深入剖析

加密解密

武新华　周义德　编著

专业讲解　尽显加密解密全程

技巧放送　出神入化触类旁通

技术专题　深入探讨传授真经

举一反三　拓展思路指点迷津

西安电子科技大学出版社

http://www.xduph.com

7

深入剖析加密解密

武新华　周义德　编著

西安电子科技大学出版社

2004

内 容 简 介

本书介绍软件的加密与解密技术，在详细讲述加密解密技术的同时，还介绍了相应的实现原理，以使读者能够对加密解密技术有更深入的了解，从而更好地提高自己的编程水平。全书共分为 10 章，包括反汇编调试的静态与动态分析、暴力破解共享软件、补丁技术及其工具、加壳/脱壳技术和反编译语言程序等内容。

本书内容丰富，图文并茂，叙述深入浅出，适用于对加密解密技术有兴趣的读者；同时可作为加密解密课程教材或参考书，也适用于软件开发从业人员及编程爱好者参考。

图书在版编目（CIP）数据

深入剖析加密解密 / 武新华，周义德编著.

—西安：西安电子科技大学出版社，2004.7

ISBN 7-5606-1392-6

Ⅰ. 深… Ⅱ. ①武… ②周… Ⅲ. ①软件-密码-加密 ②软件-密码-解密译码 Ⅳ. TP311.56

中国版本图书馆 CIP 数据核字（2004）第 037418 号

策　　划　李惠萍

责任编辑　杨宗周

出版发行　西安电子科技大学出版社(西安市太白南路 2 号)

电　　话　(029)88242885　88201467　　邮　编　710071

http: //www.xduph.com　　E-mail: xdupfxb@pub.xaonline.com

经　　销　新华书店

印刷单位　西安文化彩印厂

版　　次　2004 年 7 月第 1 版　2004 年 7 月第 1 次印刷

开　　本　787 毫米×1092 毫米　1/16　　印张　20.75

字　　数　491 千字

印　　数　1～6000 册

定　　价　28.00 元

ISBN 7-5606-1392-6/TP・0740

XDUP 1663001-1

前　言

网络的发展无疑极大地推动了共享软件的发展，但共享软件的蓬勃发展也无疑造就了另外一个行业的发展，那就是加密解密。尤其是人们知识产权意识的不断加强和有关方面法律法规的不断完善，更多学习计算机的人都对加密解密技术有浓厚的兴趣，并要求学习、掌握加密解密技术，因此，这也是我们编写本书的初衷。

我们的目的不是要教会大家如何破解别人的软件，更反对将这一技术用于非法目的，而是要大家学会如何通过跟踪软件了解别人的编程思路，从容分析它的程序原理，从而写出自己更好的程序。

为照顾初学者，本书经过精心编排，尽力做到以图例讲解代替大段大段枯燥的代码说教，使得各个层面的读者，甚至是那些从未接触过汇编和没有多少编程基础的读者也能够在阅读完本书后轻松入门。

而对于电脑加密解密高手来说，阅读本书的意义却另有不同。不断地探索未知，是每一个“高手”心目中永远的诱惑。作为一个加密解密爱好者，则注定了自己与未知的对抗，而不能掌握一些必要的软件加密解密技术，将是自己心中永远的痛。

为方便广大读者的阅读，本书完全采用图例步骤式的讲解方法，使得图文能够紧密结合，尽可能地减少了长篇累牍的枯燥代码，理论讲解深入浅出，同时强调应用技能的快速掌握，使得本书简单易读。

本书凝聚众多网际高手的加密解密经验与技巧，并在对其加密解密思想或操作步骤进行重新整理的基础上，进行了进一步的深化和剖析，从而使得这些高深的加密解密技术不再遥不可及。

我们相信，有这样一本书置于你的案头，那许许多多在你过去看来难于登天的事情，你会突然发现却原来如此简单。

本书由众多经验丰富的高校教师编写，并得到了众多网友的支持，在此一并表示衷心的感谢。本书的编写情况是：樊瑞负责第 1 章，张蕾负责第 2 章，齐伟负责第 3 章，周义德负责第 4 章，武新华负责第 5、6、7 章，安向东负责第 8 章，张海峰负责第 9、10 章；武新华、周义德统稿。

我们虽满腔热情，但限于自己的水平，书中的疏漏之处在所难免，欢迎广大读者批评指正。

最后，需要提醒大家的是：

根据国家有关法律规定，任何破解他人软件程序用于商业目的的行为都属于违法行为，希望读者在阅读本书后最好不要使用本书中介绍的破解技术对某些软件进行盗版制作，否则后果自负。

编　者

2004 年 6 月

目　录

第 1 章 加密 VS 解密——大碰撞

- 认识加密技术
- 加密 VS 解密：矛与盾的关系
- 常见软件加密保护技术
- 熟悉汇编语言的几条常用命令

在本章中，我们来了解一些有关加密解密的基础知识，如什么是加密技术、常见软件的加密保护等，以便我们对本书后面要讲解的内容能够有一个全面的认识。

1.1 认识加密技术

我国最新修订的《著作权法》和《计算机软件保护条例》中均增加了有关软件加密的内容。加密不但在保护软件知识产权方面起到重要作用，而且在保证软件及信息技术完整性、安全性方面也同样具有不可替代的作用，数据加密已成为当今信息安全技术的实力象征。

1.1.1 数据加密的基本概念

数据加密是一种限制对传输数据的访问权的技术。原始数据(也称为明文，plaintext)被加密设备(硬件或软件)和密钥加密而产生的经过编码的数据称为密文(ciphertext)。

将密文还原为原始明文的过程称为解密，它是数据加密的反向处理，但解密者必须利用相同类型的加密设备和密钥对密文进行解密。

数据加密的基本功能包括：

(1) 防止不速之客查看机密的数据文件；

(2) 防止机密数据被泄露或篡改；

(3) 防止特权用户(如系统管理员)查看私人数据文件；

(4) 使入侵者不能轻易地查找一个系统的文件。

个人用户最常见的加密就是在上网输入密码时，文本框中显示出的是星号(*)，这就是一种简单的数据加密，不管你的密码是什么，这种加密技术都会将它显示为星号。

本书中所讲述的加密与解密，只是简单介绍一些加密解密的基本原理和相关的软件使用方法，而没有妄自深论加密与解密技术，因为我们的目的只是想让大家了解那些比较流行的加密和解密技术的实现机理，起到一个软件加密与解密的启蒙作用。

众所周知，计算机软件极易被复制。一套商业软件从构思、编程、调试到完成，软件

开发者付出了很多心血，如果轻易被他人盗版，损失将是巨大的。所以软件商为了维护自己的利益，一般都采取了各种保护手段来防止非法用户盗用自己的软件。软件狗便是作为一种插在计算机并行口上的软件加密产品。软件开发者可以通过接口函数和软件狗进行数据交换(即对软件狗进行读写)，来检查软件狗是否插在并行口上；或者直接用软件狗附带的工具加密自己的 EXE 文件(俗称“包壳”)。

这样，软件开发者可以在软件中设置多处软件锁，利用软件狗做为钥匙来打开这些锁；如果没有插软件狗或软件狗不对应，软件将不能正常执行。常见的加密狗有：微狗、USB 狗、光盘狗等。

首先我们要懂得一个原则，进行软件解密在某种程度上并不仅仅是为了破坏软件，任何破解都只能是用于个人，而绝不能用于网络传播或转予他人，否则是违法的。

在下面这些场合人们往往需要进行解密：

(1) 磁盘加密技术。绝大多数的教学盘上都用磁盘加密技术进行保护正版软件的合法使用性，但这样做的最大弊端就是磁盘的反复读取会大大地缩短磁盘的使用寿命，如果我们通过使用一些解密技巧，便可以不再担心了。

(2) 密码破解。很多游戏软件都需要密码才能进入，每次翻阅说明书夹缝中的密码表是不是很烦？既然是花了钱的正版用户，为什么还要这样麻烦，而且一旦说明书丢了，那又如何办呢？因此，最简单的方式就是破解。

(3) 硬件保护。诸多的不良接触是不是令人头痛？如果我们的机箱经常使你有“触电”的感觉，你是不是恨不得扔了这个硬件保护的家伙。

(4) 最有效的学习方式。有时候为了学习软件编程高手们的加密算法应用技巧，或是为了试验一个破解工具的实用程度，我们通常会对一些常见软件进行破解，从中学习国外的软件加密算法和技巧。

(5) 遗忘口令。口令在增加安全性的同时，也增加了遗忘口令可能带来的不必要的麻烦，破解口令在这个时候就显得十分重要。

1.1.2 加密技术及其相关问题

1. 加密的分类

加密类型可以简单地分为 3 种：

(1) 根本不考虑解密问题。这种加密技术主要是针对一些像口令加密这样的类型，它只需要被加密，并与以前的加密进行比较。

(2) 私用密钥加密技术。私用密钥加密技术是利用一个密钥对数据进行加密，它需要对方在接收到数据后，采用同一密钥来进行解密。这种加密技术的特点是数学运算量小，加密速度快。其主要弱点在于密钥管理困难，而且一旦密钥泄露则直接影响到数据信息的安全性。

(3) 公开密钥加密技术。1976 年，Diffie 和 Hellman 首次提出公开密钥加密体制，即每个人都有一对密钥，其中一个为公开的，一个为私有的。发送信息时用对方的公开密钥加密，收信者用自己的私用密钥进行解密。

公开密钥算法有很多，一些算法如著名的背包算法和 McELiece 算法都被破译。目前公认比较安全的公开密钥算法主要就是 RSA 算法及其变种 Rabin 算法、离散对数算法等。在这方面，我国学者陶仁骥、陈世华提出一种基于有限自动机的公开密钥加密方法：FAPKC0、FAPKC1、FAPKC2、FAPKC3。这是国际上的第一个时序公开密钥算法，其安全性是建立在非线性有限自动机求逆的困难上的。因为从数学上，线性有限自动机已有完美的可逆性理论，但迄今为止未解决非线性有限自动机可逆性理论，它可用于保密通信、数字签名，易于实现，密钥量适中，加解密速度快，受到国际上的广泛关注。

2. 加密技术发展趋势

(1) 私用密钥加密技术与公开密钥加密技术相结合。鉴于两种密码体制加密的特点，在实际应用中可以采用折衷方案，即结合使用 DES/IDEA 和 RSA，以 DES 为“内核”，RSA 为“外壳”，对于网络中传输的数据可用 DES 或 IDEA 加密，而加密用的密钥则用 RSA 加密传送，此种方法既保证了数据安全又提高了加密和解密的速度，这也是目前加密技术发展的新方向之一。

(2) 寻求新算法。跳出以常见的迭代为基础的构造思路，脱离基于某些数学问题复杂性的构造方法。如刘尊全先生提出的刘氏算法，是一种基于密钥的公开密钥体制，它采用了随机性原理构造加解密变换，并将其全部运算控制隐匿于密钥中，密钥长度可变。它是采用选取一定长度的分割来构造大的搜索空间，从而实现一次非线性变换。此种加密算法加密强度高、速度快、计算开销低。

(3) 加密最终将被集成到系统和网络中，例如 IPV6 协议就已有了内置加密的支持，在硬件方面，Intel 公司正研制一种加密协处理器，它可以集成到微机的主机上。

3. 密钥管理

对于私用密钥加密和公开密钥加密系统来讲，并不强调对加密解密算法的保密。计算机网络加密的安全性主要是依赖于算法本身的安全性和对密钥的保护。密钥主要有会话密钥(Session Key)、基本密钥(Basic Key)和主密钥(Master Key)三种。

会话密钥是通信双方在会话中使用的密钥，此种密钥只在一次会话中有效，会话结束时密钥就失效；在网络中用来传送会话密钥的密钥，就是基本密钥；而对基本密钥进行加密的密钥则称为主密钥。网络中一般是采用这种三级密钥方案来进行保密通信的。

从技术上看，密钥管理包括密钥的产生、存储、分配、使用和销毁等一系列技术问题，密钥分配是其中最重要的问题。

网络密钥管理主要有 KDC(Key Distribution Center)和 Diffie-Hellman 两种方法。KDC 使用第三方来验证通信双方的真实性，产生会话密钥，并通过数字签名等手段分配自己的密钥。Diffie-Hellman 方法则不需要 KDC，通信发起方产生通信会话的私用密钥，并通过数字签名或零知识证明等方式安全传递通信密钥。

IETF 现在正在开发一种基于 Diffie-Hellman 技术的密钥交换协议 Okaley，从而具有更好的安全性和独立性。目前多密钥分配是有待深入研究的领域。

随着网络的应用与发展，网络安全问题日益突出。数据加密是确保计算机网络安全的一种重要的机制，虽然由于成本、技术和管理上的复杂性，目前尚未在网络中普及，但数据加密作为实现网络环境下数据安全的重要手段之一，必将得到广泛应用。设计优秀的现

代加密技术还融入了防篡改和防否认等数字签名技术，因此其不但适用于企业用户，而且也适用于其他用户。

1.2　加密 VS 解密：矛与盾的关系

为什么要解密呢？这个问题不太好回答，因为解密的缘由很多，不同的人有不同的动机。可能只是因为对解密技术感兴趣；可能就是因为想要无偿的、免费使用软件；可能是因为需要用某个软件，虽然愿意付钱，但是很难办到；可能是为了盗版，获取非法利益……总之，你有你的理由，我有我的根据，但是大家都想到了解密。

早期的软件加密的操作系统平台主要是 DOS，在那个时代，人们可以很容易写个程序控制整个计算机。因为 DOS 比较简单，安全性较差，用户可以进行各种低层操作，例如调用各种中断、读写内存区域、操作 I/O 端口等。总之，利用 DOS 本身提供的环境，几乎可以控制任何想要控制的东西，甚至控制 DOS 本身也不是难题。

那个时候的加密技术主要有磁盘加密(包括软盘和硬盘)、文件和目录的加密、各种硬件(加密卡、加密狗等)加密等。当时最流行的莫过于软盘的加密技术了，包括特殊磁道(如额外磁道、螺旋磁道、无缝磁道等)，特殊扇段(超长扇段、乱序扇段、异常 ID 等)，特殊方法(如弱位技术、FM 格式法、磁道噪声法、针孔加密技术等)。那时经常会见到很多软件都需要钥匙盘，那些钥匙盘用的就是各种各样的软磁盘加密技术。为了防止解密，还运用了多种反跟踪技术，比如修改中断向量、锁键盘、关闭视屏、覆盖技术、废指令、逆指令流、指令队列预取法等。

总之，DOS 时代你可以充分运用想象力去构造各种各样新奇的加密技术，因为 DOS 给了你很多自由发挥的空间。

到了 Windows 时代，你会觉得一切都变了，自己不再是主人，而只能在限定的范围内活动。Windows 把低层的东西都隐藏了起来，人们只能看到各种各样的壳，至于里面的东西，只有 Windows 自己才能动。

Windows 把程序分为三六九等，普通的应用程序是工作在 Ring 3(3 级)特权级，而系统关键的程序和数据工作在 Ring0(0 级)特权级下，也就是最高特权级。因为不能再像以前一样按照自己的意愿进行低层操作，所以 DOS 下的加密技术到了 Windows 下基本上都是英雄无用武之地。不同的环境有不同的方法、对策，虽然不能应用 DOS 下那些老办法来进行加密，但是还可以通过其他的方法来实现，各种技术总是会随着时间逐渐进步的，这是历史发展的方向。

1.3　常见软件加密保护技术

在这里并不是要系统讨论软件加密保护技术，只是简单介绍一些和破解相关的软件保护方式，目的是让大家搞清楚哪些是我们解密的对象。有一点是要首先明确的，并不是所有受限制的软件都可以破解的。因为作为破解一方来说，只能是将软件中不允许、受限制的功能变为可用的、没有限制的。

有两种软件可以免费使用：

(1) 一种是自由软件(Freeware)，完全免费的，没有任何使用限制。

(2) 一种是共享软件(Shareware)，可以免费试用。但如果要得到完全的功能或者服务的话，一般情况下需要注册。

现在网上有很多的免费软件，其中共享软件相对来说更多一些。当然大家应当明白一点，真正免费的午餐还是不多的。

1.3.1　常见软件加密保护技术

1. 软件注册

有些共享软件没有任何的限制，只是有可能会提示用户该注册了，如果不注册的话，并不影响软件的正常使用，不会出现功能限制，但是注册之后会有更多的支持与服务。更多的共享软件都有诸多的限制，比如使用天数限制、有效日期限制、次数限制、功能限制、未注册画面、延迟或干脆禁用软件等。

软件的注册方式具有以下一些形式：

(1) 用户 ID 或注册码；

(2) 序列号；

(3) 用户名(或用户 ID)+注册码(或序列号)；

(4) 用户名+单位名+序列号(或注册码)。

以上这几种方式都是将用户输入的信息经过一些特殊的运算，然后和正确的注册信息相比较，如果相同则注册成功。

需要说明的是，正确的注册信息并不一定是显式存在的，也就是说程序根据输入的信息计算出一个结果后，有可能直接和正确的注册信息相比较，也有可能是间接比较的。

通常对于那些直接存在于内存中的注册码，比较容易破解，但是对于那些隐式存在于内存中的注册码，则需要仔细去跟踪、分析程序，才能得到正确的注册码，对付这类软件通常需要更多的精力和耐心。这里所指的用户 ID 可能是用户名，也可能是一串软件所带的数字，也有可能就是注册码，通常根据不同的软件来定。序列号通常具有以下的形式：XXXXX-XXXXX-XXXXX-XXXXX(这里是笔者随便输入的)。

(5) 有些软件注册时会在你的电脑里搜集一些信息，让你 E-mail(或邮寄)给软件公司并缴费，然后对方提供给你一个注册码。

对于这类软件，破解时通常要彻底搞清楚程序的算法，然后再给出一个破解方案。因为在自己的电脑上简单得到一个注册码，在其他电脑上并不能用，没有什么意义。

(6) 还有一些软件注册时是通过网上或邮寄方式付款的，然后对方会提供另外一个注册过的软件，对于这类共享软件大家是没有办法破解的，因为其共享软件本身就缺乏某些功能，要得到功能完全的正式版，惟一的办法就是通过正常的渠道去注册。

(7) 序列密码保护。很多共享软件为了保护自己的权益或是为了吸引人们去购买它们，都采取了序列密码的软件保护方式。软件注册后才能将全部的软件功能提供给注册用户使用。有些软件公司为了防止软件被非法解密，甚至会采取搜集用户计算机硬件信息的方式来通过网络注册，并返回相应的注册码。

例如微软公司出品的 Windows XP 所采用的激活技术就是一个很典型的例子。当然也有

一些软件只是为了想知道自己的软件究竟到了哪些地方，有哪些人在使用而已。

2. *密码保护*

这一类也是大家常见的加密保护方式，凡是需要用户输入密码(PASSWORD)的地方都属于这类，有应用软件密码、游戏密码、文件密码等(当然能破解的目标只是其中的一部分了)。当然了，这也是最容易招致黑客破解的一种软件加密方式，凡是需要用户输入相应密码的软件都是属于这种类型。

3. Keyfile *加密方式*

严格来说，这不能算是软件加密，因为这类保护方式不需要用户输入注册码来通过程序验证是否合法，而是检查默认的文件的有效性。这个文件可能是程序中任一段程序代码，或是某一普通文件，也有可能是软件自己定义的特殊格式文件。当需要验证的部分被更改或破坏、丢失的时候，这个软件就默认为被非法修改，从而拒绝用户使用。

4. *磁盘保护*

一般是利用软磁盘做成钥匙盘，然后软件运行的时候会去校验软盘中的数据是否正确，例如大家都很熟悉的杀毒软件 KV300、KV3000 就是如此。其实说白了也就是“钥匙盘”的加密方式是一种很有效的且成本相对较低的加密方式。在软件加密保护技术中，“锁”和“钥匙”是一种基本加密设计思想。在软件中预先设置一个或多个“锁”，然后给予合法用户开启这些“锁”的“钥匙”，于是软件在验证“钥匙”正确后，可以正常运行。

最典型的例子应该是江民公司的 KV 杀毒软件系列了，使用过 KV 系列杀毒软件的用户都知道，其全部程序装载在一张 3.5 英寸软盘上，我们可以把这张盘复制到另外一张 3.5 英寸软盘上，但复制盘却不能运行 KV 杀毒软件。因为拷贝程序不能把原盘上的那些校验信息拷走，软件厂商通过钥匙盘的方式减少了非法使用。

5. *逻辑炸弹*

这是一种最无奈的软件保护方式，一个好的软件总会有黑客们去尝试破解它。逻辑炸弹就是软件设计者们为了防止软件被破解而采用的一种保护方式，当破解程序反复尝试跟踪某个程序段时，逻辑炸弹就会自动引爆，使破解工作被迫停止，严重的会使电脑停机，或自动格式化破解者的硬盘。

6. *软件加壳*

这种方式很常见，也很有效，但也是黑客最喜欢进行挑战的软件加密方式。软件加壳是指利用专门的工具使应用程序失去原有的状态，但功能无损。如果冒然用反汇编工具去进行反汇编的话，那么加壳技术就会使破解者什么也看不见。

7. *加密狗*

所谓加密狗，就是插在计算机接口上的一个数字电路，里面存有若干数据，软件通过计算机接口对这部分电路进行操作(读取，修改等)，只有正确的狗才能使软件正常运行，从而达到保护软件、防止盗版的目的。

8. *网卡加密*

这种方式主要是利用网卡的序列号来进行识别，只有随机带的网卡才能正常使用软件。

9. License 保护

这种加密多用于大型的商业软件，软件通过正确的 License 文件运行。License 文件都有固定的格式(这里所说的不是有关 License 的说明性文本文件)，一般是一个 License 对应软件的一个功能模块。

如果想要增加软件功能，只要购买相应的模块，得到一个 License，然后就可以使用了。不同的 License 许可的用户数量不同，有单用户 License，多用户 License，有单机版 License，网络版 License 等。

10. 软件压缩

软件压缩就是通常说的加壳，利用专门的压缩软件将应用程序进行压缩，程序文件失去了本来的面目。如果你用反汇编工具反汇编，那么你是什么也看不到的，因为软件本身已经被压缩，并不是真实的可执行文件代码了。

11. 光盘加密(CD KEY)

这类保护多用于游戏中，程序运行时要求将原版 CD 碟放在光驱中，然后输入光盘附带的 CD KEY，或者是程序直接检查光盘上的特殊数据(指纹等)，由此来判断使用的是否是正版光碟。

12. 其他

还有一些并不属于加密保护之列，但也常常是破解的对象。比如游戏中的生命值、经验值、法力值、钱和物品的数量等(有很多专用的游戏修改器可以达到这些目的)。

1.3.2　加密解密的相关概念

在本节中，介绍一下在学习解密过程中经常遇到的问题。

1. 断点

所谓断点就是程序被中断的地方，这个词对于解密者来说是再熟悉不过了。那么什么又是中断呢？中断就是由于有特殊事件(中断事件)发生，计算机暂停当前的任务(即程序)，转而去执行另外的任务(中断服务程序)，然后再返回原先的任务继续执行。

解密的过程就是等到程序去获取我们输入的注册码并准备和正确的注册码相比较的时候将它中断下来，然后通过分析程序，找到正确的注册码。

所以需要为被解密的程序设置断点，在适当的时候切入程序内部，追踪到程序的注册码，从而达到 Crack(破解)的目的。

2. 领空

这是个非常重要的概念，但也是初学者常常不明白的地方。大家在各种各样的破解文章里都能看到领空这个词，如果你搞不清楚到底程序的领空在哪里，那么你就不可能进入破解的大门。或许你也曾破解过某些软件，但那只是瞎猫碰到死耗子而已。

所谓程序的领空，说白了就是程序自己的地方，也就是我们要破解的程序自己程序码所处的位置。

我是在程序运行的时候设置的断点，为什么中断后不是在程序自己的空间呢？

因为每个程序的编写都没有固定的模式，所以我们要在想要切入程序的时候中断程序，就必须不依赖具体的程序设置断点，也就是我们设置的断点应该是每个程序都会用到的东西。

在 DOS 时代，基本上所有的程序都是工作在中断程序之上的，即几乎所有的 DOS 程序都会去调用各种中断来完成任务。但是到了 Windows 时代，程序没有权力直接调用中断，Windows 系统提供了一个系统功能调用平台(API)，就像 DOS 程序以中断程序为基础一样，Windows 程序以 API 为基础来和系统打交道，实现各种功能，所以 Windows 下的软件破解的断点设置是以 API 函数为基础的，即当程序调用某个 API 函数时中断其正常运行，然后进行解密。

例如，在 SoftICE 中设置断点 bPxGetDlgItemText(获取对话框文本)，当需要破解的程序要读取输入数据而调用 GetDlgItemText 时，立即被 SoftICE 拦截到，从而被破解的程序停留在 GetDlgItemText 的程序区，而 GetDlgItemText 则处于 Windows 自己管理的系统区域，如果擅自改掉这部分的程序代码，那就大祸临头了。

所以大家要从系统区域返回到被破解程序自己的地方(即程序的领空)，才能对程序进行破解。

3. API

API 是 Application Programming Interface 的简写，中文叫应用程序编程接口，是一个系统定义函数的大集合，它提供了访问操作系统特征的方法。

API 包含了几百个应用程序调用的函数，这些函数执行所有必须的与操作系统相关的操作，如内存分配，向屏幕输出和创建窗口等，用户程序通过调用 API 接口同 Windows 打交道，无论什么样的应用程序，其底层最终都是通过调用各种 API 函数来实现各种功能的。

API 通常有两种基本形式：Win 16 和 Win 32。Win 16 是原来的、API 的 16 位版本，用于 Windows 3.1；Win 32 是现在的、API 的 32 位版本，用于 Windows 9X/NT/ME/2000。Win 32 包括了 Win 16，是 Win 16 的超集，大多数函数的名字、用法都是相同的。16 位的 API 函数和 32 位的 API 函数的区别在于最后的一个字母。

例如设置这样的断点：bpxGetDlgItemText、bpxGetDlgItemTextA 和 bpxGetDlgItemTextW，其中 GetDlgItemText 是 16 位 API 函数，GetDlgItemTextA 和 GetDlgItemTextW 是 32 位 API 函数，而 GetDlgItemTextA 表示函数使用单字节，GetDlgItemTextW 表示函数使用双字节。

现在破解中常用到的是 Win 32 单字节 API 函数，就是和 GetDlgItemText A 类似的函数，其他的两种(Win 16API 和 Win 32 双字节 API 函数)则比较少见。

Win32 API 函数包含在动态链接库(DynamicLinkLibraries，简称 DLLs)中，即包含在 kernel32.dll、user32.dll、gdi32.dll 和 comctl32.dll 中，这就是为什么要在 SoftICE 中用 exp=C:\windows\system~kemel32.dll 等命令行将这些动态链接库导入&softice 中的原因。因为不这样做的话，我们就无法拦截到系统 Win32API 函数调用了。

4. 关于程序中注册码的存在方式

破解过程中大家都会去找程序中将输入的注册码和正确的注册码相比较的地方，然后通过对程序的跟踪、分析找到正确的注册码。但是正确的注册码通常在程序中以两种形态存在：显式的和隐式的。

对于显式存在的注册码，可以直接在程序所处的内存中看到它。例如我们可以直接在 SoftICE 的数据窗口中看到类似“385766523”这样存在的注册码(这里是随意写的)，对于注册码显式存在的软件破解起来比较容易。但是有些软件的程序中并不会直接将我们输入的注册码和正确的注册码进行比较，比如有可能将注册码换算成整数、或是将注册码拆开，然后将每一位注册码分开在不同的地方逐一进行比较，或者是将我们输入的注册码进行某种变换，再用某个特殊的程序进行验证等。

总之，应用程序会采取各种不同的复杂运算方式来回避直接的注册码比较，对于这类程序，我们通常要下功夫去仔细跟踪、分析每个程序功能，找到加密算法，然后才能破解它，当然这需要一定的 8086 汇编编程功底和很大的耐心与精力。

5. 关于软件的破解方式

在此将破解方式分为两大类，即完全破解和暴力破解。

所谓完全破解主要是针对那些需要输入注册码或密码的软件来说的，如果能通过对程序的跟踪找到正确的注册码，通过软件本身的注册功能正常注册了软件，这样的破解称之为完全破解。但如果有些软件本身没有提供注册功能，只是提供试用(Demo)，或是注册不能通过软件本身进行(例如需要获取另外一个专用的注册程序，通过 Internet 的注册等等)，或者是软件本身的加密技术比较复杂，软件破解者的能力、精力、时间有限，不能直接得到正确的注册码，此时就需要去修改软件本身的程序码，即人为改变软件的运行方向，这样的破解称之为暴力破解。

6. 关于破解中程序代码的地址问题

我们在阅读一些破解文章时，都会看到其中的一部分程序代码是用以帮助讲解程序的分析方法的，例如下面的一段程序代码：

```
……
0167：00408033　PUSH　00
0167：00408035　PUSH　EBX
0167：00408036　CALL　[USER32！End Dialog]
0167：0040803CJMP　0040812C
……
```

在这里，程序中的代码地址如 0167：00408033，其代码段的值(即 0167)根据不同的电脑会有区别，不一定一模一样，但偏移值应该是固定的(即 00408033 不变)，所以如果看到某些破解文章里的程序代码的地址值和自己电脑里显示的不一样，不要以为就是出了什么差错，只要自己的程序代码正确就行了。

7. 关于如何设置断点的问题

正确恰当地设置好断点对于快速有效地解密非常重要，好的断点设置可以使我们迅速找到关键的程序段，而不恰当的断点则会对解密造成不必要的精力消耗，甚至根本就不能

拦截到程序的运行。但是很难说具体什么时候用什么断点比较合适，这需要自己去累积经验，总的说来 bpxhmemcpy 这个万能断点对大多数注册码方式的软件都有用，初学者不妨多试试这个断点。

对于那些需要暴力破解的非注册码方式的软件，通常应该拦截对话框(如 bpxDialogBox)和消息框(如 bpxMessageBox(A))等。

不论对于哪一类软件，当自己设置的断点均没有效果时，可以试一下 bpx lock my task，这个断点的作用是拦截任何一个按键的动作，具体常用的一些断点设置请参阅其他资料。

另外，在注册码的破解中通常需要输入用户名和注册码，一般说来用户名和密码都可以随意输入，但是根据笔者自己的经验，很多软件对于注册码都会逐位的进行处理。例如输入“78787878”这串数字，那么在跟踪程序的时候就无法知道自己当时所看到的“78”到底是哪一个“78”，所以笔者比较喜欢用“12345678”这样的注册码输入方式，这样的话就能知道程序是在对注册码的哪一位进行运算，同样的对于那些需要输入较长序列号的软件，输入类似“12345-67890-ABCDEF”这样的序列号较好。

不过有一点大家需要特别的注意，上面讲的注册码输入方式“12345678”是针对拦截 Win32API 函数来说的，假如有些时候直接拦截 Win32API 函数难以找到程序的突破口，而要借助于“S”指令在内存中寻找我们输入的用户名或注册码时，就最好不要采用“12345678”作为注册码，因为内存中很可能有许多的“12345678”字符串，这样就没有办法知道到底要破解的程序使用的是哪一个“12345678”，因此应该选择一个不易和内存数据相同的注册码，比如：74747474，对应的搜索指令为：S 30：0L FFFFFFFF '74747474'。

当然，以上只是笔者个人的习惯而已，具体用什么样的输入形式大家完全可以根据自己的爱好、习惯来定，不必拘泥于某一固定的模式。

8. 关于如何跟踪程序的问题

初学者在开始学习解密的时候往往不知道怎样去跟踪程序，怎样找到注册码比较的地方，当面对长长的一堆程序代码时显得不知所措。

通常软件的程序内部都会利用一个子程序(即 CALL **)去验证输入的注册码正确与否，对于注册码显示存在的程序，一般都会将所输入的注册码和正确的注册码放进寄存器，然后调用验证子程序进行判断，将结果返回，应用程序根据子程序返回的结果决定是否注册成功。这样的程序经常具有如下的形式：

```
****: ******** MOV EAX,[********](或 PUSHEAX 等形式)
****: ******** MOV EDX,[********](或 PUSHEDX 等形式)
****: ******** CALL ********
****: ******** TEST EAX,EAX(或 TESTAL,AL，或没有这一句等形式)
****: ******** JNZ ******** (或 JZ********等形式)
```

其中 EAX 和 EDX 指向的内存区域就是输入的注册码和正确的注册码，这里的寄存器 EAX 和 EDX 是随意写的，也可以是 ECX，EBX，EDI，ESI 等。

对于注册码隐式存在的程序，虽然不能直接看到正确的注册码，但是通常也是先将所输入的注册码地址放进某个寄存器，然后调用子程序去验证，破解时就需要进入子程序去分析注册算法。