Table of Contents

Users and Groups Lab

1. Run Commands as `root`

In this lab, you practice running commands as `root`.

Use `su` with and without login scripts to switch users. Use `sudo` to run commands with privilege.

1. Log in to `server1.example.com` as `student` with the password `r3dh@t1!`.
2. Explore characteristics of the current student login environment.
   a. View the user and group information and display the current working directory.

```
[student@server1 ~]$ id
uid=1000(student) gid=1000(student) groups=1000(student),10(wheel)
context=unconfined_u:nconfined_r:unconfined_t:s0-s0:c0.c1023
[student@server1 ~]$ pwd
/home/student
```

   b. View the variables which specify the `home directory` and the locations searched for executable files.

```
[student@server1 ~]$ echo $HOME
/home/student
[student@server1 ~]$ echo $PATH
/usr/local/sbin:usr/local/bin:/usr/sbin:/usr/bin:/home/student/.local/b
in:/home/student/bin
```

3. Switch to **root** without the dash and explore characteristics of the new environment.

   a. Become the **root** user at the shell prompt.

```
[student@server1 ~]$ su
Password:r3dh@t1!
```

   b. View the user and group information and display the current working directory. Note the identity changed, but not the current working directory.

```
[root@server1 student]# id
uid=0(root) gid=0(root) groups=0(root)
context=unconfined_u:nconfined_r:unconfined_t:s0-s0:c0.c1023
[root@server1 student]# pwd
/home/student
```

   c. View the variables which specify the **home directory** and the locations searched for executable files. Look for references to the **student** and **root** accounts.

```
[root@server1 student]# echo $HOME
/root
[root@server1 student]# echo $PATH
/usr/local/sbin:usr/local/bin:/usr/sbin:/usr/bin:/home/student/.local/b
in:/home/student/bin
```

   d. Exit the shell to return to the **student** user.

```
[root@server1 student]# exit
exit
```

4. Switch to **root** with the dash and explore characteristics of the new environment.

a. Become the **root** user at the shell prompt. Be sure all the login scripts are also executed.

```
[student@server1 ~]$ su -
Password:r3dh@t1!
```

b. View the user and group information and display the current working directory.

```
[root@server1 ~]# id
uid=0(root) gid=0(root) groups=0(root)
context=unconfined_u:nconfined_r:unconfined_t:s0-s0:c0.c1023
[root@server1 ~]# pwd
/root
```

c. View the variables which specify the **home directory** and the locations searched for executable files. Look for references to the **student** and **root** accounts.

```
[root@server1 ~]# echo $HOME
/root
[root@server1 ~]# echo $PATH
/usr/local/sbin:usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin:/root/bin
```

d. Add the **student** user to the **wheel** group.

```
[root@server1 ~]# usermod student -aG wheel
```

By default, in Red Hat Enterprise Linux 7, users in the **wheel** group have access to run any command as **root** using the **sudo** utility.

e. On the **desktop1.example.com** system become **root** and use the **usermod** command to add **student** to the **wheel** group.

```
[student@desktop1 ~]$ su -
[root@desktop1 ~]# usermod student -aG wheel
```

f. Log completely out of the **desktop1.example.com** and **server1.example.com** hosts and then log back in again (this is required for the new **sudo** permission to take effect).

```
[root@server1 ~]# exit
logout
[student@server1 ~]$ exit
```

```
[root@desktop1 ~]# exit
logout
[student@desktop1 ~]$ exit
```

5. As **student**, run several commands that require root access.

   a. View the last 5 lines of **/var/log/messages**.

```
[student@server1 ~]$ tail -5 /var/log/messages
tail:cannot open ‘/var/log/messages’ for reading: Permission denied
[student@server1 ~]$ sudo tail -5 /var/log/messages
Feb  3 15:7:22 localhost su: (to root) root on pts/0
Feb  3 15:0:01 localhost systemd: Starting Session 31 of user root.
Feb  3 15:0:01 localhost systemd: Started Session 31 of user root.
Feb  3 15:2:05 localhost su: (to root) root on pts/0
Feb  3 15:4:47 localhost su: (to student) root on pts/0
```

   b. Make a backup of a configuration file in the **/etc** directory.

```
[student@server1 ~]$ cp /etc/motd /etc/motdOLD
cp:cannot create regular file ‘/etc/motdOLD’: Permission denied
[student@server1 ~]$ sudo cp /etc/motd /etc/motdOLD
```

   c. Remove the **/etc/motdOLD** file that was just created.

```
[student@server1 ~]$ rm /etc/motdOLD
rm:remove write-protected regular empty file ‘/etc/motdOLD’? y
rm:cannot remove ‘/etc/motdOLD’: Permission denied
[student@server1 ~]$ sudo rm /etc/motdOLD
```

   d. Edit a configuration file in the **/etc** directory.

```
[student@server1 ~]$ echo "Welcome to class" >> /etc/motd
-bash:/etc/motd: Permission denied
[student@server1 ~]$ sudo vim /etc/motd
```

e. Type **i**, then **Welcome to class**, press **ESC** then **:wq** to enter the text into **motd**.

---

2. Create Users Using Command-Line Tools

In this lab, you create a number of users on your **server1.example.com** system, setting and recording an initial password for each user.

1. Log in to **server1.example.com** as **student** with the password **student**.
2. Become the **root** user.

```
[student@server1 ~]$ su -
Password:r3dh@t1!
```

3. Add the user **juliet**.

```
[root@server1 ~]# useradd juliet
```

4. Confirm that **juliet** has been added by examining the **/etc/passwd** file.

```
[root@server1 ~]# tail -1 /etc/passwd
juliet::1001:1001::/home/juliet:/bin/bash
```

5. Use the **passwd** command to initialize **juliet**'s password.

```
[root@server1 ~]# passwd juliet
Changing password for user juliet.
New password:juliet
BAD PASSWORD:The password is shorter than 8 characters
Retype new password:juliet
passwd:all authentication tokens updated successfully.
```

6. Continue adding the remaining users in the steps below and set initial passwords.

romeo

```
[root@server1 ~]# useradd romeo
[root@server1 ~]# passwd romeo
Changing password for user romeo.
New password:romeo
BAD PASSWORD:The password is shorter than 8 characters
Retype new password:romeo
passwd:all authentication tokens updated successfully.
```

hamlet

```
[root@server1 ~]# useradd hamlet
[root@server1 ~]# passwd hamlet
```

reba

```
[root@server1 ~]# useradd reba
[root@server1 ~]# passwd reba
```

dolly

```
[root@server1 ~]# useradd dolly
[root@server1 ~]# passwd dolly
```

elvis

```
[root@server1 ~]# useradd elvis
[root@server1 ~]# passwd elvis
```

---

3. Manage Groups Using Command-Line Tools

In this lab, you add users to newly created supplementary groups.

1. Perform the following steps on **`server1.example.com`**.
2. Become the **`root`** user at the shell prompt.

```
[student@server1 ~]$ su -
Password:r3dh@t1!
```

3. Create a supplementary group called **shakespeare** with a group ID of **30000**.

```
[root@server1 ~]# groupadd -g 30000 shakespeare
```

4. Create a supplementary group called **artists**.

```
[root@server1 ~]# groupadd artists
```

5. Confirm that **shakespeare** and **artists** have been added by examining the **/etc/group** file.

```
[root@server1 ~]# tail -5 /etc/group
reba::1004:
dolly::1005:
elvis::1006:
shakespeare::30000:
artists::30001:
```

6. Add **juliet** to the **shakespeare** group as a supplementary group.

```
[root@server1 ~]# usermod -G shakespeare juliet
```

7. Confirm that **juliet** has been added using the **id** command.

```
[root@server1 ~]# id juliet
uid=1001(juliet) gid=1001(juliet) groups=1001(juliet),30000(shakespeare)
```

8. Continue adding the remaining users to groups as follows:
   a. Add **romeo** and **hamlet** to the **shakespeare** group.

```
[root@server1 ~]# usermod -G shakespeare romeo
[root@server1 ~]# usermod -G shakespeare hamlet
```

b. Add `reba`, `dolly`, and `elvis` to the `artists` group.

```
[root@server1 ~]# usermod -G artists reba
[root@server1 ~]# usermod -G artists dolly
[root@server1 ~]# usermod -G artists elvis
```

c. Verify the supplemental group memberships by examining the `/etc/group` file.

```
[root@server1 ~]# tail -5 /etc/group
reba::1004:
dolly::1005:
elvis::1006:
shakespeare::30000:juliet,romeo,hamlet
artists::30001:reba,dolly,elvis
```

---

4. Manage User Password Aging

In this lab, you set unique password policies for users.

Perform the following steps on `server1.example.com`.

1. Make sure you are not root (use `exit` until you see the $ prompt).
2. Explore locking and unlocking accounts.
   a. Lock the `romeo` account.

```
[student@server1 ~]$ sudo usermod -L romeo
```

   b. Attempt to log in as `romeo`.

```
[student@server1 ~]$ su - romeo
Password:romeo
su:Authentication failure
```

   c. Unlock the `romeo` account.

```
[student@server1 ~]$ sudo usermod -U romeo
```

3. Change the password policy for **romeo** to require a new password every 90 days.

```
[student@server1 ~]$ sudo chage -M 90 romeo
[student@server1 ~]$ sudo chage -l romeo
Last password change                                    :Feb 03, 2014
Password expires                                        :May 04, 2014
Password inactive                                       :never
Account expires                                         :never
Minimum number of days between password change          :0
Maximum number of days between password change          :90
Number of days of warning before password expires       :7
```

4. Additionally, force a password change on the first login for the **romeo** account.

```
[student@server1 ~]$ sudo chage -d 0 romeo
```

5. Log in as **romeo** and change the password to **forsooth123**.

```
[student@server1 ~]$ su - romeo
Password:romeo
You are required to change your password immediately (root enforced)
Changing password for romeo.
(current) UNIX password:romeo
New password:forsooth123
Retype new password:forsooth123
[romeo@server1 ~]$ exit
```

6. Expire accounts in the future.

   a. Determine a date 180 days in the future.

```
[student@server1 ~]$ date -d "+180 days"
Sat Aug  2 17:5:20 EDT 2014
```

   b. Set accounts to expire on that date.

```
[student@server1 ~]$ sudo chage -E 2014-08-02 romeo
[student@server1 ~]$ sudo chage -l romeo
Last password change                                    :Feb 03, 2014
Password expires                                        :May 04, 2014
Password inactive                                       :never
Account expires                                         :Aug 02, 2014
Minimum number of days between password change          :0
Maximum number of days between password change          :90
Number of days of warning before password expires       :7
```

5. Connect to a Central LDAP and Kerberos Server

In this lab, you configure your **`desktop1`** system to become a client of the LDAP server running on **`instructor.example.com`**. You configure your **`desktop1`** system to use the Kerberos infrastructure provided by **`instructor.example.com`** for additional authentication.

Lab Resources

`http://instructor.example.com/pub/example-ca.crt`

To simplify user management, your company has decided to switch to centralized user management. Another team has already set up all the required LDAP and Kerberos services. Centralized home directories are not yet available, so the system should be configured to create local home directories when a user first logs in.

Given the following information, configure your **`desktop1`** system to use user information from the LDAP server, and authentication services from the Kerberos KDC. DNS service records for the realm have not yet been configured, so you will have to configure Kerberos settings manually.

Name	Value
LDAP server	**`ldap:/instructor.example.com`**
LDAP base DN	**`dc=example,dc=com`**
Use TLS	Yes
Root CA	**`http://instructor.example.com/pub/example-ca.crt`**
Kerberos realm	**`EXAMPLE.COM`**

Kerberos KDC	`instructor.example.com`
Kerberos admin server	`instructor.example.com`

1. Reset your `desktop1` system.
   a. Start by installing the necessary packages: `sssd`, `krb5-workstation`, and `authconfig-gtk`.

```
[student@desktop1 ~]$ sudo yum -y install sssd authconfig-gtk krb5-workstation
```

2. Launch the **Authentication Configuration** application, then apply the settings from the table for both LDAP and Kerberos options.
   a. Launch `system-config-authentication` from the command line. Enter the `root` password (`r3dh@t1!`) when asked.
   b. Make sure the **Identity & Authentication** tab is open.
   c. In the **User Account Database**, select **LDAP**.
   d. Enter `dc=example,dc=com` in the **LDAP Search Base DN** field, and `instructor.example.com` in the **LDAP Server** field.
   e. Make sure the **Use TLS to encrypt connections** box is checked, then click the **Download CA Certificate…** button.
   f. Enter `http://instructor.example.com/pub/example-ca.crt` in the **Certificate URL** field, then click **OK**.
   g. Select **Kerberos password** from the **Authentication Method** dropdown, and uncheck both **Use DNS…** boxes.
   h. Enter `EXAMPLE.COM` in the **REALM** field, and `instructor.example.com` in both the **KDCs** and **Admin Servers** fields.
   i. Switch to the **Advanced Options** tab and check the **Create home directories on the first login** box.
   j. Click the **Apply** button.
3. Use both `getent` and `ssh` to verify your work. You can use the username `ldapuser1` (where `1` is your station number) with the password `kerberos`. Please note that your users will not yet have a home directory mounted.

```
[student@desktop1 ~]$ getent passwd ldapuser1
ldapuser1::170X:170X:LDAP Test User X:/home/guests/ldapuser1:/bin/bash
```

```
[student@desktop1 ~]$ ssh ldapuser1@localhost
The authenticity of host 'localhost (:1)' can't be established.
EDCSA key fingerprint is XX:X:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX:XX.
Are you sure you want to continue connecting (yes/no)? yes
Warning:Permanently added 'localhost' (ECDSA) to the list of known hosts.
ldapuser1@localhost's password:kerberos
Creating home directory for ldapuser1.
[ldapuser1@desktop1$ ]$ pwd
/home/guests/ldapuser1
[ldapuser1@desktop1$ ]$ ls -a
.   .bash_history  .bash_profile  .cache   .mozilla
..  .bash_logout   .bashrc        .config
[ldapuser1@desktop1$ ]$ logout
```