突破常规玩电脑系列丛书

智美利达

怎么样，动心了吧？如果是，请赶快翻到目录页寻找你感兴趣的内容仔细阅读。相信我们，读完之后你一定不会失望的

COMPUTER

脱壳技术大放送

计算机技术研究组　总策划

精彩内容导读

- 全面破解加壳压缩
- 如何破解加密保护
- PE&Import的破解
- 如何脱DLL的壳
- 如何保护自己的壳

【特别提示】

本书荟萃了众多破解名家的“不传之秘”，旨在与读者朋友进行经验交流，切勿用于其他目的

内蒙古大学出版社

突破常规玩电脑系列丛书

脱壳技术大放送

计算机技术研究组　总策划

内蒙古大学出版社

书　　名：突破常规玩电脑系列丛书（1-7）
编　　著：武新华
出　　版：内蒙古大学出版社（呼和浩特市大学西路235号 邮编 010021）
责任编辑：赵英
封面设计：南永夫
发　　行：全国各地新华书店
印　　刷：河南省瑞光印务股份有限公司
开　　本：850×1168　1/32
印　　张：7
字　　数：215千字
版　　期：2003年9月第1版　2003年9月第1次印刷
标准书号：ISBN 7-81074-508-5/TP·27
印　　数：1—5000册
定　　价：112.00元（本册定价16.00元）

为什么购买这本书

这是一本有关计算机加密与解密问题的书籍。这本书既是写给普通电脑迷的，同时更适合那些孜孜于计算机软硬件程序的探索者——Cracker。

对于普通的电脑迷，阅读本书的意义意味着你再也不需面对日益繁多的共享软件而茫然无措。我们知道，你曾经为它们心动，因为你确实需要它们的帮助去驾御你的计算机世界；但很多时候你又非常无奈，你无奈于你的钱财，更无奈于对加密解密技术的无知以及由此导致的对你心爱之物的不可获得。与你相似的苦恼我们也曾经历，更深知援助之手在这一时刻所具有的意义。我们无法助你钱财，但有时候知识却远比钱财重要。这本书便具有这样的作用。

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。

而对于更高一级的电脑玩家 Cracker 来说，阅读本书的意义却另有不同。去不断地探索未知，是每一个 Cracker 心目中永远的诱惑。作为一个 Cracker，注定了你与未知的对抗，而不能掌握一些必要的软件破解技术，是你无法忍受的羞耻。

的确，有那么多的共享软件，而你却不明白它们的程序原理，这是有辱于 Cracker 的称号的。当然，Cracker 的目的不是单纯地破解软件，而是通过跟踪软件了解程序思路，从而写出更好的程序。

破解也不在于数量多寡，关键是掌握方法，弄清注册码计算原理。本书中，我们集合了数十位破解高手苦心钻研出来的心得，提供了一般加密解密书籍都无法教你的秘技与妙招，我们相信，当你翻阅完这些 Cracker“老鸟”们轻易不会示人的“私藏秘典”，你也会很快地加入他们的行列。

本书就是这样一本讲解各种破解实例的教科书，这也是我们推荐你购买与阅读本书的理由。

目 录

第 1 章　全面破解加壳压缩

- 如何破解用 ASPack 加的壳
- 全面破解加壳压缩

现在的加壳压缩工具越来越多，被压缩的资源往往无法进行正常编辑。因此就需要能够在不使用脱壳工具的情况下，却能分析被压缩的资源，因此可以不受制于加壳压缩工具的版本升级问题，以期能够达到百分之百的分析被压缩的资源，对于被压缩的软件，还能为它重新建立起一份可编辑的资源，使得其他资源编辑工具（ResScope，eXeScope 等）能够对其进行正常的处理。

1.1　如何破解用 ASPack 加的壳

使用 ASPack 2000 加壳是目前共享软件开发者普遍喜欢采用的一种加密方式，但反过来说，有加壳就有脱壳，“道高一尺，魔高一丈”，在这场加密 VS 解密的持久战中，永远没有最终的赢家。

1.1.1　用 ASPack V2.000 加壳的软件怎么脱

为了练习脱壳，所以随便选一个来加壳。把 netterm1.exe 用 ASPack 2000 加壳，奇怪的是，Procdump 1.62 应该是可以脱掉 ASPack 2000，但是你试一下，好像脱不掉。

现在加壳之风已日渐盛行，若不会脱壳是不行的。通常被加壳的程式，一开始执行时，要把程式解压缩到记忆体，再进入真正程式的 Entry point，所以只要找到程式的 Entry point 就行了。

而你如果走到一个回圈，常常在那边绕来绕去，那就很有可能是在解压缩了；如果你突然走到一个地方，位址变化很大，那可能就是程式进入点了；如果找到程式进入点，就可以大笑三声了。

① 用 Soft-ICE 载入 netterm1.exe，奇怪，在电脑中无法载入，没关系，执行

procdump，选【PE Editor】→【netterm1.exe】

看到 Entry point : 0005E001

Size of image : 00061000

Image Base: 00400000

那就是说，程式一启动的 EIP 应该是 00400000 + 0005E001=0045E001，但是要如何到这边呢?

直接进入 softice 下 bpx 0045E001 ??

当然是不行的，要等程式到了 netterm1 的空在下 bpx 0045E001 才行，那要如何到 netterm1 的空呢?

有很多种方法，随便下一个 netterm1 有用到的 API，就可以拦截了。

例如说下 bpx createwindowexa，接着再执行 netterm1，立刻被拦到。

下 bd *，暂停刚刚的断点，按几下 F12 键，立刻回到 netterm1 的空了。

如何知道是不是 netterm1 的空呢?

仔细看下方的一条绿线，如果写着 netterm 1，就是到 netterm1 的空了，此时，我们只要下 bpx 0045E001，然后接着再按 F5 键，离开 Soft-ICE 就可以了。

② 然后执行 netterm1.exe，立刻被拦住在 cs:0045e001。好了，开始 Trace 了。……可能会一直绕来绕去，这时就要用一点技巧来加快 Trace，就是设中断点，看一下程式，你认为最远会跳到哪里呢?

中断点就设那边，我们再接着按 F5 键，如此循环，要是预估错误导致 netterm1 跑出来了呢?

那就算了，再执行一次啊! 好，到了。

```
cs:0045e4f2   call  0045e577
cs:0045e4f4   jnz  0045e4fc
cs:0045e4f9   ret  c
cs:0045e4fc   mov  eax,1
cs:0045e501   ret
```

走到这边时，就是快要走完了。

③ 小心地按 F10 键，到了

```
cs:0043f3cd  mov  eax,fs:[0]
```

注意看一下位址，差好多喔! 而且看一下下面几行程式码，有在呼叫 GetVersion，GetCommandLineA

如果是解压缩的 source code，是不会有这些的。所以，聪明的你，应该猜到了吧！猜他是程式进入点，就是 Entry point。

④ 下 bd *，先暂停所有的断点，接着再按 F5 键，回到 Windows 系统中，然后启动 trw2000，载入 netterm1.exe，竟然可以 load，发现 trw2000 load 的能力比 softice 还要强。OK，现在就来开始 Trace 吧！！！

不过，现在可以偷懒一下，因为用 softice 已经知道地址了：

下 g 0043f3c0，到了之后，下 pedump net1.exe，net1.exe 名字可自己取，找一下 C:\Windows；或是 netterm1 的所在目录，或是 TRW2000 的目录，就可以找到 net1.exe 了，将它 Copy 到 netterm 的目录，执行一下，发现没有错误。

⑤ 到此为止，壳已经被我们给脱掉了，美中不足的是，脱了壳还比原本的档案大，原本的档案是 359936 Byte，脱壳之后竟然是 397312 Byte，比本来还大。

1.1.2　跟我一起来破 PowerStrip 2.67

由于这个软件用 ASPack 加壳，采用了 Keyfile 保护，因此我们需要先脱壳再破解。

1. 脱壳

因为软件采用了 ASPack 加壳，而且我们也不知道它用的是 ASPack 的哪一个版本，所以要给它脱壳就需要先编辑 Procdump 的 Script.ini 文件。

首先我们需要在 Script.ini 文件中加上这些：

```
P??=ASPack　（？？是行号，注意是 16 进制的）
[ASPack]
L1=OBJR
L2=LOOK EB,?
L3=JZ 5
L4=QUIT
L5=BP
L6=WALK
L7=OBJR
L8=LOOK 61,75
L9=BP
```

LA=STEP

OPTL1=00000000

OPTL2=01010001

OPTL3=01010001

OPTL4=00030000

OPTL5=00000000

这样，只要使用 Procdump 也就可以脱去到目前为止所有用 ASPack 加壳的软件了，而用 Procdump 脱去 pstrip.exe（在 Windows 目录下）的壳，然后将脱壳后的文件复制到 Windows 目录下覆盖原文件就可以了。

2. 破解

在这里我们发现软件好象用 Keyfile 保护了，可惜就是算不出它的注册码，不过如果能够算出注册码的话，当然也就不必脱壳了。

具体破解步骤如下：

首先我们启动 pstrip，然后出现注册提示画面，接着再按 Ctrl +d 键唤出 Soft-ICE，然后下 bpx destroy window，接着再按 F5 键返回 Window 系统中，然后点击一下【OK】，会发现马上被拦下，这时按一下 F12 键，回到 pstrip 的领空，下 s 0 l 90000000 'demo'，然后搜索到 xxx:yyyyyyyy 的地址，地址应该是每个人都不同的，接着再下 bpm yyyyyyyy，然后按 F5 键返回 Window 系统中，退出 pstrip，再一次启动 pstrip，拦下后一直按 F12 键，大概按 17 次左右，反正就是来到这里：

```
:004CB57D  E84AF40000        CALL     004DA9CC
:004CB582  8B45FC            MOV      EAX,[EBP-04]
:004CB585  80B84805000000    CMP      BYTE PTR [EAX+00000548],00
:004CB58C  0F85A8000000      JNZ      004CB63A
:004CB592  8B45FC            MOV      EAX,[EBP-04]
:004CB595  80B84805000000    CMP      BYTE PTR [EAX+00000548],00
:004CB59C  7433              JZ       004CB5D1
:004CB59E  8D954CFEFFFF      LEA      EDX,[EBP-01B4]
:004CB5A4  B801000000        MOV      EAX,00000001
:004CB5A9  E80673F3FF        CALL     004028B4
:004CB5AE  8B854CFEFFFF      MOV      EAX,[EBP-01B4]
:004CB5B4  8D9550FEFFFF      LEA      EDX,[EBP-01B0]
```

```
:004CB5BA  E899B3F3FF        CALL     00406958
:004CB5BF  8B8550FEFFFF      MOV      EAX,[EBP-01B0]
:004CB5C5  BA08C14C00        MOV      EDX,004CC108
:004CB5CA  E8E982F3FF        CALL     004038B8
:004CB5CF  7569              JNZ      004CB63A
:004CB5D1  8B0D7C954F00      MOV      ECX,[004F957C]
:004CB5D7  B201              MOV      DL,01
:004CB5D9  B8D8644B00        MOV      EAX,004B64D8
:004CB5DE  E8FD7AF4FF        CALL     004130E0
:004CB5E3  A350DB4F00        MOV      [004FDB50],EAX * we stop here *
:004CB5E8  33C0              XOR      EAX,EAX
:004CB5EA  55                PUSH     EBP
:004CB5EB  6833B64C00        PUSH     004CB633
:004CB5F0  64FF30            PUSH     DWORD PTR FS:[EAX]
:004CB5F3  648920            MOV      FS:[EAX],ESP
:004CB5F6  A150DB4F00        MOV      EAX,[004FDB50]
:004CB5FB  E898A7F4FF    CALL   00415D98   *这里出现注册提示画面*
:004CB600  83F802            CMP      EAX,02
:004CB603  750F              JNZ      004CB614
```

这时候会发现停在 004CB5E3 处，当光标经过 004CB5FB 这里也就出现注册提示画面，向上看哪里可以跳过它。于是找到 004CB58C 和 004CB5CF 都可以跳过，第 1 个比较好一些，所以在 004CB585 处设断点，再来一次，在 004CB585 处拦下，这时 [EAX+00000548] 的值为 0，如果 [EAX+00000548] 的值为 1，也就行了，因此把

```
:004CB585  CMP  BYTE PTR [EAX+00000548],00
```

改成：

```
:004CB585  INC  BYTE PTR [EAX+00000548]
```

因为少了一个 byte，所以加一个 nop，别忘了下 code on 指令。

3. 用 Uedit 32 打开已脱壳的 pstrip.exe 文件

查找：80B848050000000F85A8000000

改成：FE804805000090————

然后再接着运行 pstrip，没有干扰画面，机上显示着：License #84653，破解成功！

1.1.3　怎样脱用 ASPack 2.12 加的壳

笔者在网上下载了一个 ASPack2.12，但不知它的加壳有没有变化，于是来看看。

试验对象是 Windows 自带的记事本程序，压缩前 52KB，压缩后 32.5KB。好了，我们现在就开工。

首先用 fi 检查壳的类型，系统报告显示 PE Win GUI *UNKNOWN*，没关系了，自己加的壳还用怀疑吗？

接着我们再用 Trw2000 加载程序 Load，程序被中断在如下代码处：

```
0167:0040D001 60                PUSHAD
0167:0040D002 E803000000    CALL    0040D00A
```

然后按 F8 键跟进去（这时如果按 F10 键的话程序会直接运行）：

```
0167:0040D00A 5D                POP     EBP
0167:0040D00B 45                INC     EBP
0167:0040D00C 55                PUSH    EBP
0167:0040D00D C3                RET
```

到了这时候按 1 下 F10 键将会看到下面的代码：

```
0167:0040D008 EB04              JMP     0040D00E
```

这时候我们再按 1 下 F10 键会看到下面的代码：

```
0167:0040D00E E801000000        CALL    0040D014
```

这时候我们再按 F8 键跟进去，来到下面：

```
0167:0040D014 5D                POP     EBP
0167:0040D015 BBEDFFFFFF        MOV     EBX,FFFFFFED
0167:0040D01A 03DD              ADD     EBX,EBP
0167:0040D01C 81EB00D00000      SUB     EBX,0000D000
0167:0040D022 83BD2204000000    CMP     DWord Ptr [EBP+00000422],00000000
0167:0040D029 899D22040000      MOV     [EBP+00000422],EBX
0167:0040D02F 0F8565030000      JNZ     0040D39A        (NO JUMP)
                                <-- 注意，下断点 bpx 0040D39A
0167:0040D035 8D852E040000      LEA     EAX,[EBP+0000042E]
0167:0040D03B 50                PUSH    EAX
```

```
0167:0040D03C FF954D0F0000    CALL   Near ['KERNEL32!GetModuleHandleA']
0167:0040D042 898526040000    MOV    [EBP+00000426],EAX
0167:0040D048 8BF8            MOV    EDI,EAX
0167:0040D04A 8D5D5E          LEA    EBX,[EBP+5E]
0167:0040D04D 53              PUSH   EBX
0167:0040D04E 50              PUSH   EAX
0167:0040D04F FF95490F0000    CALL   Near ['KERNEL32!GetProcAddress']
```

下断点后，接着再按 F5 键后，程序就会来到下面的代码处：

```
0167:0040D39A B8CC100000      MOV    EAX,000010CC
0167:0040D39F 50              PUSH   EAX
0167:0040D3A0 038522040000    ADD    EAX,[EBP+00000422]
0167:0040D3A6 59              POP    ECX
0167:0040D3A7 0BC9            OR     ECX,ECX
0167:0040D3A9 8985A8030000    MOV    [EBP+000003A8],EAX
0167:0040D3AF 61              POPAD      <--关键字，入口点就在附近
0167:0040D3B0 7508            JNZ    0040D3BA
0167:0040D3B2 B801000000      MOV    EAX,00000001
0167:0040D3B7 C20C00          RET    000C
0167:0040D3BA 6800000000      PUSH   00000000
0167:0040D3BF C3      RET<-- 从这返回后我们就来到程序的真正入口点
```

下面就是程序的真正入口点：

```
0167:004010CC 55              PUSH   EBP
0167:004010CD 8BEC            MOV    EBP,ESP
0167:004010CF 83EC44          SUB    ESP,00000044
0167:004010D2 56              PUSH   ESI
0167:004010D3 FF15E4634000    CALL   Near ['KERNEL32!GetCommandLineA']
```

返回到真正的入口点后，下指令 suspend 挂起调试器，然后再接着打开 PEditor，按下【tasks】按钮，在列表中选中你的程序，点右键，选择菜单命令 dump(full)把进程保存到文件中，保存后杀掉进程。

切记：**上面的方法对其它的程序也一样适用（笔者测试了好几个程序，它们在调试器所显示的代码形式都差不多），用这种方法脱壳后的程序可以直接运行。**

附上 Procdump 的 Script:

```
[ASPack2.12]
L1=OBJR
L2=LOOK 61,75
L3=BP
L4=STEP
OPTL1=00000000
OPTL2=01010001
OPTL3=01010001
OPTL4=00030000
OPTL5=00000000
```

1.2 全面破解加壳压缩

对于加壳压缩的手动破解，在很多同类的加密解密书中都可以查阅到，但如何结合一些破解的实例来使广大读者能够感觉到其近在眼前，却并不是一件很容易的事。

下面就是笔者的一些破解秘录，主要是向大家揭示一下破解加壳压缩神秘面纱下的真实面目。

1.2.1 对 PECompact.exe v1.34 的手动脱壳

目标程序：PECompact.exe 35,840 v1.34 脱壳

程序下载：http://www.cnvnet.com/download/d/pecsetup.exe

使用工具：Soft-ICE 4.00; ProcDump 1.6.2; IceDump 6.0.1.5; HexWorkShop（补充一句：以下使用的 Import table 笔者称作输入表，可能说法不是十分规范，不过因为已经习惯了，所以还是请大家原谅一下）。

（注：以下文中数字为十六进制）

破解分析：

首先我们来运行 ProcDump，然后点击【Pe-Editor】按钮，选中【PECompact.exe】，

得到程序内存映象大小：Size of Image 为 00031000 和基地址 Image Base 为 0040000。然后点击 Sections，没有发现象.idata 这样容易识别的输入表 Section，看来找输入表的地址和大小也将成为重点了。

具体破解步骤如下：

1. 首先抓取 Import table

① 首先装载 IceDump。到 IceDump 目录中运行相应 SoftICE 版本的 icedump.exe。（比如用的 SoftICE 版本为 4.00 就到 win 9x/400 目录下运行 icedump.exe）

② 然后再接着运行 Soft-ICE 的 Loader，加载 PECompact.exe。

③ 接着按 Ctrl +D 键进行 SoftICE，如下设置断点

bpx loadlibrarya do "dd esp->4"

（注：在没有明确的输入表地址的情况下，这不失为找输入表地址的一种好办法）

④ 然后再接着按 F5 键回到 Windows 系统中，单击 Loader 中的 Load 按钮。程序中断在第 1 条指令处，进入了 Soft-ICE。

（看来 PECompact.exe 不象受 Asprotect 保护的程序那样。无法用 Soft-ICE 自动中断在第 1 条指令处）

⑤ 然后再接着按 F5 键运行程序，又中断在我们设下的中断 Loadlibrarya 处，看看数据窗口。显示如下：

```
0030:004171 AE4E52454B  32334C45  4C4C442E  4C000000  KERNEL32.DLL...L
0030:004171BE 4C64616F  61726269  00417972  65470000    oadLibraryA...Ge
0030:004171CE 6F725074  64644163  73736572  56000000    tProcAddress...V
0030:004171DE 75747269  6C416C61  00636F6C  69560000    irtualAlloc...Vi
0030:004171EE 61757472  6572466C  00000065  74697845    rtualFree...Exit
0030:004171FE 636F7250  00737365  72460000  694C6565    Process...FreeLi
0030:0041720E 72617262  00000079  4D746547  6C75646F    brary...GetModul
0030:0041721E 6E614865  41656C64  47000000  6F4D7465  eHandleA...GetMo
0030:0041722E 656C7564  656C6946  656D614E  9D8B0041  duleFileNameA...
0030:0041723E 004020A6  20AA9D3B  01750040  0CB58BC3    . @.;.. @.u.....
```

⑥ 根据笔者的经验，一般第 1 次是脱壳代码自己的函数库的加载，并不是需要的输入表。因此，按 F5 键继续执行程序。

程序又一次中断，这一次数据窗口显示如下：

```
0030:0040D4A0 E52454B  32334C45  6C6C642E  49444700  KERNEL32.dllGDI
```

0030:0040D4B0 42E3233　55006C6C　33524553　6C642E32　32.dll.USER32.dl
0030:0040D4C0 4441006C　49504156　642E3233　49006C6C　l.ADVAPI32.dll.I
0030:0040D4D0 4547414D　2E504C48　006C6C64　646D6F63　MAGEHLP.dll.comd
0030:0040D4E0 3233676C　6C6C642E　45485300　32334C4C　lg32.dll.SHELL32
0030:0040D4F0 6C6C642E　41434A00　2E31474C　006C6C64　.dll.JCALG1.dll
0030:0040D500 696C7061　6C642E62　00F4006C　4C746547　aplib.dll...GetL
0030:0040D510 45747361　726F7272　01A70000　65766F4D　astError....Move
0030:0040D520 656C6946　01160041　50746547　41636F72　FileA...GetProcA
0030:0040D530 65726464　00007373　655201CB　6F436461　ddress....ReadCo

这时候就可以看到偏移地址为40D4A0，根据输入表的知识，可以搜索内存映象中的字节A0,D4,00,00。(即00D4A0=40D4A0-400000）来确定输入表的位置。下指令

s 30:400000 l ffffffff A0,D4,00,00

⑦ 搜索的结果显示

Pattern found at 0030:0040D00C (0000D00C)

用如下指令来定位输入表的起始位置（当然目前还不能确定就是输入表）：dd 0040D00C-C

以下为数据窗口的显示结果。

0030:0040D000 0000D0C8　00000000　00000000　0000D4A0　.....
0030:0040D010 0000D2B4　0000D1A0　00000000　00000000　.....
0030:0040D020 0000D4AD　0000D38C　0000D1CC　00000000　.....
0030:0040D030 00000000　0000D4B7　0000D3B8　0000D260　.....`...

现在看看到底是不是要找的输入表。下指令

dd 40D0C8

显示结果如下：

0030:0040D0C8 0000D50A　0000D51A　0000D526　0000D538　.....&...8...
0030:0040D0D8 0000D548　0000D556　0000D564　0000D57A　H...V...d...z...
0030:0040D0E8 0000D58C　0000D59E　0000D5B4　0000D5C2　.....
0030:0040D0F8 0000D5D2　0000D5DE　0000D5F4　0000D5FE　.....

继续追踪，下指令

db 40D50A

显示结果如下：