第11章 软件保护技术

11.1 软件保护技术概述

软件保护技术是软件开发者寻找各种有效方法和技术来维护软件版权，增加其盗版的难度，或延长软件破解的时间，尽可能防止软件被非法使用。

从理论上说，几乎没有破解不了的软件。所以对软件的保护仅仅靠技术是不够的，最终要靠国家法制的完善、人们的知识产权保护意识的提高。

11.2 静态分析技术

静态分析是从反汇编出来的程序清单上分析程序流程，从提示信息入手，了解软件中各模块的功能，各模块之间的关系及编程思路。从而根据自己的需要完善、修改程序的功能。

对于破解者来说，通过对程序的静态分析，了解软件保护的方法，也是软件破解的一个必要的手段。

软件密码破解示例：VC++程序

VB程序

11.2.1 文件类型分析

对软件进行静态分析时首先要了解和分析程序的类型，了解程序是用什么语言编写的，或用什么编译器编译的，程序是否有加壳保护。

常用的文件类型分析工具有Language 2000、File Scanner、FileInfo、PEiD等。

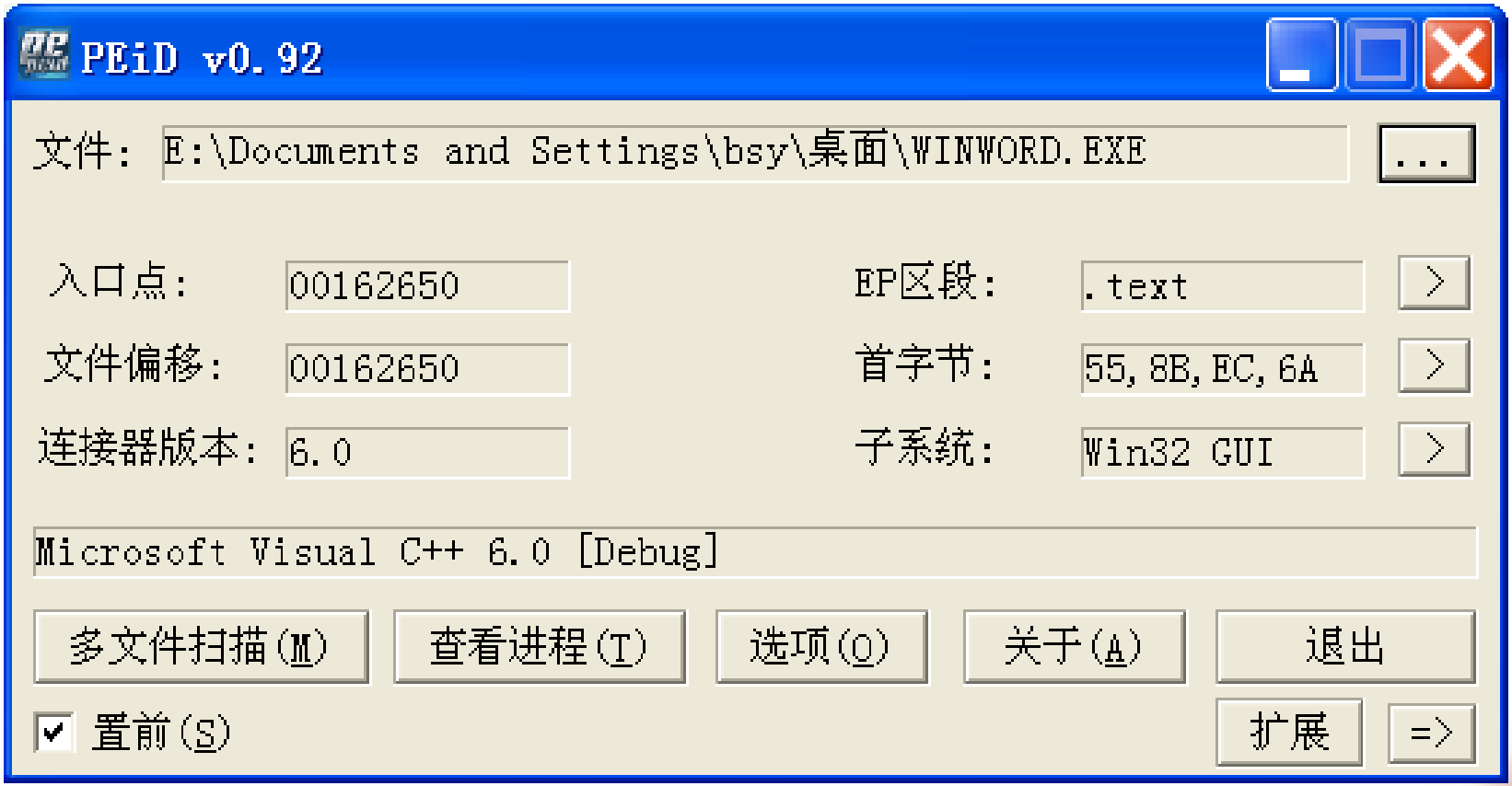
FileInfo运行时是DOS界面，支持Windows长文件名，语法：

fi <path><*.*></r></f></d-></l+></p+>

fi /r 或/s：列出当前目录和子目录所有的文件信息

fi /f ：仅列出能识别的文件

用PEiD软件对Winword.exe文件的分析结果如下图所示。



11.2.2 W32Dasm

W32Dasm针对现在流行的可执行程序进行反编译，即把可执行的文件反编译成汇编语言，以便于分析程序的结构和流程。

W32Dasm不需要安装，只要直接执行W32Dasm.exe文件。

简单介绍W32Dasm功能和使用方法。

1. 加载文件。
2. 转程序入口点。
3. 转到代码开始。
4. 转到页。
5. 转到代码位置。
6. 执行文本跳转。
7. 执行调用。
8. 输入函数。
9. 输出函数。
10. 菜单参考。
11. 对话框参考。
12. 串式数据参考。
13. 复制汇编代码文本。
14. 装载32位的汇编代码动态调试。
15. 单步跟踪程序。
16. 设置激活断点。
17. 保存反汇编文本文件。

11.2.3 IDA Pro简介

IDA相对于W32Dasm来说功能更强大、操作比较复杂。

使用IDA需要注册费用，而W32Dasm是免费的。

当分析一个简单的程序时，使用W32Dasm更为方便。

IDA能够分析加壳的程序，并以多种文件形式保存等。

目前IDA Pro最高版本为IDA Pro 4.6，支持64位处理器，具有更强大的功能。

11.2.4 可执行文件代码编辑工具

W32Dasm和IDA适合分析文件。若要对文件进行编辑、修改，可以使用专门的十六进制编辑工具。如Hiew，HexWorkshop，WinHex，**U1traEdit**等。

Hiew的界面简单、使用方便，它可以对应用程序进行反汇编，同时支持对可执行文件的十六进制代码和汇编语言代码修改。

简单介绍Hiew的使用。

11.2.5 可执行文件资源编辑工具

Windows应用程序的各种操作界面称为资源，包括加速键、位图、光标、对话框、图标、菜单、工具栏、版本信息等。

资源也是一种数据，它们一般被存储在PE文件的.rsrc块中，不能通过由程序源代码定义的变量直接访问，Windows提供的函数直接或间接地把它们加载到内存中以备使用。

对于已打包后的exe，dll和ocx等文件可以通过资源修改工具Resource Hacker、eXeScope和ResScope等修改其资源，它们也是功能强大的汉化和调试辅助工具。

一般资源修改工具具有如下功能：

1. 在已编译和反编译的格式下都可以查看Win32 可执行文件和相关文件的资源）。

2. 提取和保存资源到文件（*.res）格式，作为二进制文件或作为反编译资源脚本或图像。

3. 修改和替换可执行文件的资源。

4. 添加新的资源到可执行文件。

5. 删除资源。

简单介绍资源黑客（ Resource Hacker）软件使用。

11.3 动态分析技术

用静态分析法可以了解编写程序的思路，但是有时并不可能真正地了解软件编写的整个细节和执行过程，在对软件静态分析无效的情况下就可以对程序进行动态分析了。

动态分析就是通过调试程序、设置断点、控制被调试程序的执行过程来发现问题。如根据两个数据运算结果确定程序跳转，静态分析就不行了。

If x-y=2 then {。。。}

else {。。。}

对软件动态跟踪分析时可以分两步进行：

1．对软件进行简要跟踪

主要根据程序的顺序执行结果分析该段程序的功能，找到所要关心的模块或程序段。

2．对关键部分进行细跟踪

在获取软件中关键模块后，这样就可以针对性地对该模块进行具体而详细地跟踪分析。要把比较关键的中间结果或指令地址记录下来，直到读懂该程序为止。

动态分析技术使用的调试器可分为用户模式和内核模式两种类型。

用户模式调试器工作在Win32的保护机制Ring 3级（用户级）上，如Visual C++等编译器自带的调试器就是用户级的。

内核模式调试器是指能调试操作系统内核的调试器，它们处于CPU和操作系统之间，工作在Win32的保护机制Ring 0级（特权级）上。

常用动态分析工具有SoftICE、OllyDbg等。

11.3.1 SoftICE调试器

SoftICE是NuMega公司开发的最著名的动态调试工具，可以调试各种应用程序和设备驱动程序，还可以通过网络连接进行远程调试。

NuMega公司将SoftICE捆绑进驱动开发软件DriverStudio和SoftIC DriverSuite中发行。

DriverStudio是一个设备驱动程序和应用软件开发工具包，是一套用来加速微软Windows设备驱动程序的开发和调试的核心工具。

以SoftICE DriverSuite2.7为例介绍SoftICE的安装与配置。

1．SoftICE安装

SoftICE DriverSuite2.7安装时双击安装文件setup.exe，按照安装向导界面提示就可以完成SoftICE安装了。DriverSuite可自动识别Windows不同版本。安装过程中，安装向导会提示选择3种安装类型。

2． SoftICE启动模式

SoftICE启动模式分两种：

（1）Windows 9x系统启动模式

（2）Windows NT/2000/XP系统启动模式

SoftICE在Windows NT/2000/XP下有四种启动模式可供选择：

（a）Boot模式：SoftICE先于Windows加载，主要用于调试内核驱动程序；

（b） System模式：SoftICE后于Windows加载，主要用于调试一般的应用软件开发；

（c）Automatic模式：SoftICE先于Windows加载，但不能调试内核驱动程序；

（d） Manual模式：进入Windows系统后，需要手动执行快捷方式“Start SoftICE”或在命令行上运行“net start ntice”来装载SoftICE；

Disabled：就是禁止所有的SoftICE组件服务。

一般情况下将启动模式设置为Manual模式，

3．鼠标与显卡配置

如果使用SoftICE时发现显示或鼠标不正常，可以直接用SoftICE修改显卡和鼠标配置。

4. SoftICE环境配置

（1）常规选项（General）

General选项是设置SoftICE初始化命令和一些变量参数的 。

（2）符号选项（Symbols）

预装符号和代码，这对调试设备驱动程序很有用。

（3）导出选项（Exports）

也可通过符号预装加载更多的出口函数列表。

（4）功能键设定（Keyboard Mappings）

可以以设置SoftICE的功能键，也可自定义功能键命令。

（5）宏定义（Macro Definitions）

用宏定义可以更方便地使用SoftICE。

（6）Troubleshooting

该选项卡允许用户对键盘、鼠标和内存页等进行高级设置。

（7）高级设置（Advanced）

该选项允许用户自定义调试窗口中的鼠标右键菜单。

（8）Winice.dat配置

SoftICE初始化设置保存于文件Winice.dat文件中。

5．SoftICE使用

安装SoftICE成功后，按“Ctrl+D”键可以激活并打开一个调试窗口，当需要返回Windows系统，关闭调试窗口时，要再按“Ctrl+D”键

SoftICE调试窗口分为寄存器、浮点、数据、代码、堆栈和命令窗口等几部分。

（1）寄存器窗口。

（2）浮点窗口。

（3）数据窗口。

（4）代码窗口。

（5）系统堆栈窗口。

（6）命令窗口。

SoftICE命令十分丰富，大约有150多个，下面介绍十个最常用的命令，更多的命令请参考SoftICE的命令手册。

（1）A命令

语法: A [address]

address：指定内存地址

作用: 写入汇编代码。

用法: 如果不加地址值，可直接在当前CS:EIP处汇编。

（2）D命令

语法：D[size] [address [L length]]

size：B字节，W字，D双字，S短实型，L长实型或T 10b长实型。

address：指定内存地址

L length：指定长度

作用：显示某内存区域的内容。

用法：D 命令在所指定的内存区域开始显示指定长度的内存单元内容。

（3）G命令

语法：G [=start-address] [break-address]

start-address: 开始地址

break-address: 中断地址

作用: 执行程序。

用法：G命令属于一次性断点，F7功能键有点类似G命令。

（4）P命令

语法: P [ret]

作用: 单步执行程序。

用法: 只执行P时，单步执行程序。如果P后加RET 参数，将一直单步执行到最近的一条返回语句RET/RETF处。

P 命令可用快捷键 F10代替，P RET 命令可用快捷键 F12代替。

（5）R命令

语法：R 寄存器名

作用：显示或更改寄存器的内容。

用法：它可更改所有寄存器的值。

（6）S命令

语法：S [-cu] [address L length data-list]

address：搜索的起始地址。

Length：搜索的长度（字节数）。

data-list：可以是一系列字节，也可以是字符串；字符串可用单引号，也可以用双引号括住。

-c：使查找区分大小写

-u：查找Unicode 编码的字符串

作用：在内存中搜寻特定数据。

用法：S命令将从指定的内存地址开始查找指定内容的数据，一直到超过指定的长度为止。

（7）T命令

语法: T [=start-address] [count]

start-address： 单步跟踪起始地址

count :指定单步跟踪多少次才停止

作用: 单步跟踪

用法：T 命令是利用CPU的单步标志来进行单步跟踪的。

T 命令的快捷键是F8。

（8）U命令

语法: U [address [length] [symbol-name]

address : 段，偏移量或选择符

length : 反汇编的长度（字节数）

symbol-name : 将从指定的函数开始反汇编

作用: 反汇编指令

用法：U 命令将从指定地址开始反汇编指定长度的指令。

（9）BPX命令

语法：BPX [address]

作用：在可执行语句上设置（或清除）断点。

用法：格式为“BPX 地址”时，程序一旦执行到该地址处，SoftICE窗口就会弹出。当光标在代码窗口中时，直接键入BPX就会在光标所在语句 处设断点，再键入BPX 就取消。

（10）BMSG命令

语法: BMSG window-handle [L] [begin-msg [end-msg]][IF expression]

[DO “command1；command2；...”]

window-handle: 消息发向的窗口句柄

begin-msg，end-msg: 消息标识字的范围，如果没有end-msg，那么只在begin-msg上设置断 点，否则在区域内所有消息都会被下断点

IF-expression: 表达式的值为真时，SoftICE才弹出

DO “command1；command2；...”:当到达断点时，执行一系列SoftICE命令。

L：表示不弹出SoftICE，而是在命令窗口中记录消息

作用: 跟踪Windows消息，在Windows的消息上设置断点

用法：如果没有指定在哪个MSG上设置断点，那么所有发向该窗口的消息都会被拦截。

11.3.2 OllyDbg调试器

OllyDbg调试器是兼有动态调试和静态分析为一身的免费软件调试器，可以在Windows9X/NT/2000/XP当前各种版本下运行。

OllyDbg支持80x86、Pentium、MMX，、3DNow!、SSE 指令集、SSE2指令集。

OllyDbg文件很小，不驻留内存。运行OllyDbg.exe就可以了。

简单介绍一下Ollydbg工作界面：

1.代码窗口

2.信息窗口

3.数据窗口

4.寄存器窗口

5.堆栈窗口

11.4 常用软件保护技术

软件保护一般分为软加密和硬加密； 硬加密一般俗称加密狗或加密锁。

11.4.1 序列号保护机制

软件验证序列号的合法性过程就是验证用户名和序列号之间的换算关系，即数学映射关系是否正确的过程。

1．以用户名生成序列号

序列号＝F（用户名）

2．通过注册码来验证用户名的正确性

序列号＝F（用户名）

用户名＝F^{-1}（序列号）

3．通过对等函数检查注册码

F1（用户名）＝F2（序列号）

4．同时采用用户名和序列号作为自变量

特征值＝F（用户名，序列号）

特征值＝F（用户名1，用户名2，...序列号1，序列号2...）

映射关系越复杂，越不容易破解。

11.4.2 警告（NAG）窗口

Nag窗口是软件设计者用来不断提醒用户购买正式版本的窗口

去除警告窗口最常用的方法是利用资源修改工具来修改程序的资源，将可执行文件中的警告窗口的属性改成透明、不可见，这样就可以变相去除警告窗口了。

若要完全去除警告窗口，只要找到创建此窗口的代码，并跳过该代码的执行。

11.4.3 时间限制

时间限制程序有两类，一类是对每次运行程序的时间进行限制，另一类是每次运行时间不限，但是有时间段限制。

如使程序运行10分钟或20分钟后就停止执行，必须重新启动该程序才能正常工作。

要实现时间限制，应用程序中必须有计时器来统计程序运行的时间，在Windows下使用计时器有SetTimer（）、 TimeSetEvent（）、GetTickCount（）、 TimeGetTime（）。

11.4.4 时间段限制

这类保护的软件一般都有时间段的限制，例如试用30天等。　　安装软件的时候，或在程序第一次运行时获得系统日期，并且将其记录在系统中的某个地方。这个时间称为软件的安装日期。

程序在每次运行的时候首先读取当前系统日期，并将其与记录下来的安装日期进行比较，当其差值超出允许的天数（比如30天）时就停止运行。

为了增加解密难度，软件最少要保存两个时间值：

1．一个就是上面所说的安装时间。

2．另外一个时间值就是软件最近一次使用的日期。

11.4.5 注册保护

注册文件（Key File）是一种利用文件来注册软件的保护方式。Key File内容是一些加密过或未加密的数据，其中可能有用户名、注册码等信息。

当用户向软件作者付费注册之后，会收到注册文件，用户只要将该文件存入到指定的目录中，就可以让软件成为正式版。

为增加破解难度，可以在KeyFile中加入一些垃圾信息；对于注册文件的合法性检查可分散在软件的不同模块中进行判断；对注册文件内的数据处理也尽可能采用复杂的算法。

11.4.6 功能限制

这类程序一般是Demo（演示）版：功能限制的程序一般分为两种：

这类程序一般是Demo（演示）版，功能限制的程序一般分为两种：

1．一种是试用版和正式版的软件完全分开的两个版本，正式版只有向软件作者购买。

2．另一种是试用版和注册版为同一个文件，一旦注册之后就，用户可以使用全部功能。

11.4.7 光盘软件保护

为了能有效地防止光盘盗版，从技术来说要解决三个问题：

（1）要防止光盘之间的拷贝；

（2）要防止破解和跟踪加密光盘；

（3）要防止光盘与硬盘的拷贝。

目前防止光盘盗版技术有：

1. 特征码技术

特征码技术是通过识别光盘上的特征码，如SID（Source Ident-ification Code）来区分是正版光盘还是盗版光盘。

该特征码是在光盘压制生产时自然产生的，而不同的母盘压制出的特征码不一样。光盘上的软件运行时必须先使用该特征码，而这种特征码在盗版者翻制光盘过程中是无法提取和复制的。

2．非正常导入区

光盘的导入区TOC（Track On CD）是用来记录有关于光盘类型等信息，是由光盘自动产生的，但光盘无法复制非正常的导入区。因此，在导入区内添加重要数据以供读盘使用，便能有效地防止光盘之间的非法复制。

3. 非正常扇区

对于一般的应用软件来说，在读取光盘非正常扇区数据的时候，ECC纠错会出现错误，无法读出非正常扇区数据。但可以通过特定的方法在光盘上制造一个特殊的扇区，并在光盘上编写一个程序专门读取该扇区的数据。

如果在非正常扇区当中添加有用的数据，如应用程序的一部分或者是加密、解密的密钥。这样对于盗版者来说，在使用一般软件读该扇区时，会造成数据读出错误。同时，如果把光盘上的数据读到硬盘之后，由于密钥等在正版光盘上，通过硬盘数据来制作盗版光盘时，程序也是无法执行的。

4. 修改文档结构

光盘的文档结构是遵循ISO-9660标准所制定的， 在ISO 9660格式中包括有一种称为Directory Record记录组，记录了文件的或文件夹的名称、属性、长度、生产日期、时间等信息，若是直接修改Directory Record记录组表达的内容，就能骗过Windows等操作系统，制作出隐藏文件夹和超大文件等。

5. 使用光盘保护软件

还可以使用一些商业光盘保护软件，“光盘加密大师”能对光盘多种格式镜像文件（ISO）系统进行可视化修改，将光盘镜像文件中的目录和文件进行特别隐藏，将普通文件变为超大文件，将普通目录变为文件目录等。

（1）隐藏文件

ISO 9660 规定光盘镜像文件的每一个目录和文件都有一定的格式、规定和记录，其中第26个字节记录的就是文件夹标记项。那么光盘文件隐藏的原理就是修改ISO文件的第26个字节的位置。

让光盘产生文件确实存在，但又看不到文件和目录的特殊效果。

（2） 超大文件

超大文件是最对文件进行一些特殊的处理，让本来很小的文件的容量大于2GB（光盘容量只有700MB）。这种文件可以直接运行，但无法直接复制，如果要强制复制就会提示错误。

（3）目录变成文件目录

当目录以文件的形式存在和显示时，是无法直接进入和复制的。

（4）写入光盘密码

光盘加密大师还可以设置光盘密码，这样只有输入正确的密码才能访问光盘的内容和指定的文件等。

11.4.8 软件狗

软件狗（dongles）又称加密锁、加密狗等，是一个可安装在计算机并口、串口或USB接口上的硬件小插件。同时有一套适用于各种语言的接口软件和工具软件。

当被软件狗保护的应用软件运行时，程序向插在计算机上的软件狗发出查询命令，软件狗迅速计算查询并给出响应。如果响应正确，软件将继续运行，否则程序将停止工作。

软件狗技术属于硬加密技术，其中一般都有几十或几百字节的非易失性存储空间可供读写，现在较新的狗内部还包含了单片机。单片机里包含有专用于加密算法软件，该软件写入单片机后就不能再被读出。这样，就保证了软件狗具有硬件不可被复制、加密强度大、可靠性高等特点，软件狗广泛应用于计算机商业软件保护。　　从结构上来说，使用软件狗进行加密的软件分为三个部分：

1．软件狗的驱动程序部分

2．负责与驱动程序进行通讯的具体语言模块

3．客户软件部分。

为了提高软件狗的安全性，现在软件狗采用了一些防破译技术。如：

1．随机噪声技术是针对监视通信口工具设计的。如果试图截听通信口与软件狗的交互数据流，将会发现那里面夹杂了大量的无用随机数据，让解密者难辨真假。而应用软件和狗之间却可以按照通讯协议正常通话。

也可以采用不对称加密算法，解决通讯监听破解的难题，传统的对称算法加密，黑客只要从内存中获得其加密密钥，就可以破解整个通讯过程。

2．时间闸技术是监视程序的运行时间。如果有人想把程序停下来进行分析，软件将被时间闸切断运行或者自毁应用程序，使破解者负出沉重代价。

3．迷宫技术是在程序的入口和出口间插入了大量的跳转来迷惑破解者，使他们很难分析出程序逻辑。

4．将应用程序的一部分写到软件狗中，如果不使用软件狗，应用程序是不完整的，也就无法执行了。

软件狗破解：

（1）仿制硬件电路

（2）修改主程序。跳过访问软件狗或符合读出软件狗信息。

11.4.9 软盘保护技术

软盘保护技术的原理是用特殊的方法在软盘上建立非正常的区间，并将一些重要的信息，如密钥、加密程序存放在该区间内。软件在运行时先检验这些信息，当检验正确时软件才能使用，这种软盘就好像一把钥匙，通常称这种软盘为“钥匙盘”。制作“钥匙盘”的原理如下：

软磁盘有若干个同心圆，每个圆称为一个磁道。每个磁道分为若干个扇区。每个扇区有间隙（GAP）。

对于标准的3.5英寸软盘来说，有80个磁道，每磁道有18个扇区，每扇区可以存放512字节的数据，操作系统也只能读写标准格式化磁盘。如果采取一些特殊的手段破坏软盘的标准结构和读写方法，如改变扇区编号、扇区个数、扇区大小、磁道数、磁道接头数、磁道间隙指纹等，在软盘上制作出“非正常” 的区间。这样，用正常的拷贝命令、读写命令、删除命令是无法影响软盘非正常区间上的数据。

11.4.10 反跟踪技术

反跟踪技术是防止破解者通过直接“跟踪”软件的执行过程，如动态调试、静态反汇编等，来获取重要信息和加密方法。一个加密软件的安全性好坏很大程度上取决于软件的反跟踪能力。

下面介绍一些反跟踪技术的基本方法：

1．在应用程序启动时，先判断内存中是否有调试程序，若发现有调试程序存在，程序将拒绝运行等。

2．对重要的程序段应是不可修改的。

3．综合多种软件加密方法，交叉使用不同的加密技术。

4．设置跟踪障碍，提高破解难度。

5．一旦发现跟踪行为，可以采取自毁行为，这将大大增加破解者的成本。

6．当应用程序执行到重要程序段之前，可以采用封锁键盘输入，封锁显示器和打印机输出。待重要程序段任务完成后再解除键盘、显示器和打印机封锁。

7．为防止破解者通过修改堆栈指针的值来达到跟踪目的，可将堆栈指针设在特定的区域，使堆栈指针指向无意义的操作。

8．加密程序最好以分段的密文形式装入内存，执行完一段程序后，再解密和执行下一段程序，同时在内存中删除上一段程序。

11.4.11 网络软件保护

（1）在线方式的软件保护原理

传统的软件保护产品主要通过应用程序与本地计算机上的加密锁或许可证文件进行验证的方式，这种方式的缺点是需要安装客户端硬件、驱动或者是本地许可证，而且在软件开发商与应用程序之间没有联系，因此后期的许可升级比较麻烦。 此外，无论是客户端的加密锁硬件还是许可证文件，都较容易被破解者分析、破解。对于加密锁来说，目前硬件复制的破解方式也非常多。

随着互联网的发展，一种新的、基于互联网服务器认证的软件保护方式油然兴起。这种在线方式 (On-line Protection) 的软件保护是以新兴的互联网技术为基础，以互联网服务器来替代传统的加密锁硬件。

应用程序通过开发商发放的授权码与网络认证服务器建立连接，并调用服务器上的Web Service接口完成软件保护工作。因为这种客户端/服务器模式之前采用了高强度的类似TSL的通讯加密技术，而且服务器远离软件用户，因此它的软件保护强度可以非常高。除此之外，在线方式的软件保护不需要在客户端安装众多的模块，因此安装、部署、维护都非常简单。最重要的是，高性能的服务器可以提供除了软件保护之外的其它众多接口，如数据存储、远程通讯等，其应用范围已经大大超出了传统软件保护的范围。

在线方式的客户端通过与服务器的实时连接，为开发商收集软件用户使用状态、统计软件使用情况、快速升级与服务提供了可能。

（2）网络加密产品的原理和使用

首先要在网络上启动一个网络加密狗（或加密卡）的加密服务程序，将使得网络上所有合法用户可以访问到网络狗。当用户在客户机端运行加密后的软件时，客户机会向网络中寻找提供加密服务的网络狗。当网络狗存在并且返回正确检测信息后，用户被认为是合法的，网络软件就可以正常使用了。

网络加密产品适应能力可以表现为：

1. 支持多操作系统。
2. 支持多协议。
3. 支持复杂网络。
4. 支持多进程。

11.4. 12 补丁技术

补丁技术主要是针对已发布的软件漏洞和软件功能更新所采取的软件更新技术，也是一种软件保护方式。补丁技术主要有文件补丁和内存补丁两种。

1．文件补丁

文件补丁就是直接修改文件本身某些数据或代码，主要针对没有被加密、加壳和CRC校验的目标程序。

2．内存补丁

内存补丁主要针对被加密、加壳、CRC校验的程序，内存补丁的总体思想就是在目标程序程序解密、解压、校验等情况发生以后，在目标程序的地址空间中修改数据。

11.5 软件加壳与脱壳

11.5.1 “壳”的概念

“加壳”，就是用专门的工具或方法，在应用程序中加入一段如同保护层的代码，使原程序代码失去本来的面目，从而防止程序被非法修改和编译。

用户在执行被加壳的程序时，实际上是先执行“外壳”程序，而由这个“外壳”程序负责把原程序在内存中解开，并把控制权交还给解开后原程序。

加壳软件按照其加壳目的和作用，可分为两类：一是保护，二是压缩。

1．保护程序

这是给程序加壳的主要目的，就是通过给程序加上保护层的代码，使原程序代码失去本来的面目。它的主要目的在于反跟踪，保护代码和数据，保护程序数据的完整性，防止程序被调试、脱壳等。

2．压缩程序

这应该是加壳程序的附加功能。压缩后原程序文件代码也失去了本来的面目，可以保护软件。

注意用WinZip、WinRAR等文件压缩文件，一般是不可执行的。而这里对exe压缩程序是可执行的。

壳的一般加载过程是：

1. 获取壳自己所需要的API地址
2. 加密原程序的各个区块的数据
3. 重定位
4. HOOK-API
5. 跳转到程序原入口点

11.5.2 软件加壳工具介绍

现在用于压缩程序为主要目的的常见加壳软件有ASPacK、UPX和PECompact等，用于保护程序为主要目的的常见加壳软件有AsProtect、tElock和幻影等。下面简单介绍一些常用的加壳软件。

1．ASPack

ASPack是一款Win32高效保护性的压缩软件，文件压缩比率高达40%～70 % 。ASPack无内置解压缩，不能自解压自己压缩过的程序。

2．ASProtect

ASProtect具有压缩、加密、反跟踪代码、反-反汇编代码、CRC校验和花指令等保护措施。它使用Blowfish 等高强度的加密算法，还用RSA 1024作为注册密钥生成器。它还通过API钩子与加壳的程序进行通信。

3．幻影（DBPE）

幻影具体功能有：

（1）动态生成加密密码，对程序的代码、数据进行加密。

（2）压缩程序数据、代码。减少占用空间。

（3）对抗所有的反编译工具。

（4）程序有完整性校验，防止修改。

（5）对抗所有已知的内存还原工具。如:ProcDump，PEditor等。

（6）对抗所有已知的跟踪分析工具。如:SoftICE，Trw2000，OllyDbg等。

（7）可为软件加上运行次数限制，运行天数限制，运行有效日期限制，需要注册才能解除限制。

（8）根据每台不同电脑算出不同注册码，注册码只能在本机有效。

（9）提供接口函数，可让程序查询注册状态。

11.5.3 软件脱壳

对一个加了壳的程序，就要去除其中的无关干扰信息和保护限制，把它的壳脱去，解除伪装，还原软件的本来面目，这一过程就称为脱壳。

常用的侦壳软件有Language 2000、File Scanner、FileInfo、PEiDentifer等。

对软件进行脱壳时，可以使用脱壳软件，也可手动脱壳。手动脱壳前，需要熟悉Win32下的可执行文件标准格式，可以使用一些辅助工具，如冲击波、W32Dasm等。手动脱壳主要步骤有：查找程序入口点，获取内存映像文件，重建输入表等。

脱壳软件主要分为两大类，即专用脱壳软件和通用脱壳软件。专用的脱壳软件适用面窄，但对付特定的壳却极为有效。通用的脱壳软件往往不能精确地适用于某些软件。

脱壳成功的标志是脱壳后的文件能正常运行，并且功能没有减少。一般来说，脱壳后的文件长度大于原文件长度。即使同一个文件，当采用不同脱壳软件进行脱壳的时候，由于脱壳软件机理不同，脱出来的文件大小也不尽相同。

11.6 设计软件的一般性建议

1. 软件发行之前一定要将可执行程序进行加壳。
2. 要在自己写的软件中嵌入反跟踪的代码。
3. 增加对软件自身的完整性检查。
4. 不要采用一目了然的名字来命名与软件保护相关的函数和文件。
5. 当检测到软件破解企图之后，过一段时间后使软件停止工作。
6. 可以通过读取关键的系统文件的修改时间来得到系统时间的信息。

7．给软件保护加入一定的随机性。

8．如果试用版与正式版是分开的两个版本，而是彻底删除相关的代码。

9．如果软件中包含驱动程序，则最好将保护判断加在驱动程序中。

10．将注册码、安装时间记录在多个不同的地方。

11．采用一机一码，可以防止注册码传播。

12．最好是采用成熟的密码学算法。

13．可以采用在线注册的方法。

14．keyfile的尺寸不能太小。

本章教学要求：

（1）掌握文件静态分析和动态分析概念；

（2）知道静态分析软件W32Dasm、IDA Pro功能；

（3）知道可执行文件代码编辑工具和可执行文件资源编辑工具功能；

（4）知道动态分析软件SoftICE、OllyDbg调试器功能；

（5）熟悉常用软件保护技术；

（6）知道加壳与脱壳技术；

（7）了解使用加壳与脱壳工具；

（8）知道软件保护的一般性建议。