DTARGET_LINUX -fabi-version=2 -faligned-new -I../..../source/include/pin -I../..../source/include/pin/gen -isystem /home/kali/Downloads/extras/stlport/include -isystem /home/kali/Downloads/extras/libstdc++/include -isystem /home/kali/Downloads/extras/crt/include -isystem /home/kali/Downloads/extras/crt/include/arch-x86_64 -isystem /home/kali/Downloads/extras/crt/include/kernel/uapi -isystem /home/kali/Downloads/extras/crt/include/kernel/uapi/asm-x86 -I../..../extras/components/include -I../..../extras/xed-intel64/include/xed -I../..../source/tools/Utils -I../..../source/tools/InstLib -O3 -fomit-frame-pointer -fno-strict-aliasing -c -o obj-intel64/inscount0.o inscount0.cpp
g++ -shared -Wl,-hash-style=sysv ../..../intel64/runtime/pincrt/crtbeginS.o -Wl,-Bsymbolic -Wl,-version-script=../..../source/include/pin/pintool.ver -fabi-version=2 -o obj-intel64/inscount0.so obj-intel64/inscount0.o -L../..../intel64/runtime/pincrt -L../..../intel64/lib -L../..../intel64/lib-ext -L../..../extras/xed-intel64/lib -lpin -lxd ../..../intel64/runtime/pincrt/crtendS.o -lpin3dwarf -ldl -dynamic -nostdlib -lstdport -dynamic -lm -dynamic -lc -dynamic -lunwind -dynamic
/usr/bin/ld: warning: util_host_ia32e.os: missing .note.GNU-stack section implies executable stack
/usr/bin/ld: NOTE: This behaviour is deprecated and will be removed in a future version of the linker
make -C ../..../source/tools/Utils dir obj-intel64/cp-pin.exe
make[1]: Entering directory '/home/kali/Downloads/source/tools/Utils'
mkdir -p obj-intel64/
```

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

```
(kali@kali)-[~/Downloads/testcpp]
$ gcc -o First first.c
```

然后对 Pin 进行插桩，对 First 执行插桩命令为：

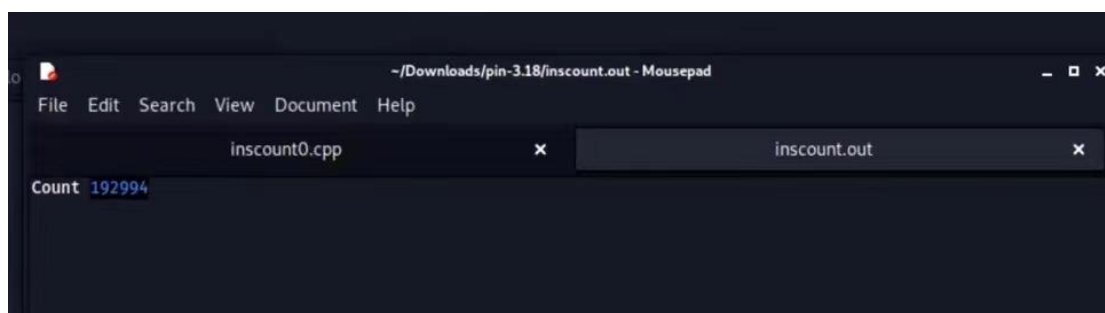
`./pin -t ./source/tools/ManualExamples/obj-intel64/inscount0.so -- ./testcpp/First`

打印 hello world! 执行成功

```
(kali@kali)-[~/Downloads/pin-3.18]
$ ./pin -t ./source/tools/ManualExamples/obj-intel64/
tCPP/First
hello world!
```

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

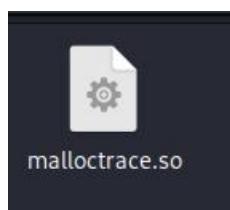


## 2、malloctrace 复现

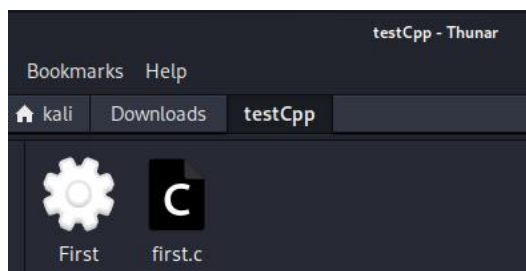
编译运行 malloctrace，产生动态链接库

```
kali@kali: ~/Downloads/pin-3.18/source/tools/ManualExamples
File Actions Edit View Help
zsh: corrupt history file /home/kali/.zsh_history
(kali@kali)~[~/pin-3.18/source/tools/ManualExamples]
$ make malloctrace.test TARGET=intel64
g++ -Wall -Werror -Wno-unknown-pragmas -D__PIN__=1 -DPIN_CRT=1 -fno-stack-protector -fno-exc
ptions -funwind-tables -fasynchronous-unwind-tables -fno-rtti -DTARGET_IA32E -DHOST_IA32E -
```

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



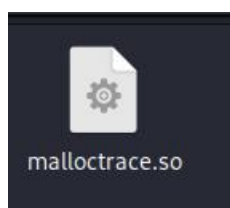
对 Pin 进行插桩，对 First 执行插桩命令为：

```
./pin -t ./source/tools/ManualExamples/obj-intel64/malloctrace -- ../testCpp/First
```

打印 hello world! 执行成功

```
$ ./pin -t ./source/tools/ManualExamples/obj-intel64/malloctrace.so -- ../testCpp/First
hello world!
```

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



```
1 malloc(0x400)
2 malloc(0x400)
3   returns 0x5598841fe2a0
4
```

### 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。