Nguyễn Thế Đức - Bkacad

Quản trị người dùng Linux

1. Quản trị tài khoản người dùng

Có hai loại tài khoản:

- Tài khoản quản trị root: có quyền quản trị cao nhất trong hệ thống, được phép làm mọi việc mà không bị kiểm soát
- Các tài khoản thông thường được tạo ra cho các mục đích:
  - Cung cấp tài khoản truy nhập cho người sử dụng hệ thống
  - Cung cấp tài khoản dùng bởi các dịch vụ hệ thống như http, samba, mysql,…

Chú ý: tránh làm việc dưới tài khoản của root cho các công việc thông thường hàng ngày

Với Linux, mỗi user có một định danh duy nhất, gọi là UID (User ID).

- *0 – 99: user có quyền quản trị*
- *>99: user khác. >= 500: không phải user hệ thống*
- *UID có khả năng sử dụng lại*

Mỗi user thuộc ít nhất một group. Mỗi group cũng có một định danh duy nhất là GID

Mỗi users cần có những thông tin: tên user, UID, tên group, GID, home directory…

Windows quản lý thông tin bằng LDAP, Kerberos. Linux quản lý thông tin bằng file text.

Có thể chỉnh sửa thông tin của users bằng công cụ, hoặc sửa trực tiếp bằng text file.

Quản lý người dùng hệ thống bằng lệnh

- *useradd: tạo user mới*
- *usermod: chỉnh sửa thông tin user.*
- *userdel -r: xóa user khỏi hệ thống*
- *passwd: đổi mật khẩu, chính sách thay đổi mật khẩu*
- *groupadd: tạo group mới*
- *groupdel: xóa group khỏi hệ thống*
- *groupmod: chỉnh sửa thông tin group.*

```
# ls -l /usr/sbin/adduser
lrwxrwxrwx 1 root root 7 Oct 21 15:18 /usr/sbin/adduser -> useradd
```

Tạo user thuộc nhóm ***users***

```
# useradd -N u1
```

-N, --no-user-group

Do not create a group with the same name as the user

```
# useradd -g users u2; useradd -g users u3
```

Kiểm tra users id

```
# id u1
```

Tạo user không có thư mục home

```
# useradd -M u4
```

Đặt thời hạn tài khoản cho user bằng thuộc tính -e cùng với ngày tháng hết hạn hoạt động

```
# useradd -e 2016-03-26 u5
# chage -l u5
```

Joint 2 file */etc/passwd* và */etc/group* dựa trên GID

```
# join -t ":" -1 4 -2 3 -o 1.1 2.1 /etc/passwd /etc/group
```

Join GID cột 4 trong /etc/passwd và cột 3 trong /etc/group

Lấy kết quả cột thứ 1 của file /etc/passưd và cột thứ 1 của file /etc/group

Xóa user

```
# userdel -r u5
```

-r Xóa cả thư mục home của user

Tạo user quyền root

```
# useradd  -u 0 -o  u5
```

Tạo user thuộc 2 nhóm **dba** và **users**

```
# groupadd dba
# useradd -g users -G dba u6
```

Mỗi user thuộc về một group chính (primary group), thay đổi group này bằng lệnh *usermod -g*

User có thể thuộc các group khác (secondary group), thay đổi group này bằng lệnh *usermod -G*

Kiểm tra user và group

```
# more /etc/passwd
# more /etc/group
```

Khoá tài khoản người dùng

Khoá user *u1*

```
# passwd -l u1
```

Hoặc

```
# usermod -L u1
```

Kiểm tra */etc/shadow*

```
# cat /etc/shadow | grep u1
```

u1:!!1fbFIdiNH$LOJdm8DqDOceykgR5GRv91:15267:0:99999:7:::

Mở khoá user *u1*

```
# passwd -u u1
```

Hoặc

```
# usermod -U u1
```

Kiểm tra */etc/shadow*

```
# cat /etc/shadow | grep u1
```

u1:1fbFIdiNH$LOJdm8DqDOceykgR5GRv91:15267:0:99999:7:::

Xoá trắng mật khẩu với lệnh passwd -d

```
# passwd -d u1
```

≅

*Mật khẩu bắt đầu bằng **!!** → tài khoản chưa tạo password*
*Mật khẩu bắt đầu bằng **!** → tài khoản tạm thời bị khóa (locked)*
*Mật khẩu bắt đầu bằng ***** → tài khoản đã bị vô hiệu hóa (disable)*

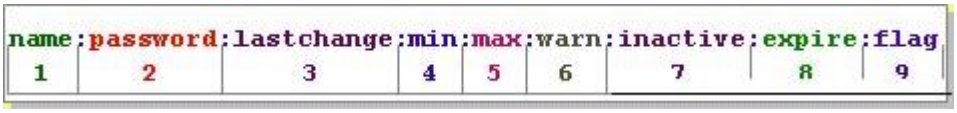
2. Quản lý thông tin user và group

/etc/passwd: chứa thông tin user login, password mã hóa, UID, GID, home directory, và login shell. Mỗi dòng là thông tin của một user.

Home directory: chứa một số file cấu hình chuẩn

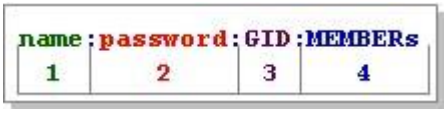
- *.bash_profile: Thực thi mỗi khi user login, thiết lập biến môi trường **PATH***
- *.bash_logout: Thực thi mỗi khi user logout*
- *.bash_history: Chứa các dòng lệnh user đã gõ*

/etc/shadow: chứa thông tin password mã hóa, thời gian sử dụng password, thời gian phải thay đổi password… mỗi một user là một dòng gồm 9 trường

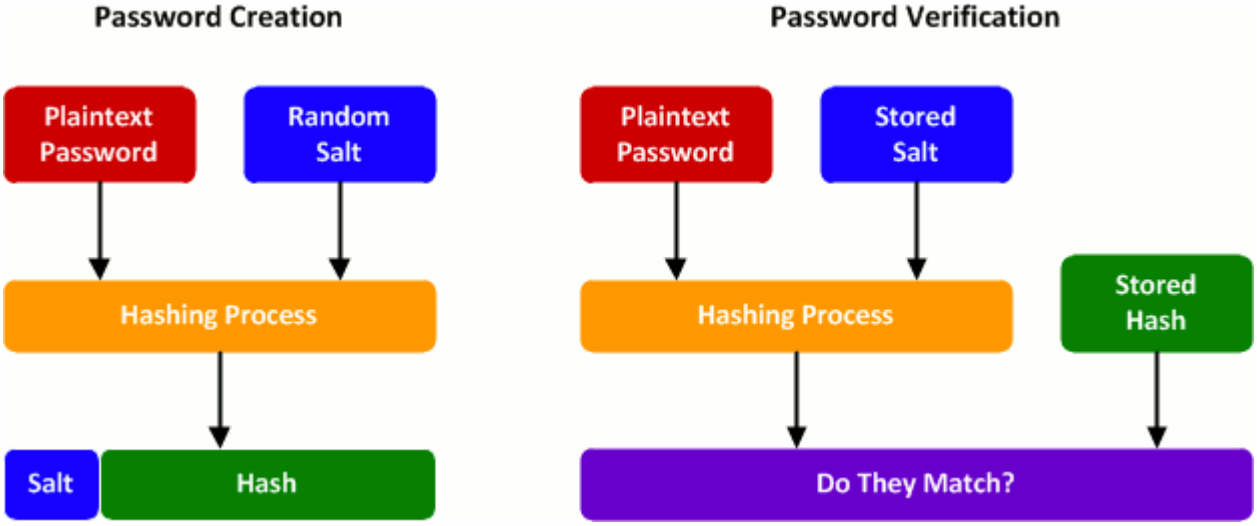


- name: username
- password: mật khẩu ở dạng đã được mã hóa
- lastchange: số ngày kể từ ngày **1/1/1970** đến ngày thay đổi password gần đây nhất
- min: số ngày tồn tại tối thiểu của password
- max: số ngày tồn tại tối đa của password
- warn: số ngày user được thông báo trước khi password hết hiệu lực
- inactive: số ngày sau khi password hết hiệu lực, account bị disable
- expire: số ngày kể từ ngày **1/1/1970** đến ngày account bị disable
- flag: trường cuối cùng chưa được dùng vào việc gì; để dành cho tương lai.

/etc/group: chứa thông tin group



Công thức hash mật khẩu trong */etc/shadow* (*one way hash function*)



- Mật khẩu của user sẽ được hash với giá trị *salt* ngẫu nhiên. Giá trị *salt* và mật khẩu đã được hash sẽ được lưu trong file */etc/shadow*
- Khi user login vào hệ thống thì giá trị *salt* sẽ được lấy ra và dùng để hash mật khẩu. Nếu giá trị mật khẩu hash trong file */etc/shadow* và giá trị vừa được hash trùng nhau thì user được xác thực.

Mật khẩu trong hệ thống Linux

Các hệ thống linux thường sử dụng thuật toán băm để tạo mật khẩu được mã hóa

Thuật toán DES 56 bit được sử dụng như thuật toán mã hoá phổ biến trước đây

```
# perl -e 'print crypt("password", "salt"),"\n"'
sa3tHJ3/KuYvI
# useradd -p sa3tHJ3/KuYvI u7
# echo "123456" | passwd u7 --stdin
```

Theo thời gian, khi phần cứng phát triển và giá thành thiết bị rẻ đi, thuật toán DES trở nên dễ dàng bị giải mã. Các bản phân phối linux hiện tại cung cấp thuật toán mã hoá mạnh hơn như: MD5, SHA-1, SHA-2 thay thế cho thuật toán DES.

Ký tự **"$"** trước mật khẩu đã được mã hóa của user cho biết mật khẩu này được mã hóa bằng một thuật toán mã hóa khác DES

*Trường mật khẩu trong **/etc/shadow** có cấu trúc dạng* ***idsalt$hash***

id=1 *mật khẩu được mã hóa bằng thuật toán MD5*

id=2 *mật khẩu được mã hóa bằng thuật toán blowfish (thuật toán mã hóa đối xứng, dùng khóa nào để mã hóa dữ liệu thì dùng chính khóa đó để giải mã)*

id=2a *mật khẩu được mã hóa bằng thuật toán eksblowfish*

id=5 *mật khẩu được mã hóa bằng thuật toán SHA-256*

id=6 *mật khẩu được mã hóa bằng thuật toán SHA-512*

salt*: 8 characters*

hash*: MD5: 22 characters, SHA-256: 43 characters, SHA-512: 86 characters*

Thuật toán SHA

Năm thuật toán SHA là *SHA-1* (160 bit), *SHA-224* (224 bit), *SHA-256* (256 bit), *SHA-384* (384 bit), và *SHA-512* (512 bit). SHA là thuật toán băm được phát triển bởi cục an ninh quốc gia Mỹ (National Security Agency hay NSA) và được xuất bản thành chuẩn của chính phủ Mỹ bởi viện công nghệ và chuẩn quốc gia Mỹ (National Institute of Standards and Technology hay NIST). Bốn thuật toán sau thường được gọi chung là *SHA-2*

SHA-1 được sử dụng rộng rãi trong nhiều ứng dụng và giao thức an ninh khác nhau, bao gồm TLS và SSL, PGP, SSH, S/MIME, và IPSec. SHA-1 được coi là thuật toán thay thế MD5 (128 bit). MD5 thường được diễn tả bằng một số hệ thập lục phân 32 ký tự.

Xem thuật toán hệ thống hiện tại sử dụng

```
# authconfig --test | grep hashing
 password hashing algorithm is md5
```

Cấu hình sử dụng SHA-512

```
# authconfig --passalgo=sha512 --update
# authconfig --test | grep hashing
 password hashing algorithm is sha512
```

Liệt kê các group và user

```
# getent group
# getent passwd
```

List các user session đang login

```
# who -Hu
NAME     LINE         TIME             IDLE          PID COMMENT
root     pts/0        2014-10-03 09:11   .          3659 (192.168.15.139)
root     pts/1        2014-10-03 09:17   .          3753 (192.168.15.139)
# ps -ef | grep bash
root      3659  3655  0 09:11 pts/0    00:00:00 -bash
root      3753  3749  0 09:17 pts/1    00:00:00 -bash
root      3792  3659  0 09:17 pts/0    00:00:00 grep bash
```

Xác định tập tin tty user đang login

```
# tty
/dev/pts/0
```

Kill user session đang login có PID ***3659***

```
# kill -9 3659
```

Lệnh **md5sum** sử dụng thuật toán *md5* để xác định chuỗi finger print của một gói phần mềm, một file hay một xâu ký tự. Với mục đích đảm bảo sự toàn vẹn của gói phần mềm từ nhà cung cấp tới người dùng

```
# echo "hello" | md5sum
b1946ac92492d2347c6235b4d2611184  -
# echo b1946ac92492d2347c6235b4d2611184  | wc -c
```

Linux cho phép quản lý user theo 2 cách

- Mật khẩu lưu trong */etc/passwd*
- Mật khẩu lưu trong */etc/shadow*

Để chuyển đổi giữa 2 kiểu, sử dụng lệnh

```
# pwconv
# pwunconv
# ls -l /etc/passwd /etc/passwd- /etc/shadow /etc/shadow-
-rw-r--r-- 1 root root 1601 Sep  4 03:44 /etc/passwd
-rw-r--r-- 1 root root 1549 Jul  3 17:00 /etc/passwd-
-r-------- 1 root root  994 Sep  4 03:44 /etc/shadow
-r-------- 1 root root  973 Jul  3 17:00 /etc/shadow-
# rm -f /etc/passwd /etc/shadow
```

Recover */etc/passwd* và */etc/shadow*

```
# cp /etc/passwd- /etc/passwd
# chmod 644 /etc/passwd
# cp /etc/shadow- /etc/shadow
```

Rebuild */etc/shadow* từ */etc/passwd*

```
# rm -f /etc/shadow /etc/shadow-
# pwconv
# passwd
```

≅ Khi rebuild */etc/shadow* từ */etc/passwd* , trường mật khẩu sẽ bị mất. Tạo lại mật khẩu user bằng lệnh *passwd*, nếu có quá nhiều user có thể tạo một mật khẩu mặc định bằng cách copy một mật khẩu bằng cách dùng tính năng tìm kiếm thay thế trong trình soạn thảo. Trường mật khẩu trong */etc/shadow* có thể xóa trắng

Thay đổi thông số mặc định

Khi sử dụng lệnh useradd hoặc groupadd, nếu không liệt kê đầy đủ các thông số cần thiết thì hệ thống sẽ lấy theo giá tri mặc nhiên đã được định nghĩa.trong các file sau:

/etc/default/useradd: các giá trị mặc định cho việc tạo user

/etc/skel: thư mục chứa các file cấu hình chuẩn sẽ tạo trong thư mục home của user

/etc/login.defs: những cấu hình mặc định cho shadow password (PASS_MAX_DAYS, PASS_MIN_DAY ,PASS_WARN_AGE, MD5_CRYPT_ENAB ...)

Công cụ chage cho phép quản trị hệ thống thay đổi các tham số lựa chọn trên:

```
chage [ -l ] [ -m min_days ] [ -M max_days ] [ -W warn ]  [ -I inactive ] [ -E expire ] [ -d last_day ] user
```

Tham số -l đầu tiên liệt kê giá trị của policy hiện thời của một người dùng. Tham số -E sẽ khoá một tài khoản người dùng tại thời điểm xác định. Định dạng ngày có thể theo định dạng **YYYY/MM/DD**

Để xem thông tin về thời gian của mật khẩu

chage -l *username*
Last password change : Apr 02, 2013
Password expires : never
Account expires : never
Minimum number of days between password change : 0
Maximum number of days between password change : 99999

3. Điều khiển truy cập trong Linux

Mặc định, hệ thống Linux có 1 hệ thống cấp quyền theo ma trận kiểm soát truy cập (Access Control List) được đơn giản hóa. Có ba quyền và ba nhóm mà các quyền có thể được gán
Quyền trong linux được phân chia như sau:

- *Quyền đọc: r (read) hay 4*
- *Quyền ghi: w (write) hay 2*
- *Quyền thực thi: x (excute) hay 1*

Các quyền được áp dụng trên ba nhóm người dùng, kí hiệu bằng ba kí tự tương ứng u, g, o

- *u = owner user = chủ sở hữu*
- *g = group = những người cùng nhóm với chủ sở hữu*
- *o = others = tất cả những người khác*

Quyền truy cập có thể thiết lập theo 2 dạng với lệnh *chmod [options] mode file*
- Mode dùng ký hiệu (symbolic): ***chmod [augo][+ -=][rwx] filename***
u (user), g (group), o (other), a (all), + thêm quyền. - bớt quyền
- Mode dùng số bát phân (octal): ***chmod [0-7][0-7][0-7] filename***

≅ *chmod -R : thay đổi cả trong thư mục con*

Chỉ những người sở hữu file mới có thể thay đổi được mức đặc quyền đối với file
chmod 777 filename
chmod g-w,o+r filename
chmod a+r filename
chmod og-x filename
chmod u+rwx filename
Thay đổi chủ sở hữu file
- Lệnh **chown** cho phép thay đổi **user** sở hữu, **group** sở hữu
- Lệnh **chgrp** cho phép thay đổi **group** sở hữu
Đổi **group** sở hữu của thư mục */tmp/data* bằng nhóm *users*
chown :users /tmp
hoặc
chgrp users /tmp
Đổi chủ sở hữu tập tin dựa trên *uid* và *gid*
chown 500:500 /tmp

≅ Không thể dùng lệnh *chmod, chown* trên phân vùng linux được định dạng *FAT32*

Access Control List

Access Control List (ACL): người dùng sở hữu file có thể gán quyền cho 1 số người dùng khác

Để xem, thay đổi ACL dùng các lệnh: setfacl, getfacl

Cú pháp: **setfacl -Rm u:<username>:rwx <path to directory>**

```
# mkdir /test
# getfacl /test
```

Thay đổi ACL user u1 và u2 trên thư mục /test

```
# setfacl -m u:u1:rwx,u:u2:rx /test
```

≅ *-m: --modify -x: --remove*

```
# getfacl /test
# su - u1
$ cd /test
$ touch abc.txt
$ exit
```

Xóa ACL user u1 trên thư mục /test

```
# setfacl -x u:u1 /test
```

≅**Câu hỏi ôn tập**

1. Nếu có một người dùng chạy lệnh *chmod -x /bin/chmod*, làm thế nào để khắc phục?

```
# chmod -x /bin/chmod
# chmod +x /bin/chmod
# cp /bin/chown /tmp/chmod
# cp /bin/chmod /tmp/chmod
# mv /tmp/chmod /bin/chmod
# ls -l /bin/chmod
# perl -e 'chmod 0755, "/bin/chmod"'
# setfacl -m u:root:rwx /bin/chmod
```

2. Cho một chương trình bash shell *hello.sh* không có quyền **execute**. Làm thế nào đê thực thi chương trình này?

```
# vim hello.sh
#!/bin/bash
echo -n "Can I have your name, please? "
read NAME
echo "Hello $NAME"
```

Thực thi bash shell

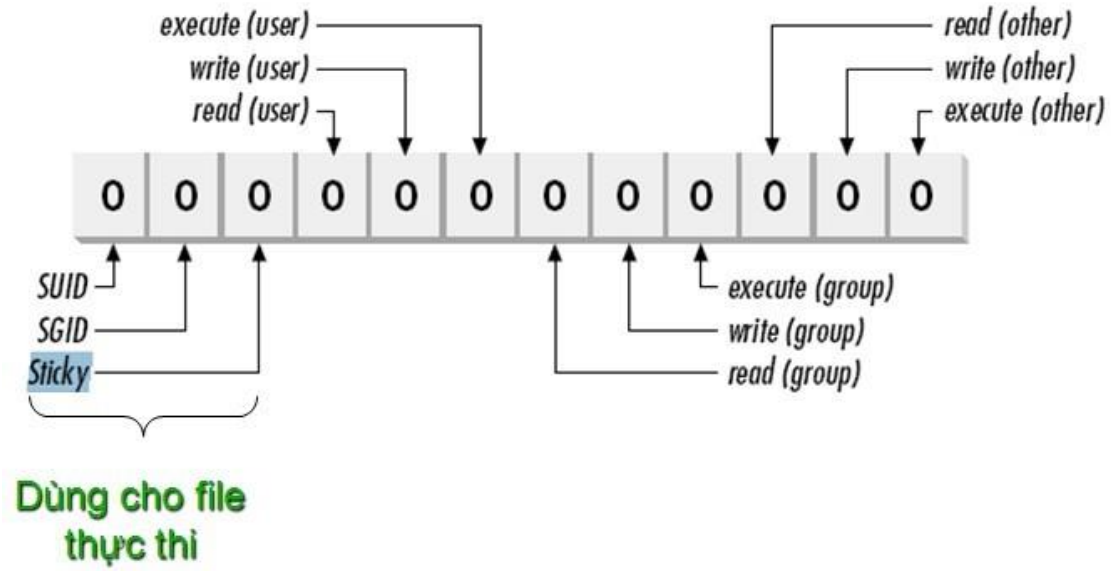
```
# bash hello.sh
# sh hello.sh
# . hello.sh
```

source hello.sh
echo $NAME
*Lệnh **.** dùng để gọi thực thi một script trong shell hiện hành với một ý nghĩa đặc biệt: thi hành và giữ nguyên những thay đổi về môi trường mà scirpt tác động*
2. Chương trình bash shell *hello.sh* không có quyền **rw** nhưng có quyền **execute**. Chương trình này có thể chạy được không?
chmod -rw,+x hello.sh
ls -l hello.sh
---x--x--x 1 root root 82 Jun 19 16:39 hello.sh
./hello.sh
Thiết lập quyền truy cập mặc định của tập tin khi chúng được tạo
Trong Linux, khi một file hay một thư mục được tạo ra thì các quyền truy cập sẽ được xác định dựa trên hai giá trị là quyền truy nhập cơ sở *(base permission)* và mặt nạ *(umask)*.
Base Permission là giá trị được thiết lập sẵn và không thể thay đổi được

- Quyền mặc định của file là **666** (rw-rw-rw-)
- Quyền mặc định của thư mục là **777** (rwxrwxrwx)

Mask là giá trị được thiết lập bởi người dùng bằng lệnh ***umask***
Giá trị Mask sẽ che đi một số bit trong Base Permission để tạo ra quyền truy cập chính thức cho file (tương tự cơ chế subnet mask)
Quyền truy cập chính thức =*"giá trị nhị phân của Base permission"AND "dạng bù 1 của mask"*
Base Permission của file bất kỳ luôn là 666 (hay 110110110 dạng nhị phân)
*Umask là 022 (hay **000010010** dạng nhị phân), dạng bù 1 **111101101***
*Quyền truy nhập của file: **110 110 110 AND 111 101 101 = 110 100 100 = 644 (rw-r–r–)***
Giá trị mask mặc định cho root = 022, quyền hạn truy cập mặc định thư mục: 755, file: 644
Giá trị mask mặc định cho user = 002, quyền hạn truy cập mặc định thư mục: 775, file: 664
Gán umask để các files được tạo ra có quyền *600*
umask 066
touch test.txt
ls -l test.txt
-rw------- 1 root root 0 Apr 7 23:25 test.txt

Quyền truy cập file đặc biệt



Dùng cho file thực thi

Working with SUID, SGID, and Sticky Bit

Permission	Numerical Value	Relative Value	On Files	On Directories
SUID	4	u+s	User executes file with permissions of file owner.	No meaning.
SGID	2	g+s	User executes file with permissions of group owner.	File created in directory gets the same group owner.
Sticky bit	1	+t	No meaning.	Users are prevented from deleting files from other users.

SUID (setuid)

```
# ls -l /usr/bin/passwd /bin/ping /usr/bin/crontab /usr/bin/at
-rwsr-xr-x 1 root root  35832 Sep 27  2009 /bin/ping
-rwsr-xr-x 1 root root  43492 Jan 27  2010 /usr/bin/at
-rwsr-sr-x 1 root root 309932 Feb 23  2012 /usr/bin/crontab
-rwsr-xr-x 1 root root  23420 Aug 11  2010 /usr/bin/passw
# chmod u-s /bin/ping
```

Thử nghiệm *ping* với user thường

SGID (setgid)

```
# mkdir /test2
# chmod 2777 /test2
# ls -ld /test2
# su - u1
$ cd /test2
$ touch abc.txt
$ ls -l
total 0
-rw-rw-r-- 1 u1 root 0 Jun  3 23:44 abc.txt
```

Bài tập phân quyền

Tạo thư mục share file cho các phòng ban

```
/data
|-- hr
|-- it
|-- public
`-- sale
```

Yêu cầu:

- *User thuộc group phòng ban nào sẽ có full quyền trên thư mục của phòng ban đó*
- *Tất cả user đều có full quyền trên thư mục* ***public***

```
# mkdir -p -m 1770 /data/sale /data/hr /data/it
# mkdir -p -m 1777 /data/public
# groupadd sale ; groupadd hr ; groupadd it
# useradd -g sale sale1 ; useradd -g hr hr1 ; useradd -g it it1
# setfacl -m g:sale:rwx /data/sale
# setfacl -m g:hr:rwx /data/hr
# setfacl -m g:it:rwx /data/it
# getfacl /data/it
# ls -l /data
```

Test

```
# su - it1 -c "mkdir /data/it/it1"
# su - it1 -c "mkdir /data/hr/it1"
# su - hr1 -c "mkdir /data/hr/hr1"
```

Chạy command bằng quyền của user xác định

```
# runuser -l "login name" -c "command"
# su - "login name" -c "command"
```

Thuộc tính immutable, append-only

Lệnh ***chattr*** thay đổi thuộc tính cho file dưới cấp độ cao cấp của một quản trị hệ thống mà lệnh ***chmod*** không thể thực hiện được

Tạo file và thiết lập thuộc tính *immutable*

```
# touch keep.me
# chattr +i keep.me
# lsattr keep.me
----i------------ keep.me
# rm -f keep.me
rm: cannot remove `keep.me': Operation not permitted
# chattr -i keep.me
```

Thiết lập thuộc tính *append-only*

```
# chattr +a keep.me
# lsattr keep.me
-----a------- keep.me
# echo "append-only set" > keep.me
-bash: keep.me: Operation not permitted
# echo "appending to file" >> keep.me
# cat keep.me
appending to file
```

Thiết lập thuộc tính append-only cho thư mục, bên trong thư mục có quyền tạo file, sửa file nhưng không có quyền xóa file

4. Chuyển đổi người dùng với lệnh su

Người dùng *root* có đặc quyền cao nhất trong các hệ thống Linux, *root* có khả năng chỉnh sửa hệ thống rất sâu theo bất kỳ cách nào, ngay cả việc chỉnh sửa các module của hệ điều hành và biên dịch lại Linux kernel, một điều không thể trên hệ thống Windows…

Mỗi một user được tạo ra trên hệ thống đều phải có một số nhận dạng gọi là UID. Linux sẽ quản lý các tài khoản người dùng thông qua UID, còn User name chỉ là tên gọi thân thiện, giúp con người dễ dàng phân biệt các user.

Tài khoản quản trị *root* được Linux tự động tạo ra với **UID=0**

```
# id root
```

Có thể sửa UID của một user về **0** để có quyền hạn ngang bằng ***root*** → để leo thang đặc quyền (***Privilege escalation***) cho một user thông thường, hacker sẽ tìm cách thay đổi **UID=0**

Lệnh **`su`** (*super user*) trên linux có chức năng thay đổi tài khoản người dùng login hiện hành Nếu không được chỉ định, lệnh sẽ yêu cầu chọn tài khoản *root* để chuyển đổi. Trong thực tế, quản trị thường dùng lệnh này để chuyển vào tài khoản *root* từ một tài khoản thông thường.

```
$ su - root
```

Cấm user *su - root*

Chỉ user thuộc nhóm ***wheel*** (*GID=10*) mới có quyền *su - root*

```
# vim /etc/pam.d/su
#%PAM-1.0
auth            sufficient      pam_rootok.so
# Uncomment the following line to implicitly trust users in the "wheel" group.
#auth           sufficient      pam_wheel.so trust use_uid
# Uncomment the following line to require a user to be in the "wheel" group.
auth            required        pam_wheel.so use_uid
# usermod -G wheel u1
```

5. Cấp quyền thực thi với sudo

Lệnh **sudo** (*do something as the supervisor*) là một thay thế nâng cao và nhiều ưu điểm hơn cơ chế chuyển sang người dùng khác bằng lệnh **su** truyền thống. Trong đó ưu điểm lớn nhất là cho phép người dùng quản trị kiểm soát được các thao tác của người dùng có can thiệp đến hệ thống, giao quyền thực thi *đúng người đúng việc*

Lệnh **sudo** cho phép cung cấp đặc quyền root thực hiện một số lệnh, thay vì tất cả các lệnh.

Các khai báo *Ai được làm gì?* được cấu hình trong file **/etc/sudoers**

```
# ls -l /etc/sudoers
-r--r----- 1 root root 3426 Dec  3 16:24 /etc/sudoers
```

Sudo là một lệnh dạng *setuid* → mọi user đều có quyền chạy lệnh này

```
# ls -l /usr/bin/sudo
---s--x--x 2 root root 174436 Mar  6  2011 /usr/bin/sudo
```

Lệnh được thực hiện qua **sudo** đều được ghi log vào file **/var/log/secure** (*Red Hat/Fedora/CentOS*) hoặc **/var/log/auth.log** (*Ubuntu/Debian*)

```
# tail -f /var/log/secure
```

Cú pháp /etc/sudoers

USER MACHINE=(TARGET-USERS) COMMANDS

- *USER: người sử dụng được quyền thực thi. Các user phân cách nhau bằng dấu phẩy ","*
- *MACHINE: máy mà quyền thực thi được áp dụng lên.*
- *TARGET-USERS: người dùng cho mượn quyền thực thi, nếu tham số này không tồn tại người dùng sudo sẽ chạy với quyền **root***
- *COMMANDS: lệnh mà người dùng sudo được quyền thực thi. Các lệnh phân cách nhau bằng dấu phẩy ","*

Nếu cấp quyền cho group, thay tham số USER bằng %GROUP.

Từ khoá *ALL* có nghĩa là tất cả các user, tất cả các nhóm, tất cả các lệnh, tất cả các máy.

Nếu các thiết lập dài hơn một dòng, có thể dùng dấu "\" để viết tiếp xuống dòng tiếp theo

Từ khoá *NOPASSWD* cung cấp khả năng thực thi lệnh mà không yêu cầu nhập password.

Tạo user quản trị **admin**

```
# useradd admin
# passwd admin
```

Khi dùng lệnh sudo, mật khẩu của root được giữ bí mật. Nếu kiểm soát vấn đề ủy quyền thông qua **sudo** không chặt chẽ thì normal user dễ dàng chiếm quyền điều khiển hệ thống với đặc quyền root

≅ Cho phép user **admin** sudo lệnh su

Thêm dòng ***admin ALL=/bin/su*** ở cuối file */etc/sudoers*

```
# visudo
admin ALL=/bin/su
```

Login với user **admin**, sau đó gõ lệnh sudo su - root rồi cung cấp mật khẩu của user admin là có quyền root mà không cần biết mật khẩu root

```
$ sudo su - root
[sudo] password for admin:
```

Password mà user admin nhập vào không phải là password root, điều này khiến cho người quản trị root biết được các hoạt động của admin.

Mặc định ***sudo*** nhớ password trong vòng **5** phút. Do đó trong khoảng thời gian này user thực hiện các thao tác khác đòi hỏi quyền root mà không cần phải nhập password.

timestamp_timeout=0 không nhớ password

timestamp_timeout=-1 chỉ cần xác thực password một lần duy nhất

≅

```
# visudo
Defaults    requiretty
```

requiretty: *If set, sudo will only run when the user is logged in to a real **tty**. When this flag is set, sudo can only be run from a login session and not via other*

*If you encounter error syntax like this: **sudo: sorry, you must have a tty to run sudo** you can disable requiretty as well in sudoers using visudo.*

Disable requiretty to particular user not to use tty

```
# visudo
Defaults:username !requiretty
```

Để xem một người dùng có thể thực thi những lệnh nào, sử dụng lệnh sau: sudo -U username -l

≅ **Bài tập cấp quyền**

Cấu hình cho phép user ***admin*** có quyền quản trị

```
# visudo
## Networking
Cmnd_Alias NETWORKING = /sbin/route, /sbin/ifconfig, /bin/ping, /sbin/dhclient,
/usr/bin/net, /sbin/iptables, /usr/bin/rfcomm, /usr/bin/wvdial, /sbin/iwconfig, /sbin/mii-tool
## Installation and management of software
Cmnd_Alias SOFTWARE = /bin/rpm, /usr/bin/up2date, /usr/bin/yum
```

Services
Cmnd_Alias SERVICES = /sbin/service, /sbin/chkconfig
Updating the locate database
Cmnd_Alias LOCATE = /usr/bin/updatedb
Storage
Cmnd_Alias STORAGE = /sbin/fdisk, /sbin/sfdisk, /sbin/parted, /sbin/partprobe, /bin/mount, /bin/umount
Delegating permissions
Cmnd_Alias DELEGATING = /usr/sbin/visudo, /bin/chown, /bin/chmod, /bin/chgrp
Processes
Cmnd_Alias PROCESSES = /bin/nice, /bin/kill, /usr/bin/kill, /usr/bin/killall
Drivers
Cmnd_Alias DRIVERS = /sbin/modprobe

admin ALL = NOPASSWD: NETWORKING, SOFTWARE, SERVICES, STORAGE, DELEGATING, PROCESSES, LOCATE, DRIVERS
Login với user ***admin*** và cài đặt package thông qua lệnh ***sudo***
$ sudo yum install mysql-server -y
$ rpm -qa | grep mysql
Khởi động ***service***
$ sudo /sbin/service mysqld start

Cấu hình cho phép user ***admin*** có toàn quyền sudo
visudo
admin ALL=(ALL) NOPASSWD:ALL
su - admin
Chuyển sang người dùng *root* với *sudo*
$ sudo -i
$ sudo -s
$ sudo bash

≅

"admin ALL=(ALL) ALL" người dùng admin, trên tất cả các máy, có thể mượn quyền tất cả các người dùng, để thực thi tất cả các lệnh.
"%admin ALL=(ALL) ALL" nhóm admin, trên tất cả các máy, có thể mượn quyền tất cả các người dùng, để thực thi tất cả các lệnh.

Tìm ra các lệnh người dùng thực thi trước đó

yum install psacct -y
service psacct start

lastcomm [commandNameHere]
lastcomm [userNameHere]
lastcomm [terminalNameHere]

Giới hạn người dùng *root* login trực tiếp hệ thống

Khai báo TTY interface mà root được phép đăng nhập

vim /etc/securetty
console
#tty1
#tty2
#tty3
#tty4
#tty5
tty6

≅ */dev/console*: TTY interface chế độ single mode