第五篇　Linux系统安全分析

第23章　系统管理员安全

本章从系统管理员的角度讨论安全问题。系统管理员是管理系统的人，其工作包括：启动系统、停止系统运行、安装新软件、增加新用户、删除老用户以及完成保持系统发展和运行的日常事务工作。

23.1　安全管理

安全管理主要分为四个方面：

• 防止未授权存取：这是计算机安全最重要的问题。用户意识、良好的口令管理（由系统管理员和用户双方配合）、登录活动记录和报告、用户和网络活动的周期检查，这些都是防止未授权存取的关键。

• 防止泄密：这也是计算机安全的一个重要问题。防止已授权或未授权的用户存取他人的重要信息。文件系统查帐、su登录和报告、用户意识、加密都是防止泄密的关键。

• 防止用户拒绝系统的管理：这一方面的安全应由操作系统来完成。一个系统不应被一个有意试图使用过多资源的用户损害。不幸的是，Linux不能很好地限制用户对资源的使用，一个用户能够使用文件系统的整个磁盘空间，而Linux基本不能阻止用户这样做。系统管理员最好用PS命令，记帐程序df和du周期地检查系统。查出过多占用CUP的进程和大量占用磁盘的文件。

• 防止丢失信息：这一安全方面与一个好系统管理员的实际工作（例如：周期地备份文件系统，系统崩溃后运行fsck检查，修复文件系统，当有新用户时，检测该用户是否可能使系统崩溃的软件）和保持一个可靠的操作系统有关(即用户不能经常性地使系统崩溃)。本书主要涉及前两个问题。

23.2　超级用户

一些系统管理命令只能由超级用户运行。超级用户拥有其他用户所没有的特权，超级用户不管文件存取许可方式如何，都可以读写任何文件，运行任何程序。系统管理员通常使用命令： /bin/su 或以 root 进入系统从而成为超级用户。在后面文章中以#表示由超级用户运行的命令，用$表示其他用户运行的命令。

23.3　文件系统安全

23.3.1　Linux文件系统概述

Linux文件系统是Linux系统的心脏部分，提供了层次结构的目录和文件。文件系统将磁盘空间划分为每1024个字节一组，称为块(也有用512字节 为一块的，如：SCO XENIX)。编号从0到整个磁盘的最大块数。

全部块可划分为四个部分，块 0称为引导块，文件系统不用该块 ;块1称为专用块，专用块含有许多信息，其中有磁盘大小和全部块的其他两部分的大小。从块 2开始是i节点表，i节点表中含有i节点，表的块数是可变的，后面将做讨论。 i节点表之后是空闲存储块（数据存储块），可用于存放文件内容。文件的逻辑结构和物理结构是十分不同的，逻辑结构是用户敲入 cat命令后所看到的文件，用户可得到表示文件内容的字符流。物理结构是文件实际上如何存放在磁盘上的存储格式。用户认为自己的文件是边疆的字符流，但实际上文件可能并不是以边疆的方式存放在磁盘上的，长于一块的文件通常将分散地存放在盘上。然而当用户存取文件时，Linux文件系统将以正确的顺序取出各块，给用户提供文件的逻辑结构。

当然，在Linux系统的某处一定会有一个表，告诉文件系统如何将物理结构转换为逻辑结构。这就涉及到i节点了。i节点是一个64字节长的表，含有有关一个文件的信息，其中有文件大小、文件所有者、文件存取许可方式，以及文件为普通文件、目录文件还是特别文件等。在i节点中最重要的一项是磁盘地址表。

该表中有13个块号。前10个块号是文件前10块的存放地址。这10个块号能给出一个至多10块长的文件的逻辑结构，文件将以块号在磁盘地址表中出现的顺序依次取得相应的块。

当文件长于10块时又怎样呢?磁盘地址表中的第11项给出一个块号，这个块号指出的块中含有256个块号，至此，这种方法满足了至多长于 266块的文件（272 384字节）。如果文件大于266块，磁盘地址表的第12项给出一个块号，这个块号指出的块中含有 256个块号，这256个块号的每一个块号又指出一块，块中含 256个块号，这些块号才用于取文件的内容。磁盘地址中和第13项索引寻址方式与第12项类似，只是多一级间接索引。

这样，在Linux系统中，文件的最大长度是16 842 762块，即17 246 988 288字节，有幸是Linux系统对文件的最大长度（一般为 1到2M字节）加了更实际的限制，使用户不会无意中建立一个用完整个磁盘区所有块的文件。

文件系统将文件名转换为 i节点的方法实际上相当简单。一个目录实际上是一个含有目录表的文件：对于目录中的每个文件，在目录表中有一个入口项，入口项中含有文件名和与文件相应的i节点号。当用户敲入cat xxx时，文件系统就在当前目录表中查找名为 xxx的入口项，得到与文件xxx相应的i节点号，然后开始取含有文件xxx的内容的块。

23.3.2 设备文件

Linux系统与本系统上的各种设备之间的通信，通过特别文件来实现， 就程序而言，磁盘是文件，调制解调器是文件，甚至内存也是文件。所有连接到系统上的设备都在 /dev目录中有一个文件与其对应。当在这些文件上执行I/O操作时，由Linux系统将I/O操作转换成实际设备的动作。例如，文件 /dev/mem是系统的内存，如果使用 cat命令显示这个文件，实际上是在终端显示系统的内存。为了安全起见，这个文件对普通用户是不可读的。因为在任一给定时间，内存区可能含有用户登录口令或运行程序的口令，某部分文件的编辑缓冲区，缓冲区可能含有用ed -x命令解密后的文本，以及用户不愿让其他人存取的种种信息。

在/dev中的文件通常称为设备文件，用ls /dev命令可以看看系统中的一些设备：

acuo	呼叫自动拨号器。
console	系统控制台。
dsknn	块方式操作磁盘分区。
kmem	核心内存。
mem	内存。

lp	打印机。
mto	块方式操作磁带。
rdsknn	流方式操作的磁盘分区。
rmto	流方式操作的磁带。
swap	交换区。
syscon	系统终端。
ttynn	终端口。
x25	网络端口。
等等	

23.3.3 /etc/mknod命令

用于建立设备文件。只有系统管理员能使用这个命令建立设备文件。其参数是文件名，字母c或b分别代表字符特别文件或块特别文件、主设备号、次设备号。块特别文件是像磁带、磁盘这样一些以块为单位存取数据的设备。字符特别文件是如像终端、打印机、调制解调器或者其他任何与系统通信时，一次传输一个字符的设备，包括模仿对磁盘进行字符方式存取的磁盘驱动器。主设备号指定了系统子程序（设备驱动程序），当在设备上执行I/O时，系统将调用这个驱动程序。调用设备驱动程序时，次设备号将传递给该驱动程序（次设备规定具体的磁盘驱动器，带驱动器，信号线编号，或磁盘分区）。每种类型的设备一般都有自己的设备驱动程序。

文件系统将主设备号和次设备号存放在 i节点中的磁盘地址表内，所以没有磁盘空间分配给设备文件（除i节点本身占用的磁盘区外）。当程序试图在设备文件上执行 I/O操作时，系统识别出该文件是一个特别文件，并调用由主设备号指定的设备驱动程序，次设备号作为调用设备驱动程序的参数。

23.3.4 安全考虑

将设备处理成文件，使得 Linux程序独立于设备，即程序不必一定要了解正使用的设备的任何特性，存取设备也不需要记录长度、块大小、传输速度、网络协议等这样一些信息，所有烦人的细节由设备驱动程序去关心考虑，要存取设备，程序只需打开设备文件，然后作为普通的Linux文件来使用。

从安全的观点来看这样处理很好，因为任何设备上进行的I/O操作只经过了少量的渠道（即设备文件），用户不能直接地存取设备。所以如果正确地设置了磁盘分区的存取许可，用户就只能通过Linux文件系统存取磁盘。文件系统有内部安全机制（文件许可）。不幸的是，如果磁盘分区设备得不正确，任何用户都能够写一个程序，读磁盘分区中的每个文件，作法很简单：读一个i节点，然后以磁盘地址表中块号出现的顺序，依次读这些块号指出的存有文件内容的块。故除了系统管理员以外，决不要使盘分区对任何人可写。因为所有者，文件存取许可方式这样一些信息存放于i节点中，任何人只要具有已安装分区的写许可，就能设置任何文件的 SUID许可，而不管文件的所有者是谁，也不必用chmod（）命令，还可绕过系统建立的安全检查。

以上所述对内存文件 mem、kmem和对换文件swap也是一样的。这些文件含有用户信息，一个耐心的程序可以将用户信息提取出来。

要避免磁盘分区（以及其他设备）可读可写，应当在建立设备文件前先用 umask命令设置文件建立屏蔽值。

一般情况下，Linux系统上的终端口对任何人都是可写的，从而使用户可以用 write命令发

送信息。虽然write命令易引起安全方面的问题，但大多数用户觉得用 write得到其他用户的信息很方便，所以系统将终端设备的存取许可设置成对所有用户可写。

/dev目录应当是755存取许可方式，且属系统管理员所有。

不允许除系统管理员外的任何用户读或写盘分区的原则有一例外，即一些程序（通常是数据库系统）要求对磁盘分区直接存取，解决这个问题的经验的盘分区应当由这种程序专用（不安装文件系统），而且应当告知使用这种程序的用户，文件安全保护将由程序自己而不是 Linux 文件系统完成。

23.3.5 find命令

find命令用于搜索目录树，并对目录树上的所有文件执行某种操作，参数是目录名表（指出从哪些起点开始搜索），还可给出一个或多个选项，规定对每个文件执行什么操作。

find / -print 将列出当前工作目录下的目录树的每一个文件。

find / -user bob -print 将列出在系统中可找到的属于bob用户的所有文件。

find /usr/bob -perm 666 -print 将列出/usr/bob目录树下所有存取许可为666的文件。若将666改为-666则将列出所有具有包含了666在内的存取许可方式的文件（如777）。

find /usr/bob -type b -print 将列出/usr/bob目录树下所有块特别文件（c为字符特别文件）。

find / -user root -perm -4000 -exec ls -l {} \; 是一个较复杂的命令，-exec COMMAND \;允许对所找到的每个文件运行指定的命令 COMMAND。若COMMAND中含有{}，则{}将由find所找到的文件名替换。COMMAND必须以\;结束。

以上举例介绍了find的用法，各选项可组合使用以达到更强的功能。

23.3.6 secure程序

系统管理员应当做一个程序以定期检查系统中的各个系统文件，包括检查设备文件和SUID和SGID程序，尤其要注意检查SUID和SGID程序，检查/etc/passwd和/etc/group文件，寻找久未登录的帐户和校验各重要文件是否被修改。（源程序清单将在今后发表。）

23.3.7 ncheck命令

用于检查文件系统，只用一个磁盘分区名作为参数，将列出 i节点号及相应的文件名。i节点相同的文件为建链文件。

注意 所列出的清单文件名与 mount命令的第一个域相同的文件名前部分将不会列出来。因为是做文件系统内部的检查，ncheck并不知道文件系统安装点以上部分的目录。也可用此命令来搜索文件系统中所有的SUID和SGID程序和设备文件，使用 -s选项来完成此项功能。

23.3.8 安装和拆卸文件系统

Linux文件系统是可安装的，这意味着每个文件系统可以连接到整个目录树的任意节点上（根目录总是被安装上的）。安装文件系统的目录称为安装点。

/etc/mount命令用于安装文件系统，用这条命令可将文件系统安装在现有目录结构的任意处。

安装文件系统时，安装点的文件和目录都是不可存取的，因此未安装文件系统时，不要将文件存入安装点目录。文件系统安装后，安装点的存取许可方式和所有者将改变为所安装的文

件根目录的许可方式和所有者。

安装文件系统时要小心：安装点的属性会改变！还要注意新建的文件，除非新文件系统是由标准文件建立的，系统标准文件会设置适当的存取许可方式，否则新文件系统的存取许可将是777!

可用-r选项将文件系统安装成只读文件系统。需要写保护的带驱动器和磁盘，应当以这种方式来安装。

不带任何参数的/etc/mount可获得系统中所安装的文件系统的有关信息。包括：文件系统被安装的安装点目录，对应/dev中的设备，只读或可读写，安装时间和日期等。从安全的观点来讲，可安装系统的危险来自用户可能请求系统管理员为其安装用户自己的文件系统。如果安装了用户的文件系统，则应在允许用户存取文件系统前，先扫描用户的文件系统，搜索SUID/SGID程序和设备文件。在除了系统管理员外任何人不能执行的目录中安装文件系统，用find命令或secure列出可疑文件，删除不属用户所有的文件的SUID/SGID许可。

用户的文件系统用完后，可用umount命令卸下文件系统。并将安装点目录的所有者改回系统管理员，存取许可改为755。

23.3.9 系统目录和文件

Linux系统中有许多文件和目录不允许用户写，如：/bin、/usr/bin、/usr/lbin、/etc/passwd、/usr/lib/crontab、/Linux、/etc/rc、/etc/inittab，可写的目录允许移动文件，这样会引起安全问题。

系统管理员应经常检查系统文件和目录的许可权限和所有者。可做一个程序根据系统提供的规则文件（在/etc/permlist文件中）所描述的文件所有者和许可权规则检查各文件。

注意 如果系统的安全管理不好，或系统是新安装的，其安全程序不够高， 可以用make方式在安全强的系统上运行上述程序，将许可规则文件拷贝到新系统来，再以设置方式在新系统上运行上述程序，就可提高本系统的安全程序。但要记住，两个系统必须运行相同的Linux系统版本。

23.4 作为root运行的程序

在Linux系统中，有些程序由系统作为 root进程运行。这些程序并不一定具有 SUID许可，因为其不少程序仅由root运行，系统管理员需要清楚这些程序做什么，以及这些程序还将运行其他什么程序。

23.4.1 启动系统

当某些Linux系统（如SCO Linux/XENIX）启动时，是以被称为单用户的方式运行的，在这种方式中普通用户不能登录，唯有的进程是init、swapper以及一些由系统管理员从控制台运行的进程。Linux系统的单用户方式启动，使系统管理员能在允许普通用户登录以前，先检查系统操作，确保系统一切正常，当系统处于单用户方式时，控制台作为超级用户，命令揭示是“#”，有些Linux系统不要确认超级用户口令就认可控制台是root，给出#提示符。这就可能成为一个安全问题。

23.4.2 init 进程

Linux系统总是以某种方式或称为某种级运行，系统有若干种运行级，这些运行级由 init进程控制。

Linux系统启动时以单用户方式运行，也叫1级或S级。对于其他用户登录进入系统，Linux有一种多用户运行方式，也叫2级。init进程控制系统运行级，它读入文件 /etc/inittab，该文件详细地规定了哪些进程在哪一级运行。当系统管理员敲入 init n（数字），系统就进入n级。init读该文件以确定终止哪些进程，启动哪些进程。

有效的运行级的数值是从0到6与s。

注意 由init建立的进程以UID为0运行（root），从/etc/inittab运行的程序也作为root运行，所以系统管理员要确保自己知道/etc/inittab中的程序做什么工作，确保这些程序以及这些程序所在的目录（直到/和/etc/inittab）除root外无人可写。

23.4.3 进入多用户

当Linux系统进入多用户方式时，将初始化一系列事件，接着开始执行 gettys，允许其他用户登录进入系统。如果再看看 /etc/inittab文件，会看到gettys定义在运行级2，至少三个外壳程序/etc/brc、/etc/bcheckrc、/etc/rc*也定义在运行级2。这些程序都在gettys启动前运行。

这些外壳程序作为root运行，也不能仅对root可写还应当检查外壳程序运行的命令，因为这些命令也将作为root运行。

23.4.4 shutdown命令

用shutdown命令关系统，shutdown 外壳程序发出警告，通知所有用户离开系统，在“给定的期限时间”到了后，就终止进程，拆卸文件系统，进入单用户方式或关机状态。一旦进入单用户方式，所有的 gettys停止运行，用户不能再登录。进入关机状态后可将系统关电。shutdown仅能由作为root登录的用户从系统控制台上运行。所以任何的 shutdown运行的命令仅能对root可写。

23.4.5 系统V的cron程序

cron在Linux系统是在多用户方式时运行的，根据规定的时间安排执行指定的命令，每隔一分钟检查一次文件 /usr/lib/crontab，寻找是否有应当运行的程序，如果找到要运行的程序，就运行该程序，否则睡眠等待一分钟。

实际的/usr/lib/crontab用于根据全天的规则时间表运行程序，也可在夜晚运行白天不愿运行怕降低其他用户速度的程序。通常由 cron运行的程序是如记帐、存文件这样的程序。 cron一般在系统进入多用户方式后由/etc/rc启动，当 shutdown运行killall命令时便终止运行。由cron运行的程序作为root，所以应当注意放什么程序在crontab中，还要确保/usr/lib/crontab和该表中列出的任何程序对任何人不可写。

如果用户需要由cron执行一个程序，系统管理员可用 su命令在crontab表中建立一个入口，使用户的程序不能获得root的权限。

23.4.6 系统V版本2之后的cron程序

在系统V版本2中，cron被修改成允许用户建立自己的crontab入口， /usr/lib/crontab文件不再存在，由目录/usr/spool/cron/crontabs中的文件代替。这些文件的格式与crontab相同，但每个文件与系统中的一个用户对应，并以某用户的名义由 cron运行。

如果想限制能建立crontab的用户，可在文件/usr/lib/cron/cron.allow文件中列出允许运行crontab命令的用户。任何未列于该文件的用户不能运行 crontab。反之，若更希望列出不允许运行crontab命令的用户，则可将他们列入/usr/lib/cron/cron.deny文件中，未列于该文件的其他用户将被允许建立crontab。

注意 若两个文件都存在，系统将使用cron.allow，忽略cron.deny。如果两个文件都不存在，则只有root可运行crontab。所以，若要允许系统中的所有用户都可运行crontab命令，应当建立一个空的cron.deny文件，如果cron.allow也存在，则删除该文件。

这个版本的cron命令的安全程度比前一个高，因为用户只能看自己的 crontab，系统管理员也不必担心其他用户的程序是否会作为root运行，由于允许每个系统登录用户有自己的crontab，也简化了对程序必须由cron运行，但不必作为root运行的系统程序的处理。

必须确保root的crontab文件仅对root可写，并且该文件所在的目录及所有的父目录也仅对root可写。

23.4.7 /etc/profile

每当用户（包括root在内）登录时，由外壳执行/etc/profile文件，应确保这个文件以及从这个文件运行的程序和命令都仅对root可写。

23.5 /etc/passwd文件

/etc/passwd文件是Linux安全的关键文件之一。该文件用于用户登录时校验用户的口令，当然应当仅对root可写。文件中每行的一般格式为：

LOGNAME：PASSWORD：UID：GID：USERINFO：HOME：SHELL

每行的头两项是登录名和加密后的口令，后面的两个数是 UID和GID，接着的一项是系统管理员想写入的有关该用户的信息，最后两项是两个路径名：一个是分配给用户的HOME目录，另一个是用户登录后将执行的外壳（若为空格则缺省为/bin/sh）。

23.5.1 口令时效

/etc/passwd文件的格式使系统管理员能要求用户定期地改变他们的口令。在口令文件中可以看到，有些加密后的口令有逗号，逗号后有几个字符和一个冒号。如： steve：xyDfcc Trt180x，My 8：0：0：admin：/：/bin/sh

restrict：pomJk109Jky41，1：0：0：admin：/：/bin/sh

pat：xmotTVoyumjls：0：0：admin：/：/bin/sh

可以看到，steve的口令逗号后有4个字符，restrict有2个，pat没有逗号。逗号后第一个字符是口令有效期的最大周数，第二个字符决定了用户再次修改口令之前，原口令应使用的最小周数（这就防止了用户改了新口令后立刻又改回成老口令）。其余字符表明口令最新修改时间。

要能读懂口令中逗号后的信息，必须首先知道如何用 passwd_esc计数，计数的方法是：.=0 /=1 0-9=2-11 A-Z=12-37 a-z=38-63

系统管理员必须将前两个字符放进/etc/passwd文件，以要求用户定期地修改口令，另外两个字符当用户修改口令时，由passwd命令填入。

注意 若想让用户修改口令，可在最后一次口令被修改时，放两个“.”，则下一次用户登录时将被要求修改自己的口令。

有两种特殊情况：

• 最大周数（第一个字符）小于最小周数（第二个字符），则不允许用户修改口令，仅超级用户可以修改用户的口令。

• 第一个字符和第二个字符都是“ . ”，这时用户下次登录时被要求修改口令，修改口令后，passwd命令将“ . ”删除，此后再不会要求用户修改口令。

23.5.2 UID和GID

/etc/passwd中UID信息很重要，系统使用 UID而不是登录名区别用户。一般来说，用户的UID应当是独一无二的，其他用户不应当有相同的 UID数值。根据惯例，从0到99的UID保留用作系统用户的UID（root、bin、uucp等）。

如果在/etc/passwd文件中有两个不同的入口项有相同的 UID，则这两个用户对相互的文件具有相同的存取权限。

23.6 /etc/group文件

/etc/group文件含有关于小组的信息， /etc/passwd中的每个GID在本文件中应当有相应的入口项，入口项中列出了小组名和小组中的用户。这样可方便地了解每个小组的用户，否则必须根据GID在/etc/passwd文件中从头至尾地寻找同组用户。

/etc/group文件对小组的许可权限的控制并不是必要的，因为系统用 UID和GID（取自/etc/passwd）决定文件存取权限，即使/etc/group文件不存在于系统中，具有相同的 GID用户也可以小组的存取许可权限共享文件。

小组就像登录用户一样可以有口令。如果/etc/group文件入口项的第二个域为非空，则将被认为是加密口令，newgrp命令将要求用户给出口令，然后将口令加密，再与该域的加密口令比较。

给小组建立口令一般不是个好方法。第一，如果小组内共享文件，而某人猜中小组口令，则该组的所有用户的文件就可能泄密：其次，管理小组口令很费事，因为对于小组没有类似的passwd命令。可用/usr/lib/makekey生成一个口令写入/etc/group。

以下情况必须建立新组：

• 可能要增加新用户，该用户不属于任何一个现有的小组。

• 有的用户可能时常需要独自为一个小组。

• 有的用户可能有一个SGID程序，需要独自为一个小组。

• 有时可能要安装运行SGID的软件系统，该软件系统需要建立一个新组。

要增加一个新组，必须编辑该文件，为新组加一个入口项。

由于用户登录时，系统从 /etc/passwd文件中取GID，而不是从/etc/group中取GID，所以group文件和口令文件应当具有一致性。对于一个用户的小组， UID和GID应当是相同的。多用户小组的GID应当不同于任何用户的UID，一般为5位数，这样在查看/etc/passwd文件时，就可根据5位数据的GID识别多用户小组，这将减少增加新组和新用户时可能产生的混淆。

23.7 增加、删除和移走用户

23.7.1 增加用户

增加用户有三个过程：

• 在/etc/passwd文件中写入新用户的入口项。
• 为新登录用户建立一个HOME目录。
• 在/etc/group中为新用户增加一个入口项。

在/etc/passwd文件中写入新的入口项时，口令部分可先设置为 NOLOGIN，以免有人作为此新用户登录。在修改文件前，应使用命令 mkdir /etc/ptmp，以免他人同时修改此文件。新用户一般独立为一个新组，GID号与UID号相同（除非他要加入目前已存在的一个新组），UID号必须和其他人不同，HOME目录一般设置在/usr 或/home目录下，建立一个以用户登录名为名称的目录做为其主目录。

23.7.2 删除用户

删除用户与增加用户的工作正好相反，首先在 /etc/passwd和/etc/group文件中删除用户的入口项，然后删除用户的HOME目录和所有文件。

rm -r /usr/loginname 删除整个目录树。

如果用户在/usr/spool/cron/crontabs中有crontab文件，也应当删除。

23.7.3 将用户移到另一个系统

这是一个复杂的问题，不只是拷贝用户的文件和用户在 /etc/passwd文件中的入口项。首先一个问题是用户的UID和GID可能已经用于另一个系统，若是出现这种情况，必须给要移动的用户分配另外的 UID和GID，如果改变了用户的 UID和GID，则必须搜索该用户的全部文件，将文件的原UID和GID改成新的UID和GID。

用find命令可以完成这一修改：

```
find.-user olduid -exec chown newuid {} \;
find. -group oldgid -exec chgrp newgid {} \;
```

也许还要为用户移走其他一些文件：/usr/mail/user和/usr/spool/cron/crontabs/user。

如果用户从一个不是本系统管理员的系统移来，则应对该用户的目录结构运行程序来检查。一个不安全系统的用户，可能有与该用户其他文件存在一起的 SUID/SGID程序，而这个SUID/SGID程序属于另一个用户。在这种情况下，如果用cpio或tar命令将用户的目录结构拷贝到本系统，SUID/SGID程序也将会拷贝到本系统而没有任何警告信息。应当在允许用户使用新系统以前先删除这种文件的SUID/SGID许可。总之，始终坚持检查所移用户的文件会更安全些。也可以用su命令进入用户的帐户，再拷贝用户文件，这样文件的所有者就是该用户，而不是root。

23.8 安全检查

像find和secure这样的程序称为检查程序，它们搜索文件系统，寻找出 SUID/SGID文件、设备文件、任何人可写的系统文件、设有口令的登录用户、具有相同 UID/GID的用户等等。

23.8.1 记帐

Linux记帐软件包可用作安全检查工具，除最后登录时间的记录外，记帐系统还能保存全天运行的所有进程的完整记录，对于一个进程所存贮的信息包括UID、命令名、进程开始执行与结束的时间、CPU时间和实际消耗的时间、该进程是否是root进程，这将有助于系统管理员了解系统中的用户在干什么。acctcom命令可以列出一天的帐目表。标明系统中有多个记帐数据文件，记帐信息保存在文件/usr/adm/pacct*中，/usr/adm/pacct是当前记录文件，/usr/adm/pacctn是以前的记帐文件（n为整型数）。若有若干个记帐文件要查看，可在acctcom命令中指定文件名：

acctcom /usr/adm/pacct? /usr/adm/pacct要检查的一个问题是：在acctcom的输出中查找一个用户过多的登录过程，若有，则说明可能有人一遍遍地尝试登录，猜测口令，企图非法进入系统。此外，还应查看root进程，除了系统管理员用su命令从终端进入root，系统启动，系统停止时间，以及由init（通常init只启动getty，login，登录外壳），cron启动的进程和具有root SUID许可的命令外，不应当有任何root进程。由记帐系统也可获得有关每个用户的CPU利用率，运行的进程数等统计数据。

23.8.2 其他检查命令

1) du 报告在层次目录结构（当前工作目录或指定目录起）中各目录占用的磁盘块数。可用于检查用户对文件系统的使用情况。

2) df 报告整个文件系统当前的空间使用情况。可用于合理调整磁盘空间的使用和管理。

3) ps 检查当前系统中正在运行的所有进程。对于用了大量CPU时间的进程、同时运行了许多进程的用户、运行了很长时间但用了很少CPU时间的用户进程应当深入检查。还可以查出运行了一个无限循环的后台进程的用户，未注销帐户就关闭终端的用户（一般发生在直接连线的终端）。

4) who 可以告诉系统管理员系统中工作的进展情况等等许多信息，检查用户的登录时间，登录终端。

5) su 每当用户试图使用su命令进入系统用户时，命令将在/usr/adm/sulog文件中写一条信息，若该文件记录了大量试图用su进入root的无效操作信息，则表明可能有人企图破译root口令。

6) login 在一些系统中，login程序记录了无效的登录企图（若本系统的login程序不做这项工作而系统中有login源程序，则应修改login）。

每天总有少量的无效登录，若无效登录的次数突然增加了两倍，则表明可能有人企图通过猜测登录名和口令，非法进入系统。

这里最重要的一点是：系统管理员越熟悉自己的用户和用户的工作习惯，就越能快速发现系统中任何不寻常的事件，而不寻常的事件意味着系统已被人窃密。

23.8.3 安全检查程序的问题

以上的检查方法没有几个能防止诱骗。如find命令，如果碰到路径名长于256个字符的文件或含有多于200个文件的目录，将放弃处理该文件或目录，用户就有可能利用建立多层目录结构或大目录隐藏SUID程序，使其逃避检查（但find命令会给出一个错误信息，系统管理员应手工检查这些目录和文件）。也可用ncheck命令搜索文件系统，但它没有find命令指定搜索哪种文件的功能。

如果定期存取.profile文件，则检查久未登录用户的方法就不奏效了。而用户用 su命令时，除非用参数，否则su不读用户的.profile文件。

有三种方法可寻找久未登录的帐户：

- Linux记帐系统在文件/usr/adm/acct/sum/login中为每个用户保留了最后一次登录日期。用这个文件的好处是，该文件由系统维护，所以可完全肯定登录日期是准确的。缺点是必须在系统上运行记帐程序以更新 loginlog文件，如果在清晨（或午夜后）运行记帐程序，一天的登录日期可能就被清除了。
- /etc/passwd文件中的口令时效域将能告诉系统管理员，用户的口令是否过期了，若过期，则意味着自过期以来，帐户再未被用过。这一方法的好处在于系统记录了久未用的帐户，检查过程简单，且不需要记帐系统所需要的磁盘资源，缺点是也许系统管理员不想在系

统上设置口令时效，而且这一方法仅在口令的最大有效期（只有几周）才是准确的。

- 系统管理员可以写一个程序，每天（和重新引导系统时）扫描 /etc/wtmp，自己保留下用户最后登录时间记录，这一方法的好处是不需要记帐程序，并且时间准确，缺点是要自己写程序。

以上任何方法都可和/usr/adm/sulog文件结合起来，查出由login或su登录帐户的最后登录时间。如果有人存心破坏系统安全，第一件要做的事就是寻找检查程序。破坏者将修改检查程序，使其不能报告任何异常事件，也可能停止系统记帐，删除记帐文件，使系统管理员不能发现破坏者干了些什么。

23.8.4 系统泄密后怎么办

如果发现有人已经破坏了系统安全，这时系统管理员首先应做的是面对肇事用户。如果该用户所做的事不是蓄意的，而且公司没有关于“破坏安全”的规章，也未造成损失，则系统管理员只需清理系统，并留心该用户一段时间。如果该用户造成了某些损失，则应当报告有关人员，并且应尽可能地将系统恢复到原来的状态。

如果肇事者是非授权用户，那就得做最坏的假设了：肇事者已设法成为 root且本系统的文件和程序已经泄密了。系统管理员应当想法查出谁是肇事者，他造成了什么损坏，还应当对整个文件做一次全面的检查，并不只是检查 SUID和SGID，设备文件。如果系统安全被一个敌对的用户破坏了，应当采用下面的步骤：

- 关闭系统，然后重新引导，不要进入多用户方式，进入单用户方式。
- 安装含有本系统原始Linux版本的带和软盘。
- 将/bin、/usr/bin、/etc、/usr/lib中的文件拷贝到一个暂存目录中。
- 将暂存目录中所有文件的校验和（用原始版本的 sum程序拷贝做校验和，不要用 /bin中的sum程序做）与系统中所有对旧的文件的校验和进行比较，如果有任何差别，要查清差别产生的原因。如果两个校验和不同，是由于安装了新版本的程序，确认是否的确安装了新版本程序。如果不能找出校验和不同的原因，用暂存目录中的命令替换系统中的原有命令。
- 在确认系统中的命令还未被篡改之前，不要用系统中原命令。用暂存目录中的外壳，并将PATH设置为仅在暂存目录中搜索命令。
- 根据暂存目录中所有系统命令的存取许可，检查系统中所有命令的存取许可。
- 检查所有系统目录的存取许可，如果用了 perms，检查permlist文件是否被篡改过。
- 如果系统Linux（/Linux）的校验和不同于原版的校验和，并且系统管理员从未修改过核心，则应当认为，一个非法者“很能干”，从暂存缓冲区重新装入系统。系统管理员可以从逐步增加的文件系统备份中恢复用户的文件，但是在检查备份中的“有趣”文件之前，不能做文件恢复。
- 改变系统中的所有口令，通知用户他们的口令已改变，应找系统管理员得到新口令。
- 当用户来要新口令时，告诉用户发生了一次安全事故，让他们查看自己的文件和目录是否潜伏着危害（如SUID文件、特洛依木马、任何人可写的目录），并报告系统管理员任何异乎寻常的情况。
- 设法查清安全破坏是如何发生的。如果没有肇事者说明，这也许是不可能弄清的。如果能发现肇事者如何进入系统，设法堵住这个安全漏洞。

第一次安装Linux系统时，可以将外壳、sum命令、所有文件的校验和存放在安全的介质上（磁带、软盘、硬盘和任何可以卸下并锁起来的介质）。这样不必再从原版系统带上重新装入文件，

可以安装备份介质，装入外壳和sum，将存在磁带上的校验和与系统中文件的校验和进行比较。系统管理员也许想自己写一个计算校验和的程序，破坏者将不能知道该程序的算法，如果将该程序及校验和保存在磁带上，这一方法的保密问题就减小到一个物理安全问题，即只需将磁带锁起来。

23.9 加限制的环境

23.9.1 加限制的外壳

这种外壳几乎与普通的外壳相同，但是该外壳能限制一个用户的能力，不允许用户有某些标准外壳所允许的行为：

- 不能改变工作目录（cd）。
- 不能改变PATH或SHELL 外壳变量。
- 不能使用含有“ / ”的命令名。
- 不能重定向输出（>和>>）。
- 不能用exec执行程序。

用户在登录时，在配置文件后系统就强加上了这些限制，如果用户在 .profile文件正被解释时按了Break键或Delete键，该用户将被注销。

这些简单的限制，使用写受限制用户的 .profile文件的系统管理员可以对用户能使用什么命令，进行完全的控制。

应当注意：系统V加限制的外壳实际上不很安全，在有敌对的用户时不要用。系统 V版本2以后的版本中加限制的外壳更安全些。但若允许受限制的用户使用某些命令（如 env、cp、ln），用户将能绕过加限制的外壳，进入非限制的外壳。

23.9.2 用chroot（）限制用户

如果的确想限制一个用户，可用 chroot（）子程序为用户建立一个完全隔离的环境。这个子程序改变了进程对根目录的概念，因此可用于将一个用户封在整个文件系统的某一层目录结构中，使用户无法用cd命令转出该层目录结构，不能存取文件系统中其他部分的任何文件。这种限制方式比加限制的外壳好得多。用户使用的命令应由系统管理员在新的 root目录中建立一个bin目录，并建立用户可用命令的链到系统的 /bin目录中相应命令文件上（若在不同的文件系统则应拷贝命令文件）。

还应建立新的passwd文件，保留系统登录帐户（为了使 ls -l正确地报告与受限制的子文件系统中的文件相关的正确登录名）和用户帐户，但系统帐户的口令改为 NOLOGIN以使受限制的用户不能取得系统登录的真实口令，使“ 破密 ”程序的任何企图成为泡影。 utmp文件是who所需要的，该文件含有系统中已登录用户的列表。

新的/etc/profile文件也不是建链文件，以便受限制的用户可以执行不同的启动命令。

/dev目录中的终端设备文件被链接到新的 /dev目录下，因为命令 who产生输出时要查看这些文件。在系统V及以后的Linux版本中，login命令有chroot（）的功能。如果口令文件中用户入口项的登录外壳域（最后一个域）是 *，login将调用chroot（）把用户的根目录设置成为口令文件中用户入口项登录目录域指定的目录。然后再调用 exec（）执行login，新的login将在新子系统文件中执行该用户的登录。

chroot（）并不是把root封锁在一个子文件系统中，所以给受限制用户用的命令时应加以考虑，具有root的SUID许可的程序可能会给予用户 root的能力。应当将这种可能减低到最小程

度，交给用户使用的命令应当取自清除了SUID陷井的系统命令。链接文件可减少磁盘占用区，但要记住，当与敌对用户打交道时，链接到chroot目录结构（尤其是命令）的系统文件是很危险的。如果建立一个像这样的限制环境，应确保对安装到新的 /bin的每条命令都做过测试，有些程序可能有系统管理员未曾想到的出乎意料的执行结果。为了使这些命令能运行，还得在加限制的子文件系统中加服务目录或文件，如： /tmp、/etc/termcap、/dev/swap、/dev/mem、/usr/lib/terminfo、/dev/kmem，用户所登录的/dev中的tty文件以及/Linux。

有些程序在子文件系统中运行时不会很好，如果将假脱机程序和网络命令拷贝到加限制的子文件系统中，并放在为两条命令专建的目录层结构下，它们可能也运行不了。

23.10 小系统安全

任何足够小的运行于办公室的Linux系统就是小系统，这类小系统也包括所有台式 Linux机器。根据安全观点，小系统的特别之处有以下几点：

- 小系统的用户比大系统的用户少，通常是很小一组用户，使系统管理员能熟悉每个人，安全问题可以直接地面对面处理。
- 由于小Linux系统管理更简单，可能只需要一个系统管理员，因而维护系统安全的责任只有一个人担负。
- 如果既是用户又是系统管理员，将不能花大量时间考虑系统安全。
- 如果自己拥有系统并且是系统管理员，就可能有权直接将违反规定的用户从系统中删除，而一般大系统的管理员没有这种权利。
- 如果自己是系统的唯一用户，则既是用户又是管理员，维护系统安全的任务就很简单了，只须确保系统中所有登录帐户的口令是好的即可。
- 如果不能将系统锁起来，就把重要的数据存放在软盘上，把软盘锁起来。
- 即使系统中有若干个用户，但如果系统的终端之间是有线连接，并且用户们保持门上锁，则系统也将是安全的，至少在本组用户内是安全的。
- 小系统通常有可移动的介质（软盘），可用mount命令将其安装到系统上，提供一种安全的方法让用户自己在系统上安装软盘，否则系统管理员要一天到晚地干这些琐碎的安装盘事务。允许用户安装软盘的通常做法是给用户一个 SUID程序，该程序与系统管理员安装用户软盘基本完成同样的操作，首先检查软盘上有无 SUID/SGID/设备文件，若发现任何奇怪的文件，则拒绝安装该软盘。
- 当小系统开电源后，系统一般在从硬盘引导以前，先试图从软盘引导。这就意味着计算机将首先试图从软盘装入程序，若软盘不在驱动器中，系统将从硬盘装入 Linux内核。软盘几乎可以含有任何程序，包括在控制台启动 root 外壳的Linux系统版本。如果破坏者有一把螺丝起子和有关系统内部的一些知识，则即便系统有被认为防止安全事故发生的特殊“ 微码 ”口令，也可能被诱骗去从软盘引导。
- 即使小系统晚上不锁，凡从不将个人的或秘密的信息存放在大系统上的人（他们不可能认识所有系统上的用户），也不会想把这样的信息存放在小系统上。
- 小系统的系统管理员在使用Linux系统方面常不如大系统管理员有经验，而安全地管理系统需要一定的使用系统的知识。

23.11 物理安全

对于运行任何操作系统的小型或大型计算机，物理安全都是一个要考虑的重要问题，物理

安全包括：锁上放置计算机的屋子、报警系统、警卫、所有安置在不能上锁的地方的通信设施（包括有线通信线、电话线、局域网、远程网、应答调制解调器）、钥匙或信用卡识别设备、给用户的口令和钥匙分配、任何前置通信设施的加密装置、文件保护、备份或恢复方案（称为安全保险方案，用作应付偶然的或蓄意的数据或计算设备被破坏的情况）、上锁的输出端、上锁的废物箱和碎纸机。物理安全中的总考虑应是：在安全方案上所付出的代价不应当多于值得保护的（硬件或软件的）价值。

下面着重讨论保护用户的各种通信线。对于任何可在不上锁的地方存取的系统，通信是特别严重的安全薄弱环节。当允许用户通过挂到地方电话公司的拨号调制解调器存取系统时，系统的安全程度就将大大地削弱，有电话和调制解调器的任何人就有可能非法进入该系统。应当避免这一情况，要确保调制解调器的电话号码不被列于电话簿上，并且最好将电话号码放在不同于本公司普通电话号码所在的交换机上。总之，不要假设没人知道自己的拨入号码 !大多数家庭计算机都能编程用一个调制解调器整天地依次调用拨号码，记录下连接上其他调制解调器的号码。如果可能，安装一个局域PBX，使得对外界的拨号产生一秒钟的拨号蜂音，并且必须输入一个与调制解调器相关联的扩展号码。

23.12 用户意识

Linux系统管理员的职责之一是保证用户安全。这其中一部分工作由用户的管理部门来完成，但是作为系统管理员，有责任发现和报告系统的安全问题，因为系统管理员负责系统的运行。

避免系统安全事故的方法是预防性的，当用户登录时，其外壳在给出提示前先执行/etc/profile文件，要确保该文件中的PATH指定最后搜索当前工作目录，这样将减少用户能运行特洛依木马的机会。将文件建立屏蔽值的设置放在该文件中也是很合适的，可将其值设置成至少将防止用户无意中建立任何人都能写的文件（ 022/026）。要小心选择此值，如果限制太严，则用户会在自己的配置文件中重新调用 umask以抵制系统管理员的意愿，如果用户大量使用小组权限共享文件，系统管理员就要设置限制小组存取权限的屏蔽值。系统管理员必须建立系统安全和用户的“ 痛苦量 ”间的平衡（痛苦量是安全限制引起的愤怒的函数）。定期地用grep命令查看用户配置文件中的umask，可了解系统安全限制是否超过了用户“ 痛苦量 ”极限。

系统管理员可每星期随机抽选一个用户，将该用户的安全检查结果（用户的登录情况简报，SUID/SGID文件列表等）发送给他的管理部门和他本人。主要有四个目的：

- 大多数用户会收到至少有一个文件检查情况的邮件，这将引起用户考虑安全问题（虽然并不意味着用户们会采取加强安全的行动）。
- 有大量可写文件的用户，将一星期得到一次邮件，直到他们取消可写文件的写许可为止。冗长的烦人的邮件信息也许足以促使这些用户采取措施，删除文件的写许可。
- 邮件将列出用户的SUID程序，引起用户注意自己有 SUID程序，使用户知道是否有不是自己建立的SUID程序。
- 送安全检查表可供用户管理自己的文件，并使用户知道对文件的管理关系到数据安全。如果系统管理员打算这样做，应事先让用户知道，以便他们了解安全检查邮件的目的。发送邮件是让用户具有安全意识，不要抱怨发送邮件。

管理意识是提高安全性的另一个重要因素。如果用户的管理部门对安全要求不强烈，系统管理员可能也忘记强化安全规则。最好让管理部门建立一套每个人都必须遵守的安全标准，如果系统管理员在此基础上再建立自己的安全规则，就强化了安全。管理有助于加强用户意识，让用户明确，信息是有价值的资产。

系统管理员应当使安全保护方法对用户尽可能地简单，提供一些提高安全的工具，如：公布锁终端的lock程序，让用户自己运行secure程序，将pwexp（检查用户口令信息的程序）放入/etc/profile中，使用户知道自己的口令时间。多教给用户一些关于系统安全的知识，确保用户知道自己的许可权限和umask命令的设置值。如果注意到用户在做蠢事，就给他们一些应当怎样做才对的提示。用户知道的关于安全的知识越多，系统管理员在保护用户利益方面做的事就越少。

23.13 系统管理员意识

23.13.1 保持系统管理员个人的登录安全

若系统管理员的登录口令泄密了，则窃密者离窃取 root只有一步之遥了，因为系统管理员经常作为root运行，窃密者非法进入到系统管理员的帐户后，将用特洛依木马替换系统管理员的某些程序，系统管理员将作为root运行这些已被替换的程序。正是因为这个原因，在Linux系统中，管理员的帐户最常受到攻击。即使su命令通常要在任何都不可读的文件中记录所有想成为root的企图，还可用记帐数据或ps命令识别运行su命令的用户。正因为如此，系统管理员作为root运行程序时应当特别小心，因为最微小的疏忽也可能“沉船”。下列一些指导规则可使系统管理员驾驶一艘“坚固的船”：

- 不要作为root或以自己的登录帐户运行其他用户的程序，首先用su命令进入用户的帐户。
- 决不要把当前工作目录排在PATH路径表的前边，那样实际是招引特洛依木马。当系统管理员用su命令进入root时，他的PATH将会改变，就让PATH保持这样，以避免特洛依木马的侵入。
- 敲入/bin/su执行su命令。若有su源码，将其改成必须用全路径名运行（即su要确认argv[0]的头一个字符是“/”才运行）。随着时间的推移，用户和管理员将养成敲/bin/su的习惯。
- 不要未注销帐户就离开终端，特别是作为root用户时更不能这样。当系统管理员作为root用户时，命令提示符是“#”，这个提示符对某些人来说可能是个红灯标志。
- 不允许root在除控制台外的任何终端登录（这是login的编译时的选项），如果没有login源码，就将登录名root改成别的名，使破坏者不能在root登录名下猜测各种可能的口令，从而非法进入root的帐户。
- 经常改变root的口令。
- 确认su命令记下的想运行su企图的记录/usr/adm/sulog，该记录文件的许可方式是600，并属root所有。这是非法者喜欢选择来替换成特洛依木马的文件。
- 不要让某人作为root运行，即使是几分钟，即使是系统管理员在一旁注视着也不行！

23.13.2 保持系统安全

- 考虑系统中一些关键的薄弱环节：
- 系统是否有调制解调器?电话号码是否公布？
- 系统是否连接到网络？还有什么系统也连接到该网络？
- 系统管理员是否使用未知来处或来处不可靠的程序？
- 系统管理员是否将重要信息放在系统中？
- 系统的用户是熟悉系统的使用还是新手？
- 用户是否很重视关心安全?
- 用户的管理部门是否重视安全？

- 保持系统文件安全的完整性。检查所有系统文件的存取许可，任何具有 SUID许可的程序都是非法者想偷换的选择对象。
- 要特别注意设备文件的存取许可。
- 要审查用户目录中具有系统ID/系统小组的SUID/SGID许可的文件。
- 在未检查用户的文件系统的SUID/SGID程序和设备文件之前，不要安装用户的文件系统。
- 将磁盘的备份存放在安全的地方。
- 设置口令时效，如果能存取Linux的源码，将加密口令和信息移到仅对root可读的文件中，并修改系统的口令处理子程序。这样可增加口令的安全。修改 passwd，使passwd能删去口令打头和末尾的数字，然后根据spell词典和/etc/passwd中用户的个人信息，检查用户的新口令，也检查用户新口令中子串等于登录名的情况。如果新口令是 spell词典中的单词，或/etc/passwd中的入口项的某项值，或是登录名的子串，passwd将不允许用户改变口令。
- 记录本系统的用户及其授权使用的系统。
- 查出久未使用的登录帐户，并取消该帐户。
- 确保没有无口令的登录帐户。
- 启动记帐系统。
- 查出不寻常的系统使用情况，如大量地占用磁盘，大量地使用 CPU时间，大量的进程，大量使用su的企图，大量无效地登录，大量的到某一系统的网络传输，奇怪的 uucp请求。
- 修改外壳，使其等待了一定时间而无任务时终止运行。
- 修改login，使其打印出用户登录的最后时间，三次无效登录后，将通信线挂起，以便系统管理员能检查出是否有人试图非法进入系统。确保 login不让root在除控制台外的任何地方登录。
- 修改su，使得只有root能以过期口令通过su进入某一帐户。
- 当安装来源不可靠的软件时，要检查源码和makefile文件，查看特殊的子程序调用或命令。
- 即使是安装来源可靠的软件，也要检查是否有 SUID（SGID）程序，确认这些许可的确是必要的。如果可能，不要让这些程序具有系统 ID（或组）的SUID（SGID）许可，而应该建立一个新用户（或给）供该软件运行。
- 如果系统在办公室中，门应上锁，将重要数据保存在软盘上或磁带上，并锁起来。
- 将secure、perms和任何其他做安全检查的外壳程序存取许可置为仅执行，更好的办法是将这些外壳程序存于可拆卸的介质上。
- 记住，只要系统有任何人都可调用的拨号线，系统就不可能真正的安全。系统管理员可以很好地防止系统受到偶然的破坏。但是那些有耐心、有计划、知道自己在干什么的破坏者，对系统直接的有预谋的攻击却常常能成功。
- 如果系统管理员认为系统已经泄密，则应当设法查出肇事者。若肇事者是本系统的用户，与用户的管理部门联系，并检查该用户的文件，查找任何可疑的文件，然后对该用户的登录小心地监督几个星期。如果肇事者不是本系统的用户，可让本公司采取合法的措施，并要求所有的用户改变口令，让用户知道出了安全事故，用户们应当检查自己的文件是否有被篡改的迹象。

如果系统管理员认为系统软件已被更改了，就应当从原版系统带（或 ;软盘）上重装入所有系统软件，保持系统安全比道歉更好。