Python开发之运维基础

讲师：王晓春

本章内容

- 解释Linux的安全模型
- 解释用户帐号和组群帐号的目的
- 用户和组管理命令
- 理解并设置文件权限
- 默认权限
- 特殊权限
- ACL

用户user

- 令牌token,identity
- Linux用户：Username/UID
- 管理员：root, 0
- 普通用户：1-65535
  - 系统用户：1-499, 1-999 （CentOS7）
    对守护进程获取资源进行权限分配
  - 登录用户：500+, 1000+（CentOS7）
    交互式登录

组group

- Linux组：Groupname/GID
- 管理员组：root, 0
- 普通组：
  - 系统组：1-499, 1-999（CENTOS7）
  - 普通组：500+, 1000+（CENTOS7）

安全上下文

- Linux安全上下文

  运行中的程序：进程 (process)

  以进程发起者的身份运行：

  root: /bin/cat

  mage: /bin/cat

  进程所能够访问资源的权限取决于进程的运行者的身份

组的类别

- Linux组的类别
  - 用户的主要组(primary group)
    - 用户必须属于一个且只有一个主组
    - 组名同用户名，且仅包含一个用户，私有组
  - 用户的附加组(supplementary group)
    - 一个用户可以属于零个或多个辅助组

用户和组的配置文件

- Linux用户和组的主要配置文件：
  /etc/passwd：用户及其属性信息(名称、UID、主组ID等）
  /etc/group：组及其属性信息
  /etc/shadow：用户密码及其相关属性
  /etc/gshadow：组密码及其相关属性

passwd文件格式

- login name：登录用名（ wang ）
- passwd：密码（x)
- UID：用户身份编号 (1000)
- GID：登录默认所在组编号（1000)
- GECOS：用户全名或注释
- home directory：用户主目录 (/home/wang)
- shell：用户默认使用shell (/bin/bash)

shadow文件格式

- 登录用名
- 用户密码，一般用sha512加密
- 从1970年1月1日起到密码最近一次被更改的时间
- 密码再过几天可以被变更（0表示随时可被变更）
- 密码再过几天必须被变更（99999表示永不过期）
- 密码过期前几天系统提醒用户（默认为一周）
- 密码过期几天后帐号会被锁定
- 从1970年1月1日算起，多少天后帐号失效

group文件格式

- 群组名称：就是群组名称
- 群组密码：通常不需要设定，密码是被记录在/etc/gshadow
- GID：就是群组的 ID
- 以当前组为附加组的用户列表(分隔符为逗号)

gshdow文件格式

- 群组名称：就是群组名称
- 群组密码：
- 组管理员列表：组管理员的列表，更改组密码和成员
- 以当前组为附加组的用户列表：(分隔符为逗号)

用户和组管理命令

- 用户管理命令
  - useradd
  - usermod
  - userdel
- 组帐号维护命令
  - groupadd
  - groupmod
  - groupdel

用户创建：useradd

- useradd [options] LOGIN
  -u UID
  -o 配合-u 选项，不检查UID的唯一性
  -g GID：指明用户所属基本组，可为组名，也可以GID
  -c "COMMENT"：用户的注释信息
  -d HOME_DIR: 以指定的路径(不存在)为家目录
  -s SHELL: 指明用户的默认shell程序
  　　可用列表在/etc/shells文件中
  -G GROUP1[,GROUP2,...]：为用户指明附加组，组须事先存在
  -N 不创建私用组做主组，使用users组做主组
  -r: 创建系统用户 CentOS 6: ID<500，CentOS 7: ID<1000
  -m 创建家目录，用于系统用户
  -M 不创建家目录，用于非系统用户

创建用户： useradd

- 默认值设定：/etc/default/useradd文件中
- 显示或更改默认设置

  useradd -D

  useradd –D -s SHELL

  useradd –D –b BASE_DIR

  useradd –D –g GROUP

用户属性修改

- usermod [OPTION] login

  -u UID: 新UID

  -g GID: 新主组

  -G GROUP1[,GROUP2,...[,GROUPN]]]：新附加组，原来的附加组将会被覆盖；若保留原有，则要同时使用-a选项

  -s SHELL：新的默认SHELL

  -c 'COMMENT'：新的注释信息

  -d HOME: 新家目录不会自动创建；若要创建新家目录并移动原家数据，同时使用-m选项

  -l login_name: 新的名字；

  -L: lock指定用户,在/etc/shadow 密码栏的增加！

  -U: unlock指定用户,将 /etc/shadow 密码栏的！拿掉

  -e YYYY-MM-DD: 指明用户账号过期日期

  -f INACTIVE: 设定非活动期限

删除用户

- userdel [OPTION]... login
  - -r: 删除用户家目录

查看用户相关的ID信息

- id [OPTION]... [USER]
  - -u: 显示UID
  - -g: 显示GID
  - -G: 显示用户所属的组的ID
  - -n: 显示名称，需配合ugG使用

切换用户或以其他用户身份执行命令

- su [options...] [-] [user [args...]]
- 切换用户的方式：

  su UserName：非登录式切换，即不会读取目标用户的配置文件，不改变当前工作目录

  su - UserName：登录式切换，会读取目标用户的配置文件，切换至家目录，完全切换
- root su至其他用户无须密码；非root用户切换时需要密码
- 换个身份执行命令：

  su [-] UserName -c 'COMMAND'
- 选项：-l --login

  su -l UserName 相当于 su - UserName

设置密码

- passwd [OPTIONS] UserName: 修改指定用户的密码
- 常用选项：
  -d:删除指定用户密码
  -l:锁定指定用户
  -u:解锁指定用户
  -e:强制用户下次登录修改密码
  -f: 强制操作
  -n mindays: 指定最短使用期限
  -x maxdays：最大使用期限
  -w warndays：提前多少天开始警告
  -i inactivedays：非活动期限
  --stdin：从标准输入接收用户密码
  echo "PASSWORD" | passwd --stdin USERNAME

创建组

- groupadd [OPTION]... group_name
  -g GID: 指明GID号；[GID_MIN, GID_MAX]
  -r: 创建系统组
  CentOS 6: ID<500
  CentOS 7: ID<1000

修改和删除组

- 组属性修改：groupmod

  groupmod [OPTION]... group

  -n group_name: 新名字

  -g GID: 新的GID

- 组删除：groupdel

  groupdel GROUP

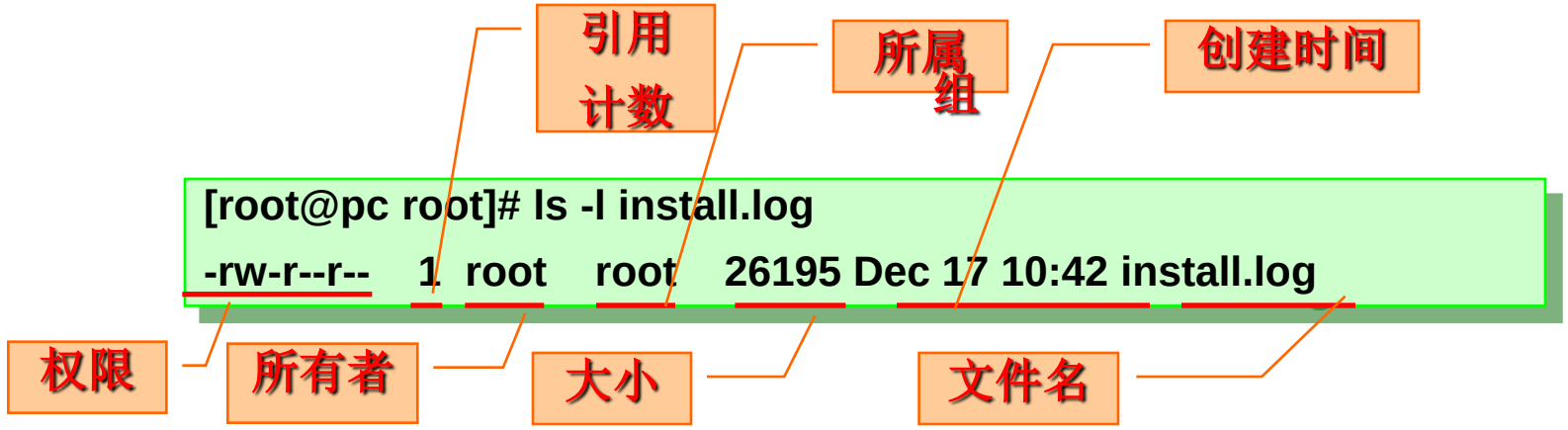
更改组密码

- 组密码：gpasswd
- gpasswd [OPTION] GROUP
  - -a user 将user添加至指定组中
  - -d user 从指定组中移除用户user
  - -A user1,user2,... 设置有管理权限的用户列表
- newgrp命令：临时切换主组

  如果用户本不属于此组，则需要组密码

文件权限

- 文件属性



- 文件属性操作
  - chown 设置文件的所有者
  - chgrp 设置文件的属组信息

修改文件的属主和属组

- 修改文件的属主：chown

  chown [OPTION]... [OWNER][:[GROUP]] FILE...

  用法：

  OWNER

  OWNER:GROUP

  :GROUP

  命令中的冒号可用.替换

  -R: 递归

  chown [OPTION]... --reference=RFILE FILE...

- 修改文件的属组：chgrp

  chgrp [OPTION]... GROUP FILE...

  chgrp [OPTION]... --reference=RFILE FILE...

  -R 递归

文件权限

- 文件的权限主要针对三类对象进行定义
  - owner: 属主, u
  - group: 属组, g
  - other: 其他, o
- 每个文件针对每类访问者都定义了三种权限
  - r: Readable
  - w: Writable
  - x: eXcutable

文件权限

- 文件：
  - r: 可使用文件查看类工具获取其内容
  - w: 可修改其内容
  - x: 可以把此文件提请内核启动为一个进程
- 目录：
  - r: 可以使用ls查看此目录中文件列表
  - w: 可在此目录中创建文件，也可删除此目录中的文件
  - x: 可以使用ls -l查看此目录中文件列表，可以cd进入此目录

文件权限操作

文件权限（rwx）

权限项	文件类型	读	写	执行	读	写	执行	读	写	执行
字符表示	（d|l|c|s|p）	(r)	(w)	(x)	(r)	(w)	(x)	(r)	(w)	(x)
数字表示		4	2	1	4	2	1	4	2	1
权限分配		文件所有者			文件所属组用户			其他用户		

文件权限操作命令

- chmod

修改文件权限

- chmod [OPTION]... OCTAL-MODE FILE...
  -R: 递归修改权限
- chmod [OPTION]... MODE[,MODE]... FILE...
  MODE：
  修改一类用户的所有权限：
  u= g= o= ug= a= u=,g=
  修改一类用户某位或某些位权限
  u+ u- g+ g- o+ o- a+ a- + -
- chmod [OPTION]... --reference=RFILE FILE...
  参考RFILE文件的权限，将FILE的修改为同RFILE

新建文件和目录的默认权限

- umask值 可以用来保留在创建文件权限
- 新建FILE权限: 666-umask
  如果所得结果某位存在执行（奇数）权限，则将其权限+1
- 新建DIR权限: 777-umask
- 非特权用户umask是 002
- root的umask 是 022
- umask: 查看
- umask #: 设定
  umask 002
- umask –S 模式方式显示
- umask –p 输出可被调用
- 全局设置： /etc/bashrc 用户设置：~/.bashrc

Linux文件系统上的特殊权限

- SUID, SGID, Sticky
- 三种常用权限：r, w, x　user, group, other
- 安全上下文
- 前提：进程有属主和属组；文件有属主和属组

(1) 任何一个可执行程序文件能不能启动为进程,取决发起者对程序文件是否拥有执行权限

(2) 启动为进程之后，其进程的属主为发起者,进程的属组为发起者所属的组

(3) 进程访问文件时的权限，取决于进程的发起者

(a) 进程的发起者，同文件的属主：则应用文件属主权限

(b) 进程的发起者，属于文件属组；则应用文件属组权限

(c) 应用文件“其它”权限

可执行文件上SUID权限

- 任何一个可执行程序文件能不能启动为进程：取决发起者对程序文件是否拥有执行权限
- 启动为进程之后，其进程的属主为原程序文件的属主
- SUID只对二进制可执行程序有效
- SUID设置在目录上无意义
- 权限设定：

```
chmod u+s FILE...
chmod u-s FILE...
```

可执行文件上SGID权限

- 任何一个可执行程序文件能不能启动为进程：取决发起者对程序文件是否拥有执行权限
- 启动为进程之后，其进程的属组为原程序文件的属组
- 权限设定：

  chmod g+s FILE...

  chmod g-s FILE...

目录上的SGID权限

- 默认情况下，用户创建文件时，其属组为此用户所属的主组
- 一旦某目录被设定了SGID，则对此目录有写权限的用户在此目录中创建的文件所属的组为此目录的属组
- 通常用于创建一个协作目录
- 权限设定：

```
chmod g+s DIR...
chmod g-s DIR...
```

Sticky 位

- 具有写权限的目录通常用户可以删除该目录中的任何文件，无论该文件的权限或拥有权
- 在目录设置Sticky 位，只有文件的所有者或root可以删除该文件
- sticky 设置在文件上无意义
- 权限设定：

  chmod o+t DIR...

  chmod o-t DIR...
- 例如：
  - ls -ld /tmp drwxrwxrwt 12 root root 4096 Nov 2 15:44 /tmp

访问控制列表

- ACL：Access Control List，实现灵活的权限管理
- 除了文件的所有者，所属组和其它人，可以对更多的用户设置权限
- CentOS7 默认创建的xfs和ext4文件系统具有ACL功能
- CentOS7 之前版本，默认手工创建的ext4文件系统无ACL功能,需手动增加

  tune2fs –o acl /dev/sdb1

  mount –o acl /dev/sdb1 /mnt/test
- ACL生效顺序：所有者，自定义用户，自定义组，其他人

访问控制列表

- 为多用户或者组的文件和目录赋予访问权限rwx
  - mount -o acl /directory
  - getfacl file |directory
  - setfacl -m u:wang:rwx file|directory
  - setfacl -Rm g:sales:rwX directory
  - setfacl -M file.acl file|directory
  - setfacl -m g:salesgroup:rw file| directory
  - setfacl -m d:u:wang:rx directory
  - setfacl -x u:wang file |directory
  - setfacl -X file.acl directory

祝大家学业有成

谢　谢