Southern Connecticut State University

SCSU

CSC424 System Administration

Instructor: Dr. Hao Wu

Week 5 User Management, Access Control and Rootly Powers

Adding Users

- Mechanically, the process of adding a new user consists of several steps required by the system and a few more that establish a useful environment for the new user and incorporate the user into your local administrative system.
- Required:
  - Edit the **passwd** and **shadow** files to define the user's account
  - Add the user to the **/etc/group** file (not really necessary, but nice)
  - Set an initial password
  - Create, **chown**, and **chmod** the user's home directory
  - Configure roles and permissions

Adding a user

- Manual maintenance of the **passwd** and **group** files is error prone and inefficient
- We use higher-level tools such as:
  - **useradd, adduser:** add a new user to system
  - **usermod:** user management
  - **passwd:** change password
  - **userdel:** delete a user

useradd, adduser

- Use ls command to check the **useradd** and **adduser** scripts

```
[root@localhost ~]# ls -l /usr/sbin/useradd /usr/sbin/adduser
lrwxrwxrwx. 1 root root      7 Feb  5 14:09 /usr/sbin/adduser -> useradd
-rwxr-x---. 1 root root 118192 Nov  5  2016 /usr/sbin/useradd
```

- adduser is actually a soft link of useradd
- Syntax for **useradd** command:

useradd [option] username

- Commonly used options:
  - g: name or ID of the primary group of the new user (group must exist)
  - G: groups

Adding a user

- Let's create a new user of your own (I'm using my name as username):
- useradd hao
- Create a passwd for your new user:
- passwd hao

```
[root@localhost ~]# passwd hao
Changing password for user hao.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

Adding a user

- When a new user is added to system using **useradd**, the system will:
  - create a home directory for user: ***/home/username***

```
[root@localhost ~]# ls -al /home
total 0
drwxr-xr-x.  3 root root  17 Feb 13 21:27 .
dr-xr-xr-x. 17 root root 224 Feb  5 14:14 ..
drwx------.  2 hao  hao   83 Feb 13 21:32 hao
```

  - copy startup files into user's home directory

```
[root@localhost ~]# ls -al /home/hao
total 16
drwx------. 2 hao  hao   83 Feb 13 21:32 .
drwxr-xr-x. 3 root root  17 Feb 13 21:27 ..
-rw-------. 1 hao  hao    9 Feb 13 21:32 .bash_history
-rw-r--r--. 1 hao  hao   18 Aug  2  2017 .bash_logout
-rw-r--r--. 1 hao  hao  193 Aug  2  2017 .bash_profile
-rw-r--r--. 1 hao  hao  231 Aug  2  2017 .bashrc
```

  - create a mailbox for user: ***/var/spool/mail/username***

Change user's password

- passwd: command to change the user's password
- Syntax:
- *passwd [username]*
  - When no username is given, system will change the password for current user
  - When username is given, system will change the password for that particular user (**Only root can change password for other users**)

Delete a user

- userdel: command to delete a user
- Syntax:
- *userdel [option] username*
- Commonly used option:
- **r:** Files in the user's home directory will be removed along with the home directory itself and the user's mail spool.

User Management

- usermod: modify a user account
- Syntax:
- *usermod [option] username*
- Commonly used options:
  - d: The user's new home directory
  - e: The date on which the user account will be disabled. The date is specified in the format YYYY-MM-DD
  - g: The group name of the user's initial login group.
  - l: Change new username
  - L: Lock a user's password
  - U: Unlock a user's password
  - m: move the content of the user's home directory to the new location.

The /etc/passwd file

- **/etc/passwd**: a file contain a list of users recognized by the system.
  - Originally, each user's encrypted password was also stored in the **/etc/passwd** file, which is world-readable
  - Now, the password is stored in **/etc/shadow**, which is not world-readable
  - The file contains 7 fields separated by colons:
    - Login name
    - Encrypted password placeholder
    - UID (user ID) number
    - Default GID (group ID) number
    - Optional "GECOS" information: full name, office, extension, home phone
    - Home directory
    - Login shell

Sample /etc/passwd file

```
root@localhost ~]# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
bin:x:1:1:bin:/bin:/sbin/nologin
daemon:x:2:2:daemon:/sbin:/sbin/nologin
adm:x:3:4:adm:/var/adm:/sbin/nologin
lp:x:4:7:lp:/var/spool/lpd:/sbin/nologin
sync:x:5:0:sync:/sbin:/bin/sync
shutdown:x:6:0:shutdown:/sbin:/sbin/shutdown
halt:x:7:0:halt:/sbin:/sbin/halt
mail:x:8:12:mail:/var/spool/mail:/sbin/nologin
operator:x:11:0:operator:/root:/sbin/nologin
games:x:12:100:games:/usr/games:/sbin/nologin
ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin
nobody:x:99:99:Nobody:/:/sbin/nologin
systemd-network:x:192:192:systemd Network Management:/:/sbin/nologin
dbus:x:81:81:System message bus:/:/sbin/nologin
polkitd:x:999:997:User for polkitd:/:/sbin/nologin
postfix:x:89:89::/var/spool/postfix:/sbin/nologin
chrony:x:998:996::/var/lib/chrony:/sbin/nologin
sshd:x:74:74:Privilege-separated SSH:/var/empty/sshd:/sbin/nologin
hao:x:1000:1000::/home/hao:/bin/bash
```

/etc/shadow file

- On Linux, the shadow password file is readable only by the superuser and serves to keep encrypted passwords safe from prying eyes and password cracking programs.

```
[root@localhost ~]# ls -al /etc/shadow
----------. 1 root root 707 Feb 13 21:34 /etc/shadow
[root@localhost ~]# ls -al /etc/passwd
-rw-r--r--. 1 root root 883 Feb 13 21:32 /etc/passwd
```

/etc/shadow file

- Nine fields for each line, separated by colons:
  - Login name
  - Encrypted password
  - Date of last password change
  - Minimum number of days between password change
  - Maximum number of days between password change
  - Number of days in advance to warn users about password expiration
  - Days after password expiration that account is disabled
  - Account expiration date
  - A field reserved for future use which is currently always empty

Sample /etc/shadow file

```
[root@localhost ~]# cat /etc/shadow
root:$6$GsG/Q1mpnPv7cnfJ$XTYn.oq2jvX896REmeQfwDfwZ2R0/
MYamT4vL0Y6uDXcNIplc0GjkX125XR0SNjUmvt7CoKXT3r9cQ.LLse730::0:99999:7:::
bin:*:17110:0:99999:7:::
daemon:*:17110:0:99999:7:::
adm:*:17110:0:99999:7:::
lp:*:17110:0:99999:7:::
sync:*:17110:0:99999:7:::
shutdown:*:17110:0:99999:7:::
halt:*:17110:0:99999:7:::
mail:*:17110:0:99999:7:::
operator:*:17110:0:99999:7:::
games:*:17110:0:99999:7:::
ftp:*:17110:0:99999:7:::
nobody:*:17110:0:99999:7:::
systemd-network:!!:17567::::::
dbus:!!:17567::::::
polkitd:!!:17567::::::
postfix:!!:17567::::::
chrony:!!:17567::::::
sshd:!!:17567::::::
hao:$6$7QdC2H5/$K.
7KBgP1zqc6XNdZ7bQClydaLHmBwYeJxjpFYVZAmbu4JOdrw5k3KKLyZARPM08gT0h1N7wlWpKih.pq0
QlNw/:17576:0:99999:7:::
```

Group Management

- groupadd: command to create a new group, syntax:
- *groupadd [options] groupName*

- groupmod: command to modify a group definition on the system, syntax:
- *groupmod [option] groupName*

- groupdel: command to delete a group, syntax:
- *groupdel [option] groupName*

The /etc/group file

- The /etc/group file contains the names of groups and a list of each group's members. Each line represents one group and contains 4 fields:
- Group name
- Encrypted password or a placeholder
- GID number
- List of members, separated by commas

Standard UNIX Access Controls

- The standard UNIX access control model has remained largely unchanged for decades. The scheme follows a few basic rules:
  - Access control decisions depend on which user is attempting to perform an operation, or in some cases, on that user's membership in a UNIX group
  - Objects(e.g., files and processes) have owners. Owners have broad (but not necessarily unrestricted) control over their objects.
  - You own the objects you create
  - The special user account called "root" can act as the owner of any object
  - Only root can perform certain sensitive administrative operations.

Standard UNIX Access Controls

- Certain systems calls are restricted to root
  - checks the identity of the current user and rejects the operation if the user is not root
- Other system calls implement different calculations that involve both ownership matching and special provisions for root
- Filesystems have their own access control system

Filesystem and process ownership

- Each file has
  - an owner
  - a group
- Each process has
  - an owner
    - can send the process signals
    - can reduce the process's scheduling priority

The root account

- The root account is UNIX's omnipotent administrative user.
- It is also known as the superuser account.
- UID of root is 0
- Traditional UNIX allows the superuser to perform any valid operation on any file or process.
- Some example of restricted operations are:
  - Creating device files
  - Setting the system clock
  - Setting system's hostname
  - Configuring network interfaces
  - Opening privileged network ports (those numbered below 1024)
  - Shutting down the system

Management of the Root Account

- Root access is required for system administration, and it's also a pivot point for system security.
- Drawbacks:
  - root logins leave no record of what operations were performed as root
    - even worse when an access was unauthorizedrized
  - log-in-as-root leaves no record of who was actually doing the work
- Most systems allow root logins to be disabled on terminals

Disable root access

- Disabling root access via any console device (tty)
  - Prevents access to the root account via the console or the network.
  - An empty /etc/securetty file prevents root login on any devices attached to the computer.
- Disabling root SSH logins
  - Prevents root access via the OpenSSH suit of tools
  - Edit the /etc/ssh/sshd_config file and set the PermitRootLogin parameter to no

su: substitute user identity

- **su**: a command to change user identity
  - a marginally better way to access the root account
  - if invoked without argument, su prompts for the root password and then starts up a root shell
  - Root privileges remain in effect until you terminate the shell by typing **<Control-D>** or the **exit** command
  - doesn't record the commands executed as root
  - create a log entry that states who became root and when
  - Can also substitute identities other than root: **su username**

sudo

- sudo: a command to execute command as root or as another restricted user
  - recommend as the primary method of access to the root account
  - /etc/sudoers lists the people who are authorized to use sudo and the commands they are allowed to run on each host
  - if the proposed command is permitted, sudo prompts for the user's own password and execute the command
  - sudo can be executed without having to type a password until a five-minute period (configurable) has elapsed with no further sudo activity
  - sudo keeps a log of the command lines that were executed, the hosts on which they were run, the people who ran them, the directories from which they were run, and the times at which they were invoked.

sudo

- Advantages
  - Accountability is much improved because of command logging
  - Users can do specific chores without having unlimited root privileges
  - The real root password can be known to only one or two people
  - Using sudo is faster than using su or logging in as root
  - Privileges can be revoked without the need to change the root password
  - A canonical list of all users with root privileges is maintained
  - The chance of a root shell being left unattended is lessened
  - A single file can control access for an entire network

chmod: change file permission

- chmod: command and system call which may change the access permissions to file system objects (files and directories). It may also alter special mode flags. The request is filtered by the umask. The name is an abbreviation of change mode.
- Syntax:
- `chmod [options] mode[,mode] file1 [file2, …]`
- Options:
  - R: recursive, i.e., files in the subdirectories
  - f: force

chmod: Octal modes

- The `chmod` numerical format accepts up to four octal digits. The three rightmost digits refer to permissions for the file owner, the group, and other users.

#	Permission	rwx
7	Read, write, execute	rwx
6	Read, write	rw-
5	Read, execute	r-x
4	Read only	r- -
3	Write, execute	-wx
2	Write only	-w-
1	Execute	- - x
0	None	—-

Chmod references

- The references (or classes) are used to distinguish the users to whom the permissions apply. If no references are specified it defaults to “all” but modifies only the permissions allowed by the `umask`. The references are represented by one or more of the following letters:

Reference	Class	Description
u	Owner	File’s owner
g	group	users who are members of the file's group
o	Other	users who are neither the file's owner nor members of the file's group
a	All	all three of the above, same as `ugo`

chmod

- The `chmod` program uses an operator to specify how the modes of a file should be adjusted. The following operators are accepted:=

Operator	Description
+	adds the specified modes to the specified classes
-	removes the specified modes from the specified classes
=	the modes specified are to be made the exact modes for the specified classes

chown: change owner

- chown: command to change the owner of the file
- syntax:
- `chown [option] [owner][:group] file [file2 …]`

- chgrp: command to change the group of the file
- syntax:
- `chgrp [option] group file`