五种注入方式：

1. **注册表注入：**

注册表注入的主要原理就是AppInit\_Dlls键值（即动态库的路径）会在调用LoadLibrary(user32.dll)时加载。

当应用程序被加载时，会将User32.dll映射到进程空间内，这时会触发User32动态库的DllMain函数中的DLL\_PROCESS\_ATTACH，进而查看注册表HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows目录下的AppInit\_DLLs是否存在需要加载的动态库路径，同时查看LoadAppInit\_DLLs的值是否为1。如果为1，那么将根据路径加载所有的动态库，完成注册表的注入。

所需的主要函数：RegOpenKeyEx打开指定注册表；RegSetValueEx将指定的动态库的路径设置在键值上；RegCloseKey关闭关闭指定注册表项的句柄。

1. **远程线程注入：**

远程线程注入的主要原理就是在指定进程中启动一个线程来执行LoadLibrary函数，加载指定的动态库路径。

在输入指定的进程名称后NtQuerySystemInformation来获取进程ID，之后通过GetProcessImageFileNameW获取完整路径。在对方进程中申请内存来保存动态库的完整路径。

最后在指定进程中创建线程来执行LoadLibrary函数，执行在指定进程中申请的内存，即动态库的完整路径。

所需的主要函数：CreateRemoteThread在指定进程中创建线程。

1. **Hook链注入：**

Hook链注入的主要原理就是通过消息机制，SetWindowsHookEx函数，在消息处理函数之前截获消息将动态库注入。SetWindowsHookEx函数把一个应用程序定义的钩子子程安装到钩子链表中时总是在Hook链的开头安装Hook子程。

同样是需要在输入指定的进程名称，NtQuerySystemInformation来获取进程ID，之后通过GetProcessImageFileNameW获取完整路径。在对方进程中申请内存来保存动态库的完整路径。最后加载动态库获取动态库中的导出函数（也就是挂钩处理函数），使用SetWindowsHookEx函数进行注入。

所需的主要函数：SetWindowsHookEx将应用程序定义的挂钩程安装到挂钩链中；UnhookWindowsHoo：删除[SetWindowsHookEx](https://learn.microsoft.com/zh-cn/windows/desktop/api/winuser/nf-winuser-setwindowshookexa)数安装在挂钩链中的挂钩过程。

1. **劫持Eip：**

劫持Eip的主要原理就是通过挂起线程，执行在指定进程中写入的ShellCode来进行注入。

同样是需要在输入指定的进程名称，NtQuerySystemInformation来获取进程ID，之后通过GetProcessImageFileNameW获取完整路径。在对方进程中申请内存来保存动态库的完整路径。

之后在指定进程中申请内存空间并写入相应的ShellCode，ShellCode中存放着动态库的完整路径以及LoadLibrary函数地址。挂起对方线程后获取到Eip，执行ShellCode，之后恢复线程并且还原之前的Eip以及线程上下背景文。

1. **APC注入：**

APC注入的主要原理就是通过QueueUserAPC函数将LoadLibrary函数插入APC队列中。当程序执行SleepEx、WaitForSingleObjectEx、WaitForMultipleObjectsEx等函数时，进入可提醒状态，系统会先检查APC队列有没有项，如有，系统不会让线程进入睡眠状态。没有时才会挂起。

同样是需要在输入指定的进程名称，NtQuerySystemInformation来获取进程ID，之后通过GetProcessImageFileNameW获取完整路径。在对方进程中申请内存来保存动态库的完整路径。

最后向目标进程中写入动态库完整路径，向目标进程的各个线程的APC队列中插入执行体。

所需的主要函数：QueueUserAPC把一个APC对象加入到指定线程的APC队列中。