Chapter: 4 Managing Local Users and Groups

OBJECTIVES

- Describe the purpose of users and groups on a Linux system.
- Switch to the superuser account to manage a Linux system, and grant other users superuser access using the sudo command.
- Create, modify, and delete locally defined user accounts.
- Create, modify, and delete locally defined group accounts.
- Set a password management policy for users, and manually lock and unlock user accounts.

DESCRIBING USER AND GROUP CONCEPTS

WHAT IS A USER?

A user account is used to provide security boundaries between different people and programs that can run commands.

There are three main types of user account: the superuser, system users, and regular users.

- The superuser account is for administration of the system. The name of the superuser is root and the account has UID 0. The superuser has full access to the system.
- The system has system user accounts which are used by processes that provide supporting services. These processes, or daemons, usually do not need to run as the superuser. They are assiged non-privileged accounts that allow them to secure their files and other resources from each other and from regular users on the system. Users do not interactively log in using a system user account.
- Most users have regular user accounts which they use for their day-to-day work. Like system users, regular users have limited access to the system.

id command to show information about the currently logged-in user.

id

The system distinguishes user accounts by the unique identification number assigned to them, the user ID or UID.

Systems use the **/etc/passwd** file to store information about local users.

WHAT IS A GROUP?

A group is a collection of users that need to share access to files and other system resources. Groups can be used to grant access to files to a set of users instead of just a single user.

The system distinguishes groups by the unique identification number assigned to them, the group ID or GID.

Systems use the **/etc/group** file to store information about local groups.

GAINING SUPERUSER ACCESS

OBJECTIVES

After completing this section, you will be able to switch to the superuser account to manage a Linux system, and grant other users superuser access through the sudo command.

SWITCHING USERS

The su command allows users to switch to a different user account. If you run su from a regular user account, you will be prompted for the password of the account to which you want to switch. When root runs su, you do not need to enter the user's password.

[user01@host ~]$ su - user02

Password:

[user02@host ~]$

If you omit the user name, the su or su - command attempts to switch to root by default.

[user01@host ~]$ su –

Password:

[root@host ~]#

MANAGING LOCAL USER ACCOUNTS

OBJECTIVES

After completing this section, you should be able to create, modify, and delete local user accounts.

MANAGING LOCAL USERS

Creating Users from the Command Line

useradd username

Modifying Existing Users from the Command Line

usermod options username

USERMOD OPTIONS:	USAGE
`-c, --comment COMMENT`	Add the user's real name to the comment field.
`-g, --gid GROUP`	Specify the primary group for the user account.

Deleting Users from the Command Line

- The **userdel username** command removes the details of username from /etc/passwd, but leaves the user's home directory intact.
- The **userdel -r username** command removes the details of username from /etc/passwd and also deletes the user's home directory.

Setting Passwords from the Command Line

The **passwd username** command sets the initial password or changes the existing password of username.

MANAGING LOCAL GROUPS

Creating Groups from the Command Line

groupadd groupname

Modifying Existing Groups from the Command Line

groupmod options groupname

Deleting Groups from the Command Line

groupdel groupname

OBJECTIVES

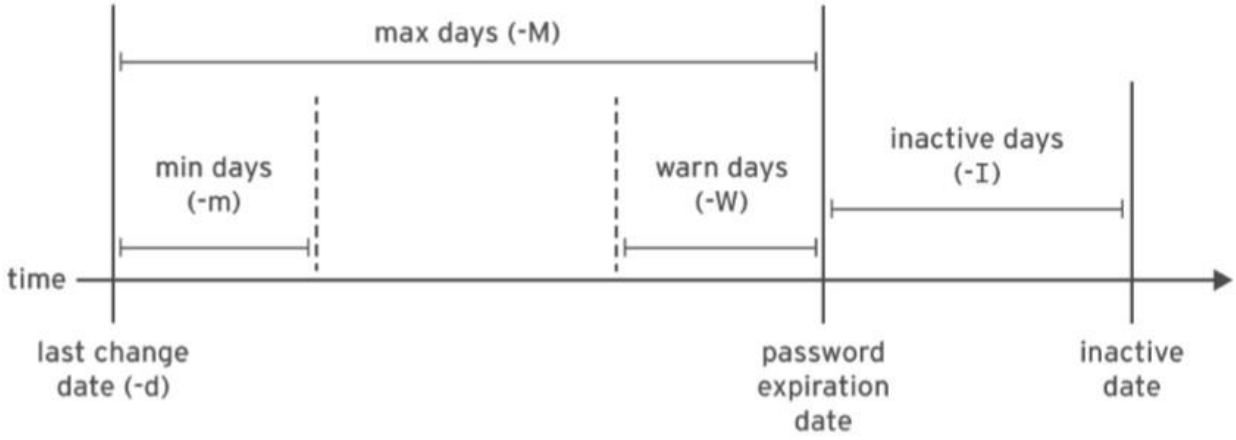
After completing this section, you should be able to set a password management policy for users, and manually lock and unlock user accounts.

SHADOW PASSWORDS AND PASSWORD POLICY

At one time, encrypted passwords were stored in the world-readable **/etc/passwd** file. This was thought to be reasonably secure until dictionary attacks on encrypted passwords became common. At that point, the encrypted passwords were moved to a separate **/etc/shadow** file which is readable only by root.

CONFIGURING PASSWORD AGING

chage options username



RESTRICTING ACCESS

The nologin

Shell The nologin shell acts as a replacement shell for the user accounts not intended to interactively log into the system.

usermod -s /sbin/nologin username

CONFIGURING SUDO

The main configuration file for sudo is /etc/sudoers. To avoid problems if multiple administrators try to edit it at the same time, it should only be edited with the special visudo command.

For example, the following line from the /etc/sudoers file enables sudo access for members of group wheel.

%wheel ALL=(ALL) ALL

In this line, %wheel is the user or group to whom the rule applies. A % specifies that this is a group, group wheel. The ALL=(ALL) specifies that on any host that might have this file, wheel can run any command. The final ALL specifies that wheel can run those commands as any user on the system.

By default, /etc/sudoers also includes the contents of any files in the /etc/sudoers.d directory as part of the configuration file. This allows an administrator to add sudo access for a user simply by putting an appropriate file in that directory.

To enable full sudo access for the user user01, you could create /etc/sudoers.d/user01 with the following content: user01 ALL=(ALL) ALL

To enable full sudo access for the group group01, you could create /etc/sudoers.d/group01 with the following content:

%group01 ALL=(ALL) ALL

It is also possible to set up sudo to allow a user to run commands as another user without entering their password:

ansible ALL=(ALL) NOPASSWD:ALL