第4章

用户与群组管理

第4章用户与群组管理

- 4.1 用户账户与群组概念
- 4.2 用户与群组文件
- 4.3 用户与群组管理
- 4.4 用户身份切换

4.1 用户账户与群组概念

理解用户账户和群组

- Linux操作系统是多用户多任务的操作系统，允许多个用户同时登录到系统，使用系统资源。**用户账户**是用户的身份标识。用户通过用户账户可以登录到系统，并且访问已经被授权的资源。系统依据账户来区分属于每个用户的文件、进程、任务，并给每个用户提供特定的工作环境（例如，用户的工作目录、shell版本以及图形化的环境配置等），使每个用户都能各自不受干扰地独立工作。

- Linux系统下的用户分为三种：

（1）普通用户：在系统中只能进行普通工作，只能访问他们拥有的或者有权限执行的文件。

（2）超级用户（root）：也叫管理员账户，它的任务是对普通用户和整个系统进行管理。超级用户账户对系统具有绝对的控制权，能够对系统进行一切操作

（3）系统用户：与系统服务相关，但不能用于登录

理解用户账户和群组

- **群组**是具有相同特性的用户的逻辑集合，使用群组有利于系统管理员按照用户的特性组织和管理用户，提高工作效率。
- 有了群组，在做资源授权时可以把权限赋予某个群组，群组中的成员即可自动获得这种权限。
- 一个用户账户可以同时是多个群组的成员，其中某个群组是该用户的主群组（私有群组），其他群组为该用户的附属群组（标准群组）。

用户和群组的基本概念

概　念	描　述
用户名	用来标识用户的名称，可以是字母、数字组成的字符串，区分大小写
密码	用于验证用户身份的特殊验证码
用户标识（UID）	用来表示用户的数字标识符
用户主目录	用户的私人目录，也是用户登录系统后默认所在的目录
登录shell	用户登录后默认使用的shell程序，默认为/bin/bash
群组	具有相同属性的用户属于同一个群组
群组标识（GID）	用来表示群组的数字标识符

root用户的UID为0：系统用户的UID从1到999；普通用户的UID可以在创建时由管理员指定，如果不指定，用户的UID默认从1 000开始顺序编号。在Linux系统中，创建用户账户的同时也会创建一个与用户同名的群组，该群组是用户的主群组。普通群组的GID默认也是从1 000开始编号。

4.2 用户账户与群组文件

用户账户文件

- 用户登录过程

1. 先找寻 /etc/passwd 里面是否有你输入的帐号？如果没有则跳出，如果有的话则将该帐号对应的 UID 与 GID（在 /etc/group 中）读出来，另外，该帐号的家目录与 shell 设定也一并读出；

2. 进入 /etc/shadow 里面找出对应的帐号与 UID，然后核对一下你刚刚输入的密码与裡头的密码是否相符

3. 核对成功，就进入 Shell 环境

用户账户文件

- **/etc/passwd**文件：在Linux系统中，所创建的用户账户及其相关信息（密码除外）均放在/etc/passwd配置文件中。用vim编辑器（或者使用**cat /etc/passwd**）打开passwd文件，内容格式如下：

```
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
user1:x:1002:1002::/home/user1:/bin/bash
```

- 文件中的每一行代表一个用户账户的资料，可以看到第一个用户是root。然后是一些标准账户，此类账户的shell为/sbin/nologin，代表无本地登录权限。最后一行是由系统管理员创建的普通账户：user1。passwd文件的每一行用“：”分隔为7个域，各域的内容如下：

用户名:加密口令:UID:GID:用户的描述信息:主目录:命令解释器（登录shell）

用户账户文件

- /etc/shadow文件：由于所有用户对/etc/passwd文件均有读取权限，为了增强系统的安全性，用户经过加密之后的口令都存放在/etc/shadow文件中。/etc/shadow文件只对root用户可读，因而大大提高了系统的安全性。shadow文件的内容形式如下：

```
root:$6$PQxz7W3s$Ra7Akw53/n7rntDgjPNWdCG66/5RZgjhoe1zT2
F00ouf2iDM.AVvRIYoez10hGG7kBHEaah.oH5U1t6OQj2Rf.:17654:
0:99999:7:::
bin:*:16925:0:99999:7:::
daemon:*:16925:0:99999:7:::
bobby:!!:17656:0:99999:7:::
user1:!!:17656:0:99999:7:::
```

用户账户文件

- shadow文件保存投影加密之后的口令以及与口令相关的一系列信息，每个用户的信息在shadow文件中占用一行，并且用“:”分隔为9个域，内容如下表

字　段	说　明
1	用户登录名
2	加密后的用户口令，*表示非登录用户，！！表示没设置密码
3	从1970年1月1日起，到用户最近一次口令被修改的天数
4	从1970年1月1日起，到用户可以更改密码的天数，即最短口令存活期
5	从1970年1月1日起，到用户必须更改密码的天数，即最长口令存活期
6	口令过期前几天提醒用户更改口令
7	口令过期后几天账户被禁用
8	口令被禁用的具体日期（相对日期，从1970年1月1日至禁用时的天数）
9	保留域，用于功能扩展

群组文件

- **/etc/group文件**：用于存放用户的组账户信息，对于该文件的内容任何用户都可以读取。每个群组账户在group文件中占用一行，并且用“:”分隔为4个域。每一行各域的内容如下（使用**cat /etc/group**）：

```
root:x:0:
bin:x:1:
daemon:x:2:
bobby:x:1001:user1,user2
user1:x:1002:
```

分别表示：群组名称:群组口令（一般为空，用x占位）:GID:群组成员列表

root的GID为0，没有其他组成员。group文件的群组成员列表中如果有多个用户账户属于同一个群组，则各成员之间以“,”分隔。在/etc/group文件中，用户的主群组并不把该用户作为成员列出，只有用户的附属群组才会把该用户作为成员列出。例如，用户bobby的主群组是bobby，但/etc/group文件中群组bobby的成员列表中并没有用户bobby，只有用户user1和user2。

群组文件

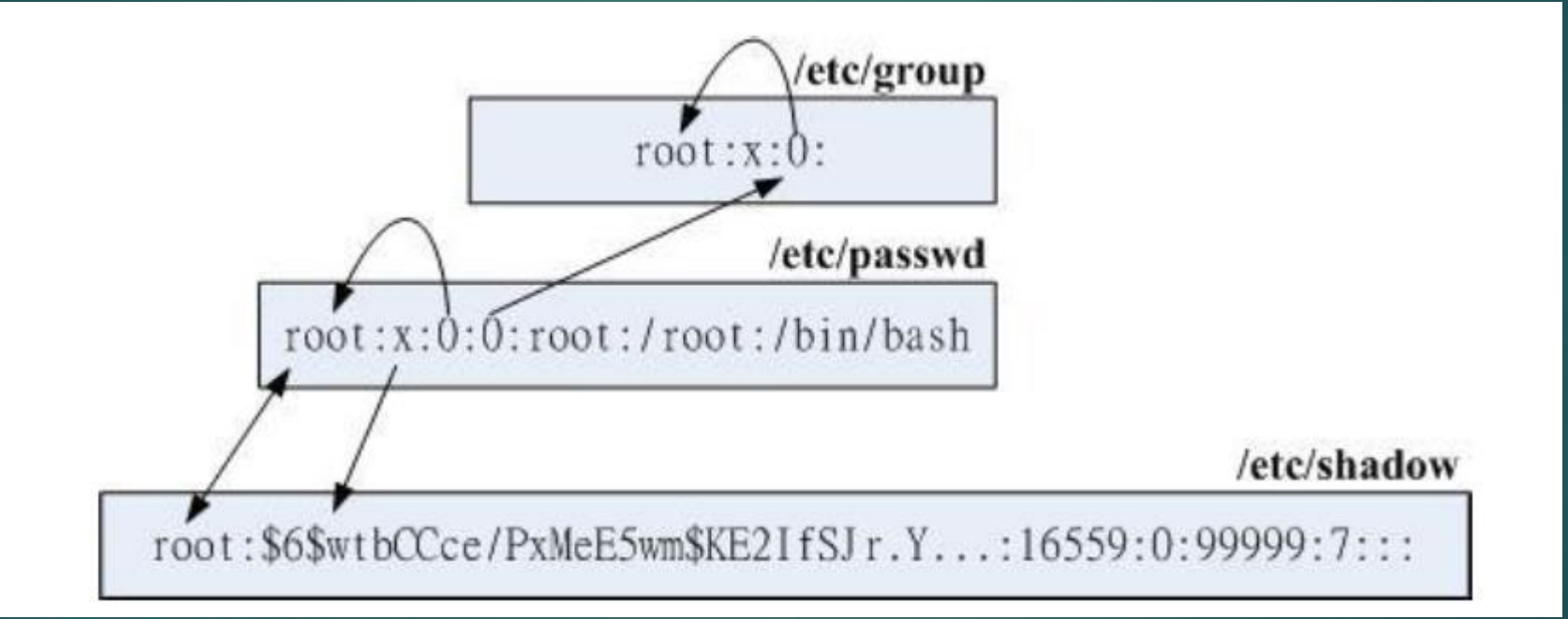
- **/etc/gshadow文件**：用于存放群组的加密口令、组管理员等信息，该文件只有root用户可以读取。每个群组账户在gshadow文件中占用一行，并以“:”分隔为4个域。每一行中各域的内容如下：

```
root:::
bin:::
daemon:::
bobby:!::user1,user2
user1:!::
```

表示：群组名称:加密后的群组口令（没有就用！）:群组的管理员:群组成员列表

群组文件

- /etc/group、/etc/shadow、/etc/passwd 文件关系



4.3 用户账户与群组管理

用户账户管理

- 1. 新建用户 useradd

作用：新建用户账户

命令格式：useradd [选项] <username>

选　项	说　明
-c comment	用户的注释性信息
-d home_dir	指定用户的主目录
-e expire_date	禁用账号的日期，格式为YYYY-MM-DD
-f inactive_days	设置账户过期多少天后用户账户被禁用。如果为0，账户过期后将立即被禁用；如果为-1，账户过期后，将不被禁用
-g initial_group	用户所属主群组的群组名称或者GID
-G group-list	用户所属的附属群组列表，多个群组之间用逗号分隔
-m	若用户主目录不存在则创建它
-M	不要创建用户主目录
-n	不要为用户创建用户私人群组
-p passwd	加密的口令
-r	创建UID小于500的不带主目录的系统账号
-s shell	指定用户的登录shell，默认为/bin/bash
-u UID	指定用户的UID，它必须是唯一的，且大于499

用户账户管理

使用举例：新建用户user3，UID为1010，指定其用户的主目录为/home/user3，用户的shell为/bin/bash，用户的密码为123456，账户永不过期。

```
[root@server1 ~]# useradd -u 1010 -d /home/user3 -s /bin/bash -p 123456 -f -
1 user3
[root@server1 ~]# tail -1 /etc/passwd
user3:x:1010:1000::/home/user3:/bin/bash
```

用户账户管理

账户创建过程

在 /etc/passwd 里面建立一行与帐号相关的资料，包括建立 UID/GID/家目录等；

在 /etc/shadow 里面将此帐号的密码相关参数填入，但是尚未有密码；

在 /etc/group 里面加入一个与帐号名称一模一样的群组名称；

在 /home 底下建立一个与帐号同名的目录作为使用者家目录，且权限为 700

查看账户信息

- 1. id

  功能：查看用户的UID、GID和用户所属群组的信息，如果不指定用户，则显示当前用户的相关信息。

  命令格式：id <username>

- 2. whoami

  功能：查看当前用户名

  命令格式：whoami

- 3. w

  功能：查看当前登录系统用户和详细信息

  命令格式：w

用户账户管理

- 2. 设置账户口令passwd

作用：指定和修改用户账户口令。超级用户可以为自己和其他用户设置口令，而普通用户只能为自己设置口令。

命令格式：passwd [选项] [username]

选 项	说 明
-l	锁定（停用）用户账户
-u	口令解锁
-d	将用户口令设置为空，这与未设置口令的账户不同。未设置口令的账户无法登录系统，而口令为空的账户可以
-f	强迫用户下次登录时必须修改口令
-n	指定口令的最短存活期
-x	指定口令的最长存活期
-w	口令要到期前提前警告的天数
-i	口令过期后多少天停用账户
-S	显示账户口令的简短状态信息

用户账户管理

使用举例：假设当前用户为root，则下面的两个命令分别为root用户修改自己的口令和root用户修改user1用户的口令。

```
//root用户修改自己的口令，直接用passwd命令回车即可
[root@server1  ~]# passwd

//root用户修改user1用户的口令
[root@server1  ~]# passwd user1
```

普通用户修改口令时，passwd命令会首先询问原来的口令，只有验证通过才可以修改。而root用户为用户指定口令时，不需要知道原来的口令。为了系统安全，用户应选择包含字母、数字和特殊符号组合的复杂口令，且口令长度应至少为8个字符。

用户账户管理

- 3. 修改用户账户 usermod

作用：修改用户账户信息。

命令格式：usermod [选项] 用户名

参　数	作　用
-c	填写用户账户的备注信息
-d -m	参数-m与参数-d连用，可重新指定用户的家目录并自动把旧的数据转移过去
-e	账户的到期时间，格式为YYYY-MM-DD
-g	变更所属用户组
-G	变更扩展用户组
-L	锁定用户禁止其登录系统
-U	解锁用户，允许其登录系统
-s	变更默认终端
-u	修改用户的UID

用户账户管理

- 使用举例

将用户user1加入root用户组中，这样扩展组列表中会出现root用户组的字样，而基本组不会受到影响,用-G参数修改用户扩展组ID：

```
[root@server1  ~]# usermod -G root user1
[root@server1  ~]# id user1
uid=1002(user1) gid=1002(user1) 组=1002(user1),0(root)
```

用-u参数修改用户的UID

```
[root@server1  ~]# usermod -u 8888 user1
[root@server1  ~]# id user1
uid=8888(user1) gid=1002(user1) 组=1002(user1),0(root)
```

修改用户user1的主目录为/var/user1，把启动shell修改为/bin/tcsh，完成后恢复到初始状态，操作如下：

```
[root@server1  ~]# usermod -d /var/user1 -s /bin/tcsh user1
[root@server1  ~]# tail -3 /etc/passwd
user1:x:8888:1002::/var/user1:/bin/tcsh
```

用户账户管理

- 4. 删除用户账户 userdel

  作用：删除用户账户

  命令格式：userdel [-r] 用户名

-r选项，在删除用户账户的同时，还将用户主目录以及其下的所有文件和目录全部删除掉。

```
[root@server1  ~]# userdel -r user1
```

用户账户管理

- 5. 禁用和恢复用户账户

作用：临时禁用一个账户而不删除

方法1：使用passwd命令

```
[root@server1  ~]# passwd -l user1
锁定用户 user1 的密码
passwd: 操作成功

//利用passwd命令的-u选项解除账户锁定，重新启用user1账户
[root@server1  ~]# passwd -u user1
```

方法2：使用usermod命令

```
//禁用user1账户
[root@server1  ~]# usermod -L user1
//解除user1账户的锁定
[root@server1  ~]# usermod -U user1
```

方法3：修改/etc/passwd文件，passwd域添加“*”

群组管理

- 1. 添加群组 groupadd

  命令格式： groupadd [-选项] 新群组名

- 2.删除群组：groupdel

  命令格式：geoupdel [-选项] 群组名

- 3.修改群组：groupmod

  命令格式：groupmod [选项] 组名

选　项	说　明
-g gid	把群组的GID改成gid
-n group-name	把群组的名称改为group-name

群组管理

- 4. 为群组添加用户

useradd命令创建用户时，会同时创建一个和用户账户同名的群组，称为主群组。当一个群组中必须包含多个用户时，则需要使用附属群组。在附属组中增加、删除用户都用gpasswd命令。 **只有root用户和组管理员才能够使用这个命令**

命令格式：gpasswd [选项] [用户] [组]

选　项	说　明
-a	把用户加入组
-d	把用户从组中删除
-r	取消组的密码
-A	给组指派管理员

```
[liu@localhost ~]$ sudo gpasswd -A liu liu
[liu@localhost ~]$ gpasswd -a jone liu
正在将用户 "jone"加入到 "liu"组中
```

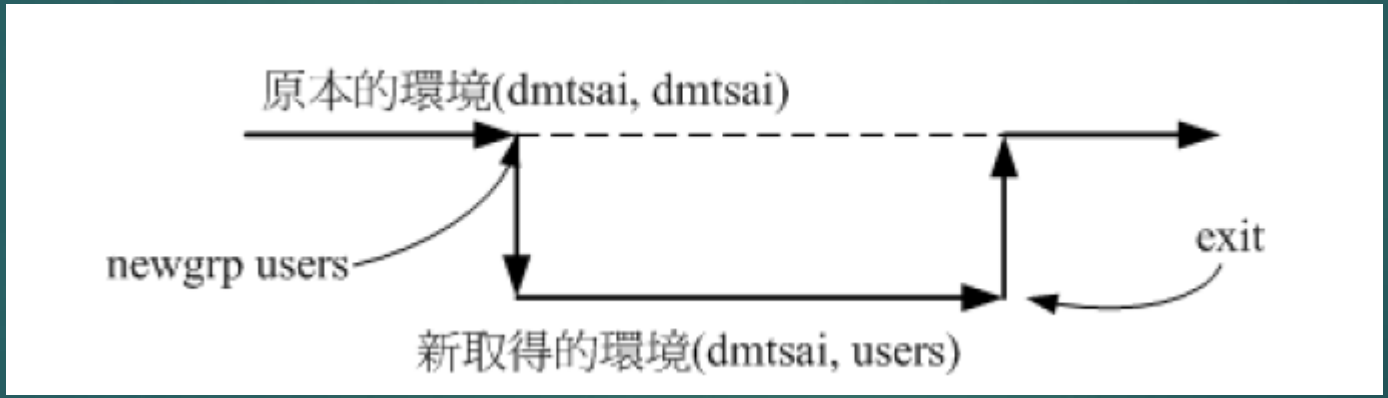
初始群组和有效群组**

- 创建用户时，用户被分配初始群组GID，加入其它群组后将拥有其它群组的权限。
- 创建文件时， 文件群组为有效群组

查看有效群组：groups

切换有效群组：newgrp 群组

原理： 分配另一个shell，用exit回到原shell



4.4 用户身份切换

用户身份切换

- 1. su命令

作用：可以解决切换用户身份的需求，使得当前用户在不退出登录的情况下，顺畅地切换到其他用户

当从root管理员切换到普通用户时是不需要密码验证的，而从普通用户切换成root管理员就需要进行密码验证。

```
[test@server1  ~]$ su - root
Password:
[root@server1  ~]# su – test
上一次登录：日 5月  6 05:22:57 CST 2018pts/0 上
[test@server1  ~]$ exit
logout
[root@server1  ~]#
```

用户身份切换

尽管像上面这样使用su命令后，普通用户可以完全切换到root管理员身份来完成相应工作，但这会暴露root管理员的密码，从而增大了系统密码被黑客获取的概率，因此上述操作并不是最安全的方案。

用户身份切换

- 2. sudo命令

作用：sudo命令用于给普通用户提供额外的权限来完成原本root管理员才能完成的任务。

命令格式：sudo [参数] 命令名称 （输入当前用户密码 5分钟内不用重复验证）

参数	作用
-h	列出帮助信息
-l	列出当前用户可执行的命令
-u用户名或UID值	以指定的用户身份执行命令
-k	清空密码的有效时间，下次执行sudo时需要再次进行密码验证
-b	在后台执行指定的命令
-p	更改询问密码的提示语

用户身份切换

普通用户不用知道其它账户密码，就能通过sudo获得额外的权限，但需要在配置文件（/etc/sudoers）中进行配置， 该文件提供集中的用户管理、权限与主机等参数，内容如下：

```
[root@server1 ~]# visudo
90 ##
91 ## Allow root to run any commands anywhere
92 root ALL=(ALL) ALL
93 test ALL=(ALL) ALL
```

谁可以使用 允许使用的主机=（以谁的身份） 可执行命令的列表

用户身份切换

除了上述配置外，还可将用户加入到wheel群组

1. visudo 去掉%wheel的注释符号“#”
2. 使用命令 usermod -G wheel user3

用户身份切换

应用举例1：

普通用户user1，查看/root目录，权限不够：

```
[user1@thispc test]$ ls /root
ls: 无法打开目录/root: 权限不够
```

用户user1， 未在sudoers中添加配置，则使用sudo命令时会出现以下提示：

```
[user1@thispc test]$ sudo ls /root
[sudo] user1 的密码：
user1 不在 sudoers 文件中。此事将被报告。
```

以root身份通过visudo，添加行user1，则可以使用sudo， 拥有root的权限：

```
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
user1   ALL=(ALL)       ALL
```

```
[root@thispc test]# su - user1
上一次登录：五 8月 14 17:36:59 CST 2020pts/2 上
[user1@thispc ~]$ sudo ls /root
[sudo] user1 的密码：
anaconda-ks.cfg       system.tar.bz2  userpass.txt  视频  下载
```

用户身份切换

应用举例2：

在sudoers文件中添加配置，仅仅让user1能够以root身份执行cat命令（**绝对路径**）：

```
## Allow root to run any commands anywhere
root    ALL=(ALL)       ALL
user1   ALL=(ALL)       /usr/bin/cat
```

user1 可以root身份执行cat，但执行其它命令仍然没权限：

```
[user1@thispc ~]$ sudo cat /etc/shadow
root:$6$zLEE8XN2$11ocLeHMcc18Ez4jf.7HP119AkfNC2iS57ZaKt4y4EZkb6SLraoD
bUFza5/xmqC/7k9ENCKrW6dF2NZwAlefr1:18354:0:99999:7:::
```

```
[user1@thispc ~]$ sudo ls /root
对不起，用户 user1 无权以 root 的身份在 thispc 上执行 /bin/ls /root。
```

批量创建用户

命令：newusers

用法：newusers 用户文件，指定包含用户信息的文本文件，文件格式要与/etc/passwd相同

用户名:x:UID:GID:用户说明:用户的家目录:所用SHELL

```
[root@localhost ~]# newusers userlist.txt
[root@localhost tail -n 5 /etc/passwd /etc/shadow
user1:x:2222:2004::/var/user1:/bin/tcsh
zhang:x:2005:2005::/home/zhang:/bin/bash
user3:x:1010:1010::/home/user3:/bin/bash
user4:x:2010:2010::/home/user4:/bin/bash
user5:x:2011:2011::/home/user5:/bin/bash
```

批量改用户密码

命令：chpasswd

用法： cat 密码文件 | chpasswd ，密码文件内容为[用户名：密码]

```
[root@localhost ~]# cat mima | chpasswd
#mima文件内容
user4:12345
user5:123123
```