


附录C InstallShield反编译

 电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY



声明：本电子文档是《加密与解密(第三版)》的配套辅助电子教程！电子工业出版社依法对本作品享有专有出版权。任何未经权利人书面许可，复制、销售或通过信息网络传播本作品的行为；歪曲、篡改、剽窃本作品的行为，均违反《中华人民共和国著作权法》，其行为人应承担相应的民事责任和行政责任，构成犯罪的，将被依法追究刑事责任。

为了维护市场秩序，保护权利人的合法权益，我社将依法查处和打击侵权盗版的单位和个人。欢迎社会各界人士积极举报侵权盗版行为，本社将奖励举报有功人员，并保证举报人的信息不被泄露。

举报电话：(010) 88254396；(010) 88258888

传 真：(010) 88254397

E-mail: dbqq@phei.com.cn

通信地址：北京市万寿路 173 信箱

电子工业出版社总编办公室

邮 编：100036

说明：本篇文档是对《加密与解密》第三版“附录 C InstallShield 反编译”补充，为了降低书的价格，直接以电子文档提供给图书购买者。

看雪软件安全网站

<http://www.pediy.com>

kanxue

2008-6-1

附录C InstallShield反编译

InstallShield（以下简称为 IS）是使用最广泛的安装制作程序，许多商用软件的安装都用它来完成。它其实也是一种解释语言，虽然仅仅是为了做安装程序用的，它的伪码放在 SETUP.INS（IS 5.x）或 SETUP.INX（IS 6.x/7.x）中。

IS 有一个集成开发环境，操作非常方便，它的脚本语言是全部制作的核心部分。IS 是专门用来编写 IS 安装程序的脚本语言。IS 遵循 C 的规则，因而使得 Visual C++ 用户编写安装脚本程序颇为得心应手。并且，IS 为用户提供了几百个内部函数，使得用户不需要太多的代码就能编写具有专业水准的安装程序。具体 IS 脚本语法请参考专门的资料。

7.4.1 安装文件构成

IS 5.x 安装过程中使用到的典型文件主要有：

SETUP.LIB
SETUP.PKG
_SETUP.DLL
SETUP.INS
SETUP.EXE

- **SETUP.LIB**：是压缩的数据库文件，包含安装过程中用到的 EXE 和 DLL。某些情况下这些 DLL 或 EXE 文件可能以独立的方式和 SETUP.EXE 放在同一个目录下，也可压缩放进 _SETUP.LIB 中。
- **SETUP.PKG**：文件是用来记录复制文件过程中需要哪些文件，对解密用处不大。
- **_SETUP.DLL**：是 IS 中包含资源的 DLL 文件，同样不重要，因为几乎所有 IS 的安装程序中都有这类东西。
- **SETUP.INS**：是已编译的安装脚本（Installation Script），这是 InstallSHILED 程序安装过程中最重要的一部分！在 Windows 9x 系统中，该文件的图标和拨号网络的连接是一样的。这个文件控制 IS 安装程序的一切动作。而 IS 6.x 的脚本编译后是 setup.inx，而不是 setup.ins。

- **SETUP.EXE**: 是安装引擎, 负责执行安装脚本, 执行所有对各个 DLL 以及磁盘访问过程的 32 位调用。

IS 制作的安装文件已将数据文件进行了压缩, 并且一般以 `setup.lib`, `*.cab`, `xxx.l-x` 和 `xxx.z` 形式存在。可以用 `icomp` 工具来解压它们。`i5comp` 支持 IS v5.x, 而 `i6comp` 支持 IS v6.x。

7.4.2 脚本语言反编译

1. IS 5.x脚本语言反编译

IS 5.x 已编译的脚本语言存放在 `SETUP.INS` 文件里, 可以用 `isDcc` 和 Windows IS Decompiler 工具反编译。

(1) isDcc

用法: `isdcc setup.ins > [output_file]`。

(2) Windows IS Decompiler

支持 `Is3.x` 和 `Is5.x` 脚本。它是图形界面, 操作非常简单直观, 但不太稳定, 经常不能正常反编译完文件。对源文件能进行一些简单的修改 (仅限运算符上), 在需修改的运算符上, 单击右键打开功能菜单, 选择相应的运算符替换原操作符。因此, 不需要其他的工具辅助, 就可轻易取消 IS 序列号保护。

2. IS 6.x/7.x脚本语言反编译

IS 6.x/7.x 脚本语言存放在 `SETUP.INX` 文件里。可以用 `SID` 和 `ISD` 工具进行反编译。

(1) SID

`SID` 支持 IS 6.x/7.x。它是图形界面, 自动检测 IS 的标准函数, 支持函数参考、串式参考显示。也能修改 `SETUP.INX` 文件, 操作基本与 Windows IS Decompiler 类似。

(2) InstallShield Decompiler

命令行界面, 支持 `InstallShiled 2.0 (*.ins)`, `InstallShiled 3.0, 5.0-5.5 (*.ins)` 及 `InstallShiled 6.0-6.21 (*.inx, *.obl, *.obs, *.dbg)`。

用法: `isd inx_inx_file [output_file] [-options]`。

7.4.3 IS解密

在理论上, IS 制作的安装程序可以实现多种类型的保护, 从序列号到狗加密。然而在实际中, 经常遇到的保护类型主要是序列号。

某软件用 IS 6.0 序列号方式保护。运行 `setup.exe` 安装程序, 在安装过程中输入一个假的序列号, 弹出一个出错的警告窗口以提示“序列号不正确。请重新输入序列号”。现在记下上面的警告消息, 退出安装。

由于 InstallShield Decompiler 对中文支持得很好, 所以先用其查看中文提示信息, 再用 `SID` 来修改 (也可先用 16 进制工具打开 `setup.inx`, 将中文提示信息改成英文, 再用 `SID` 反编译)。在 DOS 窗口里输入命令:

```
isd setup.inx 1212.txt
```

打开生成的文本文件 `1212.txt`, 查看提示信息。如下所示:

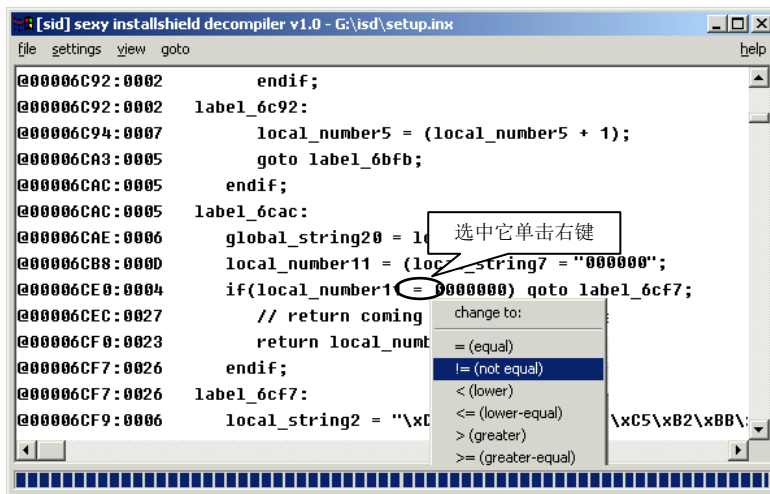
```

label_0030:
/* 00006CAE: 0006 */ g_str0014 = s0006;
/* 00006CB8: 000D */ n000A = s0006 == "000000";
/* 00006CE0: 0004 */ if(n000A = 0000000) goto label_0031;    // 关键
/* 00006CEC: 0027 */ // -- Start Return Code -- //
/* 00006CF0: 0023 */ return n0000;

// : Jump Referenced(1):
// : 00006CE0,
label_0031:
/* 00006CF9: 0006 */ s0001 = "序列号不正确。请重新输入序列号";

```

然后用 SID 反编译 setup.inx 文件，来到 00006CB8 处。将 “=” 改成 “!=”，具体情况如图一所示。



图一 改变跳转指令

最后，单击菜单 “File/Patch changes” 将修改写入 setup.inx 文件里，以后就可输入任意序列号进行安装了。

当然，分析 IS 安装程序也可用动态调试技术，难点在于 IS 使用的安装脚本是解释执行的，碰到这种解释型的东西用动态调试技术自然比较费劲。如果是在脚本中调用单独的 DLL 判断序列号，那就与跟踪普通程序没什么区别。例如，SoftICE 4.05 自己的安装序列号就是调用自带的一个 DLL 中的 DigitCheck() 函数来判断的。用 LoadLibraryA(), GetProcAddress() 等可以拦截下来，然后跟踪。IS 的脚本一般是在输入序列号的时候把序列号搬到内存的某个地方先存起来，等按 Next 按钮的时候它就不再从编辑框中取序列号，而是直接判断原先保存的那部分。所以不能把序列号全输入好了再设断点，而是在输入序列号之前就把断点设好，看它把序列号搬到哪里去了，对其设 BPM/BPR 断点。它的脚本在解释执行时不停地用 lstrcpyA() 之类的函数把序列号搬来搬去。要有耐心，再加上点运气可能会找到判断注册码的地方。