南開大學

恶意代码分析与防治技术课程实验报告

R77 隐藏效果的验证实验

学院：网络空间安全学院
专业：信息安全
学号：2113997
姓名：齐明杰
班级：信安2班

1 实验目的

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。

2 实验原理

2.1 Windows的Detours机制

Windows的Detours机制是一种软件技术，用于拦截和修改Win32函数的调用。这是通过API钩子（hooking）实现的，是一种编程技术，允许开发者拦截对动态链接库（DLL）中函数的调用。Detours 通过动态地重写目标函数的机器代码来工作。这种方法通常用于系统监控、程序调试、性能分析等方面。

在Detours机制中，当一个程序调用某个Win32 API函数时，Detours 可以劫持这个调用，并将它重定向到另一个函数。这个替代函数可以是用户自定义的，也可以是对原函数的扩展或修改。利用这种技术，开发者可以在不修改原始代码的情况下，动态地更改程序的行为。

例如，Detours 可以用于监视和记录文件操作、网络通信或系统调用。它也被用于创建安全工具，如防病毒软件，这些软件需要监视系统级活动以检测恶意行为。然而，同样的技术也可以被用于恶意软件中，用于隐藏其行为，例如隐藏文件、进程或网络连接，这就是你提到的R77程序所做的事情。

2.2 API Hooking

API（应用程序编程接口）Hooking是一种编程技术，用于改变或增强操作系统或应用程序的功能。在Windows系统中，API函数通常来自各种动态链接库（DLL）。

2.2.1 原理

- **拦截调用**：API Hooking工作原理是拦截对特定API函数的调用。这可以通过多种方式实现，包括修改函数的入口地址（例如，在导入地址表中），或者直接修改函数代码本身（如通过Inline Hooking）。
- **重定向调用**：当API调用被拦截后，它会被重定向到一个自定义函数。这个自定义函数可以在调用原函数之前或之后执行额外的代码，甚至完全替代原函数。

2.2.2 应用

- **调试和监控**：开发人员利用API Hooking来监控和记录应用程序的行为，例如文件访问、网络通信等。
- **安全软件**：安全软件（如防病毒程序）使用API Hooking来检测和阻止恶意行为。
- **系统增强**：通过API Hooking，软件可以添加或修改操作系统功能，不需要修改底层代码。

2.2.3 风险

- **稳定性问题**：不正确的API Hooking可能导致系统不稳定。
- **安全隐患**：恶意软件也可能利用API Hooking来隐藏其行为或损害系统。

2.3 隐藏进程

隐藏进程是一种技术，通常用于防止进程在常规工具（如任务管理器）中被检测到。这在恶意软件和某些类型的系统监控软件中很常见。

2.3.1 实现方法

- **修改系统结构**：通过修改操作系统的内部数据结构（如进程列表）来隐藏进程。
- **拦截系统调用**：使用API Hooking或类似技术拦截和修改系统调用，例如拦截列出进程的API调用，并从结果中删除特定进程。
- **内核模式驱动**：在内核模式下运行的驱动程序可以直接访问和修改操作系统的核心数据结构，进而隐藏进程。

2.3.2 应用

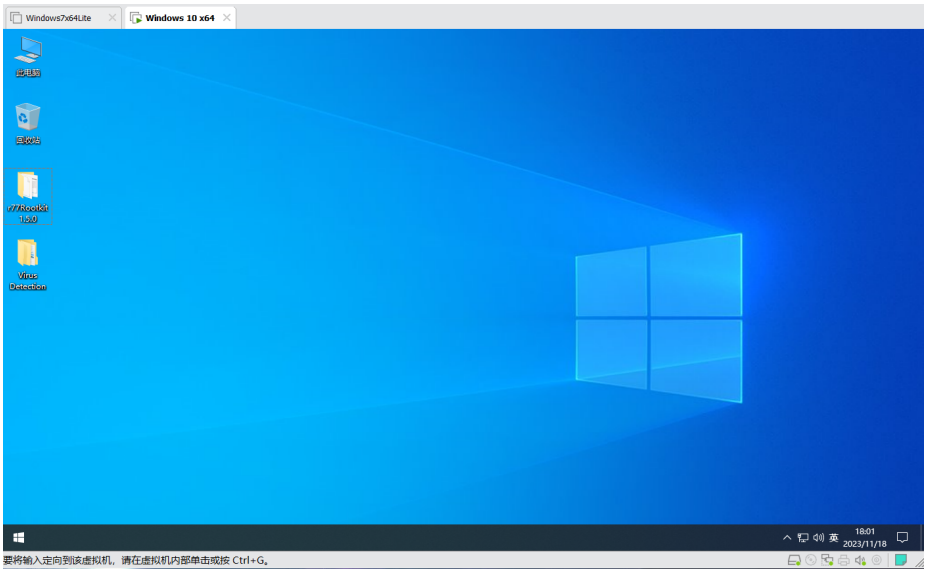
- **安全和隐私**：某些合法软件可能出于安全或隐私原因隐藏其进程。
- **恶意软件**：病毒、木马和rootkits经常隐藏其进程以避免检测。

2.3.3 风险和挑战

- **安全风险**：隐藏进程的技术可以被恶意软件利用，对用户和企业造成严重威胁。
- **检测难度**：隐藏进程的技术提高了安全软件检测和清除这些威胁的难度。

3 实验环境

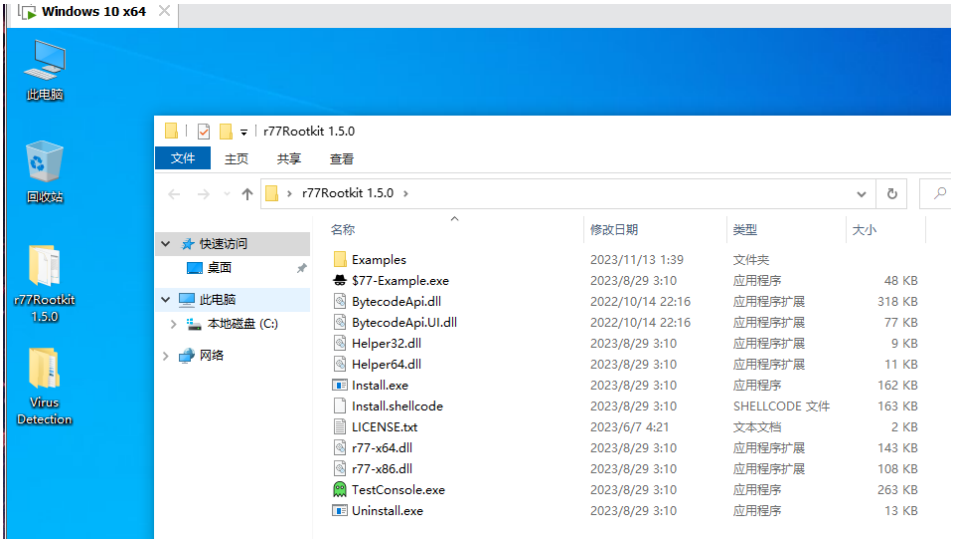
本次实验我在windows 10 x64的虚拟机上运行R77 Rootkit。



4 实验过程

4.1 安装R77

下载R77文件，并解压到桌面：



双击Install.exe，这将把 R77 注入到每个正在运行的进程中，并在系统中持久化该根工具。新进程在运行任何指令之前被注入，这是通过钩住（hooking）进程创建实现的。**安装后，R77将设置为在重启后启动，并在第一个用户登录前注入所有进程。**

由于R77已经注入到目前的所有正在运行的进程中，我们不需要进行重启。

由于R77会隐藏所有以$77为开头的文件，进程等，我们先运行Uninstall.exe复原。

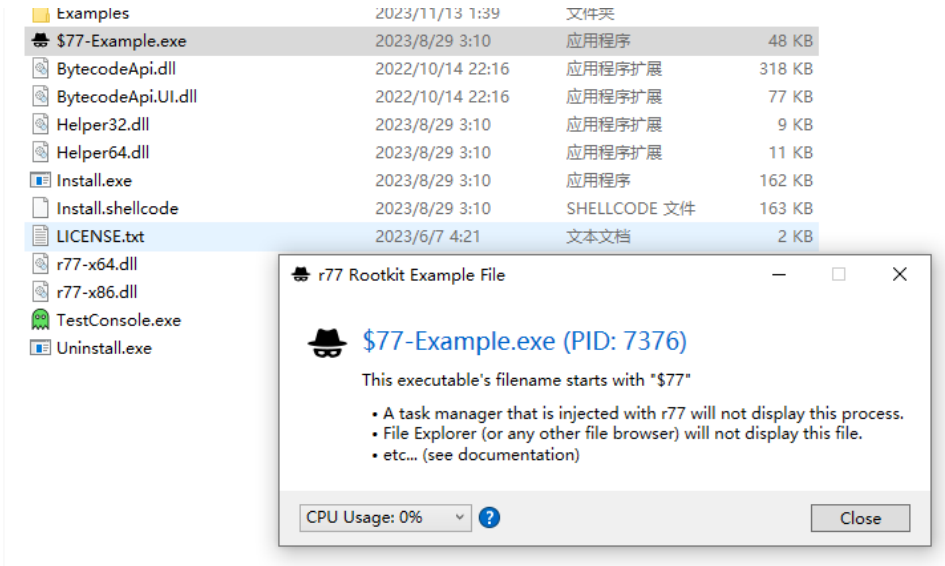
4.2 进程隐藏

R77的进程隐藏功能是一种用于使特定进程在操作系统的常规监视工具（如任务管理器）中不可见的技术。这种隐藏技术主要基于修改操作系统的行为来阻止对特定进程的检测。

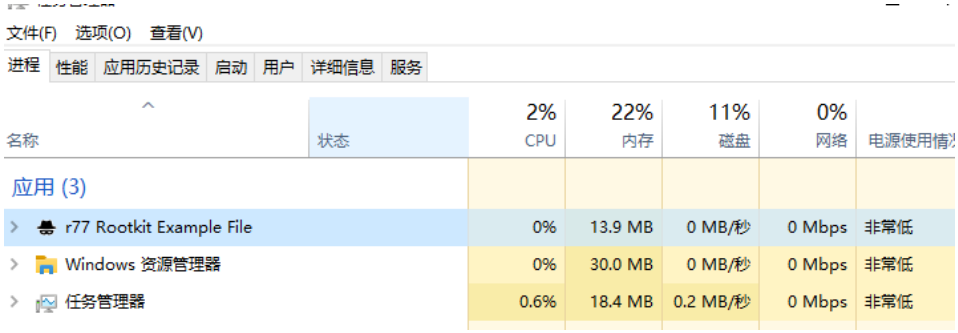
- **示例进程**

R77为我们提供了一个可执行文件`$77-Example.exe`，这个文件以$77为开头。

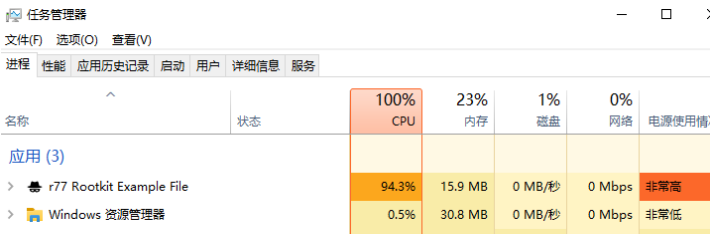
然后我们运行它：



再查看任务管理器：



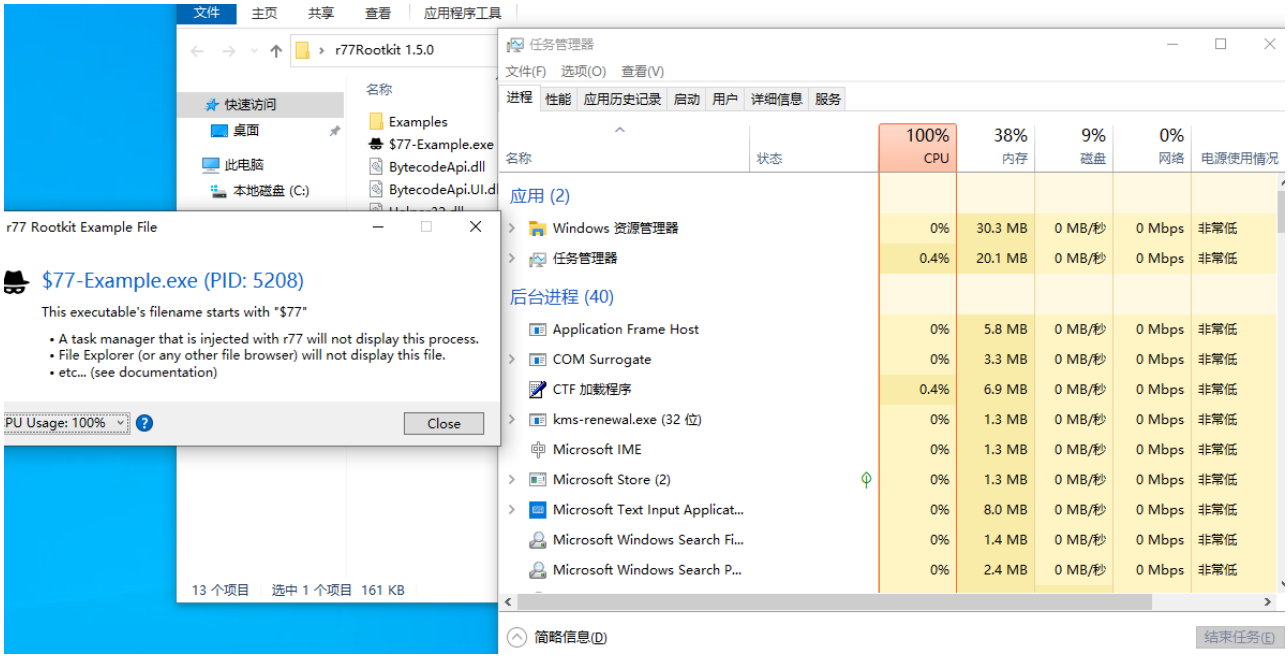
这个示例文件可以**调节CPU占有率**，如果我将其调成100%，**CPU占用率将明显上升**：



这个进程**目前是可见**的，我先选用这个exe来进行隐藏。

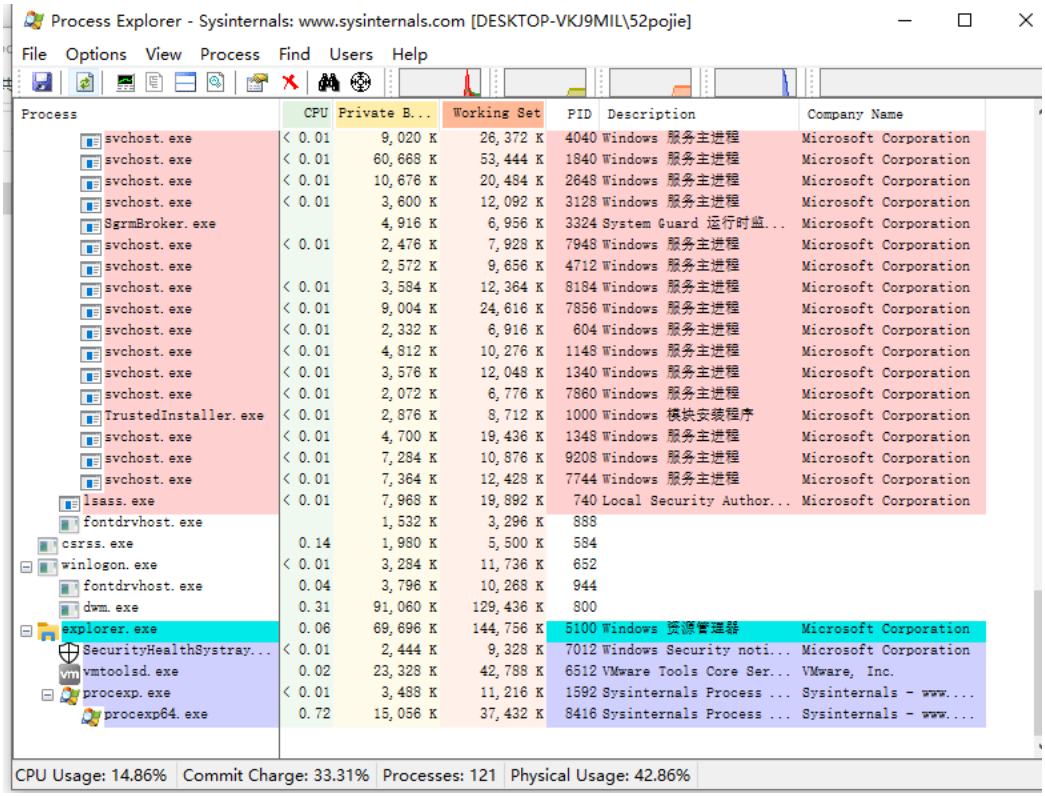
- **任意进程隐藏**

此时运行`Install.exe`，重新打开任务管理器，同时保持这个exe运行，发现**找不到这个进程**了：



在上图，为了证明这个程序仍然在运行，只是被隐藏了，我将CPU占用率调为100%。可以看到，尽管**CPU占用已经满了**，但并没有显示这个进程的存在。

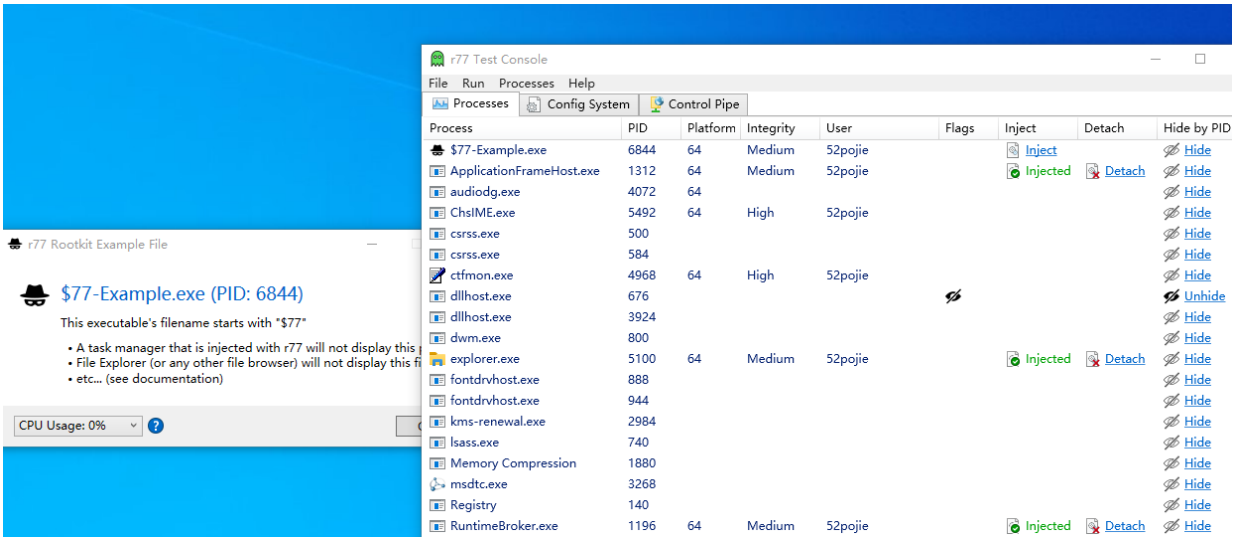
我们打开**Process Explorer**来查看进程：



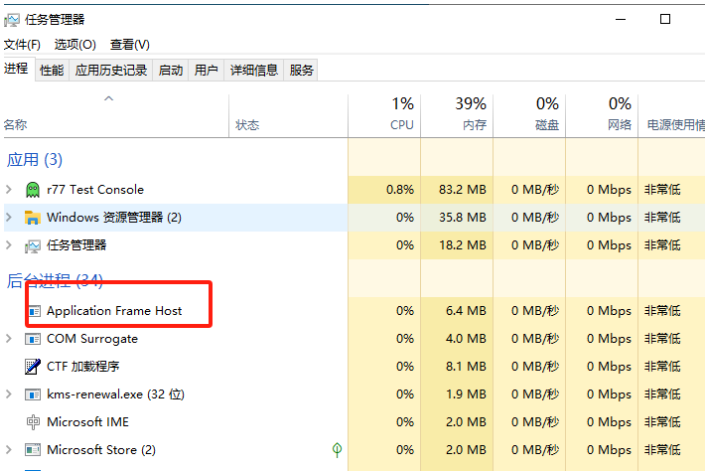
仍然没有找到这个进程，证明其确实被隐藏了。

实际上，R77默认隐藏了$77开头的进程，但其他进程也可以进行针对性隐藏。

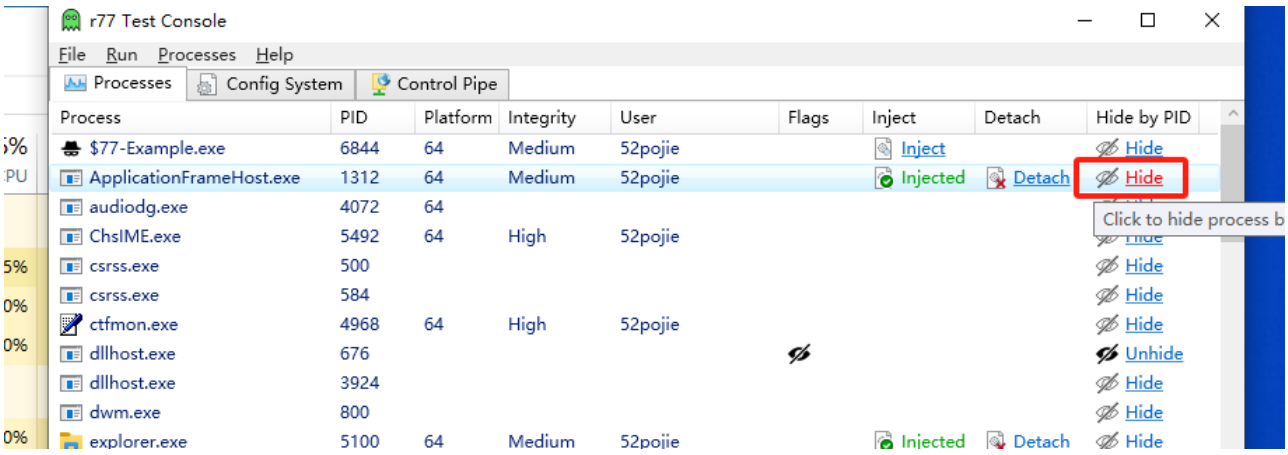
我们运行`TestConsole.exe`，可以看到进程列表，包括被隐藏的进程：



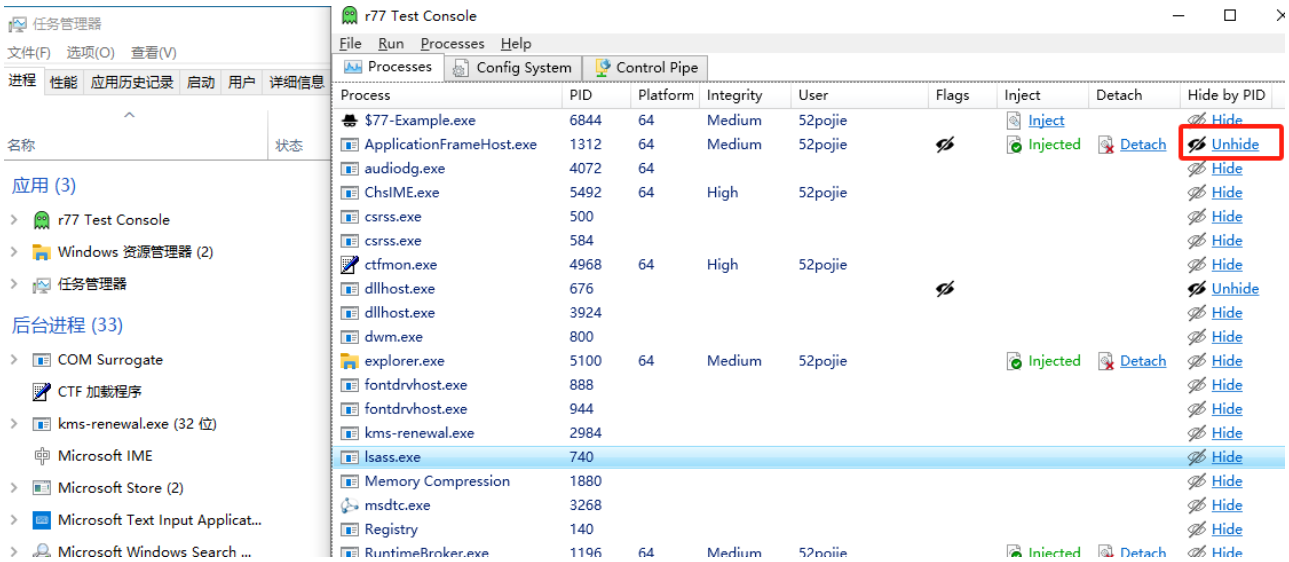
随便再找一个**任意进程名**的进程，选取下面的进程：



点击右边对应的Hide操作：



再次打开任务管理器查看，该进程已经“消失”，同时右边的**Hide选项也已经变成了Unhide**:

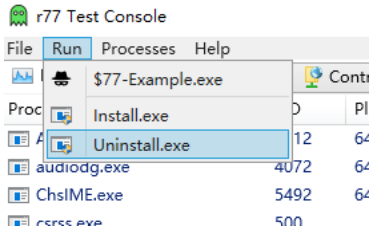


这就实现了任意指定进程的隐藏。

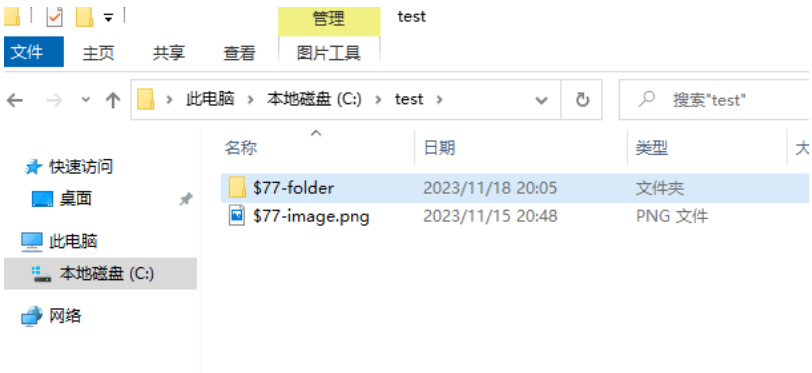
4.3 文件隐藏

R77的文件隐藏功能是一种技术，用于使特定文件或目录在操作系统的标准文件浏览工具（例如Windows资源管理器）中不可见。这通常是通过**拦截和修改系统级别的文件系统调用**来实现的。

先执行uninstall:

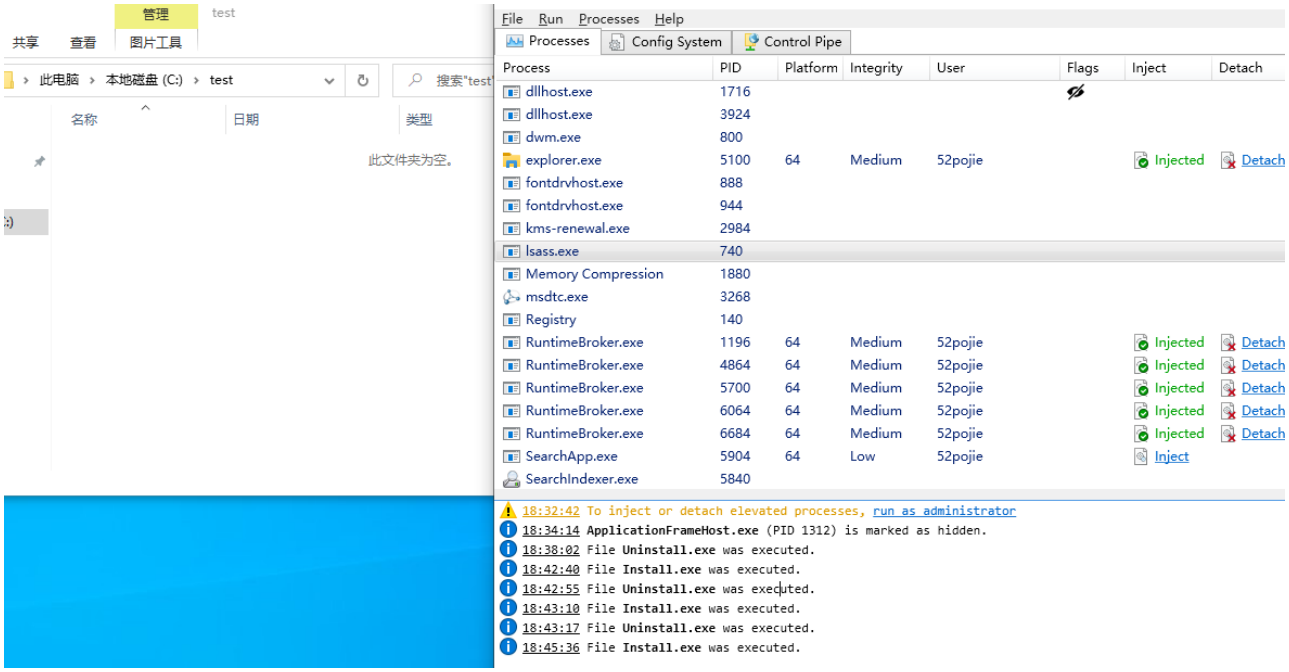


在一个文件夹里面放入我想**要隐藏的文件和文件夹**:



注意，文件和文件夹必须以$77为文件名开头才能隐藏。

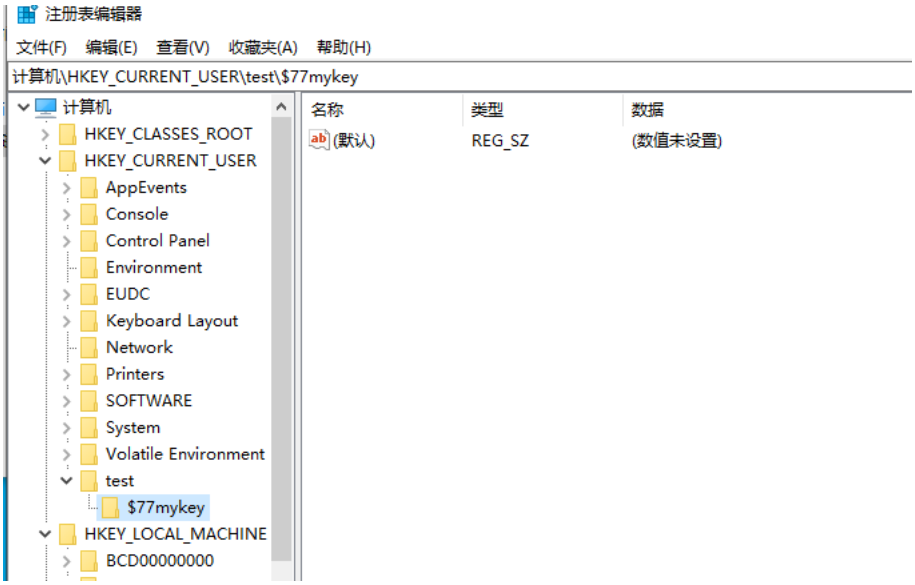
然后执行Install操作，刷新文件夹:



显示“此文件夹为空”，文件已经被隐藏成功。

4.4 注册表隐藏

执行Uninstall后，打开注册表，新建一个以$77为开头的注册表键：



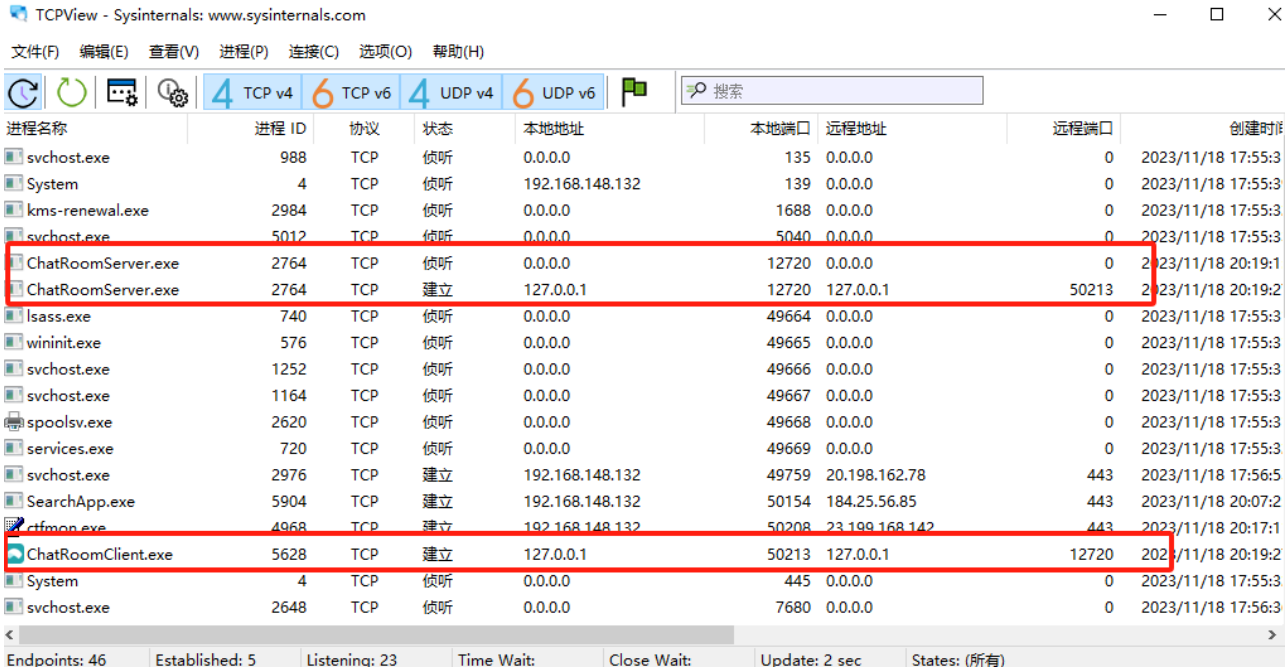
然后执行Install.exe，重新打开注册表，发现`$77mykey`**被隐藏了**：

4.5 TCP&UDP隐藏

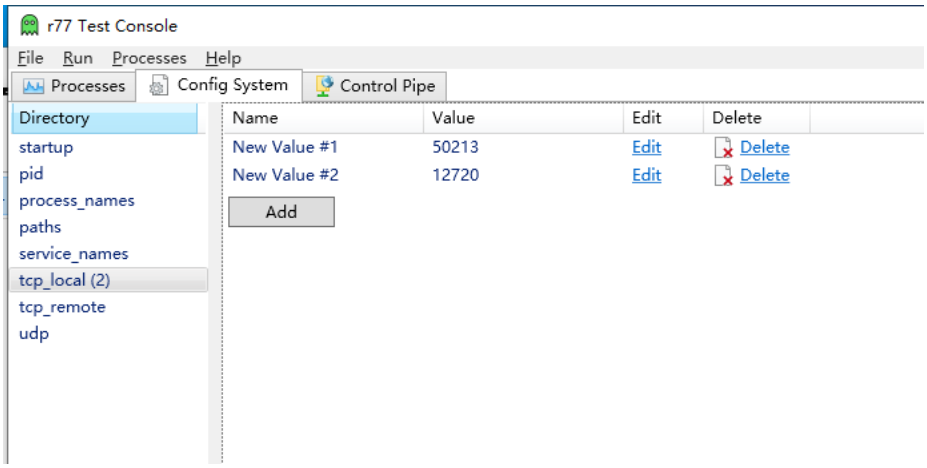
- **TCP通信隐藏**

我准备了一个聊天室程序，采用TCP编程实现，分为服务端和客户端：

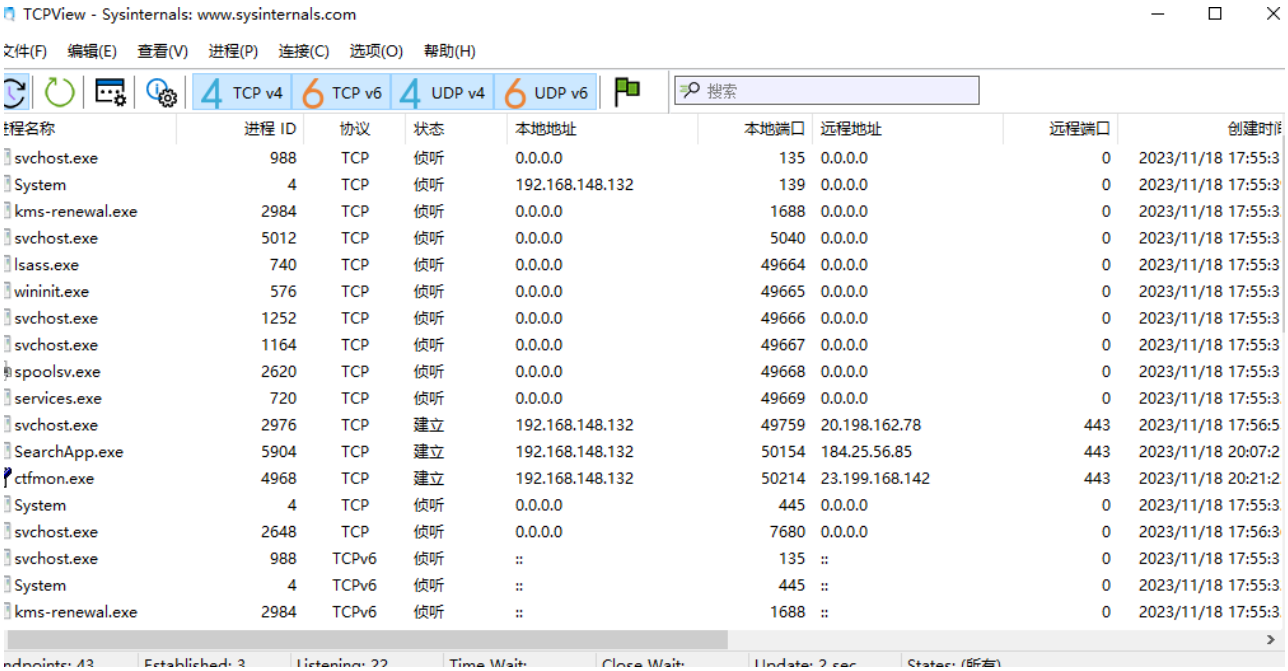
打开TCPView可以查看TCP通信状态：



那么我们打开R77面板，设置TCP的隐藏选项：



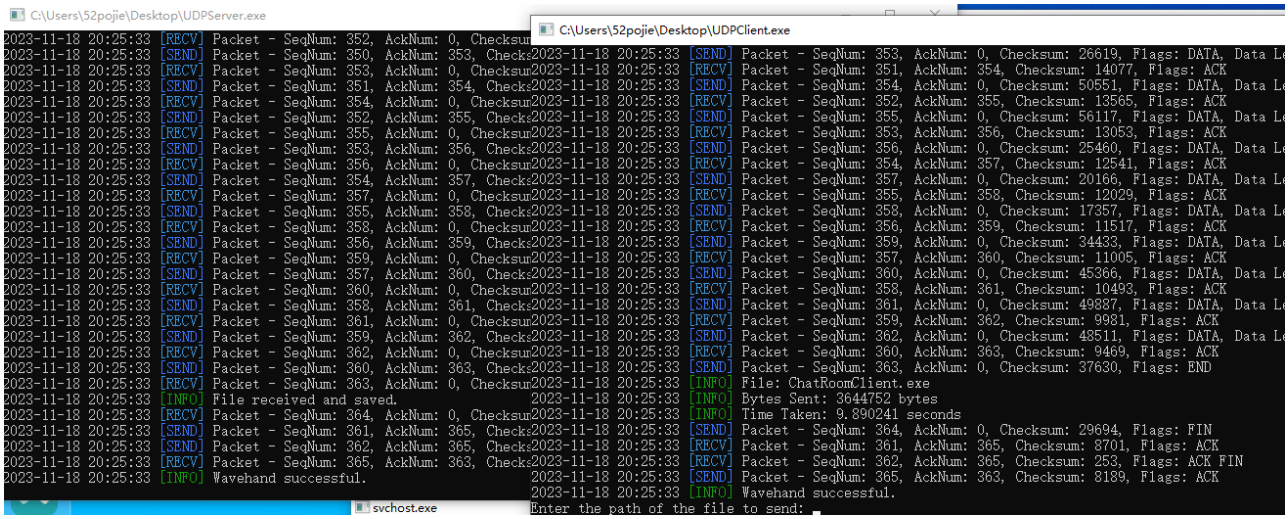
然后重新查看TCPView：



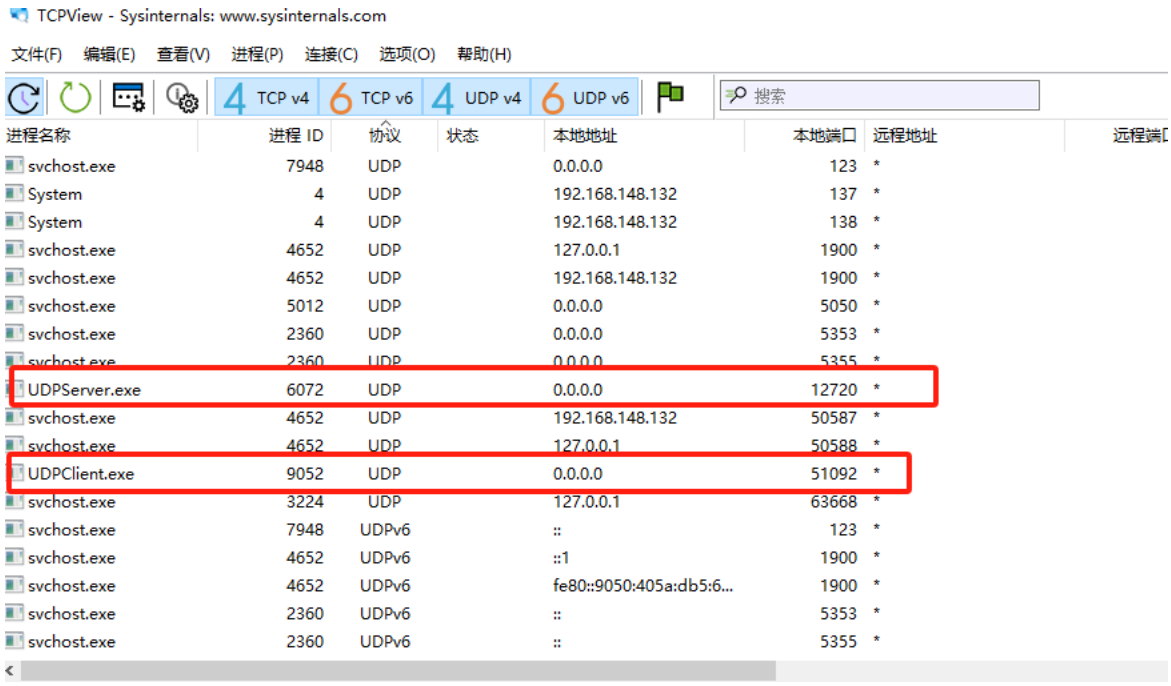
发现侦察不到相应端口的TCP通信了。

- **UDP通信隐藏**

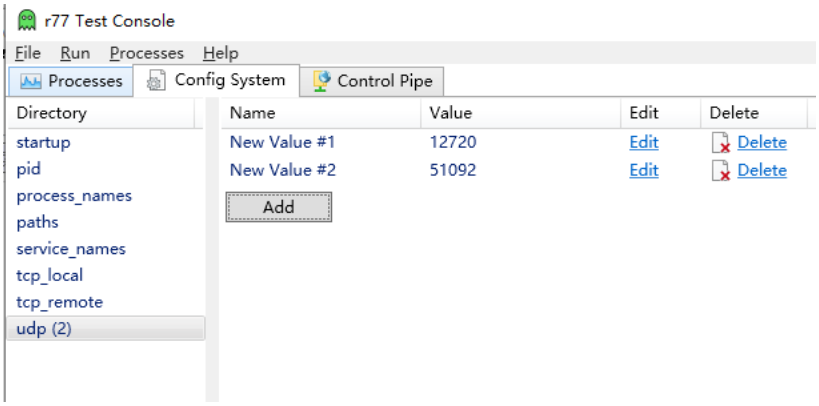
我编写了一个使用UDP协议的可靠文件传输，分为客户端(发送)和服务端(接收)：



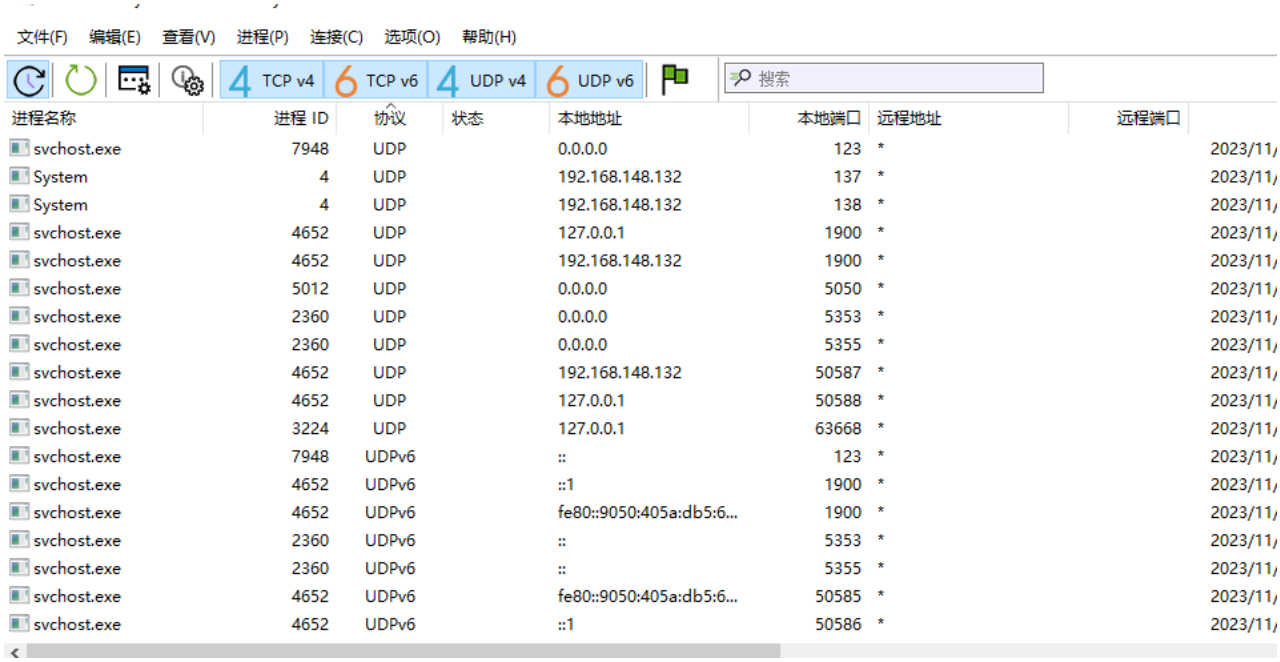
其在TCPView也可以查看端口：



修改隐藏的端口：



运行Install.exe，同样被隐藏了：



5 实验结论及心得体会

- **实验结论**

1. **技术验证**：本次实验成功验证了R77工具在隐藏进程、文件、注册表项以及TCP/UDP连接方面的有效性。通过实际操作，观察到被隐藏的对象在常规工具中不再可见，证实了R77在系统级别进行拦截和修改操作的能力。
2. **功能特性**：R77通过拦截系统级调用和修改操作系统的内部数据结构，实现了对进程、文件和网络活动的隐藏。这一过程不需要修改被隐藏对象的物理状态或数据，显示出高度的隐蔽性和灵活性。
3. **安全和隐私考量**：尽管R77为系统管理和隐私保护提供了强大工具，但同时也揭示了潜在的安全风险。恶意软件可能利用类似技术进行隐蔽活动，对用户和企业造成威胁。

- **心得体会**

1. **技术深度**：通过这次实验，我深入理解了Windows系统的内部工作原理，尤其是进程、文件系统和网络通信方面的机制。这为我日后的学习和研究奠定了坚实的基础。
2. **安全意识**：实验过程中，我意识到安全防护的重要性和复杂性。理解恶意软件的潜在行为和防御策略对于构建更安全的IT环境至关重要。

3. **责任感和伦理**：作为一名计算机专业的学生，我认识到了在使用这类强大工具时所承担的责任。遵守法律法规和伦理准则，在确保安全和隐私的前提下进行实验，是每位技术人员应当遵循的原则。