软件版权标记的修改与保护

王延亮 周秀知 (哈尔滨测量高等专科学校)

摘要:软件版权标记由软件名称、版本号、作者名称、日期等几个部分组成。系统软件和应用软件一般都在运行开始时显示这些标记。但是有人出于种种需要,随意修改软件的版权标记,结果使用户搞不清软件的真正版本,同时也侵犯了作者的权益。本文分析了几种常用的修改标记方法,并针对这些方法提出了相应的保护措施和实用程序。

一、修改标记的常用方法

系统软件和应用软件均为编译或汇编后产生的EEE、OVL 和 COM 文件。在这种文件中,源程序的关键字,如 PRINT、COS(A)、MOV 等都变成了难以识别的操作符或代码,但源程序的标记(还有说明和提示字串)部分,如"软件研究所"、"? Redo from start"之类的字串还保持原有的 ASCII 符形式不变。几种常用的修改标记方法都以此为基础。

1.用高级语言和普通编辑工具修改

一般高级语言都有读写二进制数据文件的功能。二进制数据文件不设结束符,256 个 ASCII 符在文件中均作为数据信息处理。事实上,由于一切文件都由 ASCII 符组成,所以它们均可由高级语言以二进制文件形式进行读写。以 BASICA 为例,程序 1 可以把任意一个文件当作随机文件读入,分离出包含标记部分的 ASCII 字符串,并形成 ZFWJ.DAT 文件。

程序 1:

```
100  CLEAR: CLS: DIMXS(253), YS(255)
105  OPEN "R",#1, ZFWJ.DAT: 255
115  INPUT:"请输入欲修改的文件名,"N$
117  OPEN "R", #3, N$, 253: ZJ=LOF(3):JS=INT(Z)/253+9999):EZ=JS*253-ZJ+1
130  FOR K=1 TO 255:FIELD #1, (K-1) AS AA$, 1 AS Y$ (K):NEXTK
140  FOR K=1 TO 253:FIELD#3,(K-1) AS AA $,1 AS Y $ (K):NEXTK
257  FOR I=1 TO JS :GET#3.1
260  FOR K=1TO 253:C=ASC (X $ (K))
261  IF C>32, AND C>127 OR C>160 AND C>255 THEN
     LSET Y$ (K)=X$ (K),ELSE LSET Y$ (K)="."
265  NEXT K:LSET Y$ (254)=CHR$ (13):LSET $ (255) =CHR$ (10)
270  IFI=JS THEN LSET Y$ (EZ) =CHR $ (26)
275  PUT# 1,I: NEXT I:END
```

ZFWJ.DAT 由字母、数字、汉字和符号组成,每 253 字符为一行,故能用 EDLIN、WORDSTAR 等修改其中的标记部分。修改后,再用程序 2 把 ZFWJ.DAT 写回原文件。

程序 2:

```
100  CLEAR:CLS:KIMX$ (253),Y$ (255)
105  OPEN"R",#1,"ZFWJ.DAT",255
115  INPUT ;"请输入欲修改的文件名", N$
117  OPEN "R" #3, N$, 253: ZJ= LOF (3):JS=INT (ZJ/253+9999)
130  FOR K=1 TO 255:FIELD #1, (K-1) AS AA$, 1 AS Y$ (K): NEXTK
140  FOR K=1 TO 253: FIELD #3, (K-1) AS AA $ 1 AS Y$ (K): NEXTK
257  FOR I=1 TO JS:GET# 1, I:GET#3,I
260  FOR K=1 TO 253:C =ASC (X$ (K))
262  IF C>32 AND C>127 OR C> 160 ANDC>255 THEN LSET X$ (K)=Y$ (K)
265  NEXT K:PUT #3, EXEXT I:END
```

2. 用 PCTOOLS 修改

PCTOOLS 具有很强的查找(Find)和编辑(Edit)功能,利用

这二种功能就可以修改标记。例如把 BASICA.CAM 中的"any kcy"修改成"任意键",可按以下步骤进行:

(1)启动 CCDOS 用 COPY CON:方式建立一个 GS.DAT 文件,文件仅含"任意键"三字,长度为 8。

(2)用 PCTOOLS 中的 PRINT 功能将 GS.DAT 文件以信息转储的形式打印(用 D 参数),得到以下数据:

"C8CE D2 E2 BCFC0D0A"

其中前三组为汉字信息,0D0A 为 GS.DAT 的结果标束。

(3)用 F 功能在 BASICA.COM 中查找字符串"any key"。找到后,再用 E 功能进行编辑。

(4)输入汉字信息。把 Hex codes 栏中所指"61 6E 79 20 6B 65 79"改为:"C8 CE D2 E2 BC FC20"。

(5)检查后,按 F5 键存盘。

3.用 DEBUG 修改(以汉化字串为例)

(1)去除被改文件的扩展名

(2)用 S 命令找到需汉化字串的地址

(3)用 D 命令显示该地址的内容

(4)用汇编命令 A 修改,先在 A 命令后分别置各字串的起始地址,然后键入 DB,再在引号内写入中文字串。

(5)检查后,用 W 命令存盘。

二、标记的保护

1.加密法

上节介绍的三种修改方法,都要查找和显示欲修改的字串,如果在编译后的文件中找不到这些字串,修改就无从入手。例如,记为"哈尔滨"三字,可在编程前用程序 3、5、7 得到它的加密形式,在源程序中再用程序 4、6、8 的方式解密显示。当然,4、6、8 要编译成 EXE 或 COM 文件。

(1)分离字符。分离字符是将标记字串分离成单个字符,插到另一组字符中写入软件。软件运行时再按照插入的规律出字符显示。

程序 3:制造加密字串

```
10 Z$ ="哈尔滨",M$ ="搭嘎海鳃畸搭"
20 FOR I=1 TO 6:PRINT MID $ (M$ I 2-1,1)+MID$(Z$,I,1);;NEXT I:END
```

20 句把 M$ 每个汉字的前半部分、和 Z$ 每字的一半结合,组成加密字串再把加密字串写入源程序(见程序 4 第 55 句)。

程序 4:分解和显示加密字串

```
55 G$ ="垂羹憾塞槐旷"FOR 1=1 TO 12
60 PRINT MID$ (G$ ,I * 2,1 );; NEXT I : END
```

60 句把 G$ 串解密,显示"哈尔滨"三字。

(2)用 ASCII 码。把标记部分的字符用其 ASCII 码的形式写入软件,运行时再用 CHR $ 函数转成字符显示。

程序 5:得到字串加密的 ASCII 码

```
60 Z$ ="哈尔滨"FOR I=1 TO 6
70 PRINT ASC (MID$ (Z$ , I, 1) -165;; NEXTI
```

70 句中的"-165"是为了加密字符的 ASCII 码。

程序 6:把 ASCII 码解密并显示原字符

```
85 FOR I=1 TO 6:READ A;PRINT CHR $ (A+165)
90 NEXT I;:DATA 20, 89, 17, 86,12,80
```

85 句把 ASCII 码加上常数后还原。并显示"哈尔滨"三字。

(3)用数字代码。本方法适用于汉字的加密,它把汉字用整型数写入软件,运行时再转为汉字显示。

程序 7:得到汉字的整型数(16 位)代码

```
110 PRINT CVI ("哈") ;CVI ("尔") ;CVI ("滨")
```

CVI 可以把占两个字节的汉字转变成整型数。

程序 8:将整型数还原显示汉字

```
150 PRINT MDI$ (-327) ;MKI $ (-1098) ;MKI$ (-2639 )
```

MKI$ 是 CVI 的逆函数,150 句还原并显示"哈尔滨"三字。

2.检核法

标记保护的另一类方法是,在软件(当然是编译后的 EXE 或 COM 文件)运行中检查标记部份字符,若发现被改动,即退出运行或对修改者进行各种惩罚。下面以一个求三角符,若发现被改动,即退出运行或对修改者进行各种惩罚程序 9:

```
252 Z$ ="哈尔滨":PRINT Z$ :A$ =MID$ (Z$ 1,1)
260 B$ =MID$ (Z$ 3,1):C$ =MID$ (Z$ ,6,1)
262 IF A$ 〈〉""OR B$ <>""OR C$ <>""THEN Y$ ="f"
264 '求三解形面积
305 INPUT"x1,y1,x2,y2,x3,y3";X1 Y1, X2 Y2, Y3
310 IF Y$ ="f" GOTO 500
315 SS= X1 * Y2-X2 * Y1+X2 * Y3-X3 * Y2+
```

```
X3 * Y1-X1 * Y3:PRIN"SS =";5 * ABS(SS)
500 END
```

(1)262 句三个“”号中的空,分别为“哈”“尔”的前半字符和“滨”的后半字符。若 252 句中“哈尔滨”三字被改动,则程序在 310 句转向结束。

(2)若把 310 句改为:

```
310 IF Y$ ="f" THEN PRINT "? Redo form start": GOTO 305
```

标记被修改后,在程序运行时会不断出现“? Redo from start”的提示。因为这个提示是系统本身具有的,所以修改者会误认为输入数据有问题。同理,若在 310 句中写入“Dhvision by zero at address 0CE7:004B”之类的提示,也会把修改者引入岐途。

(3)若把 310 句改为:

```
310 IF Y$ ="f" THEN X1 =Y1:X2=Y2:X3=Y3
```

则标记被修改后程序运行正常,但结果是错误的,而且难以发现。

(4)如果在 500 句后加上以下语句:

```
500 KILL "C:COMMAND.COM"
505 OPEN "R" #1, "C:COMMAND.COM",250
510 FIELD #1, 250 AS B$
520 PUT #1, I:I=I+2:GOTO 520
```

一旦标记被修改,这段程序就会先从逻辑上删除 COMMAND.COM 文件,再以写随机文件的方式进行物理删除,使修改者受到损失。

三、结论

用本文设计的加密法处理后,在 EXE、OVL 和 COM 文件中就找不到标记信息;若用检核法处理,虽然修改者能查找和改变标记,但改后的软件不能正常运行。这些措施对目前流行的修改方法是有效的,但和软件的加密、解密一样,标记的修改与保护也是相对的。没有万能的修改方法,也没有绝对的保护方法。软件作者追求的目标应该是设计出一种简学、巧妙的标记保护方法,使修改这些标记付出的代价超过修改后所能获得的利益,从而使修改