**Chapter 6 GUIDED EXERCISE**

**GAINING SUPERUSER ACCESS**

In this exercise, you will practice switching to the root account and running commands as root.

**OUTCOMES**

You should be able to:

• Use **sudo** to switch to root and access the interactive shell as root without knowing the password of the superuser.

• Explain how **su** and **su -** can affect the shell environment through running or not running the login scripts.

• Use **sudo** to run other commands as root.

**BEFORE YOU BEGIN**

Log in to workstation as student using student as the password.

On workstation, run **lab users-sudo start** to start the exercise. This script creates the necessary user accounts and files to set up the environment correctly.

[student@workstation ~]$ **lab users-sudo start**

**1.** From workstation, open an SSH session to servera as student.

[student@workstation ~]$ **ssh student@servera**

*...output omitted...*

[student@servera ~]$

**2.** Explore the shell environment of student. View the current user and group information and display the current working directory. Also view the environment variables that specify the user's home directory and the locations of the user's executables.

2.1. Run **id** to view the current user and group information. When finished, redirect the output to file ~/id.student and verify.

[student@servera ~]$ **id**

uid=1000(student) gid=1000(student) groups=1000(student),10(wheel)

context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

[student@servera ~]$ **id >~/id.student**

[student@servera ~]$ **cat ~/id.student**

uid=1000(student) gid=1000(student) groups=1000(student),10(wheel) context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

[student@servera ~]$

2.2. Run **pwd** to display the current working directory. When finished, redirect the output to file ~/pwd.student and verify.

[student@servera ~]$ **pwd**

/home/student

[student@servera ~]$ **pwd >~/pwd.student**

[student@servera ~]$ **cat ~/pwd.student**

/home/student

[student@servera ~]$

2.3. Print the values of the HOME and PATH variables to determine the home directory and user executables' path, respectively. When finished, redirect the output to file ~/HOME.student and ~/PATH.student verify.

[student@servera ~]$ **echo $HOME**

/home/student

[student@servera ~]$ **echo $PATH**

/home/student/.local/bin:/home/student/bin:/usr/local/bin:/usr/bin:/

usr/local/sbin:/usr/sbin

[student@servera ~]$ **echo $HOME >~/HOME.student**

[student@servera ~]$ **cat ~/HOME.student**

/home/student

[student@servera ~]$ **echo $PATH >~/PATH.student**

[student@servera ~]$ **cat ~/PATH.student**

/home/student/.local/bin:/home/student/bin:/usr/local/bin:/usr/bin:/usr/local/sbin:/usr/sbin

[student@servera ~]$

**3.** Switch to root in a non-login shell and explore the new shell environment.

3.1. Run **sudo su** at the shell prompt to become the root user.

[student@servera ~]$ **sudo su**

[sudo] password for student: **student**

[root@servera student]#

3.2. Run **id** to view the current user and group information. When finished, redirect the output to file ~/id.su and verify.

[root@servera student]# **id**

uid=0(root) gid=0(root) groups=0(root)

context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

[root@servera student]# **id >~/id.su**

[root@servera student]# **cat ~/id.su**

uid=0(root) gid=0(root) groups=0(root) context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

[root@servera student]#

3.3. Run **pwd** to display the current working directory. When finished, redirect the output to file ~/pwd.su and verify.

[root@servera student]# **pwd**

/home/student

[root@servera student]# **pwd >~/pwd.su**

[root@servera student]#

[root@servera student]# **cat ~/pwd.su**

/home/student

[root@servera student]#

3.4. Print the values of the HOME and PATH variables to determine the home directory and user executables' path, respectively. When finished, redirect the output to file ~/HOME.su and ~/PATH.su, and verify.

[root@servera student]# **echo $HOME**

/root

[root@servera student]# **echo $PATH**

/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin

[root@servera student]# **echo $HOME > ~/HOME.su**

[root@servera student]# **cat ~/HOME.su**

/root

[root@servera student]# **echo $PATH > ~/PATH.su**

[root@servera student]#

[root@servera student]# **cat ~/PATH.su**

/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin

[root@servera student]#

If you already have some experience with Linux and the **su** command, you may have expected that using **su** without the dash (**-**) option to become root would cause you to keep the current PATH of student. That did not happen. As you will see in the next step, this is not the usual PATH for root either.

What happened? The difference is that you did not run **su** directly. Instead, you ran **su** as root using **sudo** because you did not possess the password of the superuser. The **sudo** command initially overrides the PATH variable from the initial environment for security reasons. Any command that runs after the initial override can still update the PATH variable, as you will see in the following steps.

3.5. Exit the root user's shell to return to the student user's shell.

[root@servera student]# **exit**

exit

[student@servera ~]$

**4.** Switch to root in a login shell and explore the new shell environment.

4.1. Run **sudo su -** at the shell prompt to become the root user.

[student@servera ~]$ **sudo su -**

[root@servera ~]#

Notice the difference in the shell prompt compared to that of **sudo su** in the preceding step.

**sudo** may or may not prompt you for the student password, depending on the time-out period of **sudo**.

The default time-out period is five minutes. If you have authenticated to **sudo** within the last five minutes, **sudo** will not prompt you for the password. If it has been more than five minutes since you authenticated to **sudo**, you need to enter student as the password to get authenticated to **sudo**.

4.2. Run **id** to view the current user and group information. When finished, redirect the output to file ~/id.root and verify.

[root@servera ~]# **id**

uid=0(root) gid=0(root) groups=0(root)

context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

[root@servera ~]# **id > ~/id.root**

[root@servera ~]#

[root@servera ~]# **cat ~/id.root**

uid=0(root) gid=0(root) groups=0(root) context=unconfined\_u:unconfined\_r:unconfined\_t:s0-s0:c0.c1023

[root@servera ~]#

4.3. Run **pwd** to display the current working directory. When finished, redirect the output to file ~/pwd.root and verify.

[root@servera ~]# **pwd**

/root

[root@servera ~]#

[root@servera ~]# **pwd >~/pwd.root**

[root@servera ~]# **cat ~/pwd.root**

/root

[root@servera ~]#

4.4. Print the values of the HOME and PATH variables to determine the home directory

and the user executables' path, respectively. When finished, redirect the output to file ~/HOME.root and ~/PATH.root, and verify.

[root@servera ~]# **echo $HOME**

/root

[root@servera ~]# **echo $PATH**

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

[root@servera ~]# **echo $HOME >~/HOME.root**

[root@servera ~]# **cat ~/HOME.root**

/root

[root@servera ~]# **echo $PATH >~/PATH.root**

[root@servera ~]# **cat ~/PATH.root**

/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/root/bin

[root@servera ~]#

As in the preceding step, after **sudo** reset the PATH variable from the settings in the student user's shell environment, the **su -** command ran the shell login scripts for root and set the PATH variable to yet another value. The **su** command without the dash (**-**) option did not do that.

4.5. Exit the root user's shell to return to the student user's shell.

[root@servera ~]# **exit**

logout

[student@servera ~]$

**5.** Verify that the operator1 user is configured as to run any command as any user using

**sudo**.

[student@servera ~]$ **sudo cat /etc/sudoers.d/operator1**

operator1 ALL=(ALL) ALL

**6.** Become operator1 and view the contents of **/var/log/messages**. Copy **/etc/motd** to **/etc/motdOLD** and remove it (/etc/motdOLD). These operations require administrative rights and so use **sudo** to run those commands as the superuser. Do not switch to root using **sudo su** or **sudo su -**. Use redhat as the password of operator1.

6.1. Switch to operator1.

[student@servera ~]$ **su - operator1**

Password: **redhat**

[operator1@servera ~]$

6.2. Attempt to view the last five lines of **/var/log/messages** without using **sudo**. This should fail. When finished, redirect the output to file ~/step6.2.operator1 and verify. Note the ampersand used with redirection. The **\*** metacharacter redirects both standard out and standard error.

[operator1@servera ~]$ **tail -5 /var/log/messages**

tail: cannot open '/var/log/messages' for reading: Permission denied

[operator1@servera ~]$ **tail -5 /var/log/messages &>~/step6.2.operator1**

[operator1@servera ~]$

[operator1@servera ~]$ **cat step6.2.operator1**

tail: cannot open '/var/log/messages' for reading: Permission denied

[operator1@servera ~]$

6.3. Attempt to view the last five lines of **/var/log/messages** with **sudo**. This should

succeed. When finished, redirect the output to file ~/step6.3.operator1 and verify.

[operator1@servera ~]$ **sudo tail -5 /var/log/messages**

[sudo] password for operator1: **redhat**

Jan 23 15:53:36 servera su[2304]: FAILED SU (to operator1) student on pts/1

Jan 23 15:53:51 servera su[2307]: FAILED SU (to operator1) student on pts/1

Jan 23 15:53:58 servera su[2310]: FAILED SU (to operator1) student on pts/1

Jan 23 15:54:12 servera su[2322]: (to operator1) student on pts/1

Jan 23 15:54:25 servera su[2353]: (to operator1) student on pts/1

[operator1@servera ~]$ **sudo tail -5 /var/log/messages \ &>~/step6.3.operator1**

[operator1@servera ~]$ **cat ~/step6.3.operator1**

Dec 10 07:41:09 servera systemd[1]: Starting dnf makecache...

Dec 10 07:41:10 servera dnf[11635]: Metadata cache refreshed recently.

Dec 10 07:41:10 servera systemd[1]: Started dnf makecache.

Dec 10 07:44:02 servera su[11772]: (to operator1) student on pts/1

Dec 10 07:44:23 servera su[11811]: (to operator1) student on pts/1

[operator1@servera ~]$

**NOTE**

The preceding output may differ on your system.

6.4. Attempt to make a copy of **/etc/motd** as **/etc/motdOLD** without using **sudo**. This should fail. When finished, redirect the output to file ~/step6.4.operator1 and verify.

[operator1@servera ~]$ **cp /etc/motd /etc/motdOLD**

cp: cannot create regular file '/etc/motdOLD': Permission denied

[operator1@servera ~]$ **cp /etc/motd /etc/motdOLD &>~/step6.4.operator1**

[operator1@servera ~]$ **cat ~/step6.4.operator1**

cp: cannot create regular file '/etc/motdOLD': Permission denied

[operator1@servera ~]$

6.5. Attempt to make a copy of **/etc/motd** as **/etc/motdOLD** with **sudo**. This should succeed. When finished, redirect the output to file ~/step6.5.operator1 and verify.

[operator1@servera ~]$ **sudo cp /etc/motd /etc/motdOLD**

[operator1@servera ~]$

[operator1@servera ~]$ **sudo cp /etc/motd /etc/motdOLD \ &>~/step6.5.operator1**

[sudo] password for operator1:

[operator1@servera ~]$ **cat ~/step6.5.operator1**

[operator1@servera ~]$

6.6. Attempt to delete **/etc/motdOLD** without using **sudo**. Note: this step does not require redirection to a file.

[operator1@servera ~]$ **rm /etc/motdOLD**

rm: remove write-protected regular empty file '/etc/motdOLD'? **y**

rm: cannot remove '/etc/motdOLD': Permission denied

[operator1@servera ~]$

6.7. Attempt to delete **/etc/motdOLD** with **sudo**. This should succeed. When finished, redirect the output to file ~/step6.7.operator1 and verify.

[operator1@servera ~]$ **sudo rm /etc/motdOLD**

[operator1@servera ~]$

[operator1@servera ~]$ **sudo rm /etc/motdOLD &>~/step6.7.operator1**

[operator1@servera ~]$ **cat ~/step6.7.operator1**

[operator1@servera ~]$

6.8. Exit the operator1 user's shell to return to the student user's shell.

[operator1@servera ~]$ **exit**

logout

[student@servera ~]$

6.9. Log off from servera.

[student@servera ~]$ **exit**

logout

Connection to servera closed.

[student@workstation ~]$

**7. Evaluation**

On workstation, run the **lab users-sudo grade** script to complete this exercise.

[student@workstation ~]$ **lab users-sudo grade**

**8. Finish**

On workstation, run **lab users-sudo finish** to complete this exercise. This script deletes the user accounts and files created at the start of the exercise to ensure that the environment is clean.

[student@workstation ~]$ **lab users-sudo finish**

This concludes the guided exercise.