6.7.3 Linux User Commands and Files

This lesson covers the following topics:

- User files
- User management commands

User Files

Linux is extremely flexible regarding where user and group information is stored. The options for storing the information are:

- Local file system
- LDAP-compliant database
- Network Information System (NIS). NIS allows many Linux computers to share a common set of user accounts, group accounts, and passwords.
- A Windows domain

When files are stored in the local file system, the following files are used:

File	Description
/etc/passwd	The /etc/passwd file contains the user account information. Each user's information is stored in a single line on this file. There are two types of accounts in a Linux system: ▪ Standard accounts (these are user accounts). ▪ System user accounts (these are used by services).
/etc/shadow	The /etc/shadow file contains the users' passwords in an encrypted format. The shadow file is linked to the /etc/passwd file. There are corresponding entries in both files, and they must stay synchronized. There are password and user management utilities provided by the system that allow you to edit the files and keep them synchronized. You can use the following commands to identify errors and synchronize the files: ▪ **pwck** verifies each line in the two files and identifies discrepancies. ▪ **pwconv** adds the necessary information to synchronize the files.
/etc/group	As with Active Directory, groups can be used to simplify user access to network resources. The /etc/group file contains information about each group.

Be aware of the following configuration files when managing user accounts:

File	Description
/etc/default/useradd	The /etc/default/useradd file contains default values used by the **useradd** utility when creating a user account, including: ▪ Group ID ▪ Home directory ▪ Account expiration ▪ Default shell

	▪ Secondary group membership
/etc/login.defs	The /etc/login.defs file contains: ▪ Values used for the group and user ID numbers ▪ Parameters for password encryption in the shadow file ▪ Password expiration values for user accounts
/etc/skel	The /etc/skel directory contains a set of configuration file templates that are copied into a new user's home directory when it is created, including the following files: ▪ .bashrc ▪ .bash_logout ▪ .bash_profile ▪ .kshrc

User Management Commands

Although it is possible to edit the /etc/passwd and /etc/shadow files manually to manage user accounts, doing so can disable your system. Instead, use the following commands to manage user accounts:

If you are logged in as the root user, the commands in the table can be run by typing the applicable command and its options. However, if you are not logged in as the root user, you will need to use the *sudo* or *su* command to gain the permissions required.
For example, to create a new user named Kim Sanders, you would run: **sudo useradd -c "Kim Sanders" -m ksanders**

Command	Command Function
useradd	Create a user account. The following options override the settings as found in /etc/default/useradd: ▪ **-c** adds a description for the account in the GECOS field of /etc/passwd. ▪ **-d** assigns an absolute pathname to a custom home directory location. ▪ **-D** displays the default values specified in the /etc/default/useradd file. ▪ **-e** specifies the date on which the user account will be disabled. ▪ **-f** specifies the number of days after a password expires until the account is permanently disabled. ▪ **-g** defines the primary group membership. ▪ **-M** defines the secondary group membership. ▪ **-m** creates the user's home directory (if it does not exist). ▪ **-n** does not create a group with the same name as the user (Red Hat and Fedora, respectively). ▪ **-p** defines the encrypted password. ▪ **-r** specifies that the user account is a system user. ▪ **-s** defines the default shell. ▪ **-u** assigns the user a custom UID. This is useful when assigning ownership of files and directories to a different user.
passwd	Assign or change a password for a user. ▪ **passwd** (without a user name or options) changes the current user's password

	▪ **passwd** (without a user name or options) changes the current user's password. ▪ Users can change their own passwords. The root user can execute all other **passwd** commands. Be aware of the following options: ▪ **-S username** displays the status of the user account. LK indicates that the user account is locked, and PS indicates that the user account has a password. ▪ **-l** disables (locks) an account. This command inserts a !! before the password in the /etc/shadow file, effectively disabling the account. ▪ **-u** enables (unlocks) an account. ▪ **-d** removes the password from an account. ▪ **-n** sets the minimum number of days after a password exists before it can be changed. ▪ **-x** sets the number of days before a user must change the password (password expiration time). ▪ **-w** sets the number of days before the password expires that the user is warned. ▪ **-t** sets the number of days following the password expiration that the account will be disabled.
usermod	Used to modify an existing user account; **usermod** uses several of the same switches as **useradd**. Be aware of the following switches: ▪ **-c** changes the description for the account. ▪ **-l** renames a user account. ▪ **-L** locks the user account. This command inserts a ! before the password in the /etc/shadow file, effectively disabling the account. ▪ **-U** unlocks the user account.
userdel	Remove the user from the system. Be aware of the following options: ▪ **userdel** ***[username]*** (without options) removes the user account. ▪ **-r** removes the user's home directory. ▪ **-f** forces the removal of the user account even when the user is logged into the system.