

## 微机加密心得

April 26, 1999

Sinbad Technical Publications  
Website: <http://sinbad.zhoubin.com>  
© Copyright 1999, All Rights Reserved

---

如何防止他人进入自己的计算机？怎样保护宿舍里公用机器上自己的文件？这些问题同学们都比较关心，BBS上也经常有相关的讨论。现在我把自己的一点心得体会 post 出来，与大家共享，就当抛砖引玉吧。

## 一. 拒人于千里之外

防止别人进入系统最简单的方法就是修改 CMOS，设置上开机密码，但通用密码的存在使这项功能形同虚设，所以说不太保险。

比较方便的就是借助软件 System Commander (以下简称 SC) 来设置密码，SC 的强项是能使多个操作系统共存于硬盘，而且互相之间协调的很好。它之所以能做到这一点，是因为安装 SC 时，硬盘的 MBR (Master Boot Record, 主引导扇区) 及其他 0 磁道的扇区被作了修改，填充了大量 SC 引导各操作系统的代码，也就是说，你的机器已经交给 SC 了，在这里设置密码比较好，适合于你的个人电脑。

SC 是一个很好的工具软件，但却属于“请神容易送神难”的那种。要把它干干净净的卸掉，需要对硬盘有一定的了解。前面已经提到，SC 修改了硬盘 0 面 0 磁道包括 MBR 的前 6 个扇区。第二扇区是引导各主分区的程序，如果把 C 盘根目录下的 Syscmdr.sys 文件改名，这段程序将被执行，出现一个分区启动菜单，我觉得这比 OS2 的 Boot Manager 简洁多了。第 3 扇区是个备份，第 4、5、6 扇区填充了大量的 P，不知作什么用的。这后 5 个扇区用 Debug 的 f 命令全部填 0 即可，至于 MBR 就在 DOS 下打入“FDISK/MBR”，MBR 的代码部分就恢复了。

把 SC 从 0 磁道赶走以后，就可以操起 Debug，自己编写加密代码了。这时你必须清楚 DOS 的启动过程，并能够读懂硬盘 MBR 中代码部分和分区表各字节项的含义。我在这里简要介绍一下，详细内容请参考有关病毒和加解密的书籍。机器启动时，硬件检测成功后，通过 INT 19H 将硬盘的 MBR 读到内存指定区域 0:7C00H 中，然后转到其中执行，根据分区表的信息确定启动哪个分区内的操作系统。MBR 中有 206 个字节为空，插入相应的代码之后就可以在启动操作系统之前实现加密。下面一小段程序可以观察 MBR 的内容：

```
C:\>debug
-a
1369:0100 mov ax,201
1369:0103 mov cx,1
1369:0106 mov dx,80
1369:0109 mov bx,200
1369:010C int 13
```

```
1369:010E int 20
1369:0110
-g
Program terminated normally
-d200
```

MBR 一共 512 字节，用 d 命令 4 次看完，你会发现中间有一大段为 00 的部分，这将是我们的加密程序的所在之处。用 u 命令反汇编，可以看到两个 jmp 指令，找一个修改为 jmp 到加密程序段开始处，然后在程序段末尾再 jmp 回来就天衣无缝了。在扇区内写代码，要注意的是当时地址与执行时地址之间的换算关系，搞不清楚就死机了。

下面将介绍如何把一段加密代码植入 MBR 的详细过程，完成以后每次机器，必须输入字母\$才能引导 DOS，按其它键均死机。运行上面那段程序把 MBR 读入内存，然后按照以下步骤实现：

```
-u218
136A:0218 EA1D060000    JMP    0000:061D
-a218
136A:0218 CALL 0000:06E0    (通过调用子程序来执行加密代码)
-a2e0
136A:02E0 MOV AH,0
136A:02E2 INT 16          (从键盘接收一个字符)
136A:02E4 CMP AH,24      (是$吗?)
136A:02E7 JE 02EB       (是，返回)
136A:02E9 JMP 02E9       (不是，进入死循环)
136A:02EB RET
136A:02EC
```

最后用 INT 13H 的写功能将内容写回 MBR 就大功告成了。

有一点需要说明，装了 Win95/98 之后，MBR 中为空的部分就不到 206 字节了。所以上面从 2e0 处开始写代码可能不适合于你的硬盘，我是装完 Win95/98 之后，把 DOS 的 MBR 代码部分找回来，把新的给覆盖了。这样做可以获得更多的空余空间来嵌入加密代码，对操作系统又没什么影响，比较不错。现在 DOS 不多见了，需要的话给我写信。我曾经写了一段“十位密码确认”的代码，由于设计了比较好看的界面，MBR 放不下，还利用了 0 磁道的第 2 扇区。过几天整理一下贴出来。

## 二. 井水不犯河水

大部分宿舍里的计算机都是公用的，如何保障个人隐私是一个很棘手的难题。以前的 DOS 时代，大家通过修改 FAT (File Allocation Table, 文件分配表) 和 RDT (Root Directory Table, 根目录表) 来实现对自己目录的加密，还要用到一些磁盘工具，很是麻烦。如今是 Win98 了，也有相关的加密工具出现。以前我曾在 Win95 下用过一个对目录加密的软件，叫做 007，它好象把加密的目录作备份，如果文件太多，很是浪费硬盘空间。有没有一个既简单且加密效果又好的办法呢？

如果你经常折腾硬盘，不可能没用过 Partition Magic 吧，它有个隐藏分区的功能比较不错。经过一段时间的钻研，我终于搞明白了它的实现原理，现在就可以脱离 Partition Magic，提出一个针对宿舍里公用微机的解决方案。

这个方案综合了上面所讲的在 MBR 中嵌入代码的内容，大体如下：在扩展分区中除了 D 盘放公用程序，再划出 8 个逻辑盘给 8 个用户（一个 100M 总共才 800M，对大硬盘来说无所谓）。启动 Win98 前 MBR 中的代码先被执行，要求输入用户名和密码，确认后使用该用户的逻辑盘变为可见，其他的则处于隐藏状态。这样每个用户在进入 Win98 后，都拥有各自的 E 盘来存放自己的个人文件，保密性比较好。下面分析一下如何实现对逻辑盘的隐藏：

我们知道，硬盘分区表位于 MBR 的 1BEH 与 1FDH 处，占 64 个字节，可以容纳 4 个分区的信息，每个表项占 16 个字节，其含义见表：

偏移量	含义
0	引导标志(80h 表示活动分区，00h 表示非活动分区，其他值非法)
1	本分区的起始磁头号
2-3	本分区的起始扇区号和起始柱号
4	分区类型 (0B—Win95 FAT32; 06—DOS FAT16; 05—扩展 DOS;)
5	本分区的结束磁头号
6-7	本分区的结束扇区号和结束柱号
8-B	本分区的相对扇区号
C-F	本分区的扇区数

从表中可见，扩展 DOS 分区标识号为 05，它在分区表中占有一项。我机器上该项的内容如下：

```
00 00 41 31 05 3F FF FD C0 C3 12 00 C0 1C 2C 00
```

在偏移 2-3 处的 3141H 是整个扩展分区的起始扇区号和起始柱号，用 INT 13H 将该起始扇区读进内存，可以发现该扇区与 MBR 类似，在偏移 1BEH 到 1DDH

处，记录了两个分区表项：

```
00 01 41 31 0B 3F BF 61 3F 00 00 00 81 C3 12 00  
00 00 81 62 05 3F FF FD C0 C3 12 00 00 59 19 00
```

第一表项对应于本分区，第二表项则对应于下一分区，各字节含义同上。再读入 6281H 处的第一扇区，两个分区表项为：

```
00 01 81 62 0B 3F FF FD 3F 00 00 00 C1 58 19 00  
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
```

可见，该硬盘有 2 个逻辑分区，且都是 FAT32 的。各逻辑分区是通过各自第一扇区的分区信息表串起来形成了一个链式结构，使得 DOS 能够管理多个逻辑分区。如果有 8 个逻辑分区，就有 8 个扇区记录了分区信息，DOS 引导时（不管是从软盘启动还是从硬盘启动），都将搜索这条链，为各逻辑盘建立磁盘基数表。其实，DOS 的这种链表式数据结构到处可见，在设备管理中、在内存管理中、在文件管理中，可谓比比皆是，大家应该很熟悉的。

**Partition Magic** 隐藏分区的原理很简单，就是把第 4 偏移处的分区标识符改成了 1B，改且仅改了第一表项，作为指针的第二表项没有变化。这样，分区标识符为 1B 的逻辑分区由于 DOS 不可认，所以就处于隐藏状态。分区标识符占一个字节 00-FF，只要你选择得当，不要与其他操作系统的发生冲突，改为其他值，**Partition Magic** 也认不出来的。

写到这里，明白了隐藏分区的原理，就看如何发挥汇编水平，写出短小精悍的代码了。

以上是我平时玩电脑中的一点心得体会，当中定有许多不足及错误之处，请各路高人不吝指教。