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：周钰宸

学号：2111408

专业：信息安全

2024 年 1 月 14 日

1 实验目的

1. 配置病毒分析虚拟机；
   - VMare 虚拟机或其它虚拟机软件；
   - Windows XP 操作系统。
2. 虚拟机中安装静态分析工具：string.exe、PEView、dependency walker、IDA 等工具
3. 虚拟机中安装动态分析工具：
   - 预习教材 chapter 3 的 basic dynamic analysis；
   - OllyDBG、Process Monitor、Process Explorer、RegShot、WireShark 等工具
4. 实验报告内容：1. 虚拟机的安装和配置过程；2. 静态分析工具的功能和安装过程；3. 动态分析工具的功能和安装过程。

2 实验原理

2.1 虚拟机

2.1.1 虚拟机技术

虚拟机技术是一种将物理计算机划分为多个独立虚拟环境的技术。这些虚拟环境可以在同一台物理计算机上运行，每个环境都具有自己的操作系统、应用程序和资源。**在恶意病毒分析中，虚拟机技术可以用于创建隔离的、安全的实验环境，有助于进行恶意代码样本的分析、病毒行为观察以及安全漏洞测试**。

2.1.2 VMware

VMware 是一种常用的虚拟化软件，它允许用户在单台物理计算机上运行多个虚拟机，每个虚拟机都可以运行独立的操作系统和应用程序。在恶意病毒分析中，VMware 可以发挥以下作用：

1. 安全隔离：VMware 提供了隔离的虚拟环境，可以在单个物理计算机上运行多个虚拟机，从而使恶意代码无法对物理系统造成直接伤害。
2. 实验环境：安全地在虚拟机中运行恶意样本，观察其行为，分析其特征和攻击方式，而不影响物理系统的安全。
3. 取样与分析：在虚拟机中运行疑似恶意样本，进行样本的捕获、取样和分析，帮助研究人员深入了解恶意代码的工作原理、攻击链和传播方式。
4. 系统还原：可以使用快照和复制技术，将虚拟机系统恢复到安全状态，避免持久性损害并方便下一轮的分析。

2.1.3 VMware workstation

1. 定位：VMware Workstation 是适用于个人用户和企业的桌面虚拟化软件，旨在为开发人员、测试人员和技术爱好者提供一个功能丰富的虚拟化平台。
2. 功能：它具有强大的虚拟机创建、管理、快照、克隆和网络功能。用户可以在单个物理计算机上运行多个虚拟机，模拟不同的操作系统和网络环境。
3. 操作系统：支持在 Windows 和 Linux 主机上运行。

2.2 Windows XP Professional

Windows XP Professional 是微软公司于 2001 年推出的一款操作系统。它是 Windows XP 系列中的专业版，适用于商业和专业用户。并且作为隔离环境和恶意代码分析的环境，其具有如下优势：

1. 广泛使用基础：Windows XP 是一个历史悠久且广泛使用的操作系统，因此它是许多恶意代码的目标。分析恶意代码时了解它在这个平台上的行为非常重要。
2. 兼容性：由于其广泛的硬件和软件兼容性，恶意代码可以在其上运行，提供了一个理想的环境用于分析不同类型的恶意代码。
3. 虚拟化支持：Windows XP 可以很容易地在虚拟化环境中运行，这为安全研究人员提供了一个安全且可控的实验环境，可以在其中分析恶意代码。
4. 丰富的工具和资源：由于其广泛使用，有许多安全工具、恶意代码分析工具和资源适用于该操作系统的平台。这些工具可以帮助分析人员更好地理解、检测和对抗恶意代码。

2.3 静态分析工具及其功能

1. string.exe:

   string 是一个简单而强大的工具，用于从二进制文件中提取 ASCII 和 Unicode 字符串。它可以帮助分析人员快速定位代码中的字符串，这对于恶意代码分析和漏洞挖掘非常有用。**用于在二进制文件中查找、分析和提取字符串，**帮助分析人员了解程序的常量和信息。

2. PEView:

   PEView 可以用于**查看和分析可执行文件的工具**。它可以显示 PE 文件的结构，包括头部、节表、导入表、导出表、资源等信息，以及**执行动态链接库依赖性分析**。有助于理解二进制文件的组织结构和导入的外部依赖。

3. Dependency Walker:

   Dependency Walker 也用于查看可执行文件和动态链接库的依赖性。它可以分析文件的依赖关系，显示哪些 DLL 被程序调用，以及这些 DLL 之间的依赖关系。用途：**通过分析二进制文件对其他模块（DLL 等）的依赖**，帮助理解程序的运行时行为和可能的问题。

4. IDA:

   IDA 是一款强大的交互式反汇编工具，用于将二进制文件转换为汇编代码，并提供丰富的交互式分析和导航功能。它支持多种处理器架构，如 x86、ARM、MIPS 等。**将二进制文件转换为汇编代码，进行反汇编和分析**。分析人员可以通过 IDA 深入理解程序的功能、逻辑和漏洞，以及对恶意代码进行深入分析。

这些工具在恶意代码分析、逆向工程、软件安全性评估等领域发挥重要作用，能够帮助分析人员理解程序的结构、逻辑和行为。选择合适的工具取决于具体的分析需求和二进制文件的特征。

2.4 动态分析工具及其功能

1. OllyDBG: 这是一款强大的动态调试器，用于分析程序的运行时行为。它允许分析人员观察程序的内存、寄存器、堆栈、指令等，并能够在程序运行时进行断点设置、反汇编、修改内存等操作。**用于动态调试程序，分析程序运行时的内存状态、执行路径和行为，以及发现潜在的漏洞和恶意行为**。

2. Process Monitor: 这是一个高级的任务管理器，可以显示系统中所有进程的详细信息，包括进程的资源使用情况、打开的句柄、线程信息等。**用于查看和分析系统中运行的进程、线程、资源情况，以及进程间的关联和依赖关系**。

3. Process Explorer: 这也是一款高级系统监控工具，可实时展示系统中运行的进程、线程和资源使用情况，提供丰富的进程详细信息和系统性能图表，帮助用户管理、监控和优化系统性能。**用于进程查看与管理、系统资源监控、进程树形结构、详细信息查看、搜索与筛选、性能图表、系统信息查看和窗口查看**。

4. Regshot: 这是一个注册表快照工具，用于比较系统或程序在两个时间点的注册表状态，找出在两个时间点注册表的变化。**用于监视程序运行时对注册表的操作，以便分析程序对系统配置的影响**。

5. Wireshark: 这是一款网络数据包分析工具，能够捕获、分析和展示网络通信中的数据包。它支持多种网络协议解析和过滤功能。**用于分析程序的网络活动，包括通信的协议、数据包内容、源目标信息等，有助于理解程序的网络行为和潜在的安全问题**。

这些工具在恶意代码分析、系统性能优化、网络安全等领域发挥重要作用。选择合适的工具取决于分析需求和目标。

2.5 Windows3.2

Windows3.2 是早期的 Microsoft Windows 操作系统版本，推出于 1994 年，是 Windows 3.x 系列的一部分。这个版本在当时为 Windows 操作系统的发展奠定了基础，为后续 Windows 版本的改进奠定了技术基础。然而需要指出的是，Windows3.2 相对于现代的操作系统来说已经非常古老且不再被广泛使用。**其作为病毒分析环境工具并不太适合，**现代病毒分析和恶意代码分析更多关注当前流行的操作系统，如 Windows 7、8、10 以及各种 Linux 发行版。

2.6 其他相关原理

2.6.1 NAT

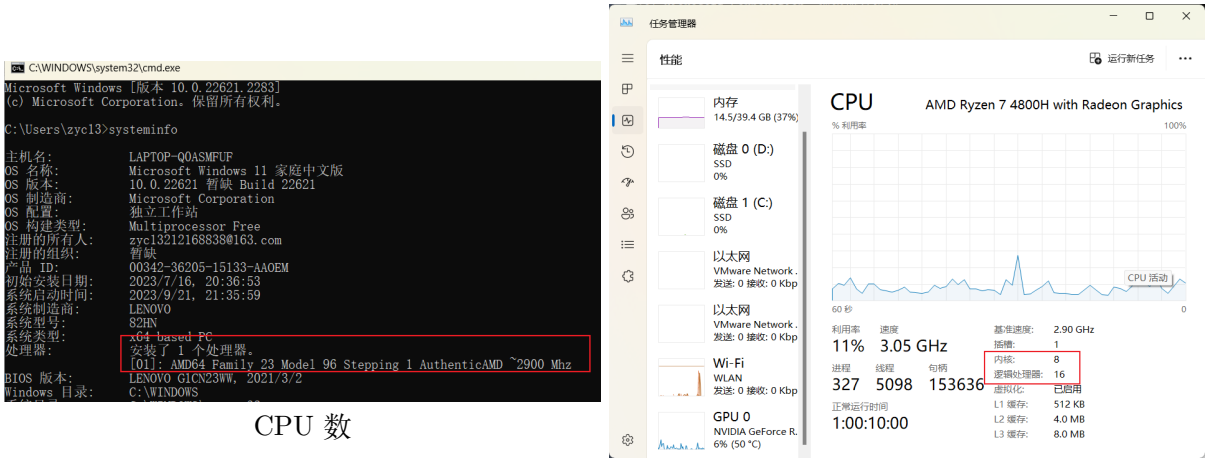
NAT（网络地址转换）是一种常用的网络连接方式，特别在虚拟化环境中，它允许多个虚拟机共享同一个真实物理网络接口，而无需为每个虚拟机分配独立的物理网络接口。

使用 NAT 连接方式可以节省 IP 地址资源，因为多个虚拟机可以共享同一个公网 IP 地址，而且实现了虚拟机之间的网络隔离，增强了安全性。更重要的是**相对于桥接方式等更为简单，不需要特别配置网络，适合初学者或对网络配置不熟悉的用户**。

2.7 处理器与内核

虚拟机安装配置的时候经常需要填写设备的处理器数目以及每个处理器的内核数量，这里实际上表述的意义是

1. 处理器数量: 物理处理器的数量，也就是 CPU 个数。消费级电脑通常都是一个 cpu，但服务器通常有多个 cpu，因此 VMware 需要模拟那些有多个 cpu 的虚拟机，**所以如果没有特殊需求建议这里大家在处理器数量这一栏通常填‘1’即可**。若为多核处理的主机，也可以将虚拟机的处理器数量设置为主机的物理 CPU 数量，以充分利用主机性能。
2. 内核数量：每个处理器的内核数量那一栏直接填写内核数量，**最好不超过本机的 50%**。



CPU 数　　内核数

图 2.1: VMware 的下载安装

以我的本机而言，CPU 个数为 1，内核数为 8。

2.8 .vmx,.iso 与.vhd 文件

这三种文件后缀通常与虚拟化、虚拟机或虚拟硬盘有关。也与本次的**彩蛋实验：Windows3.2 有关**。

2.8.1 .vmx (VM Configuration File)

1. 含义: .vmx 文件是 **VMware 虚拟机配置文件**，它包含虚拟机的配置信息，如虚拟硬件配置、网络设置、操作系统等。
2. 特点：**以文本格式保存虚拟机配置信息，可以手动编辑或通过 VMware 管理界面配置**。指定虚拟机的硬件配置，包括处理器、内存、网络等。

2.8.2 .iso (ISO Image File)

1. 含义：.iso 文件是一种**光盘镜像文件格式**，通常用于光盘映像，包括光盘的文件和目录结构。
2. 特点：包含光盘的所有文件和目录，以便用于光盘的模拟或创建实际光盘副本。**通常用于虚拟机中加载光盘镜像**。

2.8.3 .vhd (Virtual Hard Disk)

1. 含义: .vhd 文件是一种**虚拟硬盘文件格式,** 用于存储虚拟机的硬盘数据。.vhd 文件允许创建虚拟硬盘，将其连接到虚拟机，并在其中安装操作系统及应用程序。
2. 特点：可以创建固定大小或动态增长的虚拟硬盘, 包含虚拟机的硬盘数据，可以模拟实际硬盘的功能。该文件文件可以用于创建备份、快照、迁移虚拟机等。是虚拟化环境中常用的硬盘格式，提供了灵活性和可移植性。

2.9 MS-DOS

MS-DOS（Microsoft Disk Operating System）是由微软开发的一种磁盘操作系统。它最初于 1981 年发布，是微软最早的操作系统之一。

2.9.1 特点和历史

1. MS-DOS 是**基于命令行的操作系统，用户通过输入命令来执行操作，没有图形用户界面即 GUI**。
2. 它主要提供了文件管理、磁盘管理、程序运行等基本功能。
3. 早期的 MS-DOS 版本并不支持多任务处理和多用户操作。
4. MS-DOS 的**命令和操作方式相对底层，需要用户了解硬件细节和命令语法**。

2.9.2 与 Windows3.2 的关系

1. MS-DOS 在早期是微软的主要操作系统，而 Windows3.2 是基于 MS-DOS 的。**Windows3.2 需要 MS-DOS 作为底层操作系统的基础，并在其上提供了图形用户界面和更多高级功能**。
2. Windows3.2 是在 MS-DOS 上运行的图形用户界面，用户可以通过鼠标和窗口进行操作，而不仅仅依赖命令行。
3. Windows 3.2 的推出标志着 Windows 系列开始向图形化、用户友好的操作系统转变，奠定了 Windows 后续版本的基础。

总的来说，MS-DOS 是 Windows3.2 的基础操作系统，Windows3.2 是对 MS-DOS 的图形化扩展，为用户提供了更直观、易用的界面和操作方式。

3 实验过程

3.1 实验环境

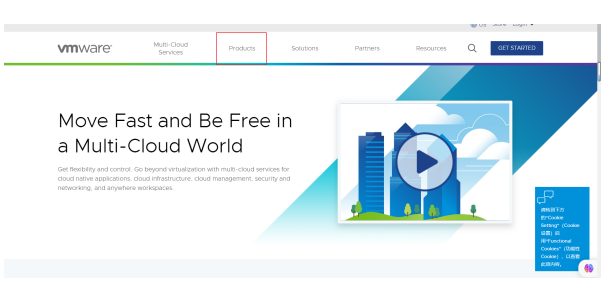
虚拟机	VMware Workstation 17 Pro
操作系统 1	Windows XP Professional
操作系统 2	Windows3.2 中文版

表 1: 本次实验环境

3.2 虚拟机安装与配置

3.2.1 虚拟机安装

由于在大二下的软件安全课程中，已经多次使用过 VMware 作为虚拟机软件。早已完成过相关的安装和配置，因此这里不再重复实验，仅给出实际操作应该的步骤：



登陆官网

下载 VMware workstation

图 3.2: VMware 的下载安装

登陆官网按照相关提示来到网页 https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html 下载对应版本即可。

3.2.2 虚拟机配置

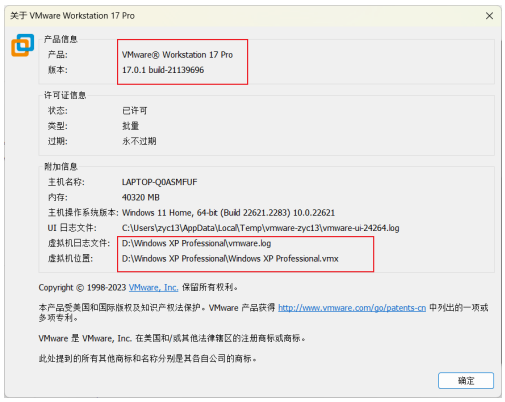


图 3.3: VMware 相关配置

图3.3给出了我的虚拟机相关配置，也是本次实验的基础环境。

3.3 Windows XP Professional

3.3.1 安装与配置

同样因为之前软件安全课程与本课程第一次实验已经多次实验了该操作系统作为实验环境，这里不再进行重装，这里只展示相关配置并解释一些细节：

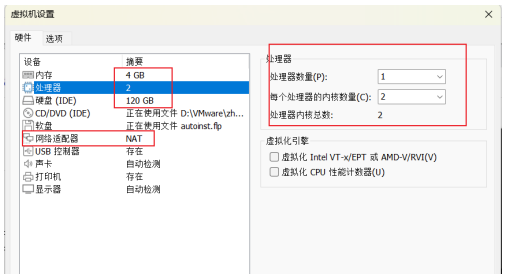


图 3.4: Windows XP Professional 相关配置

图3.4展示了一些基础的设置，比如内存与硬盘大小的分配，我这里分配给了 120GB 的充足磁盘空间用于实验。4GB 的内存以及 **NAT** 的网络连接方式以及与我的电脑适配的选择了 1 个处理器，内核我只选择了 2 个内核。(因为不到我主机内核数 8 的一半)。

3.3.2 静态分析工具安装

这里由于直接使用了拷贝的病毒分析工具集合包，所以部分工具并没有进行实际安装，而可以直接使用。这里只给出相关链接和简单演示。

1. strings.exe：可以通过该链接下载全新版本的 Windows32 位：https://www.split-code.com/strings2.html。这里只进行简单的工具演示：

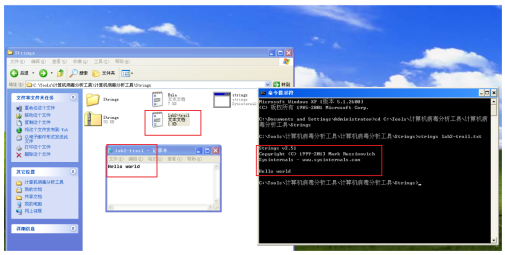


图 3.5: strings.exe 工具演示

图3.5展示了 strings.exe 的简单使用，进入 strings 的对应路径后，运行 strings.exe 文件，使用 strings+ 路径的命令即可实现。比如一个相同目录下的 lab2-trail.txt 文件中只有一行"Hello World"。通过 stirngs lab2-trail.txt 命令可以显示其中的字符串。**用于病毒中，便可以显示其特有的字符串来进行恶意代码分析，正如实验一使用的一样**。

2. PEView：可以通过如下链接下载对应版本的 PEView 即可：http://wjradburn.com/software/。这里只进行简单的工具演示：

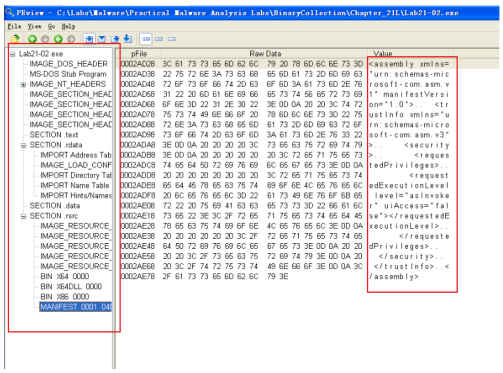


图 3.6: PEView 工具演示

图3.6展示了 PEView 的简单使用，进入 PEView 后，选择打开要查看的具有 PE 格式的文件，这里我使用了病毒 lab21-2.exe，可以看到其 PE 文件格式每一节的字符串内容，**以此作为标识来识别病毒并探索其潜在目的**。

3. Dependency Walker: 这项工具我之前在 lab1 中没有使用，这里到官网 https://www.dependencywalker.com/下载并安装。

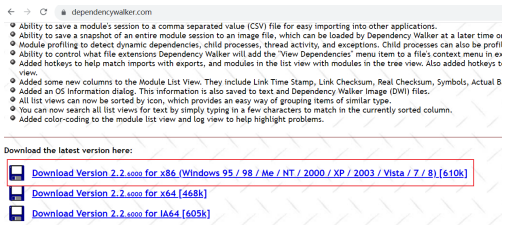


图 3.7: Dependency Walker 官网下载

在本地解压之后使用，以 lab20-02.exe 为例，简单演示其功能：

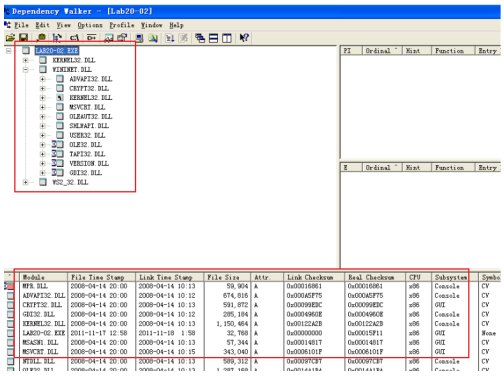
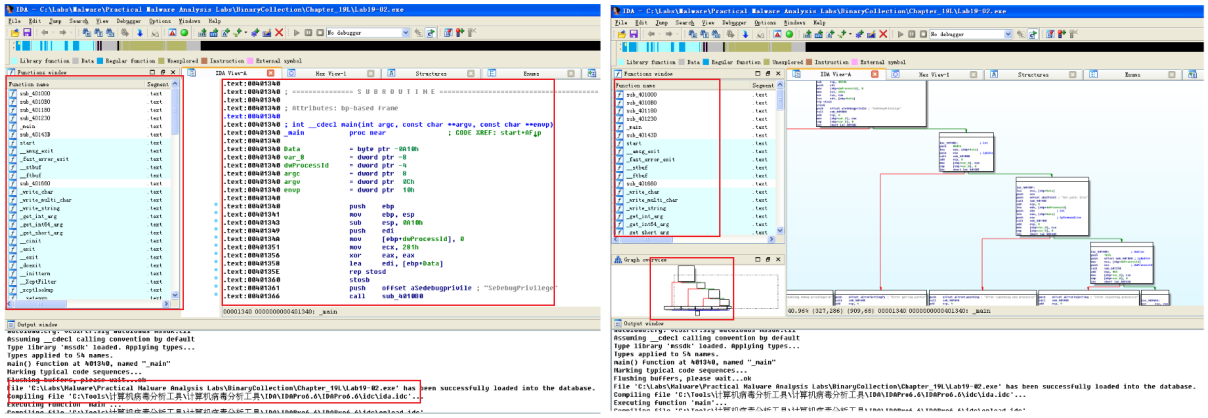


图 3.8: Dependency Walker 工具演示

图3.8可以看到通过 Dependency Walker 能够清晰地显示出**可执行文件和动态链接库的依赖关系，这有助于病毒分析者进一步分析恶意病毒的目的和功能**。

4. IDA: 这个工具由于在之前的计算机病毒分析的压缩包工具中也有，便不再重复安装，可以通过 IDA 官网下载：https://hex-rays.com/download-center/，这里以 lab19-02.exe 为例，简单演示其功能：



IDA 的汇编代码模式　　IDA 的流程图模式

图 3.9: IDA 工具演示

图3.9可以清楚地看到 IDA 的两种工作模式。其中左图中除了显示了所有的汇编函数外，还有直接反汇编成为的汇编代码；右图为图形化的导航模式，可以看到各个汇编代码块之前的跳转关系。**这些多功能的交互式分析和导航功能，可以让病毒分析人员更好地利用 IDA 深入理解病毒程序的功能，并对恶意代码进行深入分析**。

3.3.3 动态分析工具安装

这里也是由于直接使用了拷贝的病毒分析工具集合包，所以部分工具并没有进行实际安装，而可以直接使用。同样有些工具只给出相关链接和简单演示。

1. OllyDBG: 可以通过如下链接 https://www.ollydbg.de/download.html 下载 OllyDBG1.10 即可，这里以 lab18-02.exe 为例只进行简单演示：

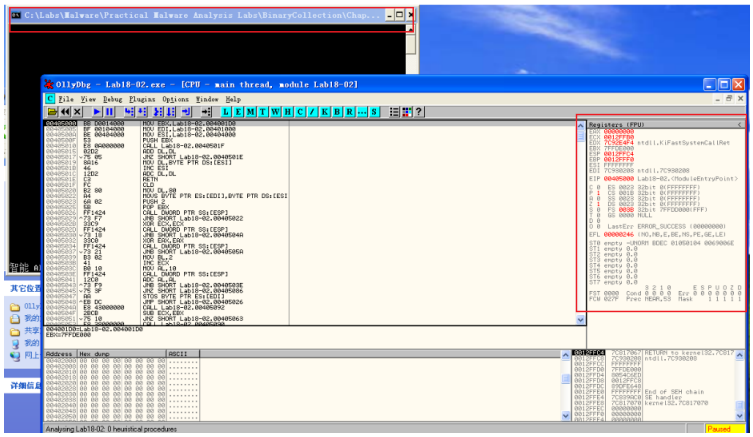


图 3.10: OllyDBG

图3.10可以看到此时的 OllyDBG 在通过实际运行病毒来进行进行动态分析，这个过程允许我们通过观察程序的各个状态和进行一些相关操作来动态调试，**以此更加清晰地分析程序可能潜在的恶意行为**。

2. Process Monitor: 可以通过以下链接 https://learn.microsoft.com/en-us/sysinternals/downloads/procmon 下载即可，这里以该 xp 操作系统为例进行简单演示：

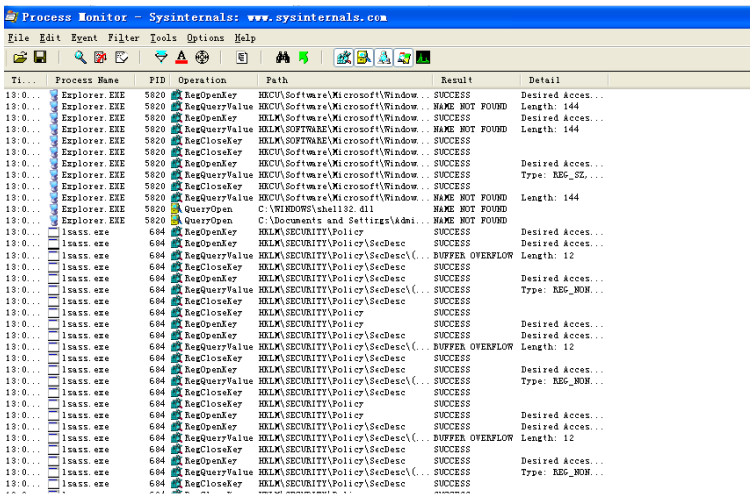


图 3.11: Process Monitor 工具演示

图3.11可以看到此时在 XP 操作系统中运行的多条进程，**以此来根据进程等详细信息和资源使用情况，来判断是否有可疑的恶意代码对其中的某些资源进行了恶意目的的利用**。

3. Process Explorer: 可以通过如下链接 https://learn.microsoft.com/en-us/sysinternals/downloads/process-explorer 下载即可，这里以该 xp 操作系统为例进行简单演示:

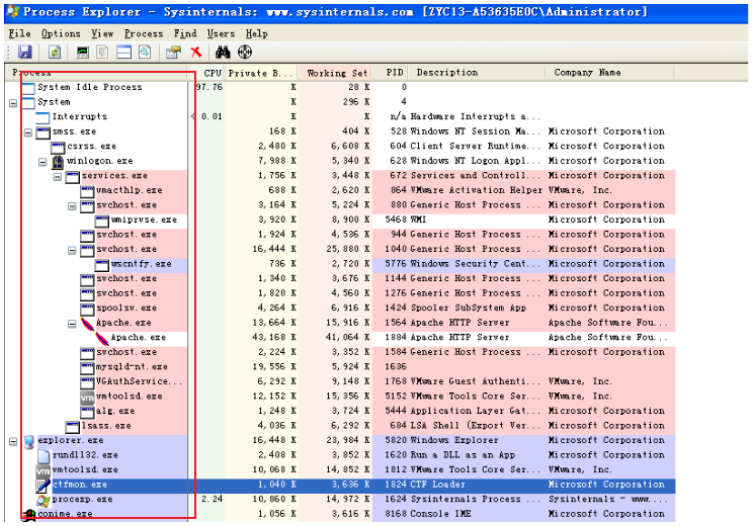


图 3.12: Process Explorer 工具演示

图3.12可以看到此时在 XP 系统中运行的一些进程和线程，以此来进行监管。**Process Monitor 更侧重于监控和记录系统的活动，而 Process Explorer 则更侧重于展示系统运行中的进程和资源信息，帮助用户了解和管理系统中的进程**。

4. Regshot: 可以通过如下链接 https://sourceforge.net/projects/regshot/下载即可，这里以 xp 操作系统为例进行简单演示。此时我的 WireShark 正好没有进行安装，借此机会来比较注册表前后的差异，先在安装 WireShark 之前保存一次快照。

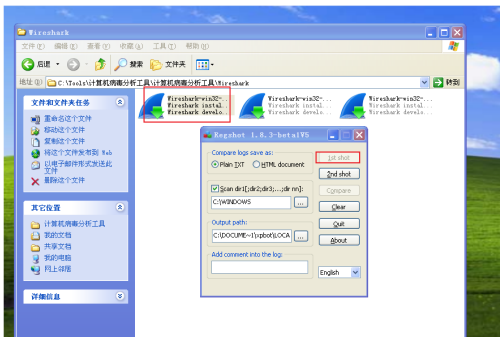
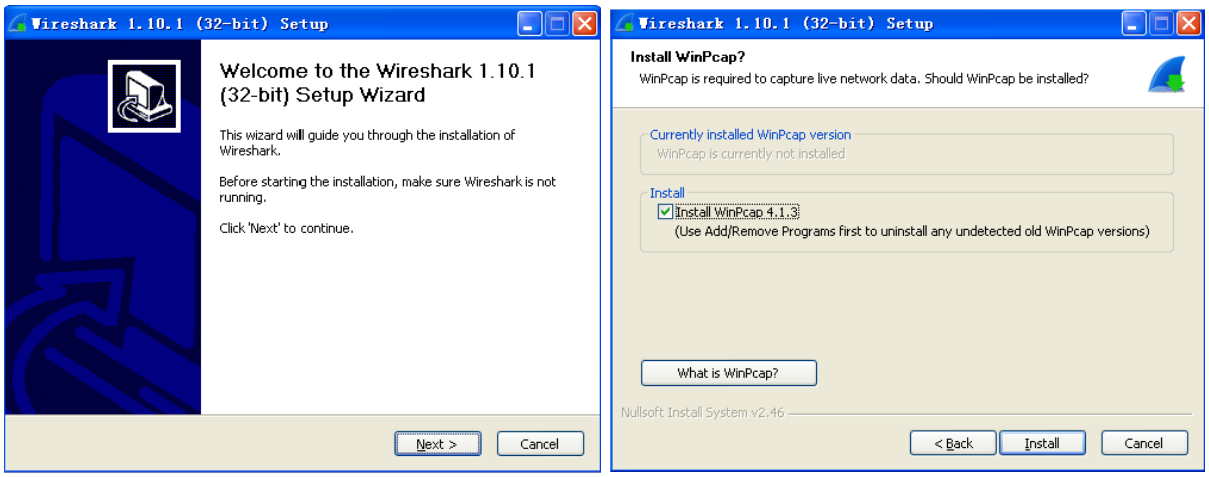


图 3.13: RegShot 拍摄改变注册表前

接下来安装 WireShark:



流程 1 流程 2

图 3.14: WireShark 安装

安装完成后再次拍摄快照，之后比较二者前后的注册表结果，将比较结果以 txt 的形式输出到 RegShot 目录下:

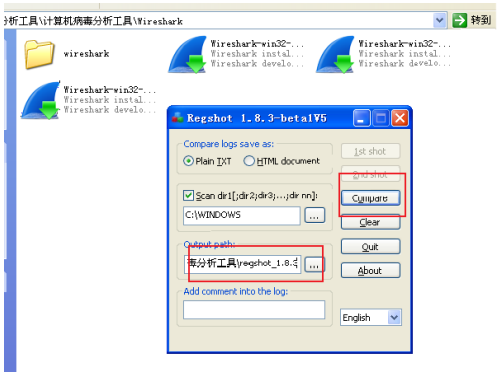


图 3.15: RegShot 拍摄改变注册表后

打开后查看相关内容:

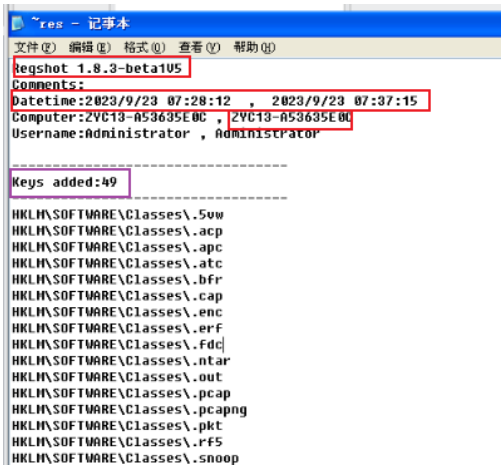


图 3.16: Regshot 比较安装 WireShark 前后注册表结果变化 1

图3.16显示了主机名字，而两次快照拍摄前后的对比，可以看到它显示了有 49 处新添加的**注册表键 (Registry Keys)。可以包含多个子键或者键值。这些很可能就是由于安装了 WireShark 导致的**。

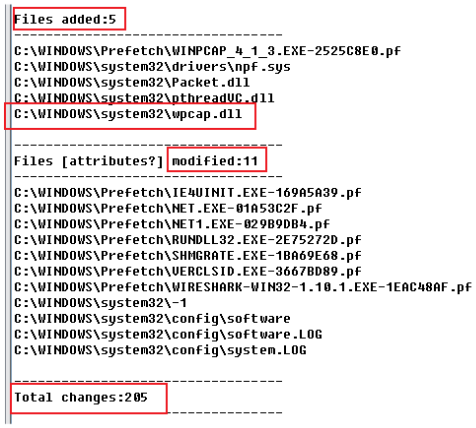


图 3.17: Regshot 比较安装 WireShark 前后注册表结果变化 2

图3.17显示了 C:_Windows 下的 **5 个新添加的文件和 11 个修改的文件**。其中可以发现有刚才安装 WireShark 时顺便安装的 wpcap，被整理为了动态链接库。同时一些系统文件的如 software 配置都发生了改变。**这些都是注册表被改变导致的差异**。

5. WireShark: 可以通过如下链接 https://www.wireshark.org/下载后安装，上面已经进行了安装，这里直接进行简单的演示。打开 WireShark 后可以看到其内部的界面有多个选项：

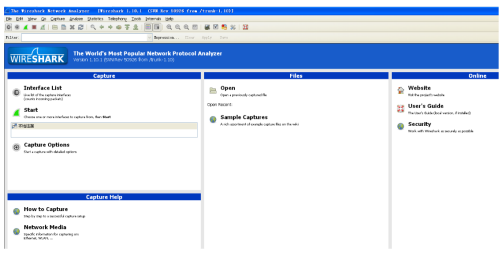
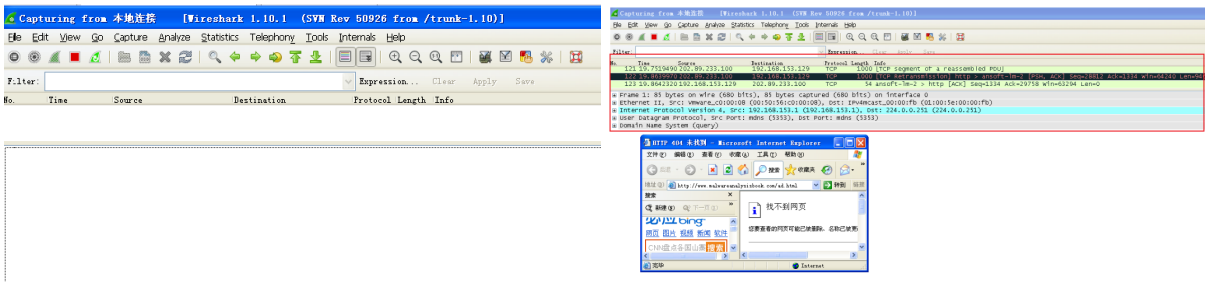


图 3.18: wireShark 工具界面

图??显示了基本的操作界面，点击 start 开始监控后，点击一个 lab1 中已经分析过具有网络行为的病毒 lab01-03.exe，观察前后的分析变化：



网络行为前 网络行为后

图 3.19: 以 lab01-03.exe 演示 WireShark 功能

图3.19可以看到一些检测到的网络包，证明了其网络行为。**进一步分析可以发现其潜在的具体网络行为，将数据包的内容和通信协议等进行抓取**。

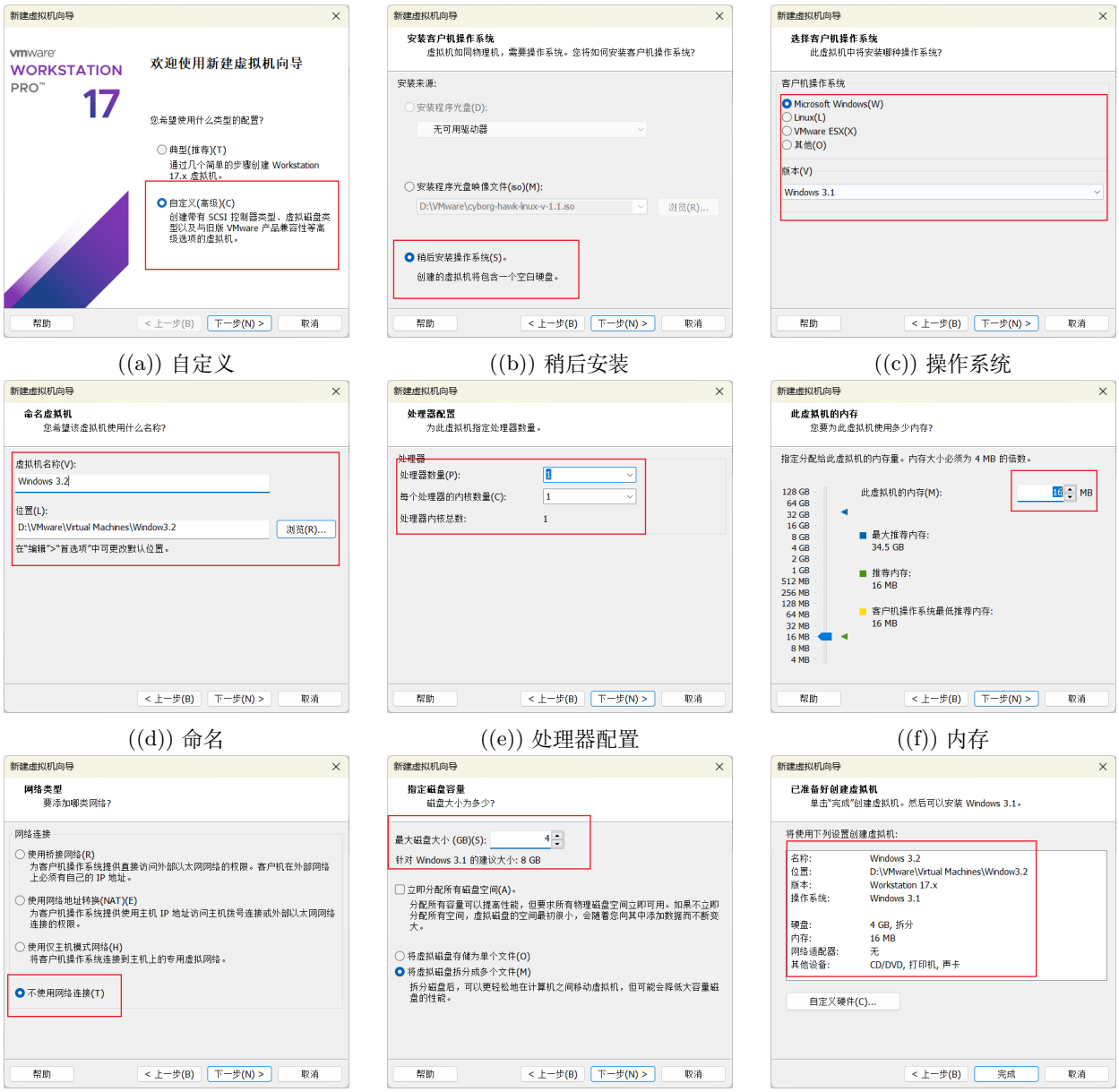
3.4 Windows3.2 中文版

这里是我的彩蛋环节了。

由于我的 xp 系统提前已经装好了。因此上面的实验其实没有进行实际在虚拟机中安装操作系统的过程，因此我想要在这里整一个活。我通过百度贴吧 bochs 吧（一个分享各种操作系统资源的贴吧），找到了一份 Windows3.2 中文版的古老镜像。**出于好奇的原因，我对其进行了加载与安装，并进行了初步的探索**。

3.4.1 新建虚拟机

这里先首先利用 VMware 自带的 Windows3.1 版本安装 Windows3.1 版本的虚拟机，选择的相关配置如下：



((a)) 自定义 ((b)) 稍后安装 ((c)) 操作系统

((d)) 命名 ((e)) 处理器配置 ((f)) 内存

((g)) 网络 ((h)) 磁盘 ((i)) 总配置

图 3.20: 虚拟机 Window3.2 中文版安装配置

图3.20展示了安装 Windows3.1 虚拟机的流程。这一步是为了创建一个基础的虚拟机，安装 Windows 3.1 操作系统。Windows3.1 是一个较早的 Windows 版本，**安装它是为了之后能够更容易升级到 Windows3.2**。

其中值得注意的是图 (g) 中**没有设置网络**，这是因为 Windows3.2 确实不具备现代意义上的网络功能。在当时，网络并不像今天这样普遍普及，因此 Windows 3.2 并未设计具备现代网络连接的功能。

3.4.2 修改配置文件

在这之后，其实安装的并不是真正的 Windows3.2 操作系统，需要进一步提供相关的镜像。

但其实该贴吧的资源并未提供镜像即.iso 文件，而只提供了一个.vhd 文件。这个文件实际上是虚拟机硬盘文件格式。这个磁盘中存储了操作系统、应用程序、文件等。通过查阅资料，我掌握了使用.vhd 文件加载虚拟机的方式。首先，**先将.vhd 文件放到和.vmx 即虚拟机配置文件相同目录下：**

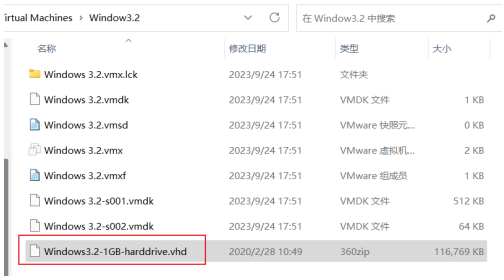


图 3.21: 将.vhd 文件复制进目录

之后，通过管理员权限打开记事本修改虚拟机配置文件.vmx 其中的内容：

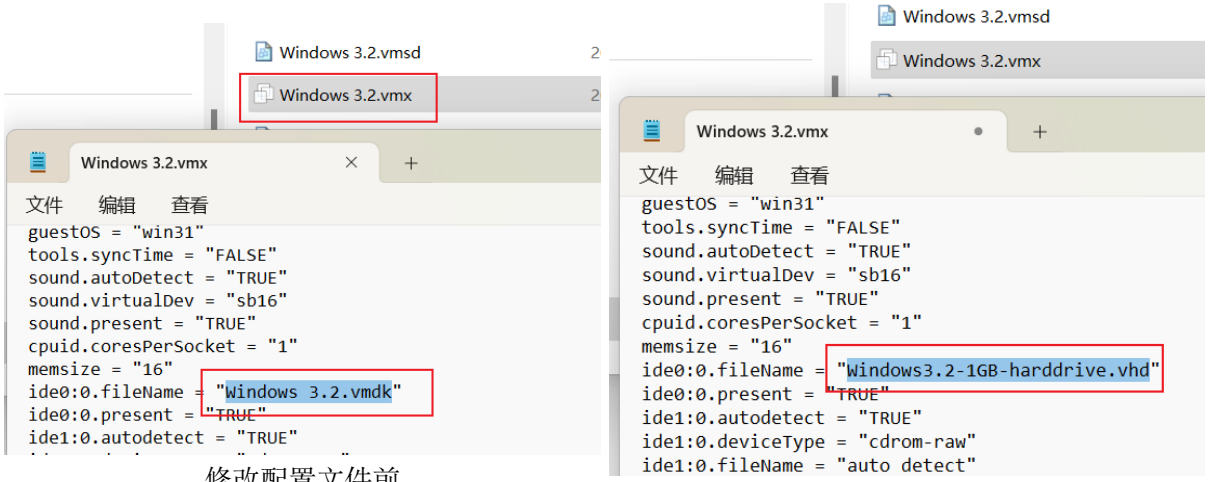


图 3.22: 替换虚拟机磁盘文件

图3.22展示了编辑.vmx 文件的内容。即通过**编辑.vmx 文件中的 ide0:0.fileName 这一项，将虚拟机的磁盘文件替换为想要安装的 Windows 3.2 的虚拟硬盘文件 Windows3.2-1GB-harddrive.vhd**。这样，当启动虚拟机时，它将加载这个新的虚拟硬盘文件作为其主要磁盘，而不再是 Windows 3.1 的磁盘。

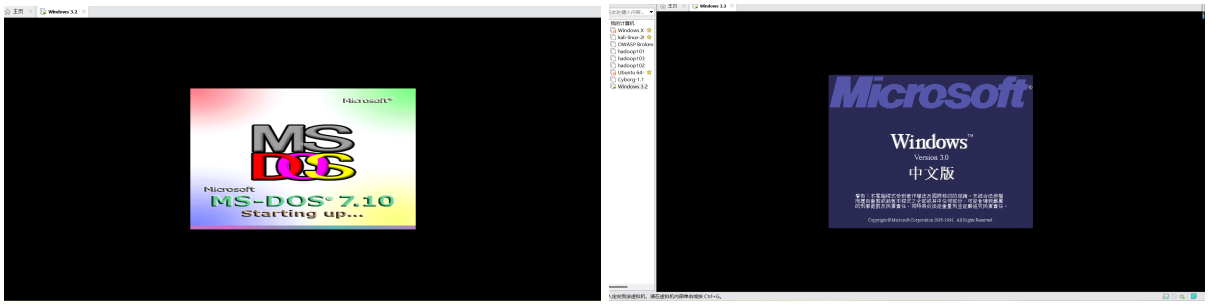
ide0:0.fileName = ”Windows3.2-1GB-harddrive.vhd” 这句话中：

- ide0:0 指定了 IDE 控制器上的第一个磁盘。
- fileName 指定了这个磁盘的文件路径。
- ”Windows3.2-1GB-harddrive.vhd” 是想要指定的新虚拟硬盘文件路径。

改变这个路径会使虚拟机在启动时加载不同的虚拟硬盘。**这样就可以让虚拟机加载其中的内容，从而运行 Windows 3.2 操作系统，而不是之前的 Windows 3.1**。

3.4.3 正式演示

之后回到 VMware，启动已经被修改过的 Windows3.2 中文版：



MS-DOS Windows3.2 中文版

图 3.23: 启动过程

图3.23可以看到此时已经正常启动了 Windows3.2 中文版，尝试其中的几个功能：

图形化界面 命令行

图 3.24: 操作界面

图3.24展示了两种 Windows3.2 的**两种操作方式，左图是图形化界面，右图是命令行界面**。

图 3.25: Windows3.2 文件系统树

图3.25可以看到 Windows3.2 中文版的**文件系统树，和现在的 Windows 版本目录有着较大的差异**。

图 3.26: 绘图游戏

图3.26是我发现的其中的小游戏，画了个 Malware 作为结束。**这就是全部的彩蛋内容啦**。

4 实验结论及心得体会

4.1 实验结论

本次实验，我使用了已经安装的 VMware 虚拟机，在其中以 Windows XP 操作系统和 Windows3.2 版本为例，进行了实验。以 Windows XP 操作系统，在其中我安装了 string.exe、PEView、dependency walker、IDA 等静态分析工具，并且都通过简单的演示，尝试了它们的基础功能；同样的还通过预习教材 chapter 3：basic dynamic analysis ，安装了 OllyDBG、Process Monitor、Process Explorer、RegShot、WireShark 等动态分析工具，并进行了二简单演示。最后以.vhd 文件的 Windows3.2 中文版操作系统为例，演示了安装虚拟机的流程，并简单探索了其内容，作为彩蛋。**总的来说，实验非常成功**。

4.2 心得体会

本次实验，虽然由于虚拟机环境和相关操作系统，甚至是一些工具我都提前学习过一些，并且有些已经能够熟练使用。但还是通过本次实验的反复探索，让我对虚拟机以及一些相关配置有了更深的理解。

1. 我掌握了 string.exe、PEView、dependency walker、IDA 等静态分析 · 工具的安装与使用；
2. 我掌握了 OllyDBG、Process Monitor、Process Explorer、RegShot、WireShark 等动态分析工具的安装与使用；
3. 我通过安装 Windows3.2 操作系统，对 VMware 中的.vmx，.vhd 文件等有了全新的认知，熟悉了怎么使用.vhd 而不是.iso 文件加载虚拟机；
4. 我对虚拟机 Windows3.2 操作系统以及 MS-DOS 操作系统等 Windows 的早期发展历史经过探索有了自己的认识。

总的来说，本次通过亲自实验让我收获颇丰，也加深了我对虚拟机软件例如 VMware 的工具的理解和使用能力。我会用好该软件，辅助我进行更多的病毒分析。