DOS 系统下批处理程序的加密

罗 升 陈洪森 （中国工商银行江苏省洪泽县支行）

磁盘操作系统 DOS 的批处理程序是由后缀.BAT 为扩展名的文本文件,该文件包括若干条 DOS 内部命令、外部命令及用户应用程序等组成。当磁盘操作系统的命令解释程序 COMMAND.COM 在命令所指定的路径上找到欲执行的批处理文件后,逐条解释并执行相应的有关命令,完毕后返回到 DOS 命令处理程序状态。由于批处理程序是以文本文件的形式存储在磁盘中,并能通过有关命令或程序如 type、ws、edlin、pe 等进行查阅、修改,所以批处理程序是完全透明的无保密性及安全性可言。为此,人们采取各种各样的保密方法、手段对批处理文本文件进行处理,以防止批处理程序被非法篡改、暴露。如在有关报刊上介绍的通过 C 语言编释程序的系统调用命令,处理批处理文本中的每条命令,以达到保密的目的。这种方法虽然可行能够抵抗 EDLIN、TYPE、WS 等命令的袭击;但是对 PCTOOLS 或 DEBUG 等工具软件却无能为力,很容易被破译,因此其安全保密性还很有限。如何才能得到一种更安全保密的批处理文件的保密方法?本人通过实践找到了解决批处理文件进行加密的一种较好方法。

用 DEBUG 或 PCTOOLS 工具软件对批处理程序进行分析,可以发现,每一条命令均以 ODH、OAH 作为该命令的结束符,而整个批处理文件以 ODH、OAH、IAH 作为该处理程序的结束符。大家都知道 ODH、OAH、IAH 分别是 ASCII 码的回车符、换行符、文件结束符,并且磁盘操作系统命令处理程序 COMMAND.COM 是以 ODH 作为批处理命令的结束符,以 IAH 作为批处理程序的结束符。通过以上分析,可以在不破坏原批处理程序文本文件组织结构的前提下,对批处理文本进行预处理,使批处理程序的换行符 OAH 全部改为返回符 OOH,从而使经过预处理的批处理程序的所有内容每一行层层覆盖;起到为批处理文件加密的作用。用此方法对批处理文本文件处理后,用 TYPE、EDLIN、WS、DEBUG、PE、SK 等命令或工具软件都无法破译批处理程序,最多只能破译最后两行的叠加结果,而不能确切得出最后两行的内容。

以下给出对批处理文本程序进行预处理的加密程序,由 Turbo C 语言编制的源程序清单。此程序不仅能解决批处理程序的安全保密,还能够运用于本文件形式的其它程序的安全保密,譬如 BASIC 解释程序、dBASE 及 MFOXPLAS 源程序。

```
#include＜dos.h＞
#include＜stdio.h＞
#include＜string.h＞
/＊开始修改为固定返回符 OOH＊/
maim( )
{
    char errorkkj[35],filename[20], * ptr, * data-all;
    static char readerror[ ]="读文件失败!";
    FILE * stream;
    int length;
    printf("请输入需要加密的文本文件名:");
    gets(filename);
    stream=fopen(filename,"rb+");  /＊打开二进制文本文件＊/
    if (stream=NULL)

        printf("007");
        printf("007");
        strcpy(errorkkj,readerror);
        strcat(errorkkj,filename);
        printf("%s",errorkkj);
        printf("＼007");
```

（下转第 58 页）

(上接第 59)

```
    }
    else
    {
      fseek(stream,2,SEEK_END);
      length = ftell(stream);
      data_all = (char * )calloc((unsigned)length,size
of(char));
      if (!data_all);
        {
          printf("出现错误,请检查后再运行!");
          exit(1);
        }
      rewind(stream);
      while (!feof (stream ) )      / * 修改为返回符
OOH * /
        fscanf(stream,"[\xOO]",data_all);
        while(ptr = strrchr(data_all,'\n'))
          strnset(ptr,'\xOO',1);
      }
      rewind(stream);
      fwrite(data_all,1,length,stream);
      fclose(stream);
```