南開大學
Nankai University

网络空间安全学院

恶意代码分析与防治技术课程实验报告

实验九：Rootkit77

姓名：周钰宸

学号：2111408

专业：信息安全

2024 年 1 月 14 日

1 实验目的

1. 复习教材和课件内第 10 章的内容。
2. 运行 R77 程序，实现对指定的进程、文件、注册表、网络连接的隐藏。对实验结果进行截图，完成实验报告。

2 实验原理

2.1 Rootkit

Rootkit 是一种恶意软件，其主要目标是在计算机系统上植入并隐藏自己，以逃避常规的安全检测和防御机制。它通常被用于维持攻击者的长期访问权限，而不被用户或系统管理员察觉。以下是 Rootkit 的一些主要行为和特点：

1. **隐藏：** Rootkits 专注于隐藏其存在，使其在操作系统上变得不可见。这包括隐藏文件、进程、网络连接和注册表项等，以防止被检测。
2. **权限提升：** Rootkits 通常试图提升其执行的权限，以绕过操作系统的安全层级。这可能涉及到提升到管理员或系统级别的权限，以执行更深层次的操纵。
3. **持久性：** Rootkits 致力于在系统中保持长期存在。它们常常会修改系统的启动项、注册表、或其他关键组件，以确保在系统重新启动后仍然存在。
4. **后门访问：** 通常会创建后门，允许攻击者在系统上执行各种操作，例如远程访问、文件上传和下载、系统控制等。
5. **内核级操作：** 一些 Rootkits 会操作在操作系统的内核级别，这使得它们更难被检测和清除，因为它们可以绕过用户空间的安全工具。

要防范 Rootkits，用户和管理员应保持系统和安全软件的更新，定期进行安全审查，使用可信任的防病毒软件，并实施最佳的网络安全实践。

然而，rootkit 并不仅仅用于恶意目的。它们也被组织和执法机构用于监视员工，使他们能够调查机器并对抗可能的网络威胁。

2.2 Rootkit77

本次实验重点研究的 **Rootkit77 就是一种特殊的 Rootkit，也被称为 r77，是一个无文件的 Ring 3 Rootkit**。通过查阅资料，我知道了它具有如下的功能：

2.2.1 隐藏内容

它能隐藏以下所有内容：

1. 文件和目录
2. 进程和 CPU 使用情况
3. 注册表键和值

4. 服务
5. TCP 和 UDP 连接
6. Junctions、命名管道、计划任务

事实上所有以"$77" 开头的内容都会被隐藏。

2.2.2 动态配置

r77 具有动态配置系统，可以通过 PID 和名称隐藏进程，通过完整路径隐藏文件系统项目，隐藏特定端口的 TCP 和 UDP 连接等。配置位于 HKEY_LOCAL_MACHINE\\SOFTWARE\\$77config，任何进程都可以写入，无需提升权限。此外，rootkit 会隐藏 $77config 键。

R77 的部署只需要一个文件：Install.exe。执行后，R77 将在系统上持久存在，并注入所有正在运行的进程。Uninstall.exe 可以完全并优雅地从系统中移除 r77。

Install.shellcode 是安装程序的 shellcode 等价物，这样，安装可以在不放置 Install.exe 的情况下集成。shellcode 可以简单地加载到内存中，转换为函数指针并执行。

综上所述，**R77 是一个 64 位上操作系统上可以运行的 Ring3 Rootkit，出于其在只能在 64 位上运行的特殊性质，本次实验选择在 Win10 操作系统上进行实验，避免不必要的情况发生**。

2.3 Windows 的 Detours 机制

Microsoft Detours 是一个用于 Windows 平台的二进制代码注入和函数重定向的工具。Detours 允许开发人员在运行时修改二进制可执行文件中的函数行为，而无需修改原始的源代码。它通常用于实现 API 挂钩（API Hooking）和函数注入，以用于监视、修改或替换目标函数的行为，如研究恶意软件、性能分析、调试和测试等场景。

2.4 API Hooking

API Hooking 是一种在运行时截获和修改应用程序对 API（应用程序编程接口）函数的调用的技术。这种技术通常用于在不修改源代码的情况下，对程序的行为进行改变、监控或扩展。

2.4.1 应用

1. 调试和逆向工程：API Hooking 常用于调试和逆向工程，以便分析程序的行为，查看函数的输入和输出，或者截获加密算法等关键操作。
2. 性能分析：通过 API Hooking，可以监测应用程序的性能，记录函数调用的频率、耗时等信息，从而进行性能优化。
3. **安全研究：在安全领域，API Hooking 可用于监测恶意软件的活动，例如截获系统调用、检测关键函数的调用等，有助于发现和分析恶意行为**。

2.4.2 风险

1. 稳定性问题：错误的 API Hooking 可能导致程序崩溃、内存泄漏或其他不稳定的行为。特别是在涉及到复杂的应用程序和系统级别的 Hooking 时，可能会引起不可预测的后果。

2. **安全性问题：恶意软件也可能使用 API Hooking 技术，以绕过安全措施、窃取敏感信息或进行其他攻击**。因此，API Hooking 的应用需要经过审慎评估，以确保不会被滥用。
3. 法律和道德问题：在某些情况下，使用 API Hooking 可能违反软件许可协议或法规，因此使用时需要注意法律和道德准则。

3 实验过程

3.1 实验环境及工具

虚拟机软件	VMware Workstation 17 Pro
宿主机	Windows 11 家庭中文版
虚拟机操作系统	Windows 10 家庭中文版
实验工具 1	OllyDBG 2.01
实验工具 2	IDAPro 6.6.14.1224
配套工具	Python 2.7.2

表 1: 本次实验环境及工具

本次实验部分过程参照 https://www.elastic.co/security-labs/elastic-security-labs-steps-throughthe-r77-rootkit。

3.2 实验环境搭建

在 VMware Workstation 17 Pro 通过官网下载媒体刻制光盘的软件，**自制一个 Windows 10 家庭中文版的 iso，部署到虚拟机中安装完如下图所示：**

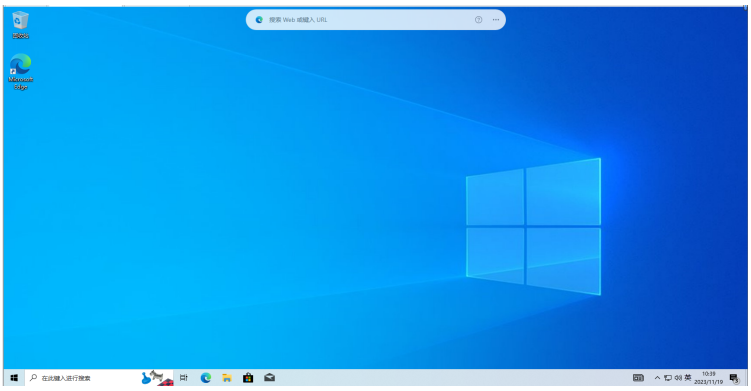

图 3.1: Windows 10 家庭中文版

花了我很长时间去安装 Win10，也占了我将近 50GB 大小空间。好麻烦，接下来去将电脑的病毒威胁检测关闭，不然压缩包一旦解压缩后，相关病毒文件会被自动删除：

图 3.2: Caption

然后我们把一会进一步分析需要的工具和程序挪入虚拟机，包含**测试隐藏进程的 Process Explorer，Procmon 和以及 TCP View**。拍好相关快照，就可以开始啦。

3.3 动态运行 R77

将 R77 压缩包放到虚拟机中，打开后查看图3.3：

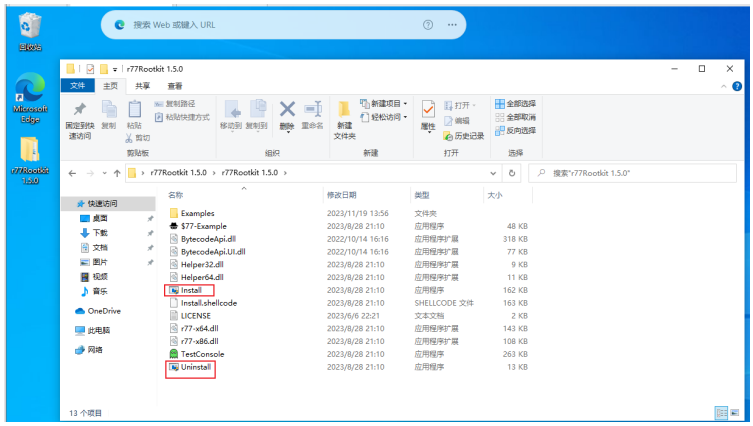

图 3.3: R77 信息

然后双击 install.exe 运行，看到可以看到原本本文件夹下的 $77-Example.exe 也不见了。这验证了它**会隐藏所有以 $77 为开头的文件，进程等的行为**。如图3.4所示。

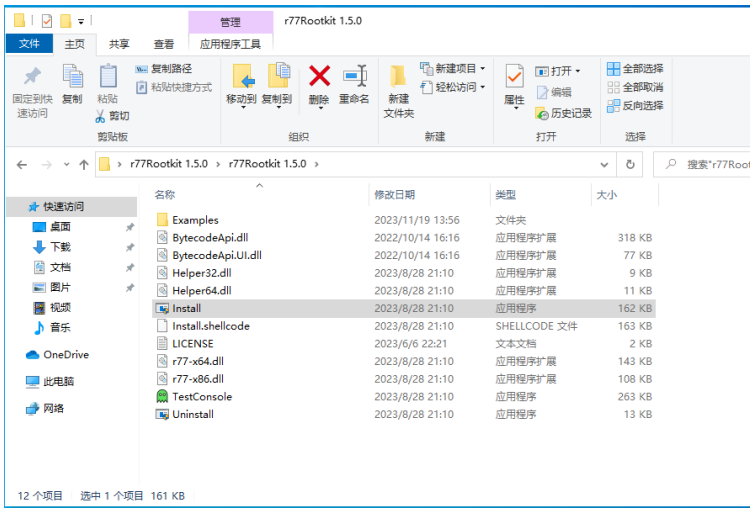

图 3.4: $77-Example.exe 消失

然后我们暂时先恢复快照，**依次对进程、文件、注册表和网络连接隐藏功能进行验证**。

3.4 进程隐藏

实际上 R77 可以实现对任意文件或者指定文件的隐藏，这里首先展示对任意文件即任意 $77 为开头的文件的隐藏：

3.4.1 任意文件隐藏

首先查看其隐藏进程的行为，利用那个给定的以 $77 为开头的 $77-Example.exe，打开运行，**为了能让他更好地体现出被隐藏但实际存在的效果，我将 CPU 占有率调为 100%!**。然后打开任务管理器：

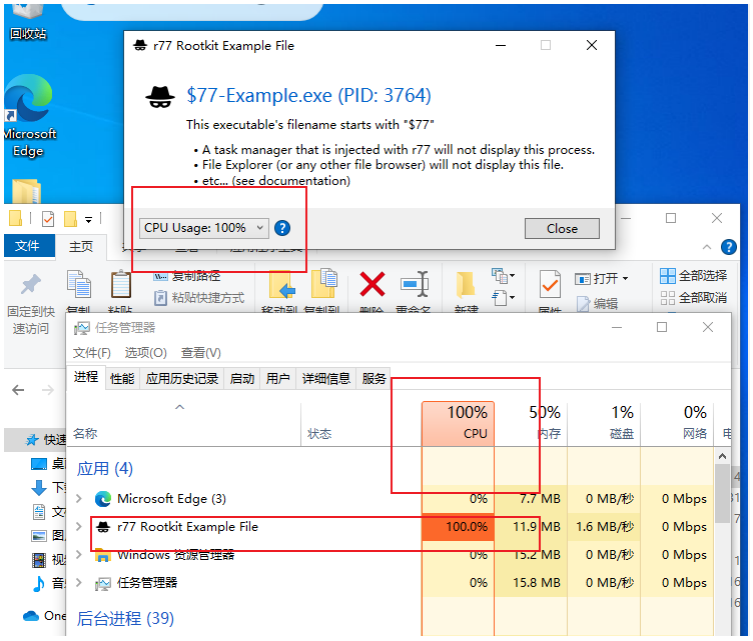

图 3.5: 运行 install 前

图3.5可以明显看到此时**这个可执行文件是完全可见的**。然后我们双击运行 install.exe：

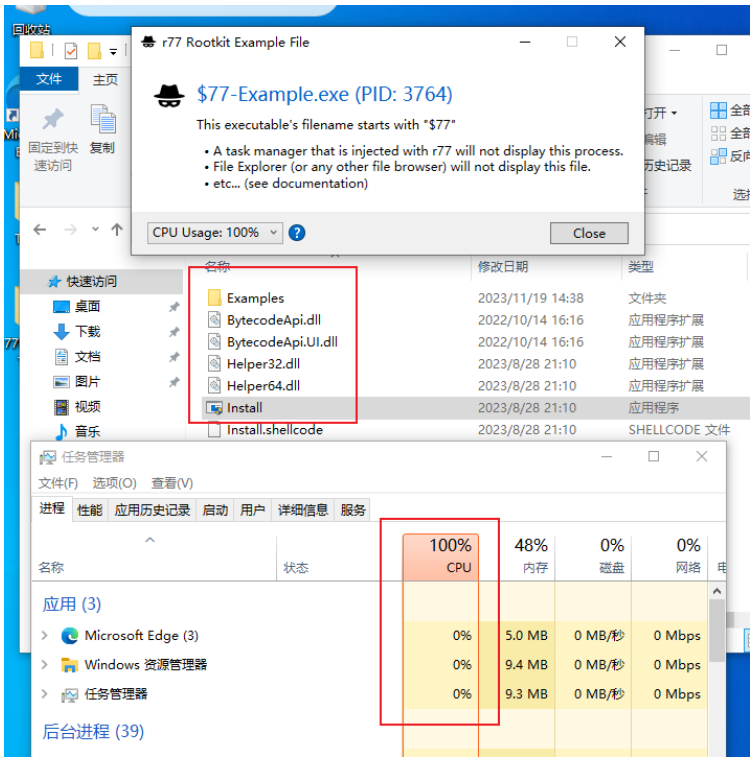

图 3.6: 运行 install 前

图3.6能明显看到此时**任务管理器中的可执行性文件 Example 已经不见了，但是由于其被我设置为了 CPU100%，我们还是能看到 CPU 此时的极高运用率，但其他三个进程都没干什么事，明显就是被隐藏了**。

除此之外，刷新文件目录，也看到 Example 文件找不到了。接下来使用 Process Explorer 查看：

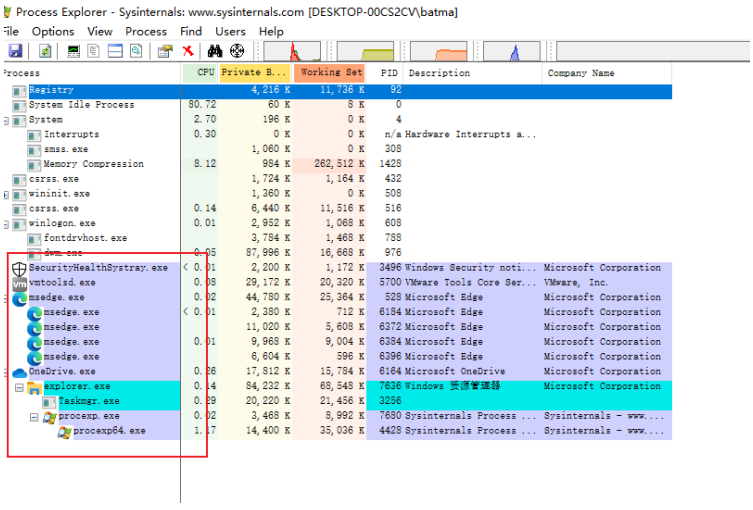

图 3.7: Process Explorer

图3.7看到此时显示的进程目录树中明显没有这个进程，确实证明其被隐藏。**不过我们接下来查看 Procmon：**

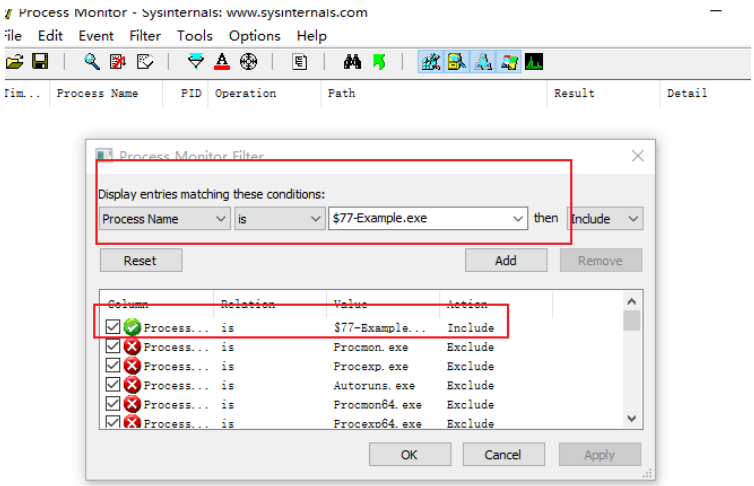

图 3.8: Procmon1

图3.8首先设置过滤器为 ProcessName is $77-Example.exe。

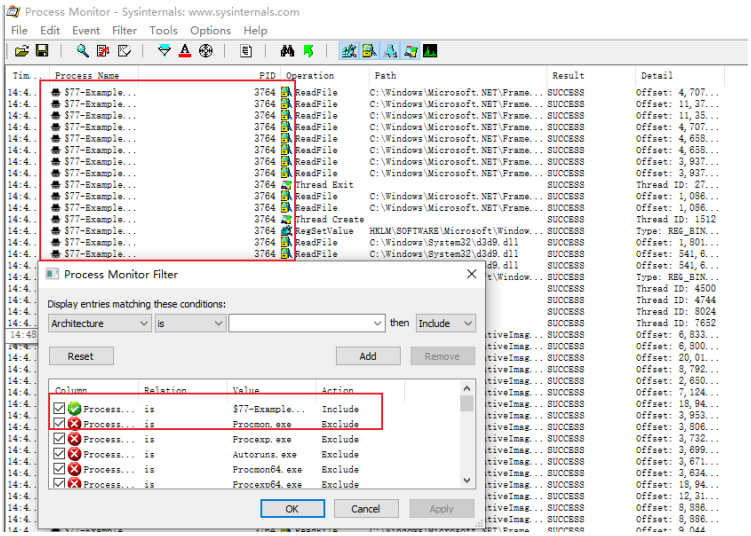

图 3.9: Procmon2

图3.9能明显看到此时**之前被隐藏的进程 $77-Example.exe 的行为被检测到了！其中包括读取文件等行文**。

这是因为如果有隐藏的进程运行，Procmon 也可以捕获它们的活动并显示其详细信息。**终究还是没逃过 Procmon 的法眼！YYDS！**

3.4.2 指定进程隐藏

除了上面说的隐藏任意 $77 开头的文件外，还可以通过 **TestConsole.exe 实现指定进程的隐藏：**

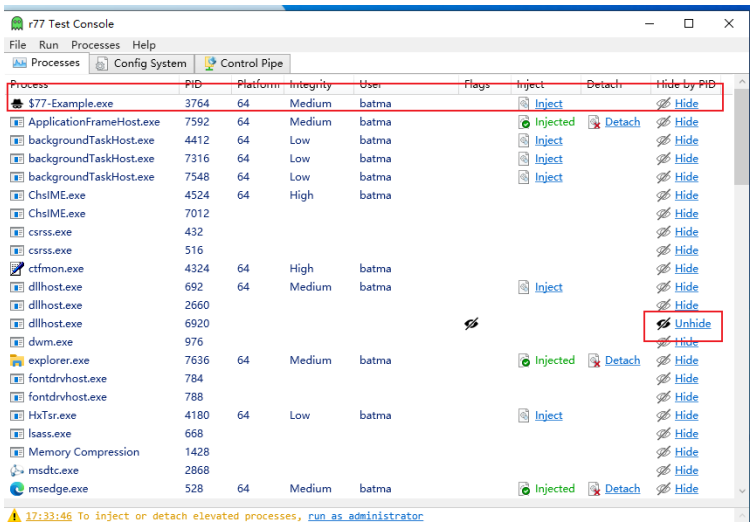

图 3.10: 查看 TestConsole

图3.10可以看到**所有进程列表，不管是被隐藏的 Example 还是别的，右边还有 Hide 选项。我这里为了实现指定进程隐藏，我把 TestConsole.exe 隐藏了**。

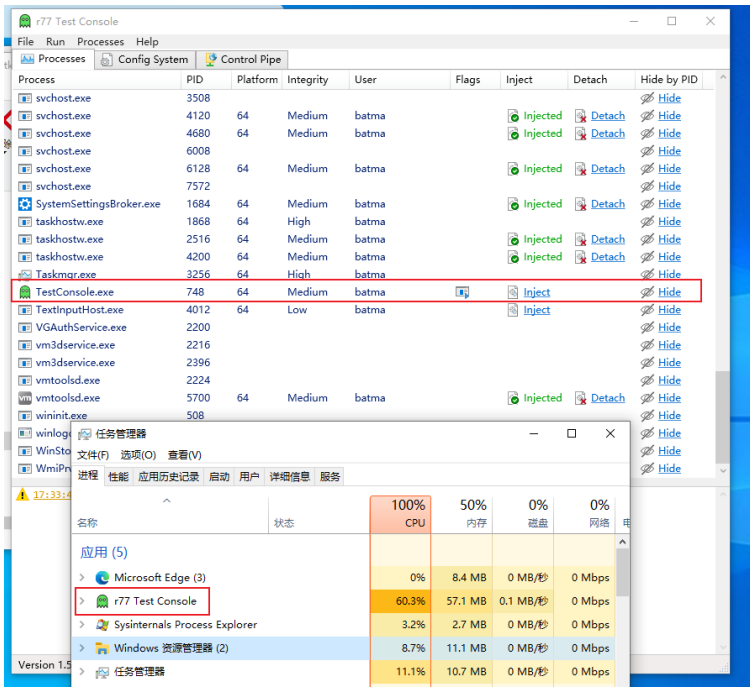

图 3.11: 隐藏 TestConsole 前

图3.11是 Hide 之前，此时还能看到任务管理器中的 TestConsole 正在运行，点击 **Hide:**

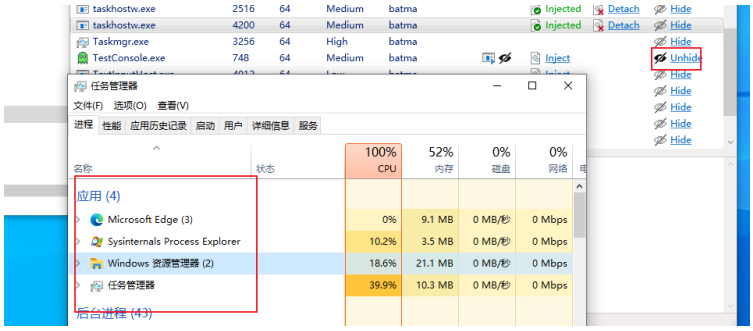

图 3.12: 隐藏 TestConsole 后

图??看到此时 Hide 变成了 UnHide，**并且任务管理器已经找不到 TestConsole 进程了，隐藏成功。 到此就实现了全部的任意或者指定进程的隐藏**。

3.5 文件隐藏

然后介绍 R77 对文件隐藏的功能，**这个功能的实现与 API Hooking 技术有关，具体而言，它通过拦截和修改系统界别的文件系统调用实现**。这样让特定文件或目录在操作系统的标准文件或者浏览工具比如 Windows 资源管理器中不可见。

具体而言**文件夹的隐藏也是通过必须 $77 为文件名开头才能隐藏**。首先 uninstall 恢复之前状态，然后：

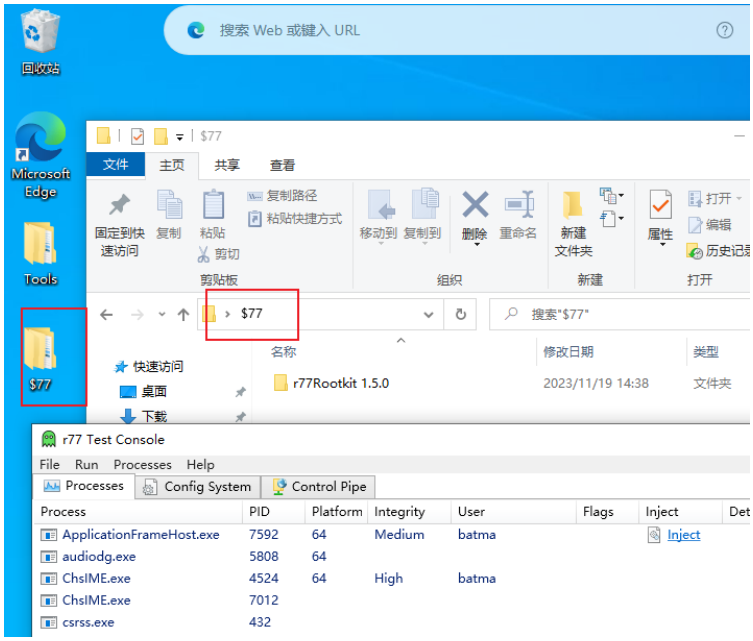

图 3.13: 隐藏 R77 文件本身

图3.13看到我**正在整活，我打算把整个 R77 文件隐藏了（笑）**。

但是为了防止隐藏后我找不到整个文件没法 uninstall，我先改好名字为 $77，然后打开 TestConsole.exe，接下来就是 install 表演时刻了！

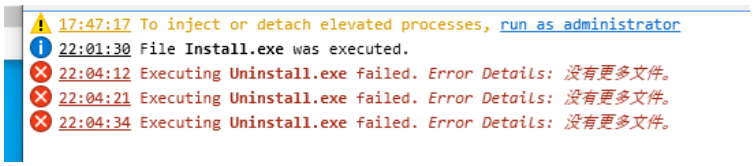

图 3.15: Uninstall 不成功

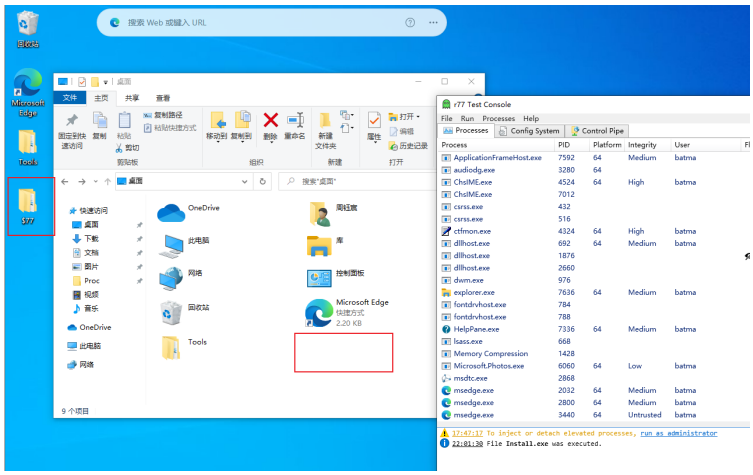

图 3.14: 文件隐藏

刷新后，如图3.14看到**装着整个 R77 的文件夹 $77 已经在文件资源管理器中不见了，但是桌面上还能看到 hhh。算是个新奇的发现，不过文件隐藏成功!**

后面发现想要 Uninstall 时候恢复不了了 hhh，报错“没有更多文件”，可能真是找不到了，幸苦我存了快照。

3.6 注册表隐藏

接下来验证 Rootkit77 的注册表隐藏功能,其配置信息存储在 HKEY_LOCAL_MACHINE\SOFTWARE$77con 中，并且可以在未提权状态下由任何进程写入。这个键的 DACL 被设置为可以给任意用户授予完整访问权 1。“$77config” 键在注册表编辑器被注入了 Rootkit 之后会自动隐藏。

并且它还可以实现以对任意以 $77 为开头的注册表键进行隐藏。

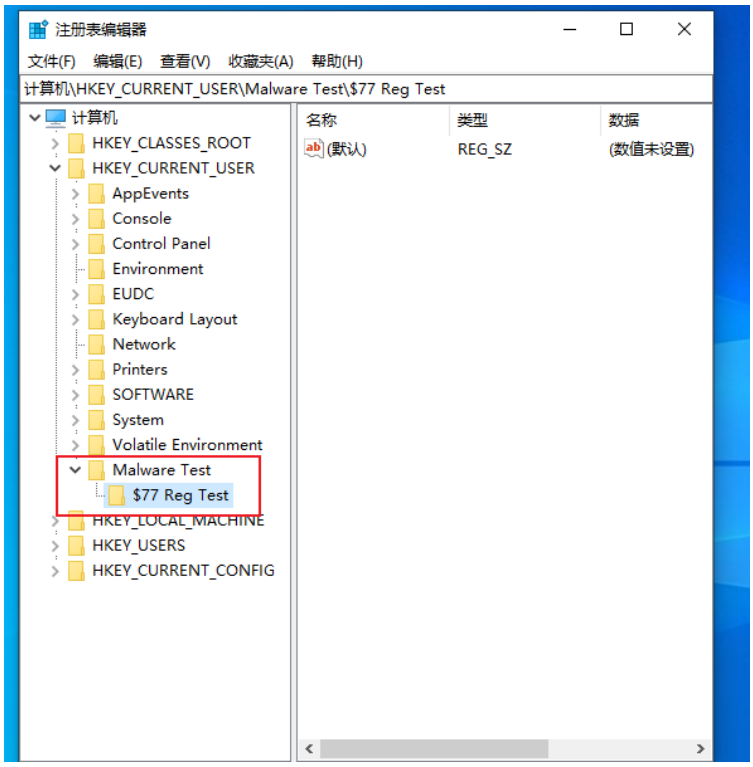

图 3.16: 创造以 $77 为开头的注册表键

图3.16所示，我打开注册表编辑器，然后创建在 HKEY_CURRENT_USER 下的一个注册表项 Malware Test 下的一个注册表项 **$77 Reg Test，以 $77 开头哦**。然后 Install 后刷新注册表：

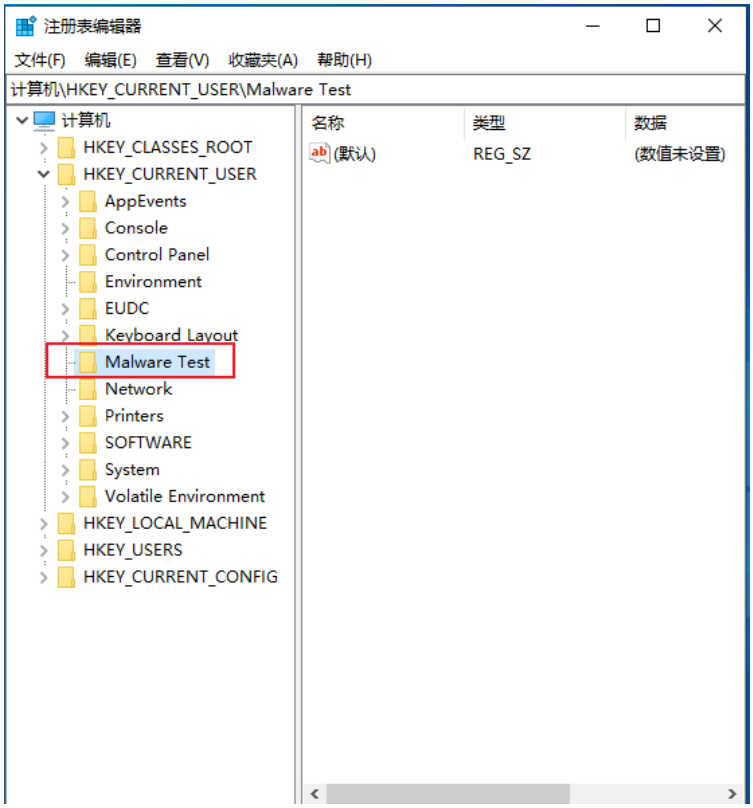

图 3.17: 注册表项被隐藏

图3.17展示了新的结果，**可以看到 Malware Test 下没有任何别的目录了，证明了 $77 Reg Test 被隐藏，验证成功。**

3.7 网络连接隐藏

最后我们 Uninstall 前面的 install，然后**验证一下 R77 对网络连接例如 TCP 和 UDP 的隐藏。**

3.7.1 UDP

直接使用 Edge 打开，搜一些奇怪的东西：

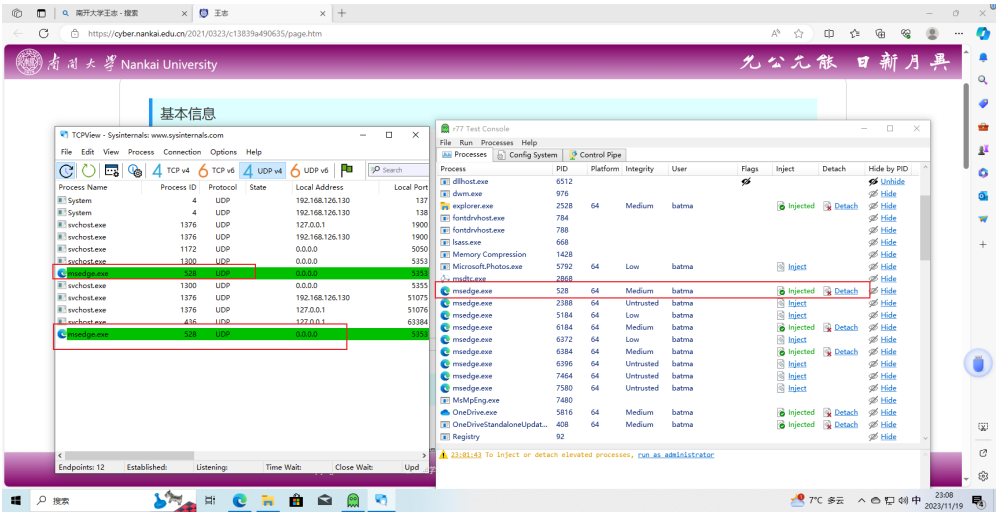

图 3.18: UDP 隐藏前

图3.18同时打开 TestConsole.exe 和 TCPView，先选中 UDP v4，查看可以看到**一个正在运行在 PID=528 的 Edge 进程，点击 Hide：**

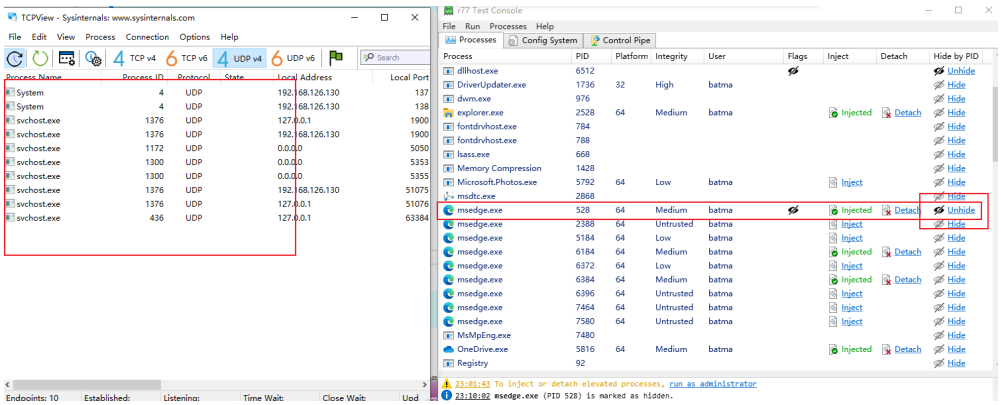

图 3.19: UDP 隐藏后

图3.19看到此时**刚才 UDP 连接的 Edge 的 UDP 连接进程已经消失了，证明 UDP 隐藏成功。**

3.7.2 TCP

接下来验证 TCP，同样观察一下，发现一个：

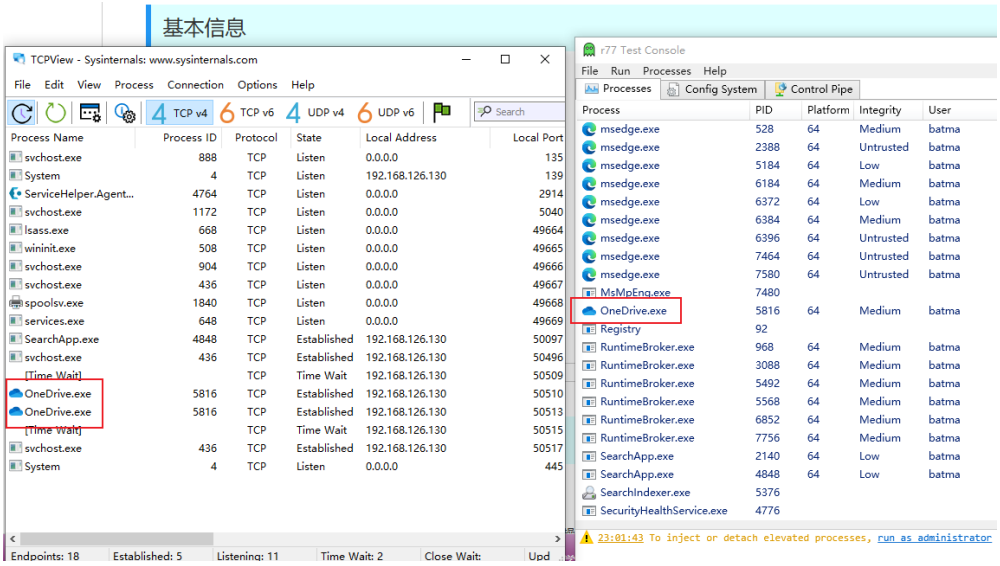

图 3.20: TCP 隐藏前

图3.20看到**有一个运行在 PID=5816 的 TCP 进程 OneDrive.exe，然后在 TestConsole 中也有**，现在 Hide 它：

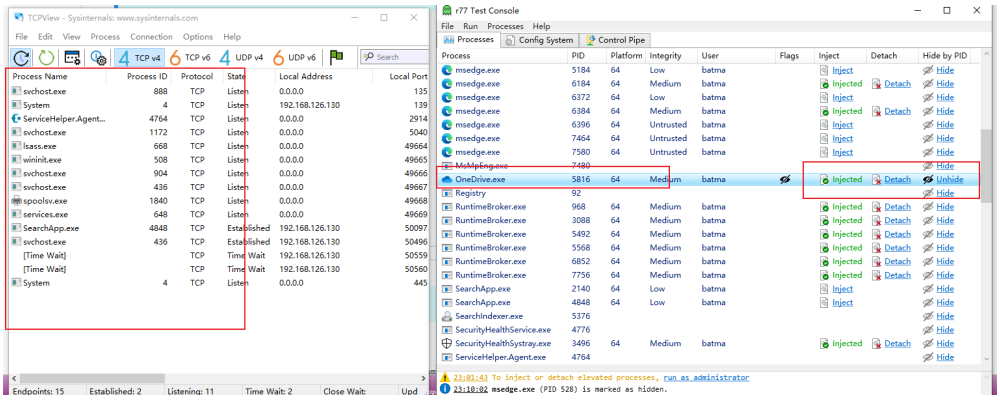

图 3.21: TC 隐藏后

图3.21看到 Unhide，并且 **TCPView 中的这个 OneDrive.exe 已经被隐藏了。到此验证了隐藏 TCP 和 UDP 的网络功能了。**

4 实验结论及心得体会

4.1 实验结论

本次实验，我对除了恶意代码书中的以外的 Rootkit 的功能进行了验证，**全部实现了隐藏进程、文件、注册表和网络连接**的功能，这让我**对它的功能之强大感到十分惊讶。 总的来说，实验非常成功。**

4.2 心得体会

本次实验，我收获颇丰，这不仅是课堂中的知识，更是我解决许多问题的能力，具体来说：

1. 首先我进一步熟练使用 Prcomon 还有 ProcessExplorer 等进行病毒分析；

2. 在这个过程中我通过自己探索，还发现了一些不能 uninstall 的奇怪地方 hhh，发挥了我的创造能力；
3. 为了本次实验，**额外下了一个 Win10 虚拟机，提升了我配置虚拟几环境进行病毒分析的能力**。
4. **最后，如此强大的病毒 Rootkit 让我不寒而栗，要好好学习恶意代码，抵制他们!**

总的来说，本次通过亲自实验让我感受到了一个强大的病毒的能力。也让我养成了充分的安全意识。培养了我对病毒分析安全领域的兴趣。我会努力学习更多的知识，辅助我进行更好的病毒分析。

感谢助教学姐审阅:)