

# 《软件安全》实验报告

姓名：李雅帆      学号：2213041      班级：信安班

## 一、实验名称

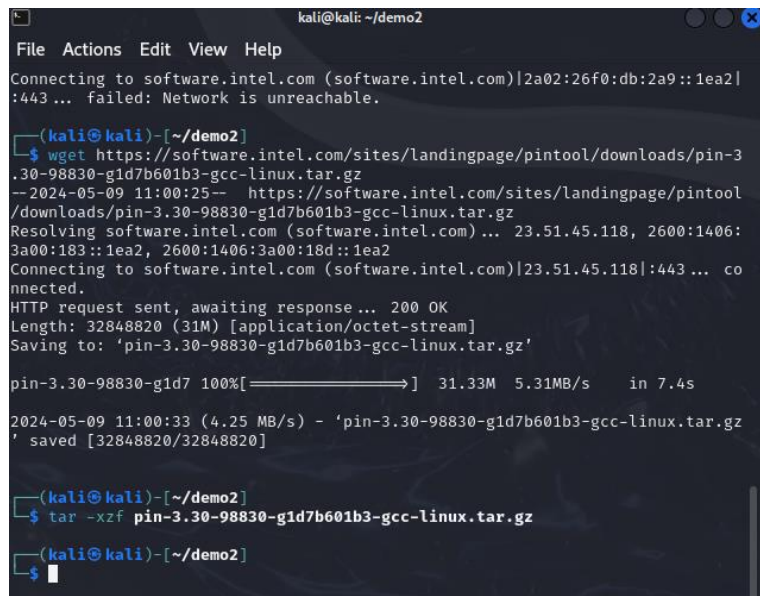
程序插桩及 Hook 实验

## 二、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。

## 三、实验过程

1. 在 kali-linux 中下载 PinTool，这里采用与 afl 同样的下载方法。下载后可以看到，demo2 中出现了 PinTool 的文件夹。Pin 中提供了一些简单的插桩函数示例。



```
kali@kali: ~/demo2
File Actions Edit View Help
Connecting to software.intel.com (software.intel.com)|2a02:26f0:db:2a9::1ea2|:443... failed: Network is unreachable.

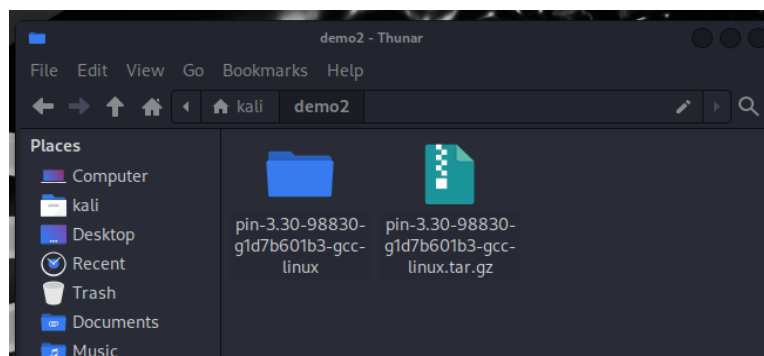
(kali@kali)-[~/demo2]
└─$ wget https://software.intel.com/sites/landingpage/pintool/downloads/pin-3.30-98830-g1d7b601b3-gcc-linux.tar.gz
--2024-05-09 11:00:25-- https://software.intel.com/sites/landingpage/pintool/downloads/pin-3.30-98830-g1d7b601b3-gcc-linux.tar.gz
Resolving software.intel.com (software.intel.com)... 23.51.45.118, 2600:1406:3a00:183::1ea2, 2600:1406:3a00:18d::1ea2
Connecting to software.intel.com (software.intel.com)|23.51.45.118|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 32848820 (31M) [application/octet-stream]
Saving to: 'pin-3.30-98830-g1d7b601b3-gcc-linux.tar.gz'

pin-3.30-98830-g1d7 100%[=====>] 31.33M 5.31MB/s in 7.4s

2024-05-09 11:00:33 (4.25 MB/s) - 'pin-3.30-98830-g1d7b601b3-gcc-linux.tar.gz' saved [32848820/32848820]

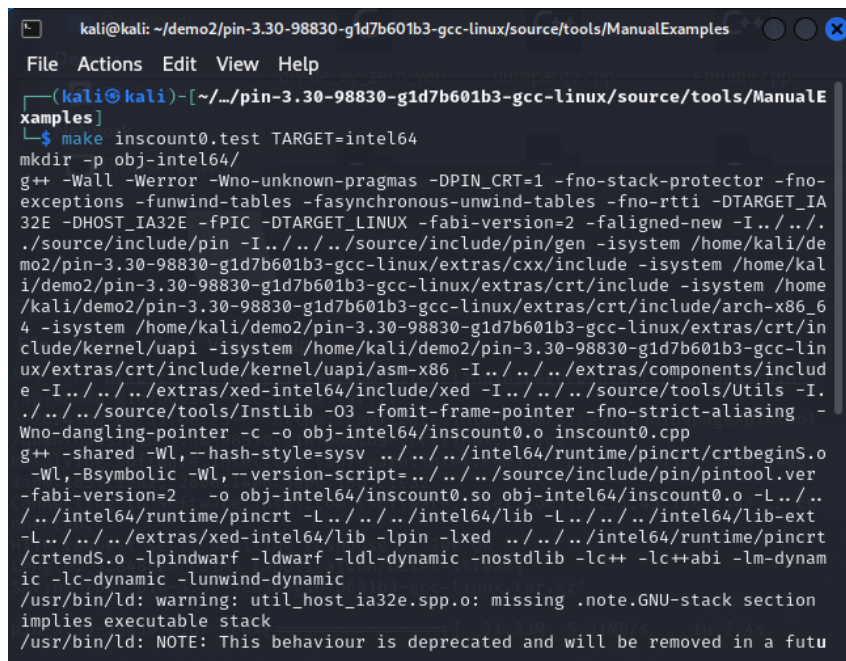
(kali@kali)-[~/demo2]
└─$ tar -xzf pin-3.30-98830-g1d7b601b3-gcc-linux.tar.gz

(kali@kali)-[~/demo2]
└─$
```



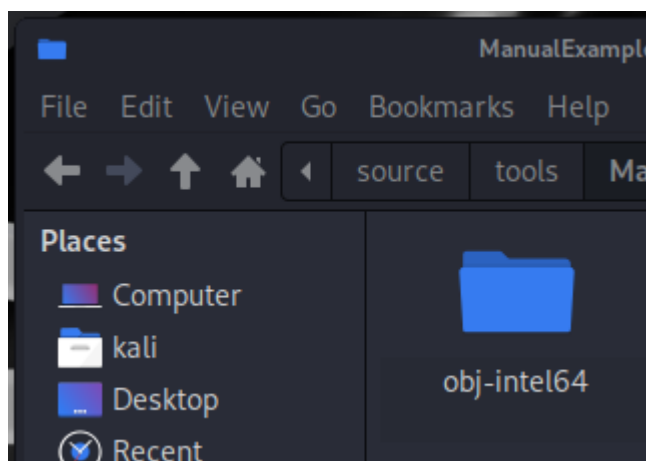
2.编译运行，产生动态链接库，使用 PinTool。

使用命令：make inscount0.test TARGET=intel64 编译 inscount0



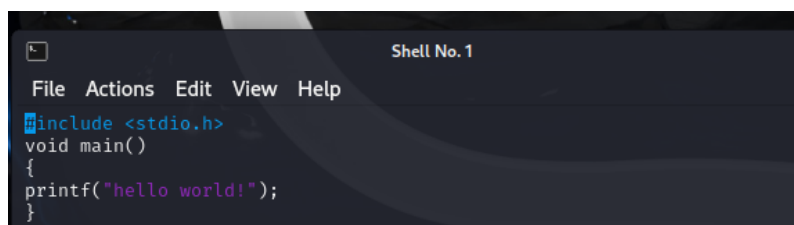
```
kali@kali: ~/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/source/tools/ManualExamples
File Actions Edit View Help
(kali@kali)~[~/pin-3.30-98830-g1d7b601b3-gcc-linux/source/tools/ManualExamples]
$ make inscount0.test TARGET=intel64
mkdir -p obj-intel64/
g++ -Wall -Werror -Wno-unknown-pragmas -DPIN_CRT=1 -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/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/cxx/include -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/crt/include -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/crt/include/arch-x86_64 -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/crt/include/kernel/uapi -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/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 -Wno-dangling-pointer -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 -lpindwarf -ldwarf -ldl-dynamic -nostdlib -lc++ -lc++abi -lm-dynamic -lc-dynamic -lunwind-dynamic
/usr/bin/ld: warning: util_host_ia32e.spp.o: missing .note.GNU-stack section implies executable stack
/usr/bin/ld: NOTE: This behaviour is deprecated and will be removed in a future
```

可以看到产生了动态链接库。



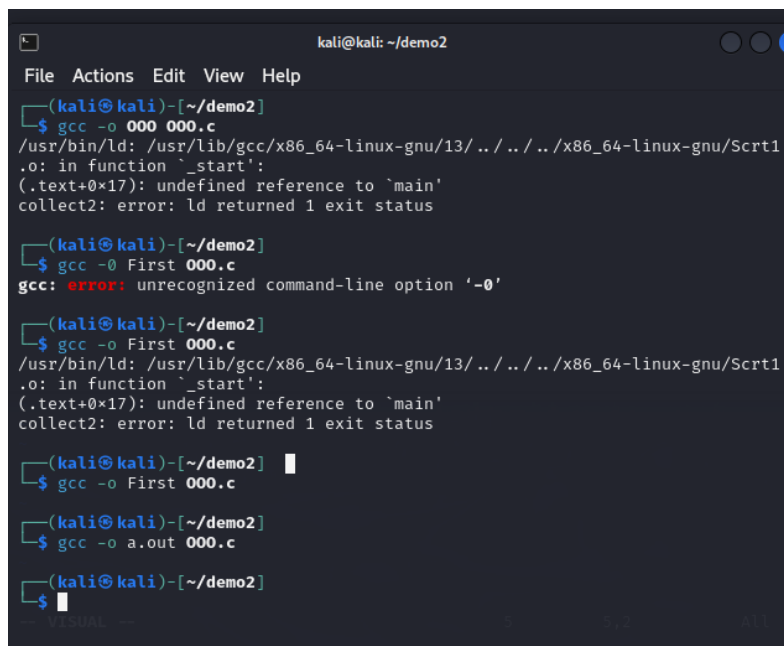
3.进行测试

编写一个简单的 hello world 程序进行测试。



```
Shell No. 1
File Actions Edit View Help
#include <stdio.h>
void main()
{
    printf("hello world!");
}
```

在 Linux 下编译该文件。



```
kali@kali: ~/demo2
File Actions Edit View Help
(kali@kali)-[~/demo2]
$ gcc -o 000 000.c
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/13/../../../../x86_64-linux-gnu/Scrt1.o: in function `_start':
(.text+0x17): undefined reference to `main'
collect2: error: ld returned 1 exit status

(kali@kali)-[~/demo2]
$ gcc -o First 000.c
gcc: error: unrecognized command-line option '-o'

(kali@kali)-[~/demo2]
$ gcc -o First 000.c
/usr/bin/ld: /usr/lib/gcc/x86_64-linux-gnu/13/../../../../x86_64-linux-gnu/Scrt1.o: in function `_start':
(.text+0x17): undefined reference to `main'
collect2: error: ld returned 1 exit status

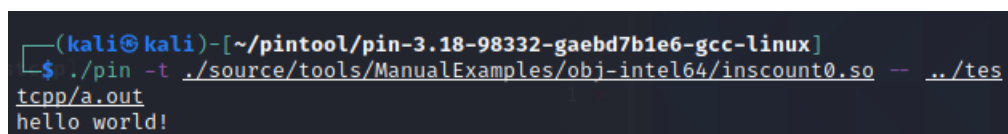
(kali@kali)-[~/demo2]
$ gcc -o First 000.c

(kali@kali)-[~/demo2]
$ gcc -o a.out 000.c

(kali@kali)-[~/demo2]
$
```

然后对 Pin 进行插桩，对 000.c 执行插桩命令为：

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



```
(kali@kali)-[~/pintool/pin-3.18-98332-gaebd7b1e6-gcc-linux]
$ ./pin -t ./source/tools/ManualExamples/obj-intel64/inscount0.so -- ./testcpp/a.out
hello world!
```

此时显示 hello world，表示执行成功。

同时产生了一个输出文件，文件内容为：



```
~/pintool/pin-3.18-98332-gaebd7b1e6-gcc-linux/inscount.out - Mousepad
File Edit Search View Document Help
count 192236
```

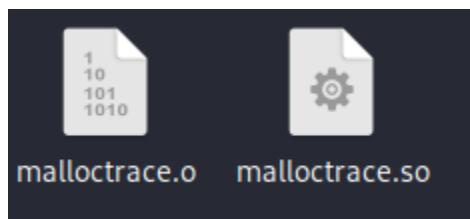
表示对指令数进行了插桩。

#### 4. 复现 malloctrace

首先进行编译运行，产生动态链接库，以使用 PinTool，这里使用命令：`make malloctrace.test TARGET=intel64` 来编译 malloctrace。

```
kali@kali: ~/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/source/tools/ManualExamples
File Actions Edit View Help
(kali@kali) - [~/pin-3.30-98830-g1d7b601b3-gcc-linux/source/tools/ManualExamples]
$ make malloctrace.test TARGET=intel64
g++ -Wall -Werror -Wno-unknown-pragmas -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/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/cxx/include -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/crt/include -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/crt/include/arch-x86_64 -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/extras/crt/include/kernel/uapi -isystem /home/kali/demo2/pin-3.30-98830-g1d7b601b3-gcc-linux/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 -Wno-dangling-pointer -c -o obj-intel64/malloctrace.o malloctrace.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/malloctrace.so obj-intel64/malloctrace.o -L../..../intel64/runtime/pincrt -L../..../intel64/lib -L../..../intel64/libext -L../..../extras/xed-intel64/lib -lpin -lxd ../..../intel64/runtime/pincrt/crtendS.o -lpindwarf -ldwarf -ldl-dynamic -nostdlib -lc++ -lc++abi -lm-dynamic -lc-dynamic -lunwind-dynamic
/usr/bin/ld: warning: util_host_ia32e.spp.o: 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
```

此时可以看到，已经产生了动态链接库。



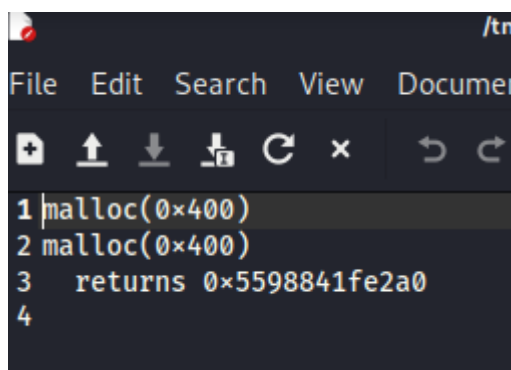
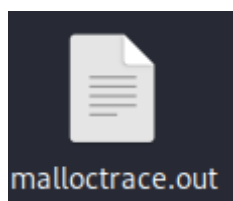
为了避免混淆，此时新建一个 First.c 的文件，同样输入 hello world 程序，并：对 First 可执行程序进行程序插桩，Pin 命令为：

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

我们可以观察到，同样输出了 hello world，表明执行成功。

```
(kali@kali) - [~/tmp/mozilla_kali0/pin-3.18-98332-gaebd7b1e6-gcc-linux]
$ ./pin -t ./source/tools/ManualExamples/obj-intel64/malloctrace.so -- ./testCpp/First
hello world!
```

产生了 malloctrace 的输出文件，并查看输出文件，此时已经进行了 Hook 插桩。



#### 四、心得体会

这次实验让我深入了解了程序插桩及 Hook 技术，在使用 Windows MyPinTool 或 Kali 中的 malloctrace 这个 PinTool 进行实验的过程中，我对 Pin 插桩工具的核心步骤和相关 API 有了更深入的理解。

我学会了如何使用 Pin 工具进行初始化操作，通过 PIN\_Init 来初始化，然后注册插桩函数，利用 IMG\_AddInstrumentFunction 实现镜像级插桩，使得在原始程序的每条指令执行前都能够进入我们注册的插桩函数中。此外，通过 PIN\_AddFiniFunction 注册程序退出时的回调函数，进行一些必要的结束处理。最后，我学会了使用 PIN\_StartProgram 启动需要进行插桩的程序。

在实验过程中，我还熟悉了一些相关的 API，例如 RTN\_FindByName 和 RTN\_InsertCall。RTN\_FindByName 能够找到执行函数的标识，而 RTN\_InsertCall 则是用来注册一个回调函数，在某一指令执行时执行指定的操作。这些 API 的灵活运用使得我们能够对程序进行更加精细的插桩和 Hook 操作，从而实现对程序行为的监控和控制。