1. **实验指导书中的行为检测样本是什么意思？**

**答：**提供5个待检测的可能存在恶意的Exe样本，后面的“复现(覆盖)被检测的行为”去掉。需要大家提供样本验证自己的检测算法或者功能。

1. **本次课设需要虚拟机靶机吗？**

答：本次课设完全可以基于Win10，VS2019开发，不用安装虚拟机。

1. **编译提示目录不合法或者\*.dll \*.exe无法写入，怎么办？**

**答：一般跟本机安装的安全工具有关，可以修改目录名，用360右键解锁dll/exe锁定,删除exe/dll解决。或则干脆卸载360,注意退出360它的内核还在工作。**

**注意：杀毒软件可能干涉钩子代码的执行，遇到目录 文件不可写的问题，建议把杀毒软件直接卸载。**

1. **注册表操作到底有哪些API呢？**

答：这个实现基本的注册表值的添加与写入截获基本可以符合要求。具体API可自行查找资料解决，关键字“注册表操作API”

1. **我发现HeapAlloc调用太频繁了，达到上万次，根本无法及时保存判断申请与释放是否一致**

答：由于HeapAlloc调用过于频繁，截获的频率太高，建议采用以下方法之一处理：

1. 在HeapCreate中记录句柄，后面的HeapAlloc判断句柄值，只处理前面自己记录的句柄，减少HeapAlloc处理范围。
2. 使用HeapCreate与HeapDestroy的句柄来做申请释放统一性或者重复释放问题检查。
3. **其它API都可以截获，为撒socket函数就是不行？**

答：注意Windows系统标准API Socket以WSA开头，比如：WSASend WSARecv，以这个为准作为截获目标。Send，recv，connect并不是标准API。

1. **我明明使用了共享内存，跟踪的时候确定把输出消息存进共享变量，但读取的时候为空，为什么？**

答：注意检查声明共享变量的时候，要初始化。否则编译器会把该变量编译为非共享变量。

#pragma data\_seg("MySeg")

char seg[1000][256] = {};

int count = 0;

volatile int HeapAllocNum = 0;

std::mutex mtx;

#pragma data\_seg()

#pragma comment(linker, "/section:MySeg,RWS")

不能是：

#pragma data\_seg("MySeg")

char seg[1000][256] ;

int count ;

volatile int HeapAllocNum ;

std::mutex mtx;

#pragma data\_seg()

#pragma comment(linker, "/section:MySeg,RWS")

1. **我的目标进程(被注入的进程)出不来了，昨天还可以出来啊，怎么办？**

答：往往是改的代码导致。首先检查直接运行TestApp.exe,是否可以执行，如果可以，则是Dll的问题，在DllMain里面把所有挂钩的代码都注释掉，重新编译，一般就可以了。在这种情况下，然后一条一条代码打开注释，找出出问题的那条代码解决！

1. **我新建一个工程，加入了Detours.h,库文件也加入了，但是编译一堆错误啊！怎么办？**

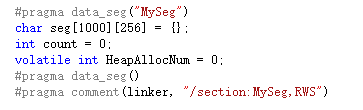
答：注意detours.h要放在其它.h文件后面，否则会有一堆错误的。因为detours用了很多需要事先定义好的类型。

1. **printf怎么没看到输出信息呢？**

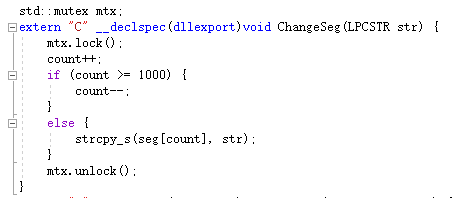
答：Dll中的printf语句在被注入的进程中运行，如果被注入的进程不是控制台程序，就无法输出。建议：

1.使用OutputDebugString输出调试语句，用Dbgview.exe查看该函数的输出；

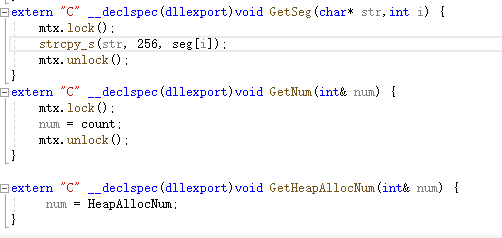
2.在Dll中设置一个进程间共享缓冲区，所有输出放入该缓冲区，注射程序通过读取该缓冲区获得调试输出：



注意做好数据操作的互斥操作：

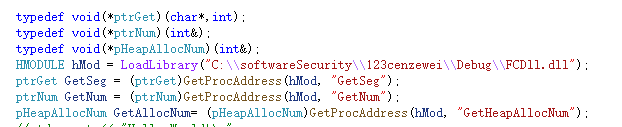


再设计一个获取这个共享内存信息的函数：



在注射器程序中调用获得被截获的API的调试输出。

先定义函数指针：



然后读取消息：



1. **指导书中的代码都是正确的吗？**

答：指导书前面部分的代码是针对VC6的代码,使用的Detours版本比较低，在VS2019编译无法通过，请参照指导书中最后的代码。注意该代码可以完成基本的MessageBoxA/W截获，主要是大家参考其基本运行原理。在扩大到其它API的时候，请仔细检查代码，遇到问题单步跟踪，排除问题。

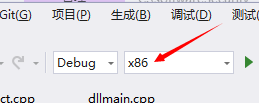
1. **我发现有截获API输出信息比预想要多的多，陷入死循环了，怎么办？**

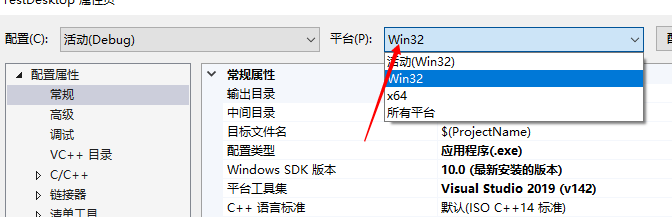
答：可能有两个原因：

1是注射器程序自身加载了Dll，导致自身的API被截获了，输出了意外的截获信息，建议在DllMain里面判断一下是否为注射器程序自身，如果是，则不要开启Detours的API截获。比如可以通过GetModuleFileName(0, Buf, 256)得到当前进程名，比较判断是否为注射器名字即可。

2.截获的API存在重入的问题，部分函数存在递归调用。导致输出无穷无尽。比如:可能在自己定义的HeapAlloc的截获函数中使用了printf,但printf函数实现中，使用了HeapAlloc,又被截获下来，这样就进入了无穷递归了。建议1.判断自己定义的HeapAlloc函数是否被重入，如果被重入，直接调用原始HeapAlloc直接返回，不要输出即可。 2.更改输出方式，避免重入函数的调用。

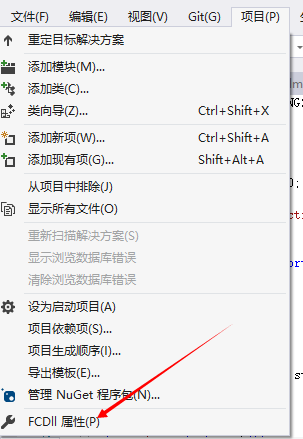
1. **我明明设置了包含目录(将inlcude目录加入Detours的include文件夹所在的路径)，也将库文件所在的目录加入，但为什么还是编译提示没有定义或者函数没有定义？**

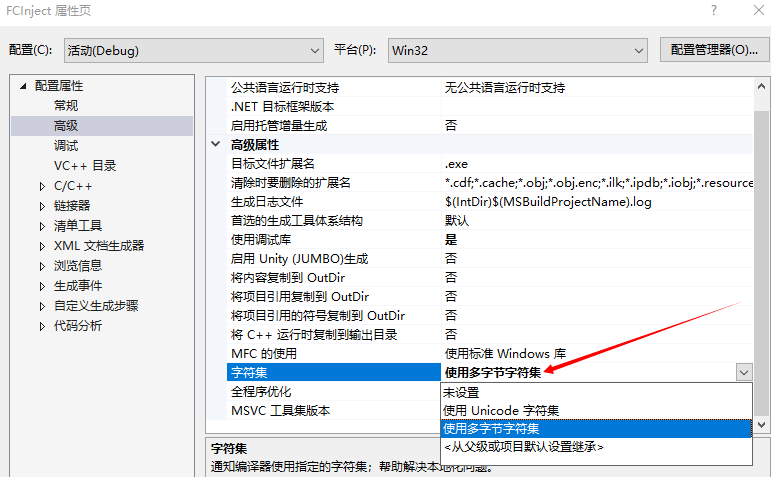
答：注意我们的实例代码是针对x86，注意检查编译目标是不是x86，不要选为x64,类似在配置项目的包含与库目录的时候，也要对应选中相同类型的目标win32，不要选为x64：



1. **为什么我使用strcpy之类标准的c语言函数编译都通不过啊？**

答：涉及到字符串的操作注意新建的工程默认为Unicode字符集，只要函数涉及到字符串操作，一般使用Unicode字符串，要做转换才能使用。如果想按照原来标准的c语言方式使用，可以在项目属性中把字符集改为使用多字节字符集。





1. **课设要求有UI吗？**

答：UI属于课程设计中单个计分点，UI可以用命令行，也可以使用图形化GUI，本课程设计是综合的安全工具开发，不同于算法竞赛、攻防竞赛，目标定位于开发出来的程序需要给用户使用，UI是使用的接口，这是软件设计的一个环节。

Btw：大家可以使用自己习惯的工具，比如：QT，python等，写UI。并且建议将行为分析、截获实时展示都放入该UI。

1. **每类行为都有A W两种，都要实现吗？**

答：作为本课程设计，实现其中一种即可。但要求测试的样例也必须是A。

1. **本课程设计被注入的目标程序可以是第三方的进程吗(不是自己开发的)？**

答：当然可以。本课程设计的最终目的就是想去分析现有的一些进程行为，如果大家开发的程序能够实战，用来分析第三方的进程，说明开发成果更具有兼容性与应用价值，建议给予加分。