

## Lab 3: User Accounts and Processes

In this lab, you will examine processes and create user accounts. Each team will turn in answers to the questions in this lab

### Boot the system

Step 1. Bring the system up to multi-user mode if it is not already.

### Explore the password and shadow files

Learn about formats of the key user account files

Step 2. Examine the password file **/etc/passwd** to become familiar with its format. Be sure to identify each of the fields.

**Q1.** How many colons are in each line in the file? \_\_\_\_\_

Step 3. Examine the shadow file **/etc/shadow** to become familiar with its format also. Be sure to identify each of the fields. Note that the first two fields match the first two fields of the **/etc/passwd** file.

**Q2.** How many fields are in the **/etc/shadow** file? \_\_\_\_\_

**Q3.** What are the other fields in the **/etc/shadow** file? \_\_\_\_\_

Step 4. Examine the group membership file **/etc/group** like the others, and identify each of the fields.

### Create a new user account

Learn how to manually add users to the system by editing the key account files.

Step 5. To create a new user, a new record in **/etc/passwd** must be created. Recall that each record has seven fields. Each of these fields needs to be filled in with values appropriate to the new user. The seven fields you need to set are shown in the table below. For this lab, use the values listed under the **Value to Use** column. Review the table to be sure you understand the values to use.

Field	Description	Value to Use
Login Name	Enter the new user's login account name here.	<b>joeuser</b>
Password	Place either the letter <b>x</b> or a <b>*</b> here. If your system has an <b>/etc/shadow</b> file, you are using shadow (hidden) passwords. Place an <b>x</b> in the encrypted password field to indicate that the password is really stored in the <b>/etc/shadow</b> file. If you do not have an <b>/etc/shadow</b> file, place the character <b>*</b> in this field, to indicate that the login is currently disabled. This will prevent a small window of time where the user has no password - the account is disabled until you are done creating it, and the real encrypted password will be filled in later by the <b>passwd</b> command.	<b>x</b>
User Id	Enter an unused UID, typically above 100.	<b>1234</b>
Group ID	Enter the GID you want the user to belong to.	<b>6789</b>
GECOS	The GECOS field is also known as the Real Name field. Enter the user's full real name.	<b>Joe User</b>
Home Directory	Enter the full path to the home directory for the user.	<b>/home/joeuser</b>
Login Shell	Enter the shell the user will use by default.	<b>/bin/bash</b>

Edit **/etc/passwd** to create a new user account for user **joeuser**. Use **vipw** (not **vi**) to edit **/etc/passwd**. This will run **vi** (or the editor you have configured with the environment variable **EDITOR**), but will create a **cooperative lock** on **/etc/passwd** to prevent simultaneous edits. The lock is cooperative meaning that it only works if everyone uses **vipw**, so get in the habit of using **vipw**! Here are some **vi** tips to help you edit the password file quickly.

Tips: Once in **vi**, the easiest way to create a new user is to copy and paste an existing entry. Go to the end of the file with the **G** command. You can quickly copy and paste **vi** by *yanking* and *putting* the line you want to duplicate. Try the following 3 keystrokes: **GYP** - this will **go** (G) to the end of the file, **yank** (Y) the current line, and **put** (P) it after

the yanked line. The copied line is now the last line in the file. Then move the cursor into each field of the new user you are creating (**f**: will bring you to the next colon). Start with the first field, the login name. You can quickly change the value in the field with the **Change Word** command - the keystrokes are **cw**. Enter **cw** and just start typing in the value, hitting the Escape key when the field is entered. Move the cursor to the beginning of the next field, and repeat the process.

Step 6. When the new password record looks like the line below, save your changes and quit the editor:

```
joeuser:x:1234:6789:Joe User:/home/joeuser:/bin/bash
```

Step 7. Our Linux installation uses **/etc/shadow**, so you have to add an entry for the user. The procedure for this is similar to that of **/etc/passwd**, but the fields of each record are not the same. Edit **/etc/shadow**, and copy the last entry using the same process as you did above. Then, change only the first two fields of your new entry (we are going to ignore the other fields for now). The first field is the user name, and the second field is the encrypted password. Set the user name to **joeuser** and the password to **\***.

Step 8. Set the user's initial password by running the **passwd** command and enter a password for the user:

```
# passwd joeuser
```

Step 9. Create the users home directory, using the **mkdir** command:

```
# mkdir /home/joeuser
#
```

Step 10. Copy the standard skeleton startup files for the user into the user's home directory. On Red Hat, they are in **/etc/skel**. Use **ls -a** to see the dot files. For your user, just copy the three **.bash\*** files.

```
# cp /etc/skel/.bash* /home/joeuser
```

Step 11. Edit **/etc/group** using your favorite editor. The fields in the **/etc/group** file are listed in the table below. Use the values specified in the table below.

Field	Description	Value to Use
Group Name	Enter the name of the new group.	<b>cis68c1</b>
Password	Group passwords typically are not used - place an <b>x</b> here to disable group passwords (which are rarely ever used).	<b>x</b>
Group ID	Enter a unique new group id.	<b>6789</b>
Group Members	Enter a comma-separated list of login names who are to be members of this group.	<b>joeuser</b>

The new group record should look like:

```
cis68c1:x:6789:joeuser:
```

Note the trailing **'**, which is normal, but opposite of the **/etc/passwd** file.

Step 12. Change the owner, group, and permissions of the user's home directory and files. Use the UID and GID values you entered in the password entry. Use **700** as the default value for permissions.

```
# chown -R joeuser /home/joeuser
# chgrp -R cis68c1 /home/joeuser
# chmod 700 /home/joeuser
# chmod 644 /home/joeuser/.??*
```

Step 13. Check your entries in **/etc/passwd**, **/etc/shadow**, and **/etc/group**. Check that you have copied the skeleton dot files. And very importantly, check that the user's file and directory permissions are correct, especially the permissions of **/home** (did you give away ownership of all of **/home** to **joemailer**?). The commands below will help you with validation.

```
# ls -la /home/joemailer
# grep joemailer /etc/passwd
# grep joemailer /etc/shadow
# grep cis68c1 /etc/group
```

Step 14. The account should now be ready. Log into the account to be sure the login works. Log off as the **root** user, and then try to log in to the newly created account. Don't use **su**, since as **root**, you will not be required to enter a password - you should validate that the password you created actually works. After you have logged in as **joemailer**, try the commands below:

```
# who am i
# last | head -2
```

Step 15. Logout of the system.

### Examine processes

*Learn to use the output from the **ps** and **top** commands to manage processes and the system*

Step 16. Login as **root**, and run the commands below. Examine the output:

```
# vi &
# ps -el
```

**Q4.** What state is the **vi** process in? \_\_\_\_\_

**Q5.** What state is the **init** process in? \_\_\_\_\_

Step 17. Find the PID for the **vi** command you started above

Step 18. Send the **vi** process an **INT** signal.

**Q6.** What is the command line you enter to do this? \_\_\_\_\_

**Q7.** What happens when the **INT** signal is sent to **vi**? \_\_\_\_\_

Step 19. Send the **vi** process an **TERM** signal.

**Q8.** What happens when a **TERM** signal is sent to **vi**? \_\_\_\_\_

**Q9.** Which process has the lowest nice value? What is its name and value \_\_\_\_\_

**Q10.** Which TTYs are being used by the **mingetty** processes? \_\_\_\_\_

**Q11.** What process is the parent of your shell process? Give the process name and its PID. \_\_\_\_\_

Step 20. Switch to another virtual terminal using **Control-Alt-F2** (hold down the Control and Alt keys simultaneously, and hit the Function 2 key). Login as user **joemailer**.

Step 21. Examine the output from **ps** again, and look at the processes owned by **joemailer**.

**Q12.** What is the relationship between the process you found for question Q11, and the **mingetty** processes? \_\_\_\_\_

\_\_\_\_\_

Step 22. Run the **top** command.

**Q13.** How much memory does the system have? \_\_\_\_\_

**Q14.** How much RAM is currently unused? \_\_\_\_\_

**Q15.** What percentage of time is the CPU currently idle? \_\_\_\_\_

**Q16.** Which process is currently the process using the most CPU time? \_\_\_\_\_

Step 23. The **top** command can filter processes – let's look at only **joeuser**'s processes. While **top** is running, type **u** followed immediately by the user name **joeuser** and hit Enter. You should now see a list of only those processes for **joeuser**.

Step 24. You can have **top** refresh more frequently. Type **s1** and then by Enter. This will set the time period to refresh every second. You can also use the space bar to force a refresh. Use **top** with care – **top** is fairly expensive to run, so you should not leave **top** processes running needlessly.