8.9.3 Linux User Commands and 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.
- NIS, network information system. NIS allows many Linux computers to share a common set of user accounts, group accounts, and passwords.
- A Windows domain.

When the 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 in this file. The syntax for the file is: *USER:PW:UID:GID:FULL_NAME:HOME:SHELL* There are two types of accounts in a Linux system: ▪ Standard accounts that are user accounts ▪ System user accounts that are used by services
/etc/shadow	The **/etc/shadow** file contains the users' passwords in encrypted format. The shadow file is linked to the **/etc/passwd** file. There are corresponding entries in both files, and they must stay synchronized. The syntax for the file is: *USER:PASSWD:LASTMOD:MINDAYS:MAXDAYS:WARN:DIS:EXP* A single exclamation mark (!) or double exclamation marks (!!) in the password field indicates that the account is locked. There are password and user management utilities provided by the system that will 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. The syntax for the file is: *GROUP:PASSWORD:GID:USERS*
/etc/gshadow	Some distributions use the **/etc/gshadow** file to store group passwords. The syntax for the file is: *GROUP:PASS:GROUP_ADMINS:MEMBERS*

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 passwords 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**

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:

Command	Command Function	Example
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. ▪ **-G** defines the secondary group membership. ▪ **-M** does not create the user's home directory. ▪ **-m** creates the user's home directory (if it does not exist). ▪ **-n, 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.	**useradd pmaxwell** creates the *pmaxwell* user account**.** **useradd -c "Paul Morril" pmorril** creates the *pmorril* account with a comment**.** **useradd -d /tmpusr/sales1 sales1** creates the *sales1* user account with the home directory located at */tmpusr/sales1*. **useradd -u 789 dphilips** creates the *dphilips* account with user ID *789*.

passwd	Assign or change a password for a user.
	▪ **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.
	  ▪ 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 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.
	▪ **-i** sets the number of days following the password expiration that the account will be disabled.
		passwd jsmith changes the password for the *jsmith* account.
		passwd -d removes the password from an account.
		passwd -d jsmith removes the password from the *jsmith* account.
		passwd -x 40 jsmith requires *jsmith* to change his password every 40 days.
		passwd -n 10 jsmith makes it so that *jsmith* cannot change his password for 10 days following the most recent change.
		passwd -w 2 jsmith means that *jsmith* will be warned two days before his password expires.
		passwd -i 7 jsmith disables the *jsmith* account after seven days if the password is not changed.
		passwd -l jsmith locks the *jsmith* account.
		passwd -u jsmith unlocks the *jsmith* account.

usermod	Modify an existing user account. **usermod** uses several of the same switches as **useradd**. Be aware of the following switches:	**usermod -c "Paul Morril" pmorril** changes the comment field for user *pmorril*. **usermod -l esmith -d /home/esmith -m ejones** renames the *ejones*

	▪ **-c** changes the description for the account.
	▪ **-l** renames a user account. When renaming the account:
	  ▪ Use **-d** to rename the home directory.
	  ▪ Use **-m** to copy all files from the existing home directory to the new home directory.
	▪ **-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.
		account *esmith*, renames the home directory, and moves the old home directory contents to the new location.
		usermod -s /bin/tsch esmith points the shell for *esmith* to */bin/tsch*.
		usermod -U esmith unlocks the *esmith* 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 in to the system.	**userdel pmaxwell** deletes the *pmaxwell* account while leaving the home directory on the hard drive. **userdel -r pmorril** removes both the account and the home directory.