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

实验二：虚拟机技术

姓名：王峥

学号：2211267

专业：信息安全

指导教师：王志、邓琮弋

2024 年 9 月 27 日

目录

1 实验目的

本实验旨在熟悉配置病毒分析虚拟机并安装静态和动态分析工具的过程，以便进行后续的恶意软件样本的分析和逆向工程。通过完成此实验，我能够理解虚拟化环境的重要性，以及如何使用不同的工具来执行病毒分析。

2 实验原理

1. 虚拟机配置：使用虚拟机软件（如 VMware 17）创建一个虚拟计算机环境，其中安装了 Windows XP 操作系统。这个虚拟环境将用于安全分析，以避免对真实计算机系统造成潜在的风险和威胁。

2. 静态分析工具：

- string.exe：用于在二进制文件中查找可打印字符的工具，有助于识别潜在的关键信息。
- PEView：用于查看和分析可执行文件（PE 文件）的工具，可帮助分析二进制文件的结构和属性。
- Dependency Walker：用于分析可执行文件的依赖关系，识别它们所依赖的动态链接库（DLL）。
- IDA：交互式反汇编器，用于分析二进制文件的汇编代码，以识别其功能和行为。

3. 动态分析工具：

- OllyDBG：一款用于动态分析可执行文件的调试器，用于跟踪程序的执行过程、内存内容和函数调用。
- Process Monitor：用于监视系统进程和文件系统活动，可帮助识别恶意软件的行为和修改。
- Process Explorer：用于查看系统进程、资源使用情况和线程信息，有助于检测异常进程或行为。
- RegShot：用于比较系统注册表的快照，以检测恶意软件对注册表的修改。
- Wireshark：用于网络流量分析，可帮助分析恶意软件是否与恶意服务器通信或进行其他网络活动。

3 实验过程

3.1 虚拟机安装与配置

(1)：VMware 虚拟机的安装

首先我们在官网上下载 VMware, 并进行安装，由于之前课程中已安装过 VMware, 因此无需再次安装，无安装详细过程，此处直接贴当前软件的版本信息。

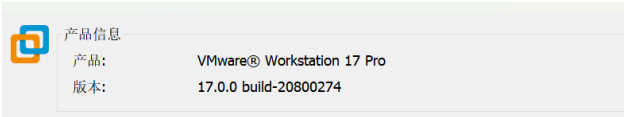

图 3.1: 安装 VMware 的版本信息

安装后，我们直接进入 VMware，进行 windows XP 的安装在**文件-新建虚拟机**中，我们将相关虚拟机的映像文件（.ios）导入

图 3.2: 导入镜像文件

接下来跟随导向，进行相关配置

图 3.3: 虚拟机位置

图 3.4: 网络类型设置

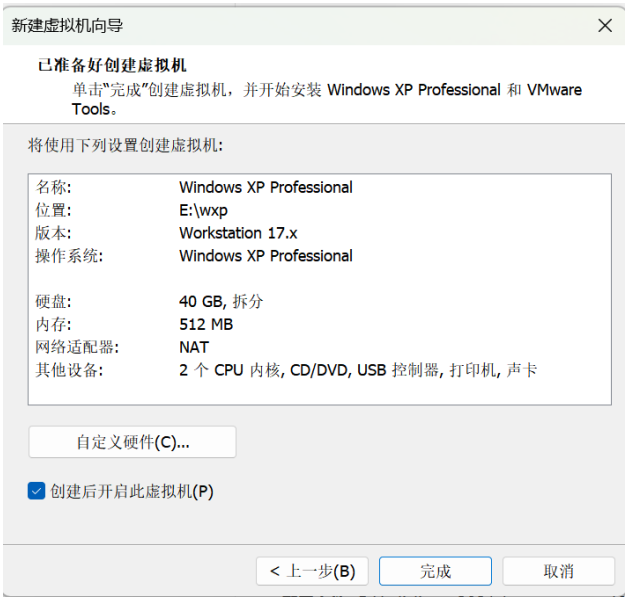

图 3.5: 虚拟机的详细信息

进入 Window XP, 输入产品密钥后，进行安装

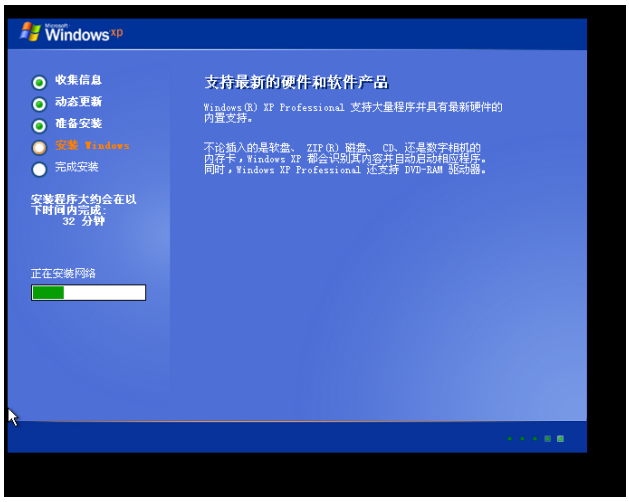

图 3.6: Windows XP 系统安装

3.2 静态分析工具的功能与安装

在虚拟机中安装 VMware tools 后，可以开始本机与 VMware 互传文件操作。把本机中的文件都拖拽到虚拟机中。

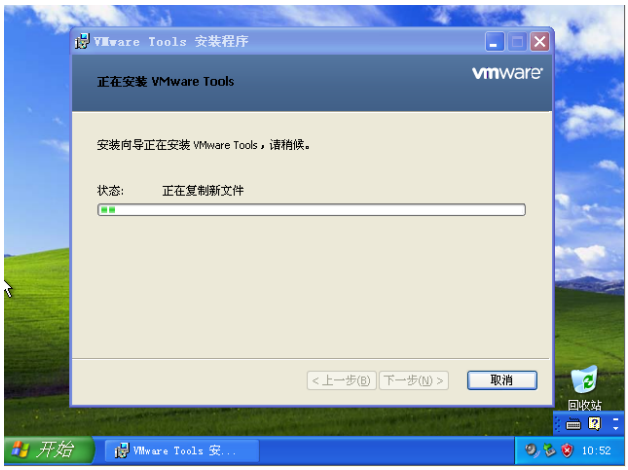

图 3.7: VMware Tools 安装

(1) string.exe

string.exe 通常用于查找二进制文件中的可打印字符序列。其主要功能包括

- 字符串提取：string.exe 扫描二进制文件，可以识别其中的文本字符串和可打印字符序列，并提取出来。这对于识别程序中的硬编码字符串、配置信息、URL、文件路径等非常有用。
- 熵值计算：工具通常还会计算字符串的熵值，以评估字符串的随机性和复杂性。较高的熵值可能表示更加随机的数据或加密的字符串。
- 可疑代码检测：通过查找特定的字符序列、模式或关键字，string.exe 有助于揭示潜在的恶意代码。例如，它可能会标识出用于动态代码加载、加密解密操作的函数名或库调用。
- 资源定位：工具能帮助定位和提取程序中嵌入的资源文件，如图片、音频、视频或其他类型的媒体文件。这对于分析程序的界面元素、功能完整性或潜在的安全漏洞可能很重要。

- 逆向工程：在逆向工程中，string.exe 可以帮助分析人员理解程序的内部逻辑和功能，尤其是在没有源代码的情况下。

以下是在虚拟机中运行 Strings

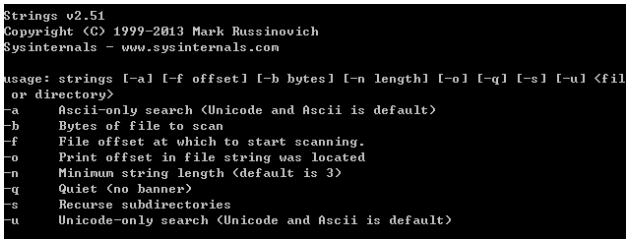

图 3.8: 在 Windows xp 中运行 string.exe

尝试利用 strings 分析我们的实验 1 中 Lab01-01.exe：

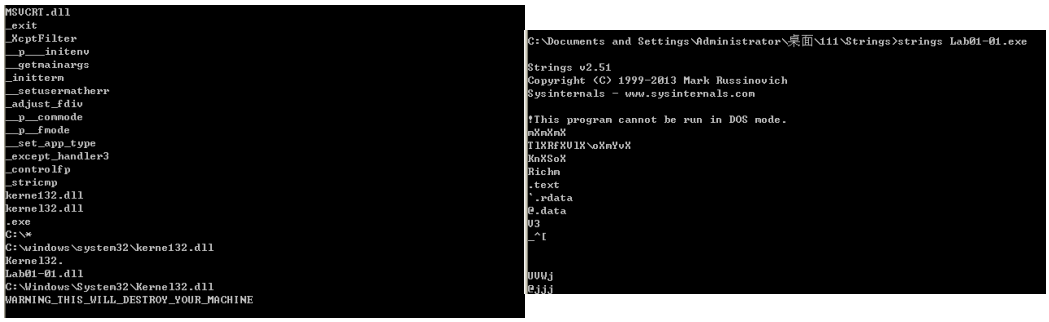

图 3.9: 使用 PEiD 打开 Lab01-02.exe 部分结果

(2) PEView

PEview 是一款功能强大的 PE（Portable Executable，可移植执行文件）和 COFF（Common Object File Format，通用对象文件格式）文件查看器，主要运用于 WIndows 系统中。PEview 的主要功能包括：

- 查看 PE 文件头信息：PEview 允许用户查看 PE 文件的头部信息，包括文件标志、节表、入口点地址、数据目录等。
- 显示文件属性：可以查看 PE 文件的属性，如文件大小、创建日期、修改日期等。
- 依赖分析：尽管 PEview 本身可能不直接提供 PE 模块依赖性的详细分析功能，但它通过展示文件的导入表和导出表，可以间接帮助用户理解文件与其他模块之间的依赖关系。
- 查看节表信息：用户可以查看 PE 文件的节表，了解文件的不同部分（节）以及它们的属性，如代码节、数据节等。
- 检查 PE 文件的完整性：PEview 可以检查 PE 文件的完整性，确保文件没有损坏或受到篡改。
- 兼容性：PEview 支持多种操作系统平台上的 PE 和 COFF 文件查看，是逆向工程、恶意软件分析、软件安全等领域中不可或缺的工具之一。
- 导出信息：可以将 PE 文件的信息导出为文本文件，以便后续分析和记录。

以下是在虚拟机中使用 PRView 分析实验 1 中的 Lab01-02.exe:

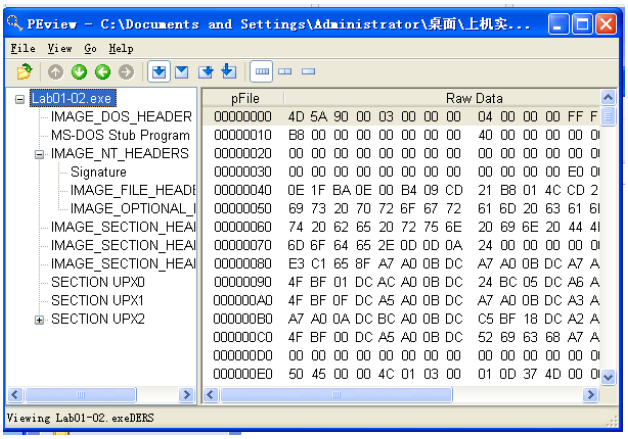

图 3.10: 在 Windows xp 中使用 PEView 打开 Lab01-02.exe

(3) dependency walker

Dependency Walker 是一种用于分析和查看可执行文件或动态链接库（DLL）的依赖关系的工具。是一种用于分析和查看可执行文件或动态链接库（DLL）的依赖关系的工具，其主要功能特点：

- 依赖关系分析: Dependency Walker 能够扫描任何 32 位或 64 位的 Windows 模块，并构建出所有依赖模块的层次树状图。这使得用户可以清晰地看到应用程序所依赖的所有动态链接库（DLL）及其之间的层级关系。详细依赖信息：对于每个依赖模块，Dependency Walker 列出了该模块的路径、版本信息、CPU 架构等详细信息，帮助用户了解模块的来源和兼容性。
- 函数导入导出分析: Dependency Walker 能够列出每个模块导入的函数及其来源的 DLL，帮助用户理解模块之间的函数调用关系。同时，它也能列出每个模块导出的函数及其内存地址，这对于理解模块的功能和接口非常有帮助。
- 依赖分析：尽管 PEview 本身可能不直接提供 PE 模块依赖性的详细分析功能，但它通过展示文件的导入表和导出表，可以间接帮助用户理解文件与其他模块之间的依赖关系。
- 问题检测与诊断: Dependency Walker 能够检测到缺失的模块，并以红色叉号标记，提示用户需要安装或修复相应的组件或库文件。对于无法加载的模块，Dependency Walker 会提供详细的错误信息，帮助用户快速定位问题原因，如路径问题、版本不匹配等。
- 性能分析: Dependency Walker 可以模拟应用程序的启动过程，以识别在运行时可能出现的模块加载问题。并通过 “Profile” 菜单，用户可以对文件进行性能分析，了解应用程序在加载和执行过程中的性能瓶颈。
- 兼容性与灵活性: Dependency Walker 具有广泛兼容性，支持在 Windows 95、98、Me、NT、2000、XP、2003、Vista、7 和 8 等多个操作系统版本上运行，能够处理任何 32 位或 64 位的 Windows 模块。并具有多种运行方式，既可以作为图形应用程序运行，也可以作为控制台应用程序运行，满足用户不同的使用需求。

(4) IDA

IDA（Interactive DisAssembler）是一款逆向工程工具，广泛应用于软件逆向、漏洞分析、恶意代码分析等领域。其主要功能包括：

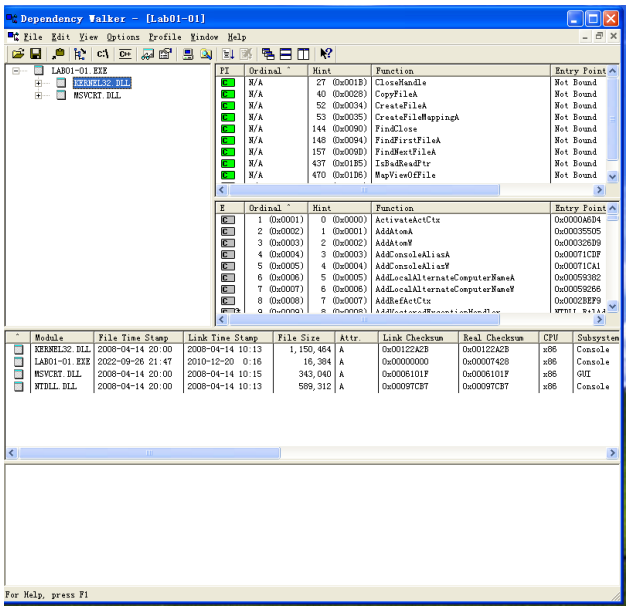

图 3.11: 在 WIndows xp 中使用 Dependency Walker

- 反汇编功能: IDA 能够将二进制文件中的机器代码转换为汇编语言代码，使得开发者能够理解程序的执行逻辑和结构。这是逆向工程的基础步骤，通过反汇编，用户可以获取到程序的底层指令集。同时，IDA 支持多种处理器架构的二进制文件，包括 x86、ARM、MIPS、PowerPC 等，使其在不同平台和指令集下都能发挥作用。IDA 还具有高级反编译功能，将汇编代码转换为接近原始源代码的高级语言代码（如 C 语言）。虽然不能完全还原原始代码，但可以提供对代码功能的更高层次理解。

- 分析和标注功能: IDA 提供了强大的分析功能，可以自动识别和标注函数、变量、字符串、常量等，帮助逆向工程师快速理解程序结构和数据流。并且能够生成函数调用图、控制流图等图形化表示，帮助用户更直观地理解程序的执行路径和逻辑。

- 插件和脚本支持: IDA 支持通过插件和脚本扩展功能，用户可以使用 Python 或 IDC 脚本语言编写自定义插件和自动化任务，提高逆向分析的效率。这种扩展性使得 IDA 能够适应不同的分析需求，并随着用户技能的提升而变得更加强大。

- 导入/导出功能: IDA 支持导入和导出多种文件格式，如 ELF、PE、Mach-O 等，方便与其他工具进行数据交换和协作分析。这种兼容性使得 IDA 能够轻松集成到现有的逆向工程工作流程中。

- 交互式界面: IDA 提供了友好的图形用户界面，支持多种操作方式，如图形视图、文本视图和交互式调试等，方便用户进行逆向分析。这种交互性使得 IDA 不仅适合经验丰富的逆向工程师，也适合想我这样的初学者逐步学习和掌握逆向工程技能。

- 自动化脚本：使用 Python 或 IDC（IDA 的内置脚本语言）等，可以编写自动化脚本来执行各种任务，从简化重复工作到执行自定义分析操作。

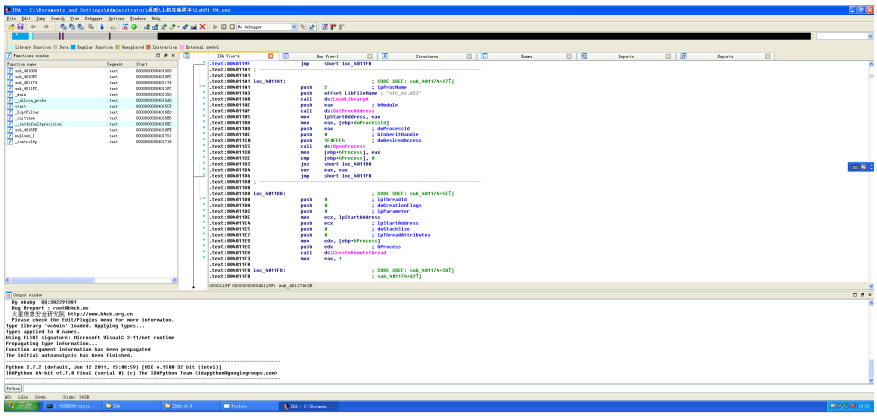

图 3.12: 在 Windows xp 中使用 IDA

3.3 动态分析工具的功能与安装

(1) OllyDBG

OllyDbg（简称 OD）结合了动态调试和静态分析的特性，并拥有一个可视化界面，使得调试过程更加直观和易于操作。OllyDbg 的主要功能如下：

- 调试与跟踪: OllyDbg 支持多种类型的断点，包括代码断点、内存断点、访问断点和条件断点。这些断点允许用户在程序的特定位置暂停执行，以便检查和分析程序的状态。用户可以单步执行程序，逐条查看程序的执行情况，包括寄存器状态、内存状态等，这对于精确控制程序执行流程非常有用。并且 OllyDbg 允许用户查看程序的调用栈，了解程序的函数调用关系和执行路径，帮助用户理解程序的逻辑结构。

- 反汇编与代码分析: OllyDbg 内置了强大的反汇编引擎，可以将二进制代码转换为汇编语言代码，使用户能够直接查看和分析程序的底层指令。OllyDbg 能够高亮不同类型的指令和操作数，并提供注释功能，帮助用户快速理解代码的功能和目的，同时能够自动分析函数过程、循环语句、选择语句等，增加代码的可读性，减少出错的可能性。

- 内存与寄存器操作: 用户可以在程序运行时搜索内存中的特定值、字符串或十六进制模式，并直接编辑内存中的内容，这对于修改程序的运行时行为或调试复杂问题非常有用。OllyDbg 提供了寄存器窗口，显示当前所选线程的 CPU 寄存器内容，用户可以查看和修改寄存器的值，以影响程序的执行流程。

- 插件与脚本支持: OllyDbg 支持第三方插件，这些插件可以添加新的功能、增强现有功能或提供特定于应用程序的工具，使得 OllyDbg 的功能更加丰富和强大。同时，OllyDbg 支持脚本语言（如 Python），用户可以使用脚本自动执行任务、处理数据或执行复杂操作，提高调试效率。

- 跨平台与兼容性: OllyDbg 可以在各种版本的 Windows 操作系统上运行，包括 Windows 95、98、ME、NT、XP 等，并且支持 32 位和 64 位应用程序（尽管在 64 位应用程序上可能有限制）。同时，程序能够识别 Borland 和 Microsoft 格式的调试信息，包括源代码、函数名、标签、全局变量、静态变量等，这些信息对于调试和分析程序非常有用。

- 兼容性：PEview 支持多种操作系统平台上的 PE 和 COFF 文件查看，是逆向工程、恶意软件分析、软件安全等领域中不可或缺的工具之一。

- 导出信息：可以将 PE 文件的信息导出为文本文件，以便后续分析和记录。

使用 Ollydbg 打开实验 1 中 Lab01-04.exe:

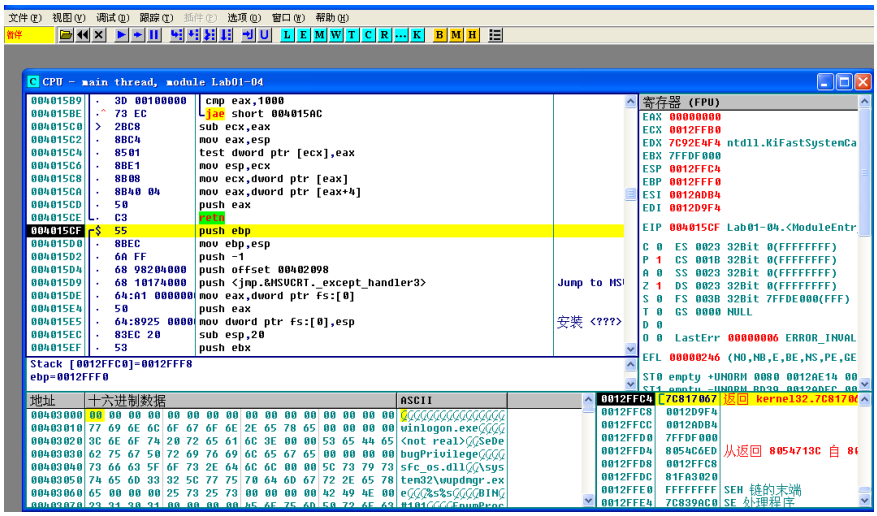

图 3.13: 在 Windows xp 中使用 Ollydbg

(2) Process Monitor

Process Monitor 是一款 Windows 系统的高级监视工具，它结合了 Sysinternals 实用程序 Filemon 和 Regmon 的功能，并添加了一系列增强特性。Process Monitor 的主要功能：

- 实时监视和记录: Process Monitor 能够实时显示和记录 Windows 文件系统的所有活动，包括文件的创建、读取、写入、删除等操作。这有助于用户监控和分析文件系统的使用情况，以及发现潜在的文件操作异常。它还能实时记录注册表的读写操作，包括注册表的创建、查询、设置、删除等。这对于诊断注册表相关的系统问题或发现恶意软件活动非常有用。

- 进程和线程监视: Process Monitor 跟踪所有进程和线程的创建、退出以及 DLL 和设备驱动程序的加载操作。用户可以通过这些信息了解系统的进程和线程状态，以及它们之间的交互关系。通过扫描系统中所有活动的线程，Process Monitor 可以为每个线程生成性能分析事件，记录内核模式和用户模式的 CPU 时间消耗，以及线程自上次分析以来执行的环境开关数量。这有助于用户分析系统的性能瓶颈和优化系统性能。

- 强大的过滤和搜索功能: Process Monitor 允许用户在不丢失任何数据的情况下设置过滤器，以便仅关注特定的文件系统、注册表或进程/线程活动。我们也可以通过搜索功能快速定位到特定的文件系统操作、注册表操作或进程/线程活动，并可以通过 “Jump To” 功能跳转到相关的文件、注册表项或进程/线程详细信息。

- 详细的事件信息: Process Monitor 提供了丰富的事件属性信息，包括事件序列号、类别、操作类型、日期和时间、路径、结果、附加信息等。这些信息有助于用户深入理解每个事件的具体内容和上下文。而对于每个进程，Process Monitor 都能捕获其详细信息，包括映像路径、命令行、公司名称、产品描述、版本号、进程 ID、线程 ID、用户名、会话 ID 等。这些信息对于诊断进程相关的系统问题或分析进程行为非常有用。

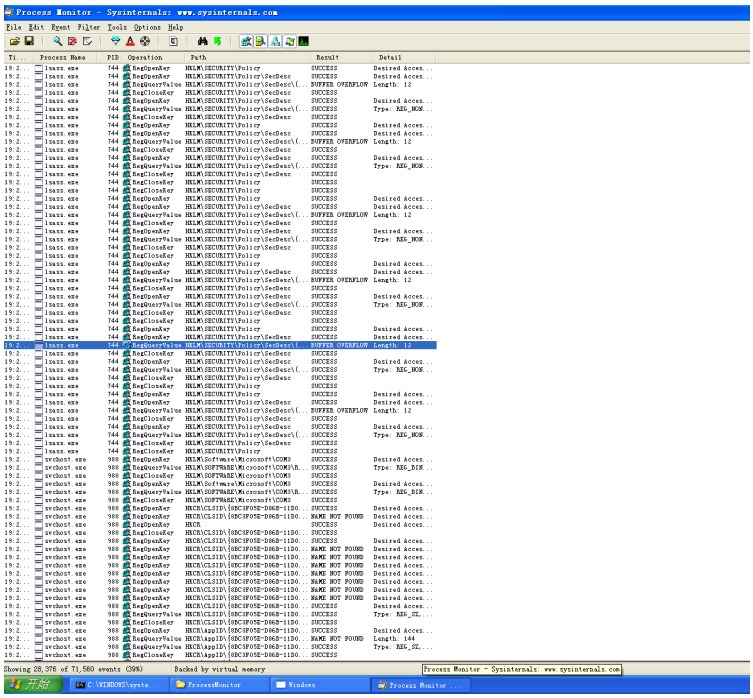

图 3.14: 在 WIndows xp 中打开 Process Monitor 查看

(3) Process Explorer

Process Explorer 是一款功能强大的 Windows 系统和应用程序监视工具，功能包括：

- 进程管理: Process Explorer 能够显示当前 Windows 中正在运行的所有进程的详细信息，包括进程 ID、父进程 ID、映像名称、命令行参数、用户账户等。这些信息有助于用户了解每个进程的来源和用途。通过树形结构展示进程之间的父子关系，使用户能够清晰地看到进程之间的关联和依赖。支持结束任何进程，包括系统级别的关键进程。此外，还可以挂起、恢复、重启进程，以及杀死线程。

- 资源监控: 工具提供详细的资源使用情况图表，包括 CPU、内存、磁盘 I/O 等，帮助我们了解系统的资源分配和使用情况，并通过曲线图表展示 CPU 和内存等使用情况的变化趋势，有助于用户分析系统的性能瓶颈和优化方案。

- DLL 和句柄查看: Process Explorer 能显示当前进程所加载的所有 DLL 文件，帮助用户了解进程所需的所有依赖关系，并显示进程打开的所有句柄，包括文件、注册表项、网络连接等，有助于用户发现句柄泄露等问题。

- 搜索和定位: Process Explorer 提供搜索功能，用户可以根据进程名称、PID 等条件快速定位到特定的进程，该工具也支持设置过滤器，以便仅显示符合特定条件的进程或事件。

- 系统监视: Process Explorer 会监视系统的磁盘和网络活动，显示文件的读写操作和网络连接状态，并显示和监视 Windows 服务，帮助我们了解服务状态并管理服务。

- 安全性保障: Process Explorer 显示进程的安全令牌信息，包括用户账户、权限等，有助于我们了解进程的安全上下文，同时该工具可以用于诊断系统性能、查找恶意软件等方面，帮助用户维护系统的安全性和稳定性。

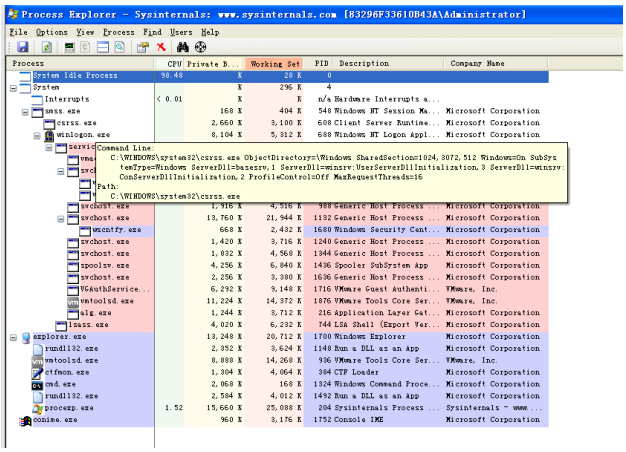

图 3.15: WIndows xp 中使用 Process Explorer

(4) RegShot

RegShot 是一款注册表比较工具，具有多种实用功能，主要包括以下几个方面：

- 注册表快照与对比：RegShot 能够扫描并保存注册表的“快照”，即在特定时间点捕捉注册表的状态。这为用户提供了注册表在某一时刻的详细记录。再通过对比两次不同时间点的注册表快照，RegShot 能够自动找出快照间存在的不同之处。这种对比功能对于分析注册表变化、诊断系统问题或监控软件安装卸载等场景非常有用。
- 文件系统快照比较：除了注册表，RegShot 还能够捕捉文件系统的快照。这使用户能够跟踪文件的创建、修改和删除，从而检测到系统或应用程序的文件操作。
- 详细报告生成：RegShot 生成详细的报告，以清晰地展示注册表和文件系统的更改。这有助于用户跟踪和分析系统在不同时间点的状态变化。
- 自动化测试：RegShot 可以用于自动化测试环境中，以确保应用程序或系统在不同配置下的行为一致性。它能够帮助开发人员检测潜在的问题和不一致之处。

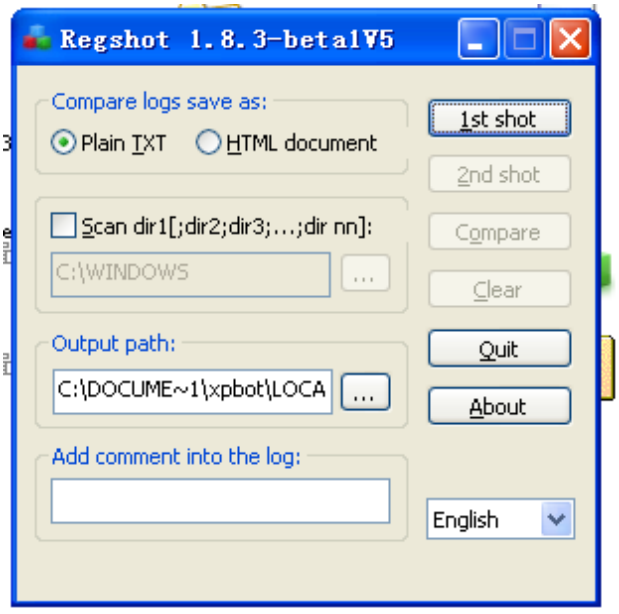

图 3.16: 在 Windows xp 中启动 RegShot

(5) WireShark

Wireshark 是一款开源的网络协议分析工具，其主要功能包括：

- 网络流量监控：Wireshark 作为强大的网络分析工具，能够全面捕获并监视网络上的双向数据流动，让用户洞悉每一笔数据传输的细节，确保网络活动的透明度。
- 深度数据包审查：该工具允许用户深入探索捕获的数据包，细致分析源地址、目标地址、端口信息及使用的协议类型等关键要素，有效辅助诊断网络通信中的疑点或异常行为。
- 多协议支持与解析：Wireshark 集成了对 TCP、UDP、HTTP、DNS 等多种网络协议的解析能力，能够直观展示数据包内部结构及协议间的交互细节，为深入理解网络通信提供了坚实基础。
- 灵活流量筛选：通过预设或自定义的过滤器，用户可以高效地从海量数据包中筛选出感兴趣的流量，聚焦于特定协议、地址或通信模式，简化分析流程。
- 实时网络性能统计：Wireshark 提供的统计功能，能够即时反馈网络流量的总体概况，包括数据包总数、协议分布、传输速率等关键指标，助力用户快速定位网络瓶颈与性能瓶颈。
- 高效网络故障排查：借助对数据包内容的详尽分析，Wireshark 成为网络故障排除的得力助手，帮助用户迅速锁定问题根源，加速网络恢复与优化进程。
- 增强的网络安全监控：在网络安全领域，Wireshark 同样发挥着重要作用，通过监控数据包中的异常模式与潜在威胁，为网络安全团队提供实时情报，助力构建更加坚固的安全防线。

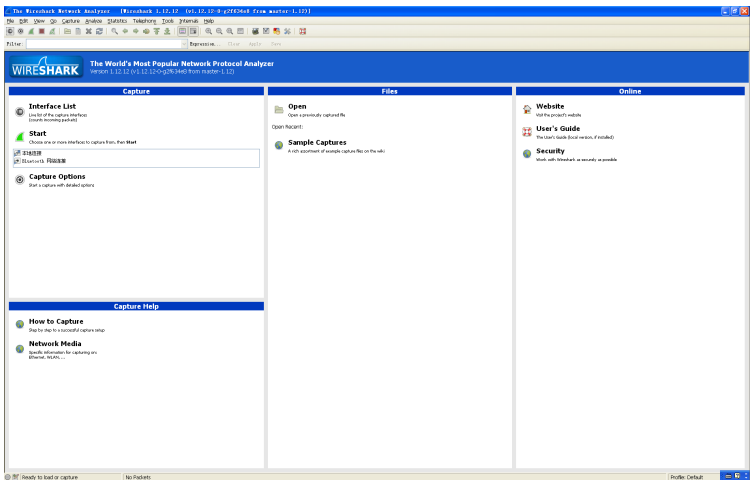

图 3.17: 在 Windows xp 中启动 WireShark

4 实验结论及心得体会

4.1 实验结论

通过本次实验经历，我深刻体会到了在恶意软件分析领域，构建并利用虚拟化环境其不可或缺的价值。这一环境如同一个安全的沙箱，让我们能够在不触及实际系统安全边界的前提下，深入探索恶意软件的行为特性，有效规避了潜在的系统损害风险。

实验过程中，我系统学习了静态与动态分析两大核心工具集的应用，它们各自扮演着解析软件本质与追踪运行时动态的关键角色。静态分析工具，诸如 string.exe、PEView 及 IDA Pro，为我们展示

二进制代码的内部构造与隐藏信息，可以说它们是理解恶意软件“骨架”的得力助手。而动态分析工具，如 OllyDbg 调试器、Process Monitor 进程监控器及 Wireshark 网络分析工具，则让我能够亲眼目睹恶意软件在运行时的状态，通过捕捉实时行为数据，如文件操作、注册表修改及网络通信等，为识别恶意意图提供了宝贵线索。

尤为重要的是，我认识到数据收集与分析在恶意软件分析流程中的基石地位。无论是静态分析中对二进制文件的细致剖析，还是动态分析中实时捕获的系统活动记录，都是构建恶意软件行为画像、识别其攻击模式与目的不可或缺的一环。通过精准捕捉并分析这些关键数据，我们能够更加准确地评估威胁等级，制定有效的防御策略。

综上所述，本次实验不仅为我搭建了恶意软件分析的基本框架，更在实战中锻炼我的技能与思维，为后续深入探索该领域打下坚实的基础。掌握这些工具的使用，将提升我在面对复杂多变的恶意软件威胁时的检测与应对能力。

4.2 心得体会

总体来说，这次实验的难度不大，实验前半部分的虚拟机安装在之前的课上已经熟练掌握，对于静态和动态工具，有一部分在汇编与软件安全课程中也有涉及，但我对于动态分析工具的使用其实还不是很熟练，需要在未来的学习中不断熟悉，了解分析原理。