第 5 章

Linux 安全机制

5.1 Linux 安全机制概述

Linux 是 UNIX 克隆(UNIX clone)或 UNIX 风格(UNIX alike)的操作系统,在源代码级上兼容绝大部分 UNIX 标准,是一个支持多用户、多进程、多线程且实时性较好的功能强大而稳定的操作系统。

5.1.1 用户标识和鉴别

Linux 中权限最高的是超级用户 root,其在系统中的地位类似于 Windows NT 的管理员 Administrator 用户,可以执行任何操作,管理一切资源,root 用户一般在安装系统时创建。其他用户通常是在系统安装完成后由 root 用户创建,这类用户只能访问和管理有限资源,也称受限用户。

在系统内部具体实现中,系统会为每个用户分配一个唯一的标识号 UID(User ID),UID 是一个数值,比如 root 用户的 UID 为 0,普通用户的 UID 通常从 1000 开始顺序编号,小于 1000 的 UID 是系统保留用以启动某些服务。具有相似属性的多个用户可以分配到同一组内,用组标识符(Group ID,GID)来唯一标识,每个用户可以属于一个或多个用户组。用户号(UID)和用户组号(GID)决定了用户的访问权限。

与用户相关的信息存储在系统中的/etc/passwd 文件中,这个文件的拥有者是 root 用户,可以读写该文件,而普通用户只有读的权利。

```
#ls -l /etc/passwd
# -rw-r--r-- root root
```

那么,在/etc/passwd 文件中具体包含哪些内容呢? 使用 cat 命令查看 passwd 文件:

```
#cat /etc/passwd
```

如图 5.1 所示,该文件是一个典型的数据库文件,每一行都由 7 部分组成,每两部分之间用冒号分隔开,这 7 部分分别描述了以下信息:用户名、口令、用户 ID、组 ID、用户描述、用户主目录、用户的登录 Shell。

其中用户描述部分用于描述用户信息,如用户全名,办公电话等;主目录是用户登录后进入的家目录的绝对路径,是用户存放个人文件和配置文件的默认位置,普通用户的家目录通常位于/home 下;登录 shell 是用户登录成功后默认启动的命令行解释器的绝对路径。

为了防止口令被非授权用户盗用,对口令的设置应以复杂、不可猜测为标准。一个好的

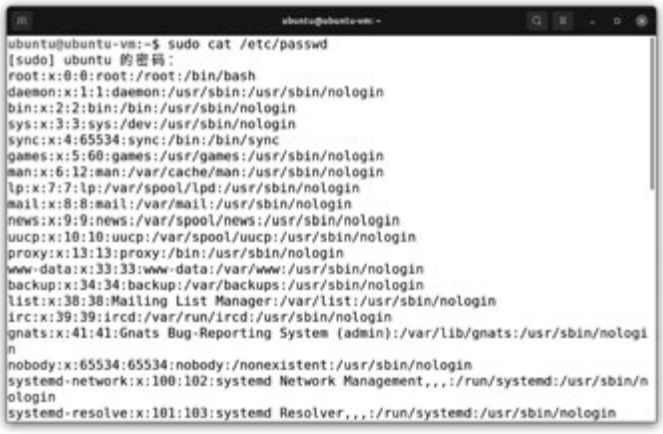

图 5.1　/etc/passwd 文件内容

口令应该满足长度和复杂度要求，并且定期更换，通常，口令以加密的形式表示，由于/etc/passwd 对任何用户可读，故常成为口令攻击的目标，在后期的 UNIX 版本以及所有 Linux 版本中，引入了影子文件的概念，将密码单独存放在/etc/shadow 中，而原来/etc/passwd 文件中存放口令的域用 x 来标记。文件/etc/shadow 只对 root 用户拥有读权，对普通用户不可读，以进一步增强口令的安全性。

/etc/shadow 每一行记录包含 9 个字段，用分号隔开，如图 5.2 所示，分别描述用户名、加密后的口令信息、口令的有效期等信息。

```
#cat /etc/shadow
```

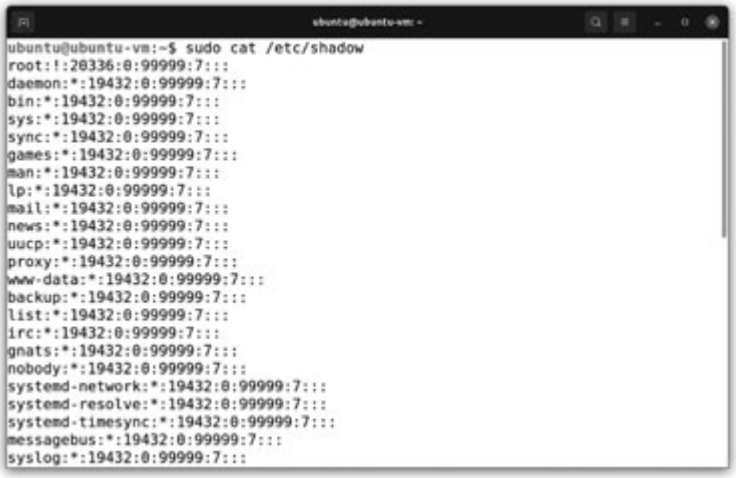

图 5.2　/etc/shadow 文件内容

5.1.2　访问控制

Linux 系统的访问控制是基于文件的，在 Linux 系统中各种硬件设备、端口甚至内存都是以设备文件的形式存在的，虽然这些文件和普通文件在实现上是不同的，但它们对外提供

的界面是一样的,这样就给 Linux 系统资源的访问控制带来了实现上的方便。

Linux 提供的访问控制机制为自主访问控制,早期的 Linux 将用户进行分组授权,一般分为属主用户、同组用户和其他用户三类,这种访问控制的粒度比较粗,无法实现对单个用户的授权。目前的 Linux 在支持粗粒度自主访问控制的基础上,利用访问控制列表(Access Control List,ACL)机制,能够实现对单个用户的授权。

1. 访问权限

命令 ls 可列出文件(或目录)对系统内不同用户所给予的访问权限,如:

```
-rw-r--r--  1  root root 1397  Mar  7 10:20 passwd
```

图 5.3 给出了文件访问权限的图示解释。

访问权限位共有 9 位,分为三组,用以指出不同类型的用户对该文件的访问权限。

权限有以下三种。

(1) r:允许读。

(2) w:允许写。

(3) x:允许执行。

用户有以下三种类型。

(1) owner:文件的属主,表示文件是由该用户创建的。

(2) group:与该文件属主同组的用户,即同组用户。

(3) other:除以上两者外的其他用户。

图 5.3 表示文件的属主具有读、写及执行权限(rwx),同组用户允许读和执行操作(rx),其他用户没有任何权限。在权限位中,-表示相应的访问权限位不允许。为操作方便,可以通过数字表示法对文件权限进行描述,这种方法将每类用户的权限看作一个 3 位的二进制数值,具有权限的位置用 1 表示,没有权限的位置用 0 表示,图 5.3 中的 rwxr-x---用数字表示为 750。

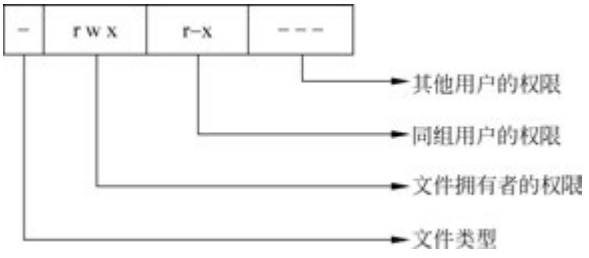

图 5.3 Linux 自主访问控制

上述授权模式同样适用于目录,目录的文件类型为 d。目录的读权限是指用列出目录中内容的权限;写权限是指在目录中增删文件的权限;执行权限是指进入目录或将该目录作路径分量的权限。因此要使用任一文件,必须要有搜索该文件所在路径上所有目录分量的权限,仅当要打开一个文件时,文件的权限才开始起作用,而 rm、mv 只要有目录的搜索和写权限,并不需要有文件的权限,这一点应尤为注意。

root 用户对任何文件或目录均可进行任何操作,具有最高权力,这样方便了管理员对系统的管理,但同时也是一个潜在的安全隐患,对于 root 账户的使用,需要注意以下几点。

(1) 除非必要,尽量避免以 root 用户身份登录。

(2) 不要随意将 root Shell 留在终端。

(3) 不要以 root 身份运行其他用户的或不熟悉的程序。

2. 改变权限

利用 chmod 命令可以改变文件或目录的访问权限,格式为:

```
chmod [选项] 访问权限 文件/目录
```

其中选项为可选参数,用于控制权限更改的行为,常用选项如下。

-R 或--recursive:递归地更改权限。当目标是一个目录时,使用此选项可以同时更改该目录及其内部所有子目录和文件的权限。如 chmod -R 755 /shared/data/命令将/shared/data/及其下所有内容的权限统一设置为 755。

-v 或--verbose:详细模式。详细说明权限改变的情况。

访问权限有两种表示方法:符号表示法或者数字表示法。

(1) 符号表示法。

利用符号表示法设置权限的命令格式为:

```
chmod [用户类别][操作符][权限] 文件/目录
```

其中用户类别中的 u 表示该文件的属主,g 表示与属主同组的用户,o 表示其他用户,a 表示所有用户;操作符中的+表示增加权限、-表示取消权限、=表示设置权限;权限类型中的 r 表示读权限,w 表示写权限,x 表示执行权限。例如命令 chmod a+r file.txt 为所有用户增加对文件 file.txt 的读权限,命令 chmod o-x ex.py 取消其他用户对文件 ex.py 的执行权限。

(2) 数字表示法。

Linux 权限的数字表示法使用三个八进制数分别对应文件属主、所属组和其他用户的权限。每个数字由读($r=4$)、写($w=2$)、执行($x=1$)三个权限的数值相加而得,这是一种高效且常用的权限设置方式。例如,命令 chmod 764 file.txt 将为属主设置读、写、执行权限(7=4+2+1),为所属组设置读、写权限(6=4+2),为其他用户设置只读权限(4)。

3. 特殊权限位

Linux 系统还包括 SUID、SGID 以及 Sticky 权限,用来表示文件的一些特殊性质。

1) SUID

有时用户需要完成只有拥有特定权限才能执行的任务,如对于普通用户,当通过/usr/bin/passwd 命令修改自己的口令时会涉及对/etc/passwd 文件的修改操作,而普通用户不拥有修改/etc/passwd 文件的权限,通过对可执行文件/usr/bin/passwd 设置 SUID(Set User ID)可以解决这个问题。利用 ls -l 命令可以看到文件属主的执行位(x)被替换为 s。

```
# ls -l /usr/bin/passwd
# -rwsr-xr-x 1 root root
```

/usr/bin/passwd 就是一个设置了 SUID 的程序,有时又称为 s 位程序,普通用户执行该程序时,将暂时拥有/usr/bin/passwd 这个可执行文件的属主 root 的权限,因此可以修改自己的口令。

用“chmod u+s 文件名”和“chmod u-s 文件名”来设置和取消 SUID 权限位。

SUID程序会使普通用户权限得到提升,从而给系统安全带来威胁,为了保证SUID程序的安全性,系统管理员应对系统中所有设置SUID的程序进行定期的检查和监视,严格限制功能范围,不能有违反安全性规则的SUID程序存在,并且要保证SUID程序自身不能被任意修改。

2) SGID

该属性既可作用在可执行文件上也可作用在目录上,当作用在可执行文件上时,它将使执行该文件的进程拥有同组用户的权限,但这个功能几乎不用。当SGID属性作用到目录时,可以用于设置该目录下创建的文件和子目录的默认组权限。

默认情况下,一个用户创建一个文件,用户的有效主组就设置为该文件的组属主,这种默认设置在有些情况下并不方便。想象这样的场合,用户linda和lori在会计部门工作,共享目录/account,为授权方便,将他们都设置为组account的成员。默认情况下,这些用户是以用户名命名的组的成员,两个用户又同时是account组的成员,但account组只能是这些用户的第二备选组。当一个用户在/account目录下创建了文件,用户的主组成为文件的组拥有者,但是如果为该/account目录设置了SGID权限,并且设置组account作为目录的主组,所有在该目录下创建的文件和子目录将组account作为默认的组拥有者,这样通过为account组设置文件操作权限,方便文件在同组用户之间共享。

可以利用"chmod g+s 文件名"和"chmod g-s 文件名"命令来设置和取消SGID权限位。

3) Sticky

该权限位的主要作用是当有多个用户对同一目录具有写权限时,为了防止某个用户的误操作而删除其他用户创建的文件。当为共享目录设置了sticky属性时,用户仅可在以下情况下删除文件。

(1) 用户是文件的属主。

(2) 用户是文件所在目录的属主。

5.1.3 审计

审计是Linux安全机制的重要组成部分,它通过对安全相关事件进行记录和分析,发现违反安全策略的活动,确保安全机制正确工作并能对系统异常及时报警提示。审计记录常写在系统的日志文件中,丰富的日志为Linux的安全运行提供了保障。常见的日志文件与说明如表5.1所示。

表5.1 日志文件与说明

日志文件	说明
acct或pacct	记录每个用户使用过的命令
aculog	筛选出modems(自动呼叫部件)记录
lastlog	记录用户最后几次成功登录的事件和最后一次登录失败的事件
loginlog	记录不良的登录尝试记录
messages	记录输出到系统主控台及由syslog系统服务程序产生的信息
sulog	记录su命令的使用情况

续表

日 志 文 件	说　　明
utmp	记录当前登录的每个用户
wtmp	记录每一次用户登录和注销的历史信息，以及系统关和开
xferlog	记录 ftp 的访问情况

Linux 系统中传统的审计机制是 Syslogd 和 Klogd。Syslog 是一个应用层的审计机制，允许应用程序将审计信息传递给系统日志守护程序 Syslogd，由 Syslogd 根据配置文件(/etc/syslogd.conf)将收到的信息按类型作相应处理，写入不同的日志文件，如图 5.4 所示。另外，也允许内核消息守护进程 Klogd 将内核中通过 Printk 打印出的消息写入日志文件。

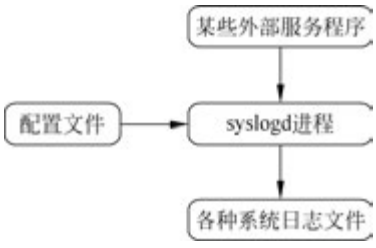

图 5.4　审计机制

Linux 传统的审计方式具有很大的局限性。首先，它不能提供系统级审计记录。Syslogd 只能接收由应用程序产生的日志信息，而 Klogd 只能接收内核中由 Printk 打印出的消息，这两种方式审计信息量获取有限，审计记录不够详细。其次，就应用层审计而言，Syslog 也是有局限性的。Syslog 产生的审计信息完全依赖应用程序。如果入侵者熟悉 Syslog 的工作方式，就可以模仿与某个应用程序相同的方式写入日志，伪造出虚假的审计数据。一旦某个收集审计数据的外部服务程序被恶意用户杀掉后，由该服务程序所收集的某类审计记录就不会产生，这样审计系统也就达不到记录所有安全相关系统活动的目的了。

5.2　Linux 标识和鉴别实验

5.2.1　实验目的

掌握用户账户安全设置方法；掌握账户锁定策略、账户注销策略的设置方法。

5.2.2　实验任务

对 Linux 系统进行账户管理、设置账户锁定策略、设置账户注销策略等基本安全配置。

5.2.3　实验环境

主流配置计算机一台，安装 Ubuntu 20.04。

5.2.4　实验任务分析及实验步骤

【任务分析】

在 Linux 安装完成后，需要对用户账户进行适当的配置以提高系统的安全性，主要涉及清除多余账户、清除多余用户组、锁定系统伪账户、检查是否存在空口令账户、对 root 账户进行保护等；当前 Linux 采用的最常用的身份认证方式是口令认证，为防范对于口令的字

典攻击和暴力破解,需要对用户鉴别机制进行相应的安全设置,包括设置账户锁定策略、设置账户注销策略等。

【实验步骤】

1. 清除多余账户

账户是黑客入侵系统的突破口,系统的账户越多,黑客们得到合法用户权限的可能性一般也就越大。我们需要定期查看账户、口令文件,与系统管理员确认后删除一些不必要的账户。删除用户的命令为 userdel,利用该命令删除 sync、news、uucp、games 等账户,这些账户为系统默认创建但很少使用的账户,具体用法如图 5.5 所示。

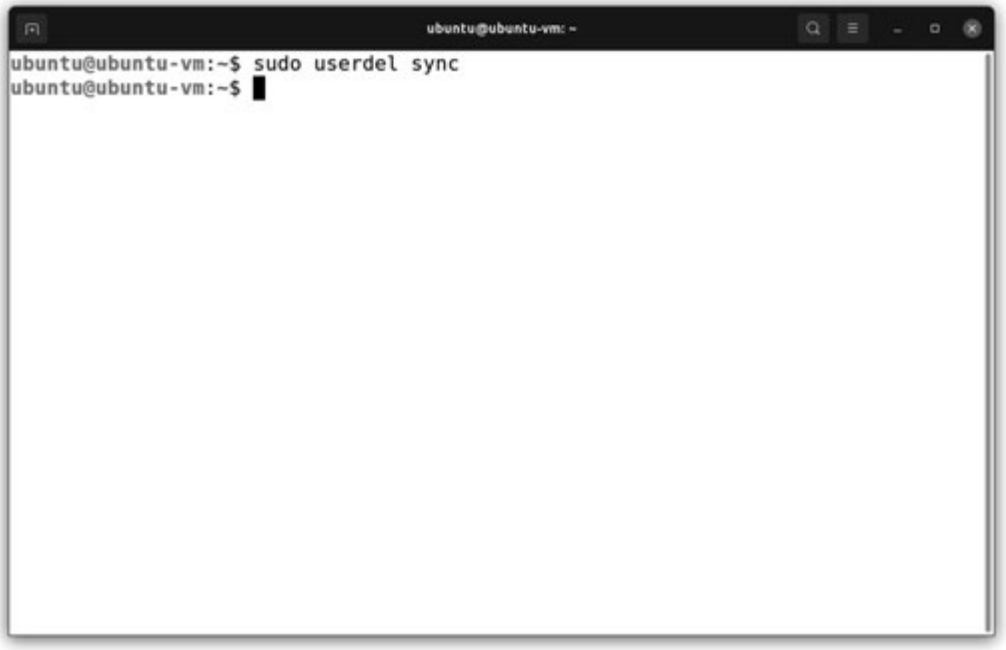

图 5.5 删除用户

删除用户通常由管理员账户 root 执行,在 Ubuntu 20.04 中,root 用户默认没有启用,可以参考相应的资料查看如何开启 root 用户,也可通过 sudo 命令临时以 root 用户的权限执行部分操作,这种方式更符合安全策略的要求。

2. 清除多余的组

同样,应该删除系统安装时默认创建但很少使用的组账号,如 adm、dip 等以减少系统受攻击的风险,删除组用命令 groupdel,其用法如图 5.6 所示。

3. 锁定账户登录

如果某些账户一段时间内不用,为了防止被恶意用户利用,可以锁定这些账户以禁止其登录,在需要时可以解锁。锁定账户的命令为 passwd -l <用户名>;解锁账户的命令为 passwd -u <用户名>,可以用 passwd -S 查看用户状态,如图 5.7 所示。

4. 禁用 root 之外的超级用户

执行命令:#cat /etc/passwd

图 5.6　删除用户组

图 5.7　锁定用户

检查账户文件中的用户 ID，若用户 ID＝0，则表示该用户拥有超级用户权限，禁用除 root 之外的超级用户。

5. 保护 root 账户的安全策略

由于 root 账户具有最高权限，一旦被攻击者获取 root 权限，将对系统造成巨大的危害，

用户需要具备以下对 root 账户的保护意识。

(1) 除非必要,避免以超级用户登录。

(2) 严格限制 root 只能在某一终端登录,远程用户可以使用/bin/su -l 来成为 root。

(3) 不要随意把 root shell 留在终端。

(4) 若普通用户确实需要以 root 来运行命令,则考虑安装 sudo 这样的工具,它能使普通用户以 root 身份来运行个人命令并维护日志。

(5) 不要把当前目录(".")和普通用户的 bin 目录放在 root 账户的环境变量 PATH 中。

(6) 不以 root 运行其他用户或不熟悉的程序。

6. 用户口令策略的设置

为了防范字典攻击和暴力破解,用户应该设置强口令,为了强制用户设置安全口令,Linux 系统提供了类似 Windows 的密码策略,在配置文件/etc/login.defs 中进行设置,如图 5.8 所示,口令策略涉及如下参数。

```
PASS_MAX_DAYS 99999      ##密码设置最长有效期(默认值)
PASS_MIN_DAYS 0          ##密码设置最短有效期
PASS_WARN_AGE 7          ##提前多少天警告用户密码即将过期
```

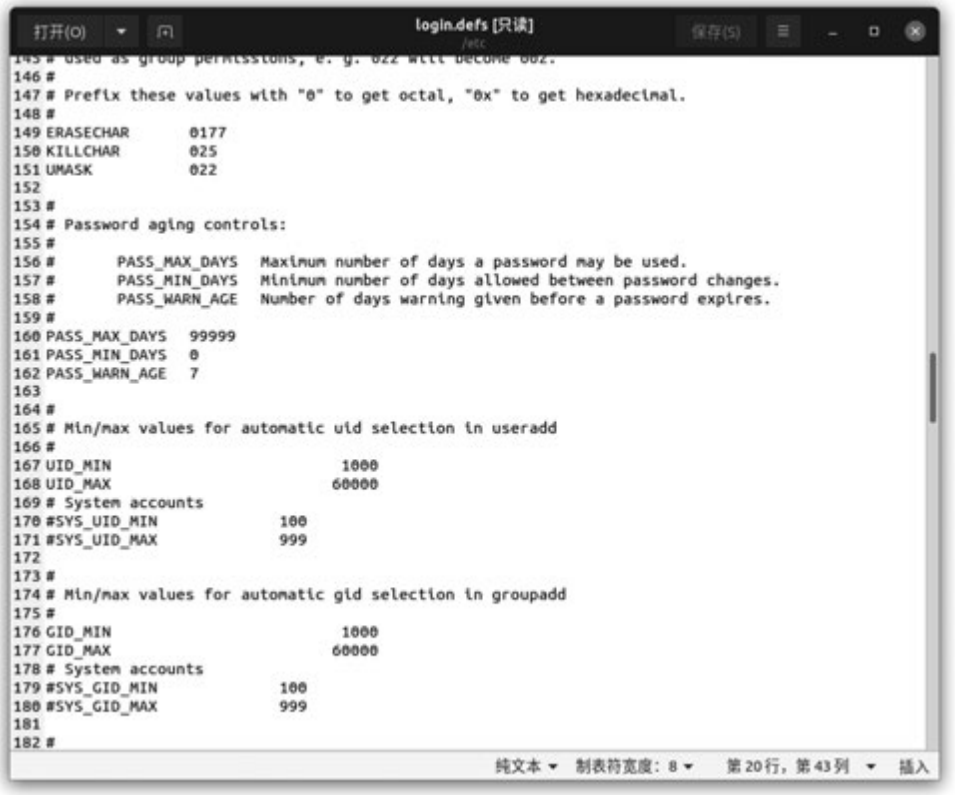

图 5.8 设置密码策略

其中,口令的最小长度设置可以在/etc/login.defs 文件中通过设置参数 PASS_MIN_

LEN 进行限制，也可以利用 Linux PAM 机制，下载 libpam_pwquality 模块，基于该模块在 /etc/pam. d/common-password 文件中可以设置更加精细的口令策略，比如口令必须包含大小写字母的个数、特殊字符的个数、口令由几种类型的字符组成、新口令与旧口令不同的位数等。

7. 修改自动注销账户时间

在 Linux 系统中 root 账户具有最高特权，如果系统管理员在离开系统之前忘记注销 root 账户，将会带来很大的安全隐患，安全起见，应该让系统自动注销用户。通过修改账户中的 TIMOUT 参数，可以实现此功能，编辑/etc/profile 后加入下面这行。

```
export TMOUT = 300
```

300 表示 300 秒，也就是 5 分钟，如图 5.9 所示。这样，如果系统中登录的用户在 5 分钟内都没有动作，那么系统会自动注销这个账户。

图 5.9　设置用户注销时间

而在 Redhat 等版本的 Linux 中，打开/etc/profile 文件，找到配置项 HISTSIZE，在它的下一行添加 TMOUT＝300 即可。

8. 设置账户登录失败锁定次数、锁定时间

在 Ubuntu 20. 04 中账户登录失败锁定的次数也在配置文件/etc/login. defs 中设置，如图 5. 10 所示，但是如果系统启用 PAM 登录机制，相应的锁定策略由 PAM 机制设置。

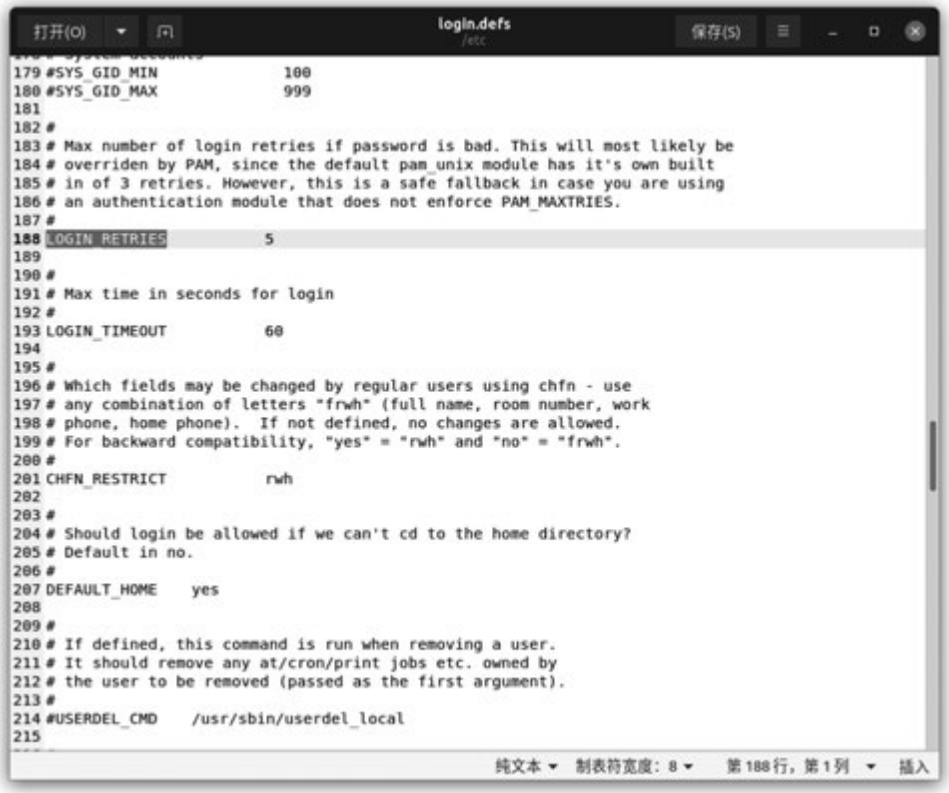

图 5.10 设置用户登录失败次数

而在 Redhat 等版本的 Linux 中，可以编辑文件/etc/pam. d/system-auth，设置 auth required pam_tally. so 为需要的策略。

```
auth required pam_tally. so onerr = fail deny = 6 unlock_time = 300
```

设置密码连续错误 6 次锁定账户，且锁定时间为 300 秒。

5.3 Linux 访问控制实验

5.3.1 实验目的

掌握自主访问控制的概念；掌握文件和目录自主访问控制的设置。

5.3.2 实验任务

利用 Linux 自主访问控制机制实现用户授权；利用 Linux 访问控制列表实现用户授权；目录文件的访问权限控制。

5.3.3 实验环境

主流配置计算机一台，安装 Ubuntu 20.04。

5.3.4　实验任务分析及实验步骤

【任务分析】

访问控制是 Linux 系统提供的一种非常重要的安全机制，用以确保用户只能执行权限范围内的操作。Linux 采用自主访问控制，由文件所有者进行授权。当创建一个文件时，系统会为用户分配默认权限，如果需要调整权限，由文件所有者根据安全需求利用指令 chmod 进行授权，授权需要遵循最小授权原则，否则会带来安全隐患。为了授权方便，Linux 将用户划分为属主、同组用户、其他用户三类，利用 chmod 可以为这三类用户授权。Linux 也支持为某个用户单独授权。

【实验步骤】

1. 利用 Linux 自主访问控制机制实现用户授权

(1) root 创建目录/data 和用户 linda 和 lucy，并分别为两个用户设置密码，如图 5.11 所示。

图 5.11　新建用户

(2) 在/data 目录中，linda 用 touch 命令创建文件 helloworld，查看 helloworld 的文件权限，测试用户 lucy 能否写，可以看到文件写入失败，请求被拒绝，如图 5.12 所示。

(3) 用户 linda 用 chmod 命令修改文件 helloworld 的权限，使得 lucy 能写文件，如图 5.13 所示。

2. 利用 Linux 访问控制列表实现用户授权

(1) 以 lucy 用户身份创建文件 worldhello，查看文件权限，可以看到用户 linda 对文件 worldhello 没有写权限。lucy 利用 setfacl 设置用户 linda 的权限，并通过 getfacl 查看文件

图 5.12 创建文件并测试文件默认权限

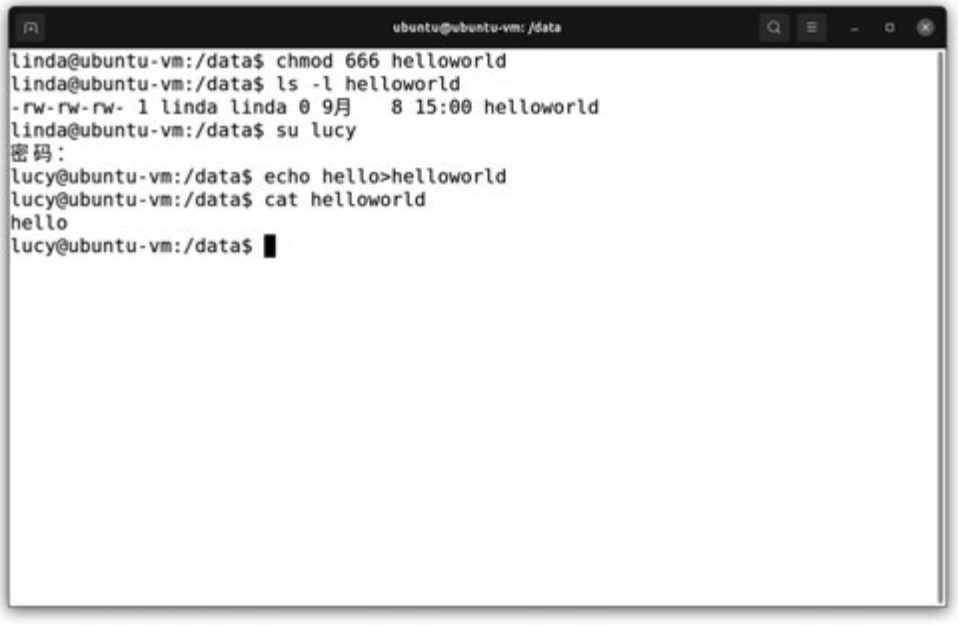

图 5.13 修改文件权限并测试

worldhello 的权限。切换到 linda 用户,尝试往该文件中写入内容,可以看到文件写入成功,如图 5.14 所示。

(2) 使用-x 选项删除 linda 对文件 worldhello 的访问权限,然后再次尝试以 linda 用户身份对文件 worldhello 进行写操作,可以看到写操作被拒绝,如图 5.15 所示。

图 5.14　设置单个用户的访问权限并测试

图 5.15　删除单个用户的访问权限并测试

3. 目录文件的访问权限控制

(1) 用户 linda 在/data 目录下创建目录 data，查看目录权限，可以看到用户 lucy 对于目录/data/data 拥有读和执行的权限，在新建目录下用 touch helloworld 创建文件，切换当前用户为 lucy，由于拥有对 data 目录的读权限，因而可以利用 ls 列出目录/data/data 的内

容,由于具有对 data 目录的执行权限,因而可以利用 cd 可以进入目录,但由于不具有对目录的写权限,所以不能在该目录下创建文件 worldhello,如图 5.16 所示。

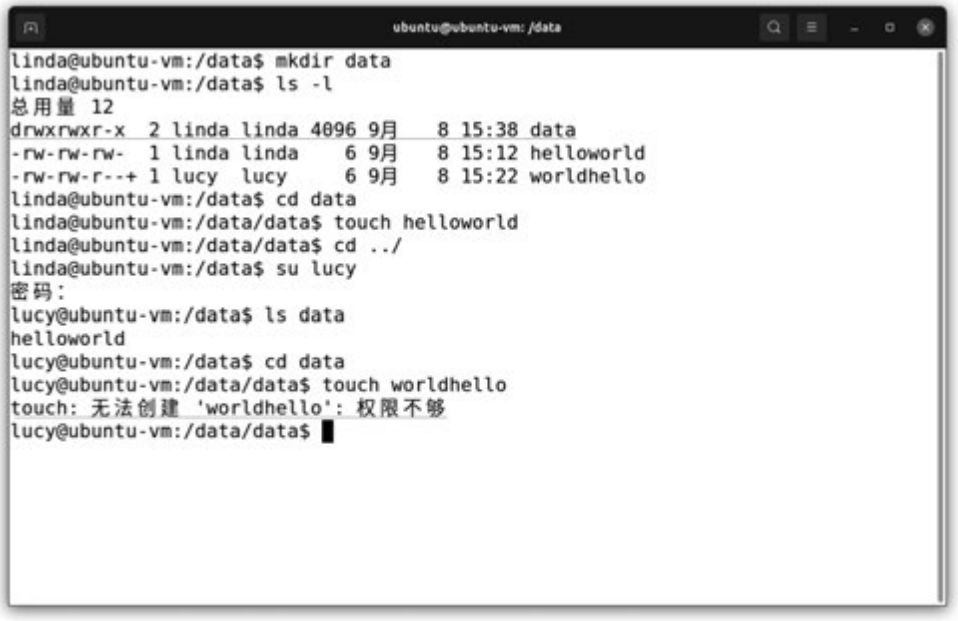

图 5.16 创建目录并测试默认访问权限

(2) 用户 linda 修改其他用户对目录/data/data 的权限,依次去掉其他用户对目录的读权限和执行权限,验证 luxy 是否能执行 ls 和 cd 操作,由此可见对于目录的读权限对应的是列出目录下内容的权限,目录的执行权限是进入目录的权限,如图 5.17 所示。

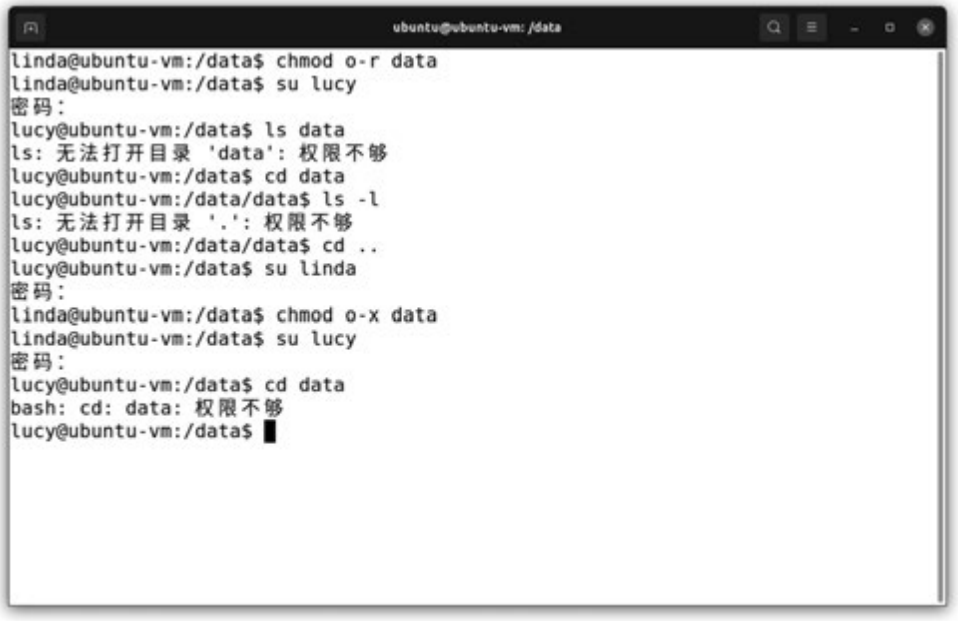

图 5.17 验证目录权限的含义

(3) 用户 linda 修改其他用户对目录/data/data 的权限,使其他用户具有对目录的读、写和执行权限,并验证,如图 5.18 所示。

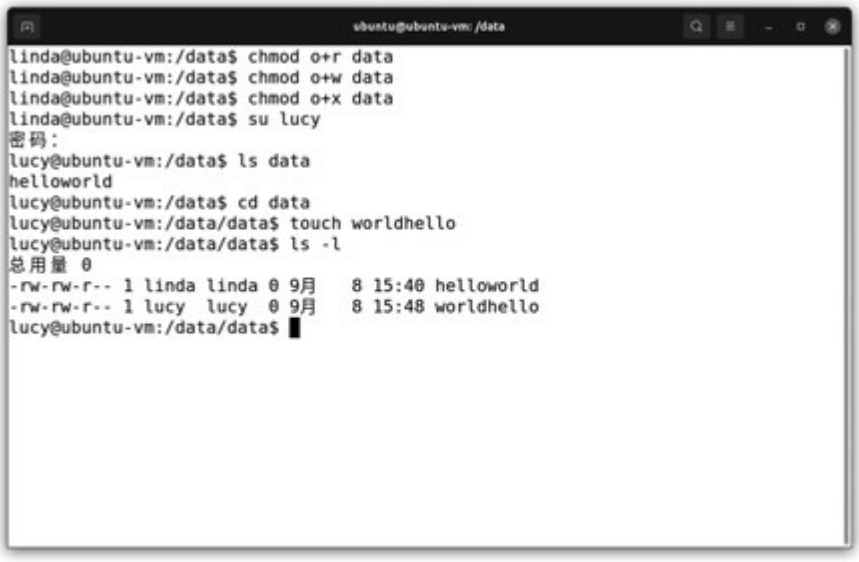

图 5.18　修改目录访问权限并测试

5.4　Linux 特殊权限实验

5.4.1　实验目的

掌握 Linux 中 SUID、Sticky 特殊权限的含义和配置方法。

5.4.2　实验任务

SUID、Sticky 等特殊权限的设置。

5.4.3　实验环境

主流配置计算机一台,安装 Ubuntu 20.04。

5.4.4　实验任务分析及实验步骤

【任务分析】

为了实现一些特殊的安全需求,Linux 提供了 SUID、SGID、Sticky 权限,理解这些权限的含义是正确使用这些权限的前提。本次任务主要设置常用的 SUID 和 Sticky 权限,这两个权限分别设置在可执行程序和目录上,用户在执行设置了 SUID 的程序时会临时以文件属主的身份执行,如果程序属主是 root 用户,则普通用户在执行程序时会临时提权,给系统带来一定的安全隐患。为了使得用户在某个目录下创建的文件不被其他用户删除,可以在目录上设置 Sticky 权限位。

【实验步骤】

1. SUID 特殊权限

1) 查找系统中所有设置了 SUID 权限位的可执行文件

Linux 中有一类具有特殊权限的可执行文件,当执行该类文件时会使普通用户的权限得到提升,给系统带来很大的安全隐患,这样的特殊权限是通过给可执行文件设置 SUID 位(有时简称 s 位)来实现的。

find path -perm 用于根据文件权限在指定路径下查找文件,有以下三种形式。

(1) find path -perm mode。

(2) find path -perm -mode。

(3) find path -perm +mode。

其中 find path -perm mode 表示严格匹配,也即文件权限必须跟 mode 完全一致才能匹配成功; find path -perm -mode 表示文件权限包含 mode 权限的文件都能匹配成功; find path -perm +mode 表示文件权限被 mode 权限包含的文件都能匹配成功。

在 5.1.2 节中介绍到用户的普通权限可以表示成 3 位十进制数值型,分别表示属主、同组用户和其他用户的权限,而对于特殊权限 SUID、SGID、Sticky 也可以表示成十进制数值,其中 4 表示 SUID、2 表示 SGID、1 代表 Sticky 权限值,这里我们查找的是设置了 SUID 位的所有程序,可以通过用 find / -perm -4000 命令找到所有具有 s 位权限的文件。

2) 关闭/usr/bin/gpasswd 的 s 位权限

利用 chmod 命令可以关闭可执行文件的 s 位权限,如图 5.19 所示。

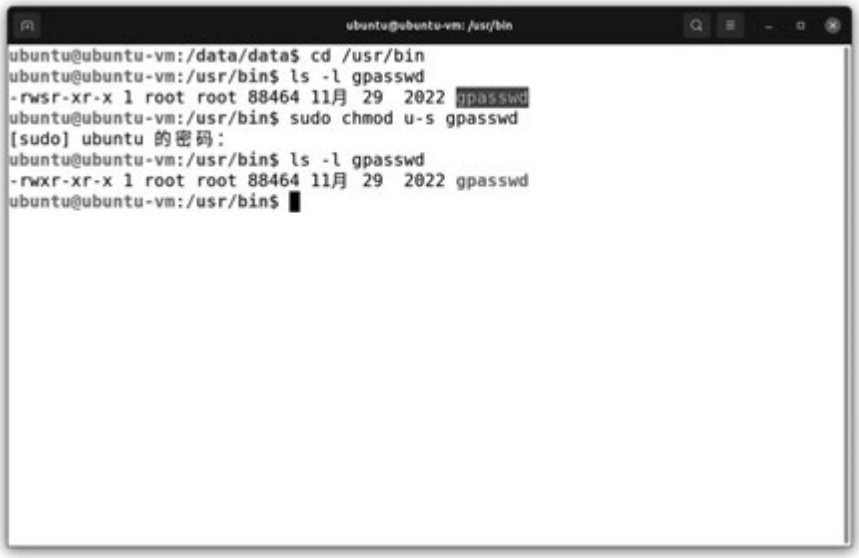

图 5.19 关闭/usr/bin/gpasswd 的 s 位权限

2. Sticky 特殊权限

如果多个用户对同一目录具有写权限,则一个用户可以删除另一个用户创建的文件,给系统带来安全隐患。如果在目录上设置 Sticky 特殊权限,则可以防止一个用户删除另一个

用户创建的文件，而只有文件或目录的属主可以删除该文件。

（1）root 用户创建目录/bookstore，查看目录权限，可以看到在默认情况下目录对非同组的其他用户只有读和执行的权限，因而其他用户无法在该目录下新建文件或子目录，如图 5.20 所示。

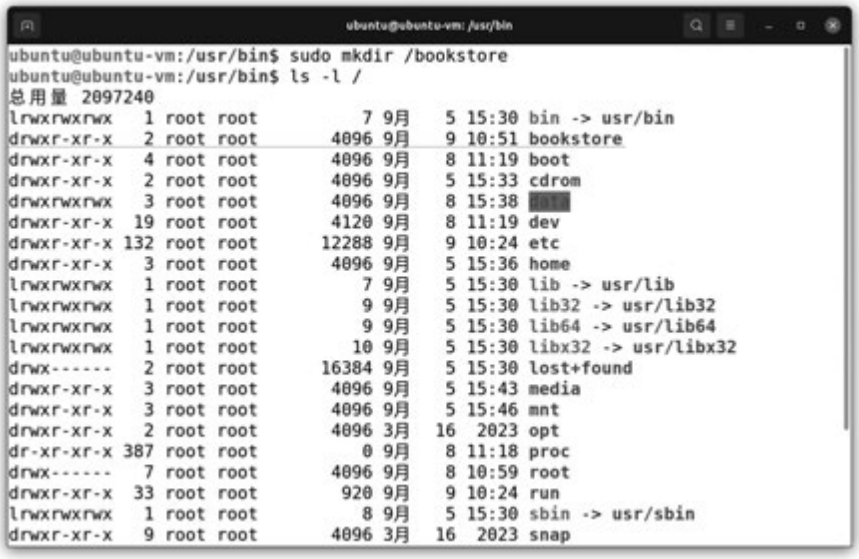

图 5.20　目录 bookstore 的默认权限

通过指令 chmod 777 修改新建目录的权限，使得所有用户都具有所有权限。以 linda 用户的身份登录系统，进入/bookstore 目录后创建文件 helloworld，切换 lucy 用户，可以看到 lucy 可以删除 linda 创建的文件，存在安全隐患，如图 5.21 所示。

图 5.21　目录没有设置 sticky 位带来的安全隐患

(2) root 用户利用 chmod 1777 命令或 chmod o+t 命令设置/bookstore 目录的 Sticky 权限,用户 linda 再次在该目录下创建 helloworld 文件,可以发现 lucy 无法删除该文件,如图 5.22 所示。

图 5.22 目录设置 sticky 位可防止文件被其他用户删除

5.5 练 习 题

(1) 在 5.3 节的实验中分别利用自主访问控制机制和访问控制列表为用户授予对文件的写权限,两种授权有何不同? 哪种更安全?

(2) 在 5.3 节的实验中,用户 test 创建了文件 aa,通过授权允许用户 zhang 读写,但不允许用户 Li 读写,用户 Li 能绕过访问控制最终读取到文件 aa 的内容吗? 请通过实验验证你的结论。

(3) 查找资料进一步学习 find 指令的用法,利用该指令查找所有设置了 SGID 位、Sticky 位的文件或目录。

(4) SGID 通常设置在目录上,设置了 SGID 的目录下创建的文件或子目录的 GID 为目录的 GID,请设计实验验证 SGID 的权限。