轻松学习 Linux

主讲：孙胜利

第三回 Linux 用户与用户组管理

1）/etc/passwd 文件

2）/etc/shadow 文件

3）用户组管理

4）用户管理

5）密码管理

6）用户身份切换

7）查询当前登录的用户

8）查询所有登录过系统的用户信息

9）每个账户的最近登录时间

1、/etc/passwd 文件

- 这个文件里每一行都是一个帐号信息，有几行就代表系统中有几个帐号
- 由“:”分割成 7 个字段

1）第 1 个字段

用户名

2）第 2 个字段

早期的 Unix 系统的密码是放在这个文件中的， 但是因为这个文件的特性是所有的程序都能够读取，所以非常危险， 因此后来就将这个字段的密码数据给他改放到 /etc/shadow 中了，这边就用 x 占位代替了

3）第 3 个字段：

用户 id，简称 uid

概念：

uid 为 0 是分配给 root 的，1～499 是属于系统用户的，500～4294967295 是分配给普通用户的！

4）第 4 个字段：

用户所在用户组 id，简称 gid

5）第 5 个字段：

注释说明

6）第 6 个字段：

用户的家目录

7）第 7 个字段：

用户登录系统后使用的 shell（这个概念后面再说）

现在只要知道 如果是/sbin/nologin 表示这个用户不允许登录帐号

注：这边说的不允许登录并不是说这个用户就一无是处，虽然外人没有办法通过这个帐号登录，但是 Linux 系统可以用这个用户的身份来运行一些服务，或者运行一些程序，也就是这个账户依然可以使用系统的资源，仅仅是没法外部登录而已。

举个例子：

比如我们提供 web 服务的软件 apache，他在系统里运行的时候，系统肯定是希望对它的行为进行约束和控制的，如果不约束，那么这个程序不就是想干嘛就干嘛吗？太危险了！

怎么约束呢？当然是利用系统里非常强大的用户权限机制，给他一个身份，一旦他有了身份那么他在系统中可以操作那些文件，不可以操作那些文件就被严格的约束了，这样不就可以有效的控制了吗？

所以系统需要以一些用户的身份来运行一些特殊的软件，比如我们用 sifangku 这个用户的身份来运行 apache，那么 apache 在运行时所能够操作的文件就是 sifangku 这个用户他所拥有的权限决定的了。

操作系统只是需要的是以他的身份来运行软件,以达到控制约束这个软件的行为 ,至于他能不能从外部登录，那就无所谓了，你外人利用这个身份登录系统反而会无事生非，所以干脆就别让你外部利用这个身份登录了，相当于这个身份就是给我操作系统内部使用的，你外人想用没门！

2、/etc/shadow 文件

- 由“:”分割成 9 个字段

1）第 1 个字段

用户名与/etc/passwd 对应

2）第 2 个字段

用户密码，已经经过加密

- 此字段可以为空，即该用户在的登录时，不需要密码
- 以叹号开始的密码字段意味着密码被锁定

3）第 3 个字段

上次更改密码的日期，这个数字是 1970 年 1 月 1 日到上次更改密码的日期的天数

- 如果为 0 表示用户应该在下次登录系统时必须更改密码
- 为空则表示密码年龄功能被禁用

4）第 4 个字段

密码的最小年龄，即经过多少天才能更改密码，默认为 0，即随便什么时候都能改密码

- 为空也表示随便什么时候都能改密码

5）第 5 个字段

最大密码年龄，到了年龄之后必须更改密码。

- 密码年龄到了最大年龄之后，密码仍然可用，但是用户将会在登录的时候被要求更改密码
- 为空表示没有最大密码年龄

6）第 6 个字段

密码警告时间段，如果设置为 7,则在密码到期前 7 天系统会开始提醒用户密码即将到期

- 为空 或者 0 表示没有密码警告期

7）第 7 个字段

密码禁用期（通牒期），如果设置为 3 则表示一旦密码到达最大年龄且用户并没有修改密码，则再过 3 天，这个帐号便不能再使用当前的密码登录！你也可以理解为这个是密码达到最大年龄后的宽限期，如果宽限期内依然不更改密码，则通牒期过后该用户的不能再使用当前密码登录，此时用户必须要联系系统管理员，自己已经无法自助操作了！

- 空字段表示不强制密码过期

8）第 8 个字段

账户过期的日期（帐号的生命周期，注意是账号不是密码），这个数字也是距离 1970 年 1 月 1 日的天数，表示该帐号在这个日期到达后将不允许登录。

注：账户过期不同于密码被锁定。账户过期时用户将不被允许登录（哪怕是同其他验证方式比如 ssh 密钥认证）；而密码被锁定，用户将不被允许使用其密码登录，但是依然可以通过密钥认证来登录

- 空字段表示账户永不过期

9）第 9 个字段

系统保留将来使用！

注： 此文件绝对不能让普通用户可读，为了安全可以把这个文件权限设置为 000

/etc/shadow- 是 /etc/shadow 的备份文件

3、用户组管理

1）新增用户组

groupadd [-g GID] 组名称

2）删除用户名

groupdel 组名称

3）/etc/group

- 由“:”分割成 4 个字段

①第 1 个字段

用户组名称

②第 2 个字段

用户组密码，这边用 x 占位，实际密码在/etc/gshadow 中，用于用户组管理员功能使用，“用户组管理员”功能用的极少，所以这边不做介绍

③第 3 个字段

用户组 id，即 GID，与/etc/passwd 第 4 个字段对应

④第 4 个字段

- 这个字段列出的是用户名，多个用户名用“,”分割！
- 什么意思呢？
  - 一个用户可以所属多个用户组，其中/etc/passwd 文件里第 4 个字段里的 GID，指的这个用户的初始用户组，如果这个用户还加入了其他用户组，那么是怎么记录的呢？就是通过这个字段记录的！
  - 如果某个用户加入了这个用户组，那么就会记录在这个字段里，当然如果一个用户的初始用户组就是这个用户组那么该用户可以不记录在这里.
  - 专业术语：
    - 初始用户组
    - 支持用户组：用户除了初始用户组之外还加入了其他用户组，那么那些其他的用户组就叫这个用户的

4、用户管理

1）新增用户

useradd [-u UID] [-g GID] [-d HOME] [-M] [-s] 用户名

-u：自定义 UID

-g：后面接组 id，设置用户所属组，当然也可以直接写组名字

-G：后面接次要用户组,多个用逗号分隔

-d：自定义家目录

-M：默认会建立加目录，加上这个选项则不建立家目录

-s：自定义 shell

用户名格式说明：

可以使用大小写、数字、减号（不能出现在首位）、点、下划线

不建议使用点和减号

PS：其实还有很多选项，不过用的不多，所以有需要自己 man useradd 查询吧

2）删除用户

userdel [-r] 用户名

3）修改用户

usermod [-dgGus]

参数含义同 useradd

4）查看所在用户组

groups [用户名]

- 一个用户如果加入了多个用户组，那么只要是那些用户组拥有的权限，这个用户就都拥有！
- 问题在于：如果这个用户想新建一个文件，那么这个文件到底是属于哪个用户组呢？

这个就是“有效用户组”的概念了！“有效用户组”决定了，groups 命令输出的用户组里，第一个用户组就是这个用户的“有效用户组”。即“有效用户组”决定一个用户新建一个文件时这个文件所属的用户组！

5）切换自己的有效用户组

newgrp 用户组

前提：用户必须事先在这个用户组里

6）id [username]

查询某人或自己的信息

5、密码管理

1）passwd [用户名]

默认创建的用户是未设置密码的，所以需要使用 passwd 设置密码否则无法登录！

若后面不接用户名则设置的是**设置用户自己的密码**，root 可以设置其他人的密码，普通用户不行哦！

选项：

-S：后接用户的名字，显示密码信息，仅 root 才能使用

```
root PS 2018-04-12 0 99999 7 -1 (Password set, SHA512 crypt.)
```

- 用户名，/etc/shadow 第 1 个字段
- 密码状态简写
- 上次更改密码日期，/etc/shadow 第 3 个字段
- 最小密码年龄，密码至少用多少天才能修改，/etc/shadow 第 4 个字段
- 最大密码年龄，密码最长用多少天必须修改，/etc/shadow 第 5 个字段
- 密码警告时间段，/etc/shadow 第 6 个字段
- 密码最后通牒期，/etc/shadow 第 7 个字段
- 密码状态说明

-l：锁住密码，使得密码失效，仅 root 才能使用

/etc/shadow 第二个字段最前面加上！，这样密码就失效了。

注： 帐户并不是被完全锁定，用户仍然可以通过其他认证方式（如 ssh 密钥认证验证）登录。

使用 chage -E 0 user 名称，可完全将帐户锁定，其实这条命令是修改/etc/shadow 第 8 个字段为 0，则表示账户在 1970 年 1 月 1 日就过期了。取消账户锁定：chage -E '' user 名称

chage 的其他用法请自行 man 查询！

-u：与-l 相反，即解锁密码

-d：删除帐户密码，它会设置指定的帐户无密码（密码字段会变成空），此时该账户登录时就无需输入密码，仅 root 才能使用

-e：使帐户密码过期，用户会在下次登录尝试期间被迫更改密码，仅 root 才能使用

/etc/shadow 第 3 个字段被设置为 0

-n：设置密码的最短生存期，以天为单位，即多久不可修改密码，仅 root 才能使用

/etc/shadow 第 4 个字段，会被编辑为你设置的天数

-x：设置密码的最长生存期，以天为单位，即多久必须要修改密码，仅 root 才能使用

/etc/shadow 第 5 个字段，会被编辑为你设置的天数

-w：设置提前警告天数，即密码过期前的警告天数，仅 root 才能使用

/etc/shadow 第 6 个字段，会被编辑为你设置的天数

-i：密码最后通牒期，仅 root 才能使用

/etc/shadow 第 7 个字段，会被编辑为你设置的天数

--stdin：接受管道数据来作为密码输入，一般用于 shell 脚本

例如：echo ‘123456’ | passwd --stdin linpingzhi

因为命令有历史记录，直接密码明文写在这里不太安全，所以不建议直接用 echo 的方式

2）生成随机密码

openssl rand 密码字节数 -base64

注：这边 1 个字节长度为 1 位，这 1 位长度的字符需要经过 base64 编码之后才能正常显示，否则看到的是乱码

生成随机密码的方法非常多，大家如果不喜欢这种可以自己寻找其他喜欢的方法。

举例：openssl rand 14 -base64

配合 passwd 的--stdin 选项会很方便！

openssl rand 14 -base64 | tee passwd-linpingzhi | passwd --stdin linpingzhi

3）密码的存储

虽然有些可以存储密码的软件，但是密码是比较敏感的物品，为避免密码泄露给第三方，所以如果你管理的账号和密码比较多我建议你可以自己搭建一个数据库服务器，然后把密码记录在里面，做好备份，这样比较安全。

6、用户身份切换

root 的权限极大，为了防止误操作以及无事生非，很多时候我们都是使用普通用户在操作 Linux，但是有些事情只有 root 才能做，普通用户是没有权利的，这时就需要临时的切换到 root 身份。

怎么切换呢？

1）su [-] 用户名

- 切换到指定用户，普通用户不接用户名则切换到 root 用户
- -：

加上-之后 切换时会初始化目的用户的各种环境变量

- 老实讲 su 命令在实际应用中并不多，为什么？想必你自己也能感受到！

2）sudo

- sudo 命令 可以让某个用户以其他用户的身份执行命令（比如 root），并且关键在于它不需要其他用户的密码哦，只需要输入自己的密码来确认操作即可！那么有同学问了：不需要指定用户的密码，那岂不是整个系统的用户权限系统都形同虚设了？
- 当然不是你想象的这样的！并不是所有人都能够执行 sudo 的，能够使用 sudo 命令的用户都是需要经过 root 用户钦点的！具体的且听我细细道来！
- 仅有/etc/sudoers 文件内的指定的用户可以执行 sudo 命令，因此我们要使用超级管理员 root 去编辑这个文件来钦定可以使用 sudo 命令的人，不过这个文件是具有格式规范的，Linux 系统为了防止牛 X 的你 粗心大意改错了，为你提供了 visudo 命令去专门的修改/etc/sudoers 文件，通过 visudo 命令编辑/etc/sudoers 文件会在编辑完离开时自动的去检查文件的格式规范！
  - sudo [-u 用户名] 要执行的命令

    -u：后接欲切换的用户，若不加这项则默认切换的 root 身份来执行
- visudo
  - 注：输入:然后按 i 表示开始编辑，编辑完了请按:输入 wq 保存退出，或者 q 不保存退出，其实这个就是调用的 vi 编辑器来编辑了，vi 编辑器的具体操作我们以后会专门讲
  - 格式：

    用户名　　来源主机名=(允许以谁的身份)　　允许执行的命令
  - 说明：
    - 用户名：指定可以使用 sudo 命令的帐号名称
    - 来源主机名：这个帐号可以以哪台主机连接到本机，比如这边你可以指定一个 ip 地址或 hostnames（较少配置）
    - 允许以谁的身份：指的是这个帐号可以，以谁的身份来执行命令（较少配置）
    - 允许执行的命令：指定以指定身份能够执行的命令，多个用“,”分割，**必须写绝对路径（which 查看）**
- 常用场景：

  1）假如希望 sunshengli 用户可以切换到任何有效用户以执行所有命令：

  sunshengli　　ALL=(ALL)　　ALL

  注意：ALL 严格区分大小写！

  2）假如希望某个用户组的所有用户都能够拥有 sudo 权利怎么办？

  %用户组名比如%wheel　　ALL=(ALL)　　ALL

  这样子只要是指定用户组里的用户都可以执行 sudo 了

  3）可否无需输入自己密码就可执行 sudo？

  sunshengli　　ALL=(ALL)　　NOPASSWD:ALL

  4）指定可以执行的命令

  命令可以加参数,如果命令前面加上!表示不能执行指定的命令，而这边列出的其他命令可以执行（注意命令的先后顺序，这涉及到逻辑关系）

  ```
  %wheel  ALL=(ALL)       NOPASSWD: ALL, !/bin/cat
  ```

  - 指定命令那边的 ALL 表示可以执行所有命令，!/bin/cat 表示不能执行/bin/cat 命令，所以合起来就是可以执行除/bin/cat 之外的所有命令

- 注意逻辑关系，是从左到右，不要把 ALL 与!/bin/cat 写倒了，这样逻辑关系就不对了，就达不到效果了
- 问：以下写法什么意思？

  ALL,!/usr/bin/passwd,/usr/bin/passwd [A-Za-z]*,!/usr/bin/passwd root

  答案：避免 root 密码被修改

5）利用别名简化、灵活配置信息

  User_Alias USERS_GOOD=sushengli,sifangku

  Cmnd_Alias CMND_LIST=/usr/bin/passwd,...

- 然后可以这样：

  USERS_GOOD ALL=(ALL) CMND_LIST

- 注意：

  定义的别名必须全部用大写

  还可以定义 Host_Alias 即 来源主机别名

  - 注：一次使用 sudo 在 5 分钟内再次使用无需输入密码确认

7、查询当前登录的用户

w

  显示已经登录的用户

who

  显示已经登录的用户

who am i

  显示自己登录的用户（su 并不会影响这个命令输出的结果）

whoami

  显示自己当前的身份（会受 su 的影响）

8、查询所有登录过系统的用户信息

last

注：可以看到登录以及登出时间

选项：

  -w：显示完整用户名

  -F：显示完整日期和时间

注：后面讲管道符就知道怎么翻页浏览了

9、每个账户的最近登录时间

lastlog

选项：

  -u 用户名：查看指定用户