[ScyllaHide] 02 InjectorCLI源码分析

原创

夜猫逐梦 于 2021-12-21 19:14:35 发布 35 收藏
分类专栏： # Windows 文章标签： 逆向 反调试

[ScyllaHide] 文章列表-看雪地址：

InjectorCLI源码分析

从项目名字，我们可以看出，该项目是注入功能的命令行程序。它实现了HookLibrary.dll的命令行注入，并启动反调试功能。

面对一个开源的、强大的反调试工程，我做了几个简单的测试：

1. 执行命令：`InjectorCLIx86.exe MyTestAntiDebuger.exe HookLibraryx86.dll`，将HookLibraryx86.dll注入MyTestAntiDebuger.exe进程，完成反调试测试环境搭建。
2. 使用procexp.exe查看进程MyTestAntiDebuger.exe的dll，我们并没有发现HookLibraryx86.dll，猜测InjectorCLI含有自加载dll的功能。
3. 使用PCHunter检测MyTestAntiDebuger.exe进程发现只hook了下面地址：

```
7781EA50  - FF25 28B28B77   jmp     dword ptr [Wow64Transition]      ; wow64cpu.77737000
```

那么问题来了，这一个hook如何实现各种反调试及时呢？

4. InjectorCLIx86.exe 能实现隐藏dll的注入，尝试注入其它dll（自己随便写了个MessageBox的dll）岂不是一个很好的反检测功能？然而尝试结果是，直接崩溃了。为什么呢？？？

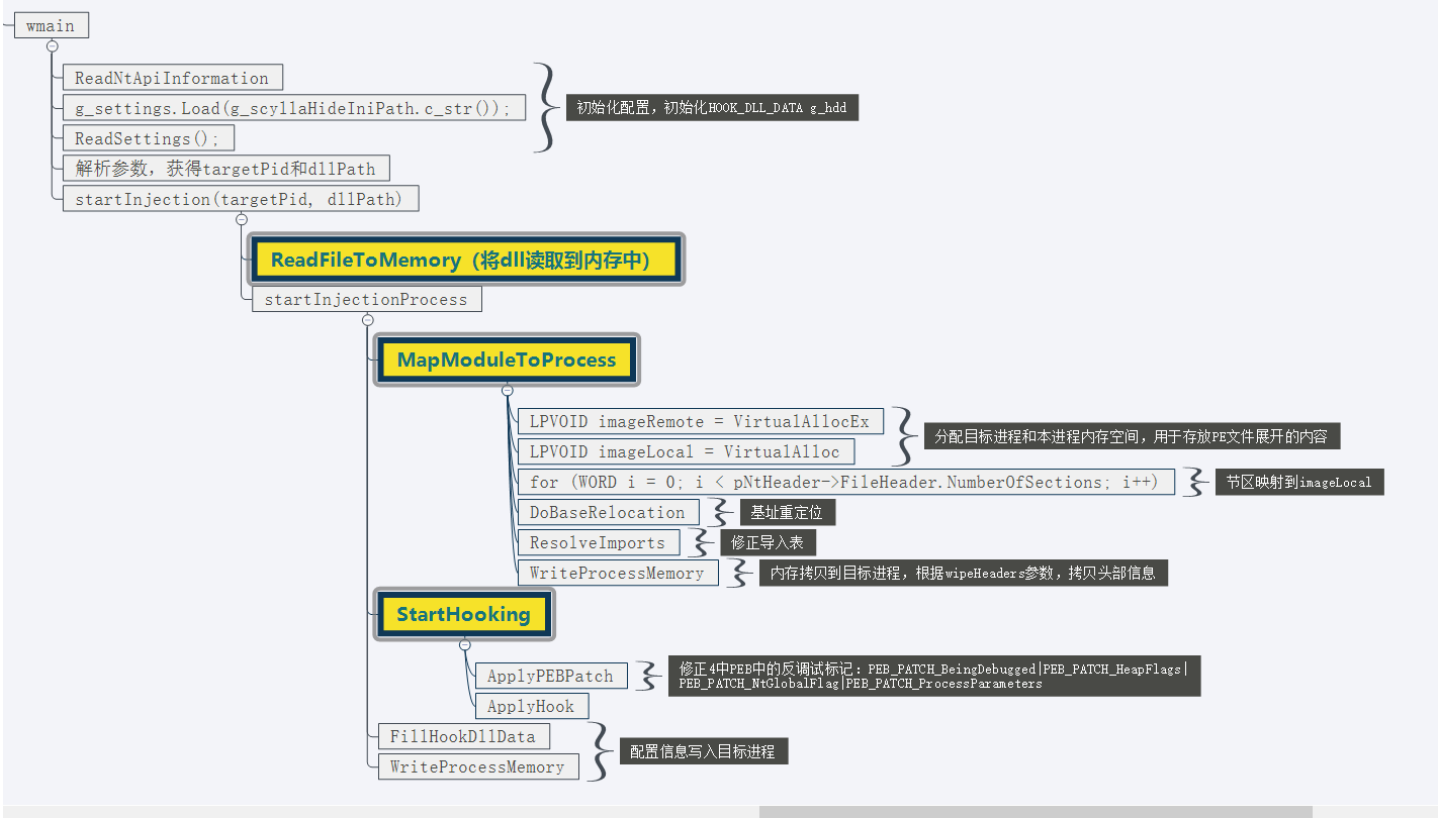
带着这些问题，我们继续走下去

章节目标

- 了解InjectorCLI整体函数调用关系
- 了解InjectorCLI注入原理（需要有PE文件基础），限制条件。
- InjectorCLIx86.exe为何不能注入其它dll
- 注入后如何实现反调试

源码分析

先上一张总的函数调用图：



wmain分析

wmain做的三件事儿：

1. 初始化配置，初始化HOOK_DLL_DATA g_hdd
2. 解析参数，获得targetPid和dllPath
3. startInjection(targetPid, dllPath)

初始化配置，包含scylla_hide.ini和NtApiCollection.ini，都是基于Windows ini相关api进行操作的。

scylla_hide.ini先通过Section[SETTINGS]的键CurrentProfile的值获得配置类型。

```
[SETTINGS]
CurrentProfile=VMProtect x86/x64
[VMProtect x86/x64]
BlockInputHook=0
DLLNormal=1
...
[Obsidium x86]
BlockInputHook=0
DLLNormal=1
DLLStealth=0
...
```

比如上面的配置，程序将读取Section[VMProtect x86/x64]中所有的键值作为配置项。

startInjection做的三件事（黄色底色的内容）

1. ReadFileToMemory（将dll读取到内存中）
2. MapModuleToProcess（dll注入到目标进程）
3. StartHooking（执行反调试逻辑）

MapModuleToProcess的逻辑是本章的重点，代码量并不多，不过需要扎实的PE知识（可以通过Windows PE权威指南获得相关知识，很全面）。该函数主要有4个步骤，如下：

1. 节区映射到imageLocal
2. 基址重定位
3. 修正导入表
4. 内存拷贝到目标进程，根据wipeHeaders参数，拷贝头部信息

通过MapModuleToProcess将dll映射到目标进程，但是并未在目标进程执行任何代码；而StartHooking将dll变成了有意义的任务（反调试）该内容将在下节进行阐述。
为了分析方便，我将Scylla隐藏注入相关代码提了出来，上传到了 github 上，有兴趣的可以去看下。

问题回顾

现在回过头看下文章开头的问题。

问题一：了解InjectorCLI注入原理（需要有PE文件基础），限制条件。

Scylla的dll注入对PE文件进行了重定位，是否适用于所有的dll呢，从代码中我们可以发现，还有很多PE相关的结构未处理，如IMAGE_DIRECTORY_ENTRY_TLS、IMAGE_DIRECTORY_ENTRY_DELAY_IMPORT等，也就是说，这样的dll，注入后会出现异常情况的。（这些未做验证，如有问题，请各路大神指正）

问题二：InjectorCLIx86.exe为何不能注入其它dll

这个其实调试一下就会发现程序崩溃在StartHooking函数中了，该函数对特定的dll导出项（HookData）进行了检测和内存操作，所以崩溃了。看源码之前，一直以为自己使用错了InjectorCLIx86.exe，各种查资料，其实最真实的资料就是源码，一行行代码比什么都直接。

问题三：注入后如何实现反调试

这个下节再议，新年快乐。

广而告之

夜猫出品，欢迎吐槽。更多精彩，可以前往博客地址。