PFSafetyGuard（改进版）是一个基于DLL注入与API Hook的行为监控系统，用于检测Windows程序运行过程中的可疑或恶意API调用行为。项目通过共享内存通信与可视化界面联动，实现了实时监控与安全预警功能。

编译说明

本项目使用 Visual Studio + Qt 开发，依赖项如下：

- Qt Creator 16.0.1

- Detours（微软开源 DLL Hook 库，已在项目中）

- Windows SDK

- Visual Studio 2022

建议使用Release模式进行编译，确保各模块路径正确配置（DLL路径、注射器路径等）。

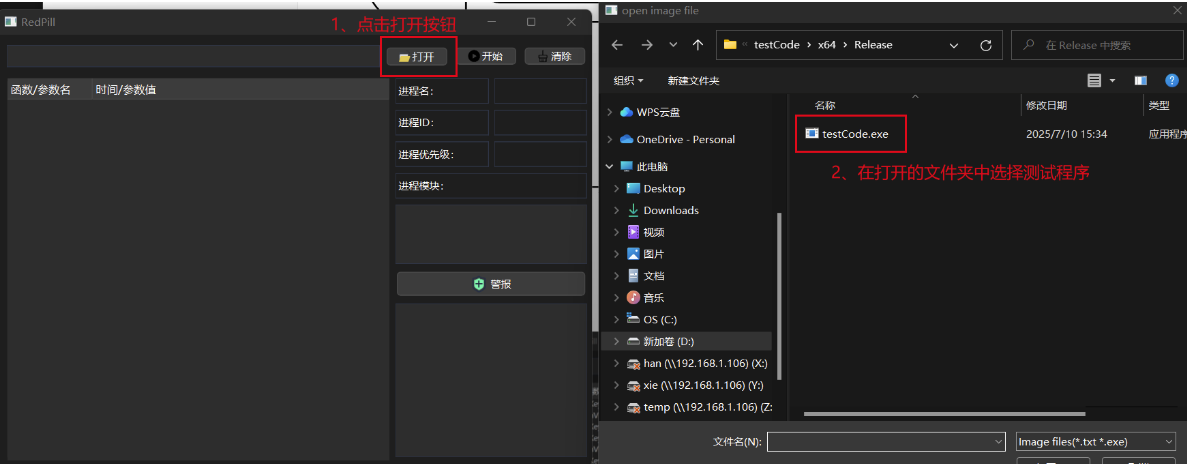
操作说明

将项目解压到D盘根目录，即确保是这样的目录结构：

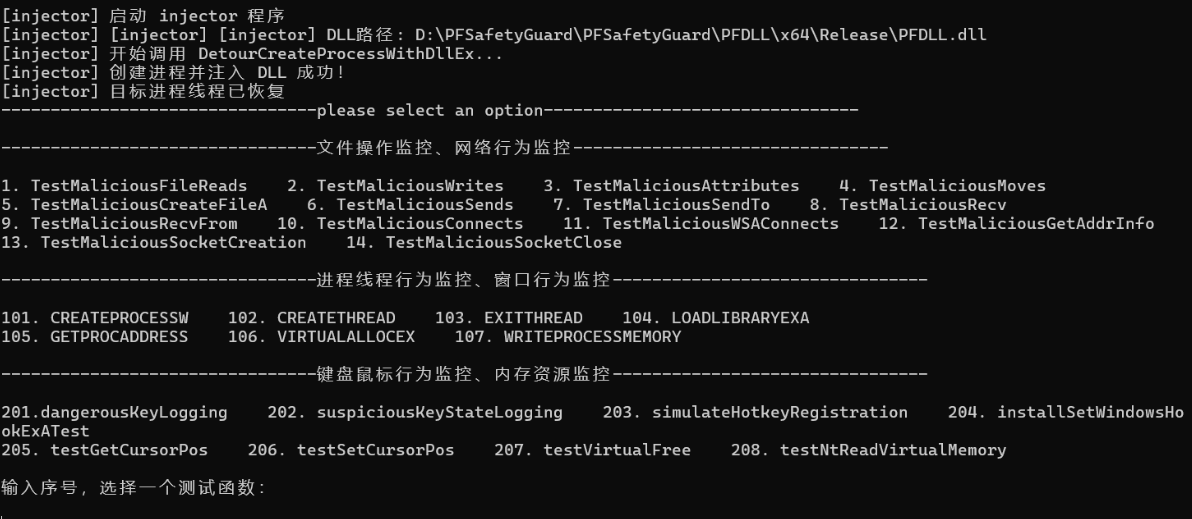


这样就不需要再改变其他设置即可运行程序。

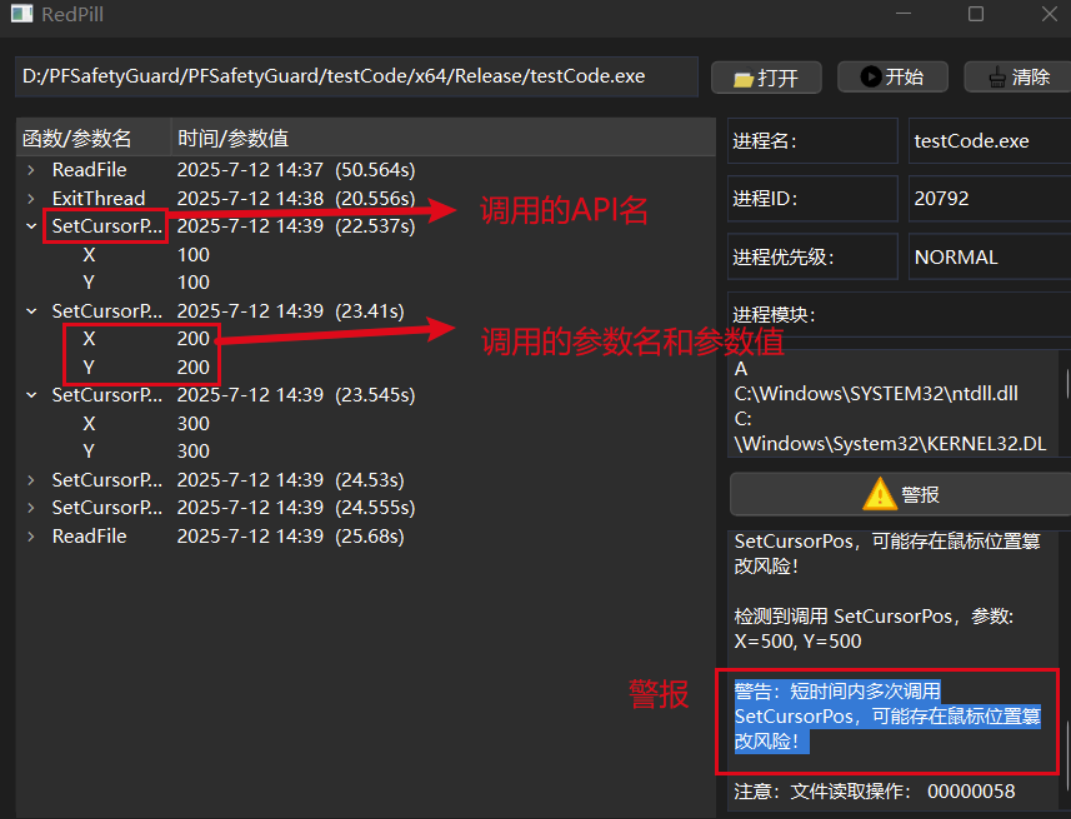
在Qt中运行项目，并按下面操作：



随后点击开始按钮，打开测试程序：



选择一个恶意函数的序号，该工具会记录调用的Windows API的参数并在检查后给出警报：



项目特点

- DLL注入技术：使用Detours框架实现对目标程序的API拦截。

- 可视化界面：基于Qt开发，实时展示进程信息、模块信息及API调用日志。

- 安全检测逻辑：内置对键盘记录、钩子安装、热键劫持、内存操作等敏感行为的检测规则。

- 共享内存通信：通过信号量与共享内存模块实现注入程序与界面模块的数据交换。

项目结构

PFSafetyGuard/

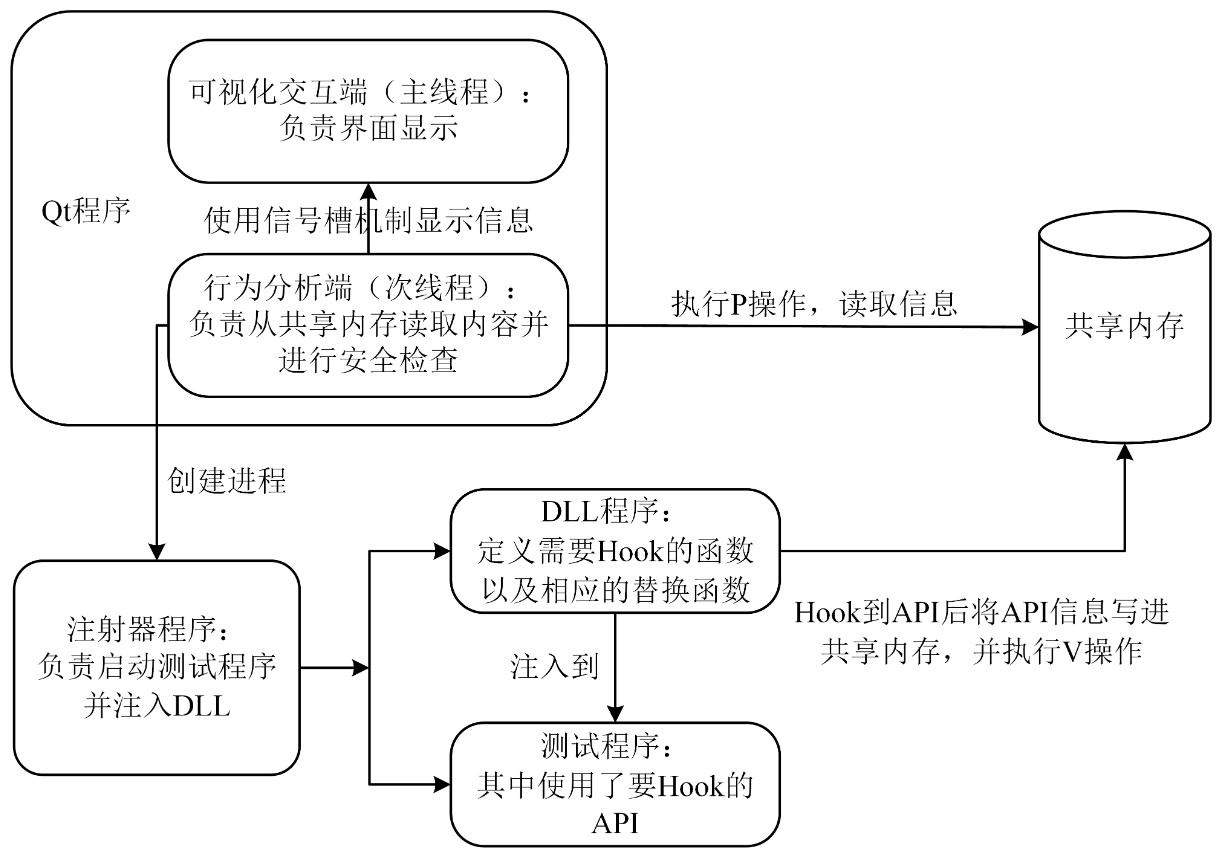
├── PFDLL/ # DLL注入模块（Hook API并写入共享内存）

├── syringe/ # 注射器程序（将DLL注入目标进程）

├── testCode/ # 模拟恶意行为的测试程序

├── PFSafetyGuard/ # Qt可视化项目（展示监控数据）

└── README.md # 项目说明文档



实现功能

1. 自动注入目标程序并监控其运行过程；

2. 实时记录并显示敏感API的调用（如：GetAsyncKeyState、SetWindowsHookExA等）；

3. 对典型恶意行为发出警报提示（如键盘监听、热键注册、内存访问等）；

4. 支持查看目标进程模块、进程ID、优先级等基本信息；

5. 安全检测策略可拓展，具备一定的可维护性和可移植性。

测试用例举例

通过模拟程序触发以下 API，测试系统的检测能力：

- GetAsyncKeyState, GetKeyState：模拟键盘记录器；

- RegisterHotKey：模拟快捷键劫持行为；

- SetWindowsHookExA：模拟钩子注入；

- GetCursorPos, SetCursorPos：鼠标行为监控；

- VirtualFree, NtReadVirtualMemory：模拟内存操作行为。