CompTIA Security+® Lab Series

Lab 17: Authentication, Authorization and Access Control

CompTIA Security+® Domain 5 - Access Control and Identity Management

Objective 5.3: Explain the fundamental concepts and best practices related to authentication, authorization and access control.

Document Version: 2013-08-02

Organization: Moraine Valley Community College
Author: Jesse Varsalone

The original works of this document were funded by the National Science Foundation's (NSF) Advanced Technological Education (ATE) program Department of Undergraduate Education (DUE) Award No. 0702872 and 1002746; Center for Systems Security and Information Assurance (CSSIA) at Moraine Valley Community College (MVCC).

Contents

Introduction

This lab is part of a series of lab exercises designed through a grant initiative by the Center for Systems Security and Information Assurance (CSSIA) and the Network Development Group (NDG), funded by the National Science Foundation's (NSF) Advanced Technological Education (ATE) program Department of Undergraduate Education (DUE) Award No. 0702872 and 1002746. This work has been adapted by The Department of Labor (DOL) Trade Adjustment Assistance Community College and Career Training (TAACCCT) Grant No. TC-22525-11-60-A-48. This series of lab exercises is intended to support courseware for CompTIA Security+® certification.

By the end of this lab, students will be able to implement permissions on files and folders to both allow and restrict users from accessing them. The ability to use file and folder security is critical for keeping an operating system and it resources secure.

This lab includes the following tasks:

2 - Adding Groups, Users and Passwords
3 - Symbolic Permissions
4 - Absolute Permissions

Objective: Explain the fundamental concepts and best practices related to authentication, authorization and access control

You may have read articles online describing situations where information was improperly accessed on systems. Information security means just that - information needs to be secure. That goal is achieved when individuals understand how to effectively implement permissions.

passwd file –User accounts on a Linux system are listed in the passwd file, which is stored in the /etc directory. The passwd file has less restrictive permissions than the shadow file because it does not store the encrypted password hashes. On most Linux systems, any account has the ability to read the contents of the passwd file.

shadow file – The shadow file also stores information about user's accounts on a Linux system. The shadow file also stores the encrypted password hashes, and has more restrictive permissions than the passwd file. On most Linux systems, only the root account has the ability to read the contents of the shadow file.

chmod – The chmod command can be used to change permissions on a file or folder. The chmod command can be used regardless of whether permissions are set using absolute or symbolic permissions. The root and other accounts can use chmod.

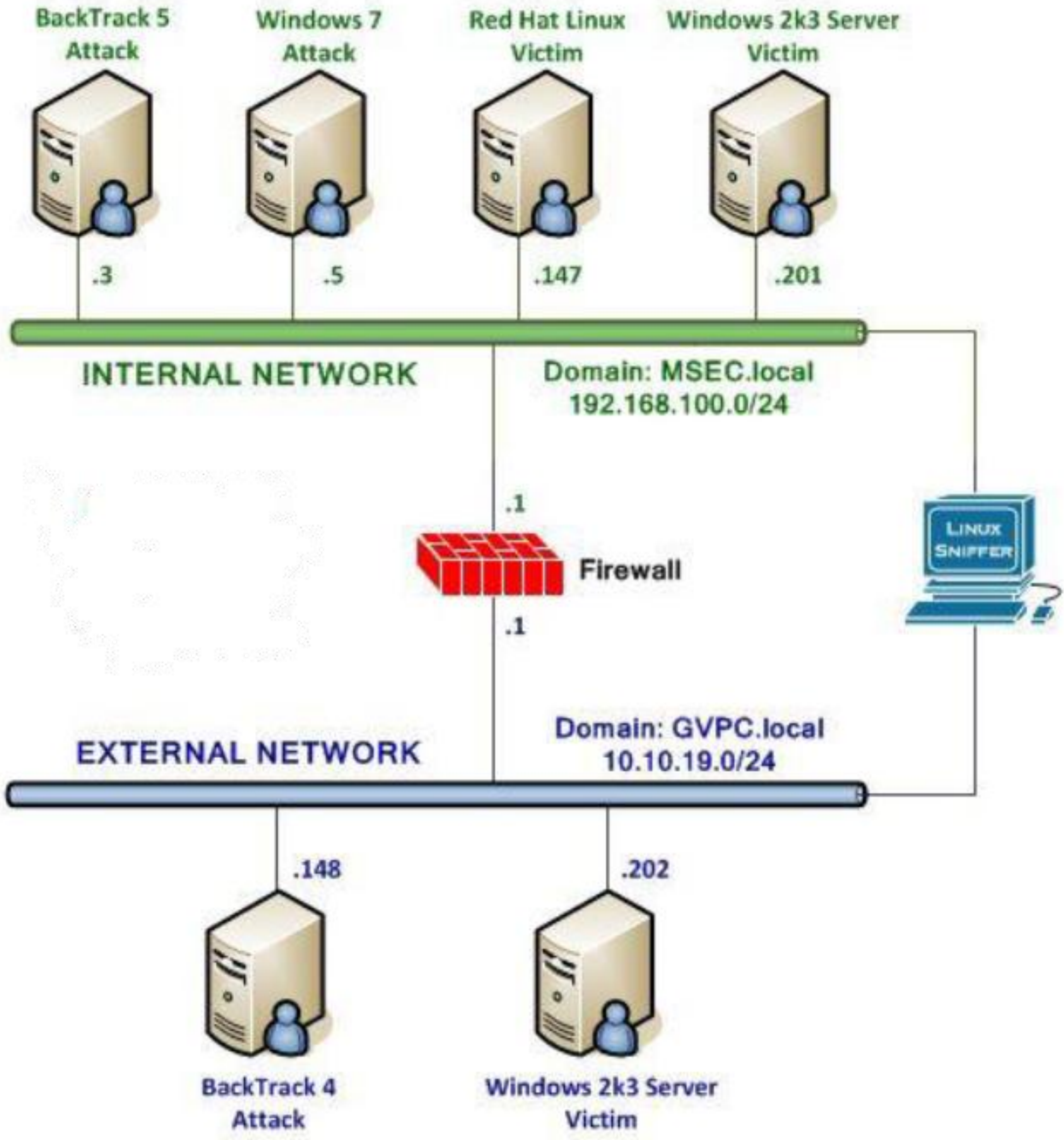
useradd – the useradd command can be used to add a user to the system. When the useradd command is utilized, a directory is created for the user in the /home folder.

groupadd – Creating the groups before the users is generally a good practice for Linux administrators. The groups are stored in the group file in the /etc directory.

Symbolic Permissions: Notation used with the chmod command to change the permissions of files and directories, using *r*, *w*, and *x* to represent *read*, *write*, and *execute* permissions. The first character indicates file type. The remaining nine characters are separated into three sets, each representing a class of permissions as three characters. The first set represents the *user* class. The second set represents the *group* class. The third set represents the *others* class. See 3 for examples of symbolic permissions.

Absolute Permissions: To specify permissions using absolute mode a three-digit octal number is used to represent the permissions for owner, group, and other. Set 4 for examples of absolute permissions.

Pod Topology



1

Figure 1: Topology

Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Required Virtual Machines and Applications

Log in to the following virtual machine before starting the tasks in this lab:

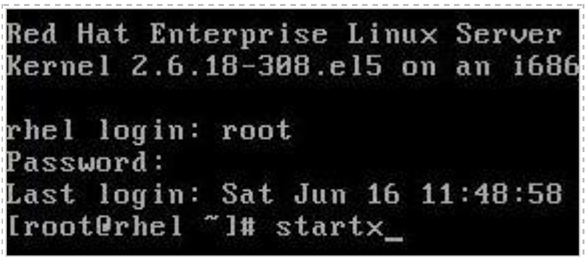
Red Hat Linux Internal Victim Machine	192.168.100.147
Red Hat Linux root password	password

Red Hat Enterprise Linux Internal Victim Login:

1. Click on the **Red Hat Linux Internal Victim** icon on the topology.
2. Type **root** at the **rhel login:** prompt and press **Enter**.
3. Type **password** at the **Password:** prompt and press **Enter**.

For security purposes, the password will not be displayed.

4. To start the GUI, type **startx** at the **[root@rhe ~]#** prompt and press **Enter**.



```
Red Hat Enterprise Linux Server
Kernel 2.6.18-308.el5 on an i686

rhel login: root
Password:
Last login: Sat Jun 16 11:48:58
[root@rhel ~]# startx_
```

Figure 2: RHEL login

2 Adding Groups, Users and Passwords

Performing account administration on a Linux system is a straightforward process, requiring several basic steps, which will be illustrated in the sections below:

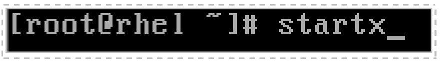
- Create the groups
- Create user accounts, adding them to the group as they are created
- Assign passwords to the accounts

Keep in mind that Linux commands are case sensitive. The commands below must be entered exactly as shown.

2.1 Adding Groups, Users and Passwords to a Linux System

1. Click on the Red Hat Linux Internal Victim Machine. Type the following command to launch the Graphical User Interface:
   [root@rhel ~]#**startx**

If you have already logged into the machine, you may skip this step.



```
[root@rhel ~]# startx_
```

Figure 3: Starting the X Server

2. In the Red Hat Linux Internal Victim Machine, right-click on the desktop and select **Open Terminal** to open a terminal. Another way to open the terminal is by clicking on the blue box next to the **System** tab.

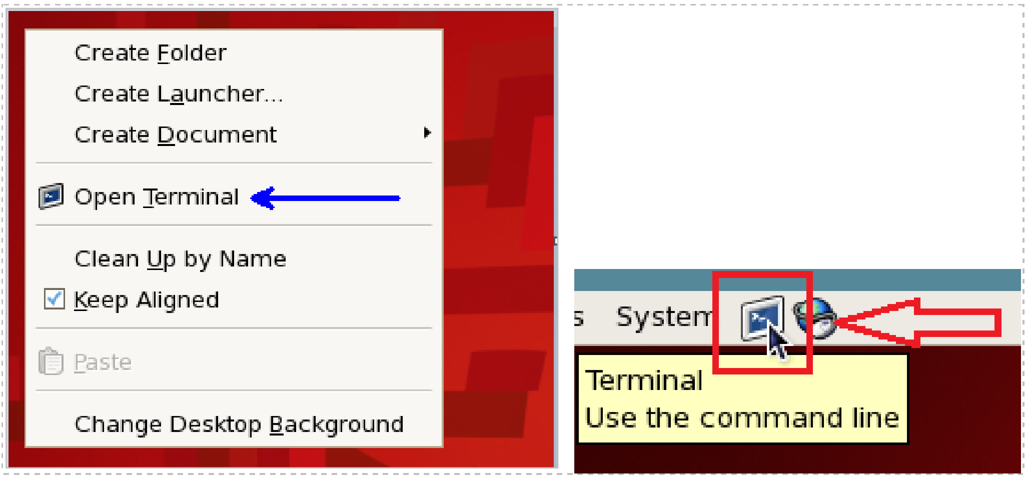
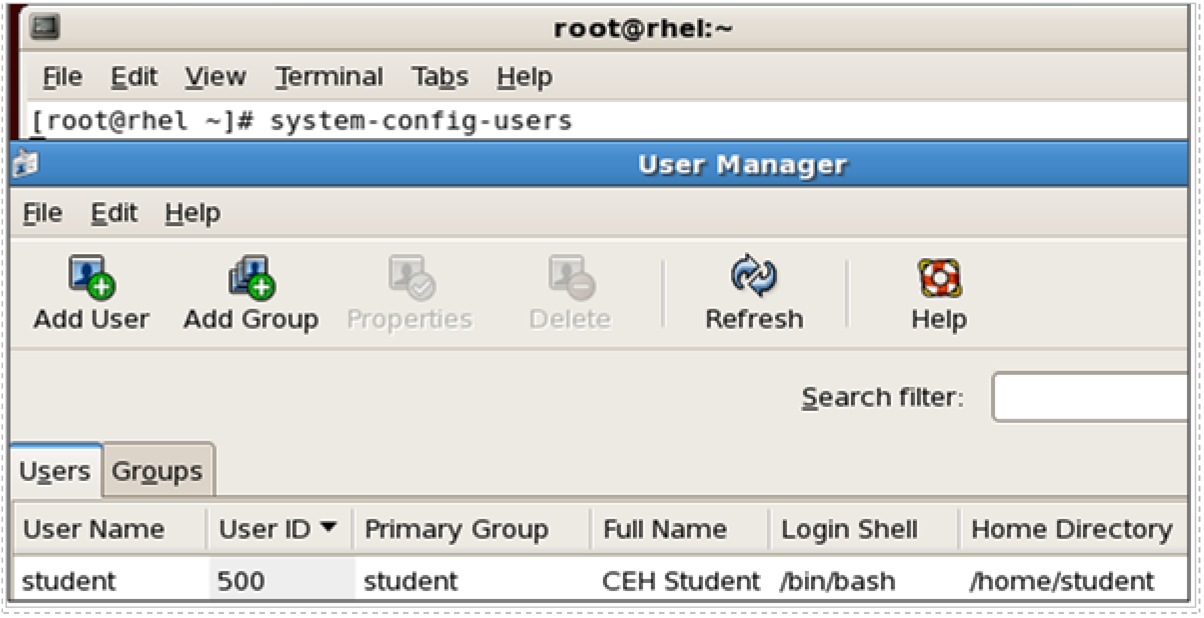


Figure 4: Opening a Terminal on Linux

3. Type the following command to view the user accounts on the system:
   [root@rhel ~]#**system-config-users**



```
[root@rhel ~]# system-config-users
```

User Name	User ID ▾	Primary Group	Full Name	Login Shell	Home Directory
student	500	student	CEH Student	/bin/bash	/home/student

Figure 5: User Accounts on the Linux System

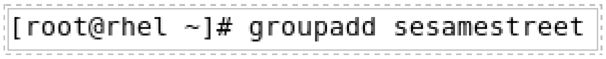
First, we will be creating two groups, **sesamestreet** and **simpsons**. We will be creating a total of six users with three users in each group. After creating each user and putting them in their corresponding group, we will assign each user account a password.

The charts below show a summary of the users, groups and passwords for our accounts.

Group: sesamestreet	
User	**Password**
elmo	red
cookie	blue
oscar	green

Group: simpsons	
User	**Password**
bart	boy
lisa	girl
homer	man

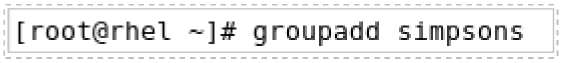
4. Type the following command to add the group **sesamestreet**:
   [root@rhel ~]#**groupadd sesamestreet**



```
[root@rhel ~]# groupadd sesamestreet
```

Figure 6: Adding the Group sesamestreet

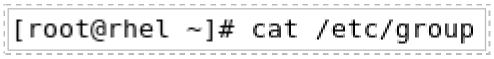
5. Type the following command to add the group **simpsons**:
   [root@rhel ~]#**groupadd simpsons**



```
[root@rhel ~]# groupadd simpsons
```

Figure 7: Adding the Group simpsons

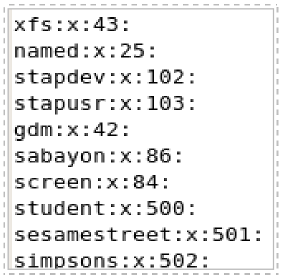
6. Type the following command to add the view the group file:
   [root@rhel ~]#**cat /etc/group**



```
[root@rhel ~]# cat /etc/group
```

Figure 8: Viewing the Group File

If you scroll to the bottom of the group file, you will see the groups that were created along with their corresponding unique group number. Note: The root group has an id of zero.

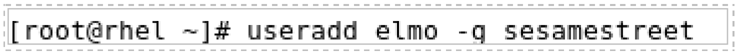


```
xfs:x:43:
named:x:25:
stapdev:x:102:
stapusr:x:103:
gdm:x:42:
sabayon:x:86:
screen:x:84:
student:x:500:
sesamestreet:x:501:
simpsons:x:502:
```

Figure 9: The group file

You can add users to the system in Linux by typing the **useradd** command. The **–g** command switch can be used to add the user to a group at the time of creation. The **useradd** command will automatically create a directory with that user's name within the */home* directory. When the user logs in, they will be placed into their directory within */home*. This is analogous to a user's profile in Windows except that the user's profile is not created until a user log into a Windows machine.

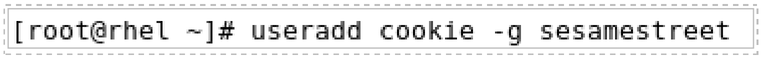
7. To add a user named **elmo** and put him in the **sesamestreet** group, type:
   [root@rhel ~]#**useradd elmo –g sesamestreet**



```
[root@rhel ~]# useradd elmo -g sesamestreet
```

Figure 10: Adding the user elmo

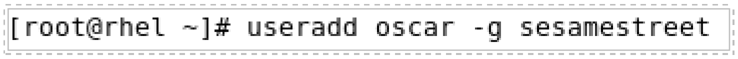
8. To add a user named **cookie** and put him in the **sesamestreet** group, type:
   [root@rhel ~]#**useradd cookie –g sesamestreet**



```
[root@rhel ~]# useradd cookie -g sesamestreet
```

Figure 11: Adding the user cookie

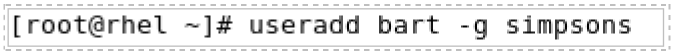
9. To add a user named **oscar** and put him in the **sesamestreet** group, type:
[root@rhel ~]#**useradd oscar –g sesamestreet**



```
[root@rhel ~]# useradd oscar -g sesamestreet
```

Figure 12: Adding the user oscar

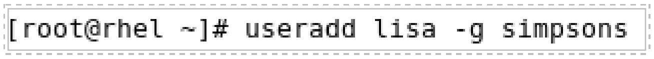
10. To add a user named **bart** and put him in the **simpsons** group, type:
[root@rhel ~]#**useradd bart –g simpsons**



```
[root@rhel ~]# useradd bart -g simpsons
```

Figure 13: Adding the user bart

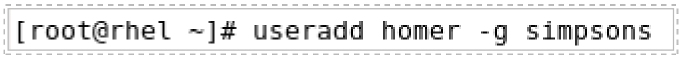
11. To add a user named **lisa** and put her in the **simpsons** group, type:
[root@rhel ~]#**useradd lisa –g simpsons**



```
[root@rhel ~]# useradd lisa -g simpsons
```

Figure 14: Adding the user lisa

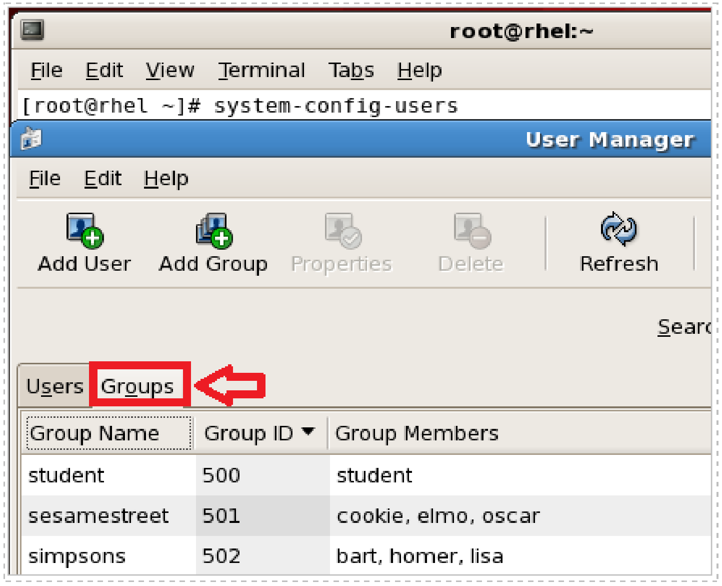
12. To add a user named **homer** and put him in the **simpsons** group, type:
[root@rhel ~]#**useradd homer –g simpsons**



```
[root@rhel ~]# useradd homer -g simpsons
```

Figure 15: Adding the user homer

13. Type the following command to view the user accounts on the system:
[root@rhel ~]#**system-config-users**



```
[root@rhel ~]# system-config-users
```

Group Name	Group ID ▼	Group Members
student	500	student
sesamestreet	501	cookie, elmo, oscar
simpsons	502	bart, homer, lisa

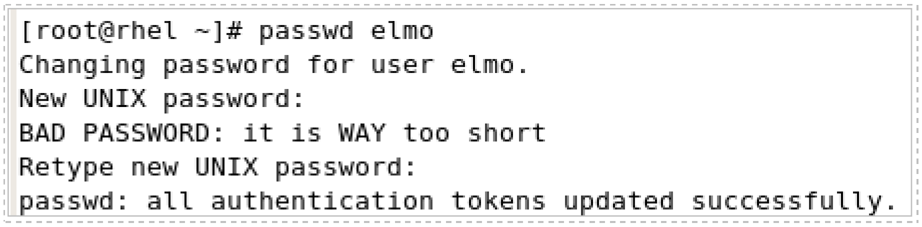
Figure 16: Viewing the Groups in the User Manager

When groups are added first, followed by users being added and put into the groups as they are created, you will have a structure where permissions can be set effectively.

Next, we will give each user a password. We will use simple passwords for this exercise, but that should never be done on a production system. Avoid dictionary words because attackers can use programs like *John the Ripper* to crack short passwords or passwords that are found in a dictionary. Stick to passwords with a minimum of eight characters, uppercase and lowercase letters, and special characters. When you use a simple password with the **passwd** command, you will be warned that the password is a "BAD PASSWORD: it is WAY too short". Retype the password again and it will be accepted.

For security reasons, passwords will not be displayed when you type them.

14. Type the following to give **elmo** a password. Type **red** twice as the password.
    [root@rhel ~]#**passwd elmo**

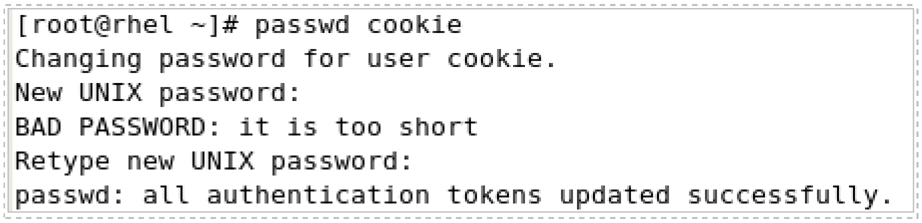


```
[root@rhel ~]# passwd elmo
Changing password for user elmo.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Figure 17: Giving the user a Password

You should receive the message, *all authentication tokens updated successfully*.

15. Type the following to give **cookie** a password. Type **blue** twice as the password:
    [root@rhel ~]#**passwd cookie**

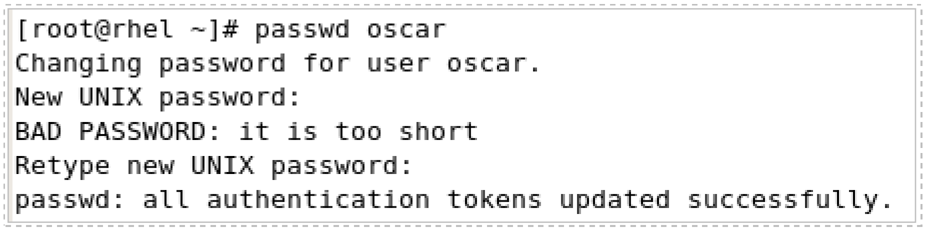


```
[root@rhel ~]# passwd cookie
Changing password for user cookie.
New UNIX password:
BAD PASSWORD: it is too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Figure 18: Giving the user a Password

You should receive the message, *all authentication tokens updated successfully*.

16. Type the following to give **oscar** a password. Type **green** twice as the password:
    [root@rhel ~]#**passwd oscar**

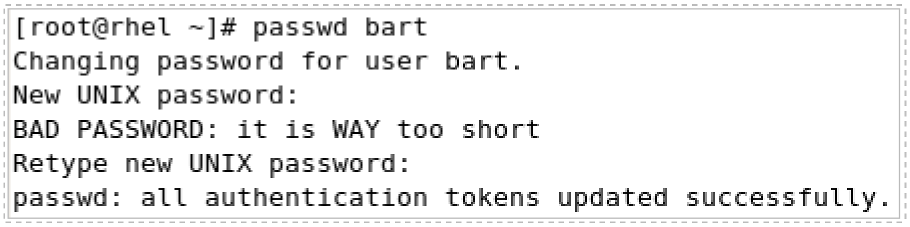


```
[root@rhel ~]# passwd oscar
Changing password for user oscar.
New UNIX password:
BAD PASSWORD: it is too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Figure 19: Giving the user a Password

You should receive the message, *all authentication tokens updated successfully*.

17. Type the following to give **bart** a password. Type **boy** twice as the password:
    [root@rhel ~]#**passwd bart**

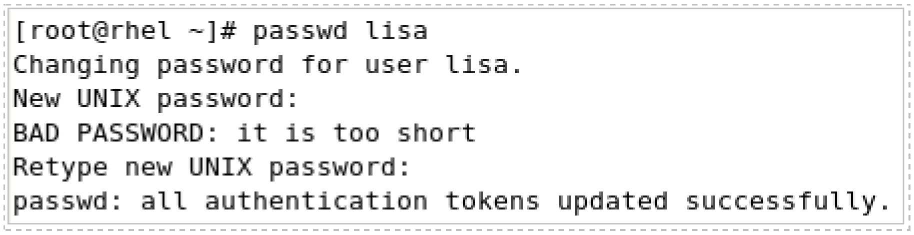


```
[root@rhel ~]# passwd bart
Changing password for user bart.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Figure 20: Giving the user a Password

You should receive the message, *all authentication tokens updated successfully*.

18. Type the following to give **lisa** a password. Type **girl** twice as the password:
    [root@rhel ~]#**passwd lisa**

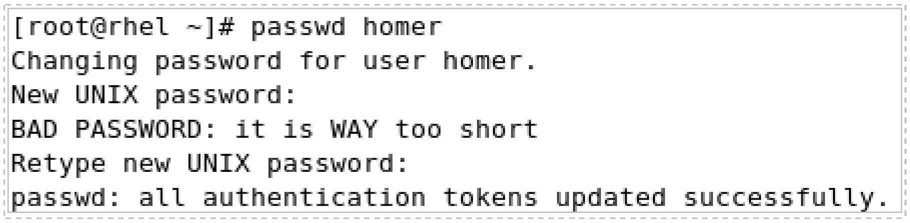


```
[root@rhel ~]# passwd lisa
Changing password for user lisa.
New UNIX password:
BAD PASSWORD: it is too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Figure 21: Giving the user a Password

You should receive the message, *all authentication tokens updated successfully*.

19. Type the following to give **homer** a password. Type **man** twice as the password:
    [root@rhel ~]#**passwd homer**

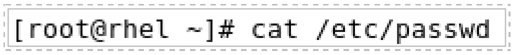


```
[root@rhel ~]# passwd homer
Changing password for user homer.
New UNIX password:
BAD PASSWORD: it is WAY too short
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

Figure 22: Giving the user a Password

You should receive the message *all authentication tokens updated successfully*.

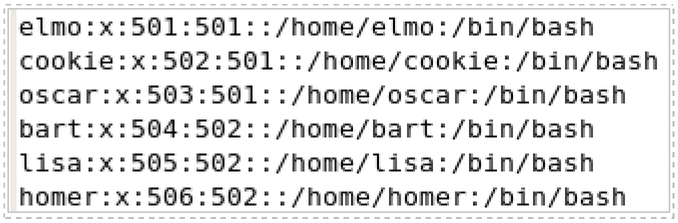
20. Type the following command to view the created users in the passwd file:
    [root@rhel ~]#**cat /etc/passwd**



```
[root@rhel ~]# cat /etc/passwd
```

Figure 23: Displaying the passwd file

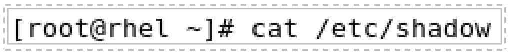
The bottom of the passwd file will display all of the newly created users.



```
elmo:x:501:501::/home/elmo:/bin/bash
cookie:x:502:501::/home/cookie:/bin/bash
oscar:x:503:501::/home/oscar:/bin/bash
bart:x:504:502::/home/bart:/bin/bash
lisa:x:505:502::/home/lisa:/bin/bash
homer:x:506:502::/home/homer:/bin/bash
```

Figure 24: The passwd file

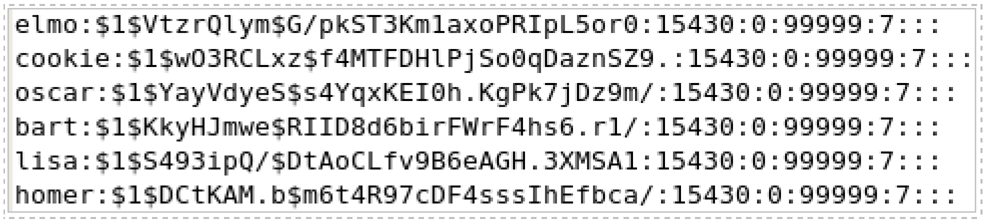
21. Type the following command to view the created users in the shadow file:
    [root@rhel ~]#**cat /etc/shadow**



```
[root@rhel ~]# cat /etc/shadow
```

Figure 25: Displaying the shadow file

The bottom of the shadow file will display all of the user's password hash.



```
elmo:$1$VtzrQlym$G/pkST3KmlaxoPRIpL5or0:15430:0:99999:7:::
cookie:$1$wO3RCLxz$f4MTFDHlPjSo0qDaznSZ9.:15430:0:99999:7:::
oscar:$1$YayVdyeS$s4YqxKEI0h.KgPk7jDz9m/:15430:0:99999:7:::
bart:$1$KkyHJmwe$RIID8d6birFWrF4hs6.r1/:15430:0:99999:7:::
lisa:$1$S493ipQ/$DtAoCLfv9B6eAGH.3XMSA1:15430:0:99999:7:::
homer:$1$DCtKAM.b$m6t4R97cDF4sssIhEfbca/:15430:0:99999:7:::
```

Figure 26: The shadow file

22. Do not close the Red Hat terminal. This exercise will be continued in 3.1.

2.2 Conclusion

The command to create a group on a Linux system is groupadd. After groups are added to the system, you can add users with useradd, and place them in the group as you create them using the –g command switch. Users are given passwords with the passwd command. The users will appear in the passwd and shadow files. The encrypted password hashes are stored in the shadow file. By default, the shadow file can only be viewed by the root account.

2.3 Discussion Questions

1. What is the command to add a group to the system in Linux?
2. What is the command to give a user a password in Linux?
3. What is the command to add a user to the system in Linux?
4. Where is a user's encrypted password hash stored on a Linux system?

3 Symbolic Permissions

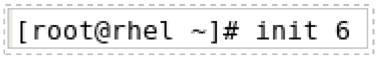
Adding permissions to files and folders can be done by using absolute or symbolic permissions. The symbolic permissions are easier for beginners to use.

Symbolic permissions use *r*, *w*, and *x* to represent *read*, *write*, and *execute* permissions. The first character indicates file type. The remaining nine characters are separated into three sets, each representing a class of permissions as three characters. The first set represents the *user* class. The second set represents the *group* class. The third set represents the *others* class. Examples of symbolic permissions are included below.

3.1 Using Symbolic Permissions

Continue using the terminal from 2.1.

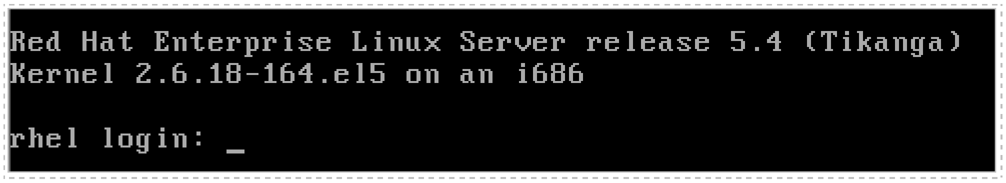
1. Type the following command as the root user to restart the system:
   [root@rhel ~]# **init 6**



```
[root@rhel ~]# init 6
```

Figure 27: Restarting the System using the init 6

You should arrive at the Red Hat Enterprise Linux Server login screen.



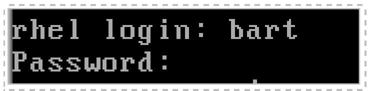
```
Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an i686

rhel login: _
```

Figure 28: The Red Hat Linux Login Screen

2. Log in as the user **bart** with the password of **boy**.

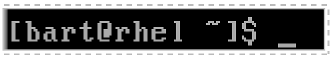
For security reasons, the password will not be displayed when you type it.



```
rhel login: bart
Password:
```

Figure 29: Logging in as bart

After a successful login, you will see the **[bart@rhel ~]** prompt followed by the dollar sign.

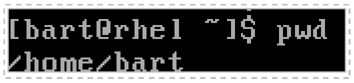


```
[bart@rhel ~]$ _
```

Figure 30: Regular users have a $ prompt

Only the root account will get the **#** prompt. Other users will get a **$** prompt.
When a user logs on to a Linux system, they are "put" into their folder within the */home* directory. Their folder is created when the account is added, using the **useradd** command.

3. To view your present working directory, type:
   [bart@rhel ~]$ **pwd**



```
[bart@rhel ~]$ pwd
/home/bart
```

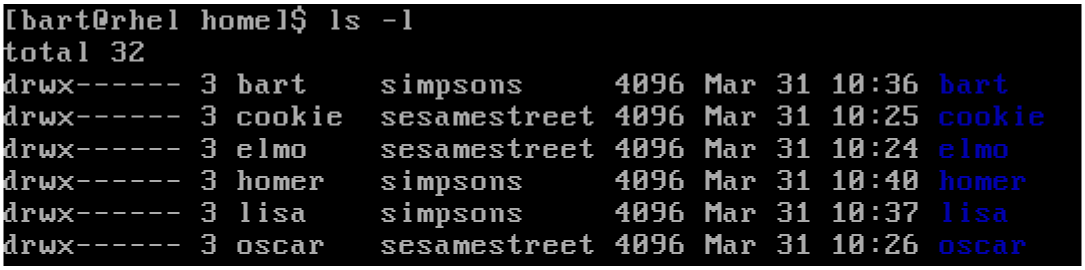
Figure 31: The Present Working Directory of a Linux System

4. Type the following command to go back one directory to the */home* directory:
   [bart@rhel ~]$ **cd ..**

```
[bart@rhel ~]$ cd ..
```

Figure 32: Moving Back One Directory

5. Type the following command to list all of the directories and their permissions:
   [bart@rhel ~]$ **ls -l**



```
[bart@rhel home]$ ls -l
total 32
drwx------ 3 bart    simpsons     4096 Mar 31 10:36 bart
drwx------ 3 cookie  sesamestreet 4096 Mar 31 10:25 cookie
drwx------ 3 elmo    sesamestreet 4096 Mar 31 10:24 elmo
drwx------ 3 homer   simpsons     4096 Mar 31 10:40 homer
drwx------ 3 lisa    simpsons     4096 Mar 31 10:37 lisa
drwx------ 3 oscar   sesamestreet 4096 Mar 31 10:26 oscar
```

Figure 33: The Permissions of the Home Directory

The Linux operating system has a total of 10 letters or dashes in the permissions fields:

- The first field is a dash for a file and a d for a directory
- The 2nd through 4th fields are for the user
- The 5th through 7th fields are for the group
- The 8th through 10th fields are for others (accounts other than those in the group)

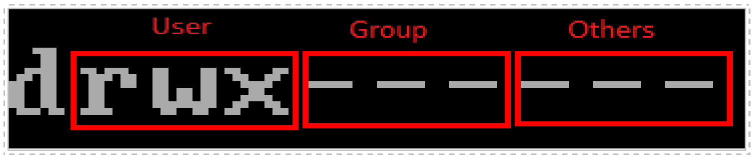
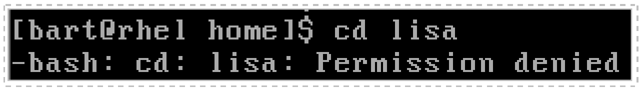


Figure 34: Linux Permissions

There is one exception to the permissions rule. The root account has full access.

In Linux, the default is for only the user to have access to their folder within home. **Bart** will be denied if he tries to enter a folder from another account in his group.

6. Try to enter lisa's folder as bart by typing the following command:
   [bart@rhel ~]$ **cd lisa**

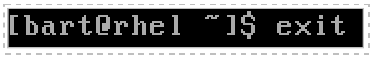


```
[bart@rhel home]$ cd lisa
-bash: cd: lisa: Permission denied
```

Figure 35: Permission is Denied

Bart is denied from accessing the lisa folder.

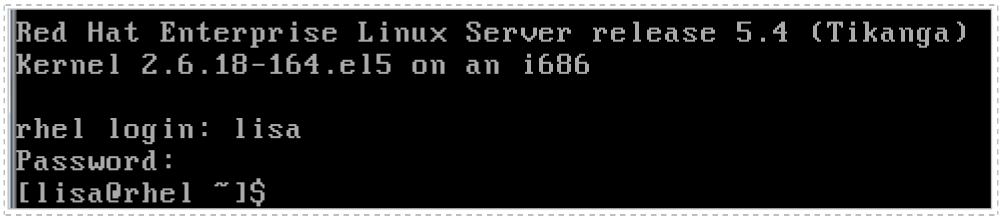
7. To log out as bart and end his session, type the following command:
   [bart@rhel ~]$ **exit**



```
[bart@rhel ~]$ exit
```

Figure 36: Typing exit

8. Log in as the user **lisa** with the password of **girl**.



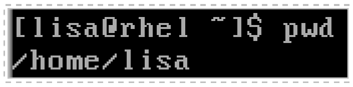
```
Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an i686

rhel login: lisa
Password:
[lisa@rhel ~]$
```

Figure 37: Logging in a lisa

After a successful login, you will see the **[lisa@rhel ~]** prompt followed by the dollar sign.

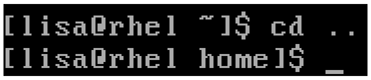
9. To view your present working directory or print your working directory, type:
   [lisa@rhel ~]$ **pwd**



```
[lisa@rhel ~]$ pwd
/home/lisa
```

Figure 38: The Present Working Directory of a Linux System

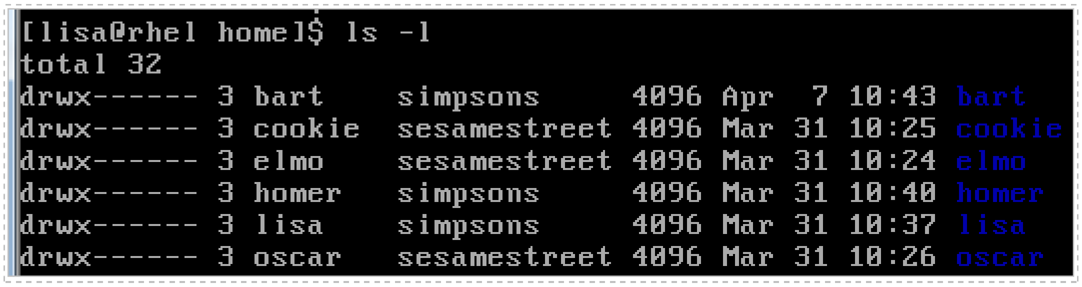
10. Type the following command to go back one directory to the */home* directory:
    [lisa@rhel ~]$ **cd ..**



```
[lisa@rhel ~]$ cd ..
[lisa@rhel home]$ _
```

Figure 39: Moving Back One Directory

11. Type the following command to list all of the directories and their permissions:
    [lisa@rhel ~]$ **ls -l**

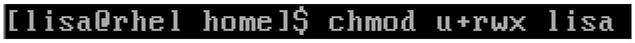


```
[lisa@rhel home]$ ls -l
total 32
drwx------ 3 bart    simpsons     4096 Apr  7 10:43 bart
drwx------ 3 cookie  sesamestreet 4096 Mar 31 10:25 cookie
drwx------ 3 elmo    sesamestreet 4096 Mar 31 10:24 elmo
drwx------ 3 homer   simpsons     4096 Mar 31 10:40 homer
drwx------ 3 lisa    simpsons     4096 Mar 31 10:37 lisa
drwx------ 3 oscar   sesamestreet 4096 Mar 31 10:26 oscar
```

Figure 40: The Permissions of the Home Directory

Lisa can give the other members of her group, bart and homer, permission to enter her folder by using the **chmod** command.

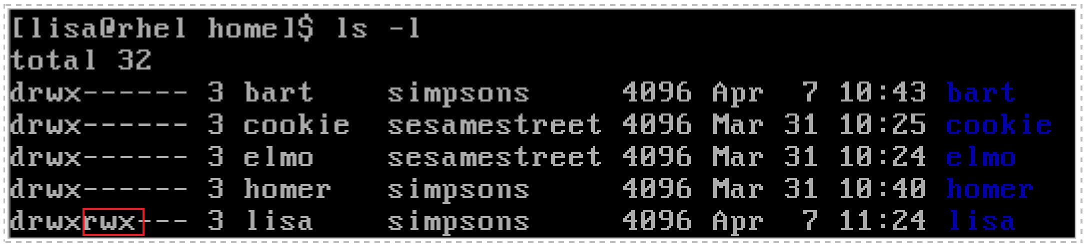
12. Type the following to add **read**, **write** and **execute** permissions for lisa's group:
    [lisa@rhel ~]$ **chmod g+rwx lisa**



```
[lisa@rhel home]$ chmod u+rwx lisa
```

Figure 41: Changing the Permissions chmod g+rwx lisa

13. Type the following command to list all of the directories and their permissions:
    [lisa@rhel ~]$ **ls –l**



```
[lisa@rhel home]$ ls -l
total 32
drwx------ 3 bart    simpsons     4096 Apr  7 10:43 bart
drwx------ 3 cookie  sesamestreet 4096 Mar 31 10:25 cookie
drwx------ 3 elmo    sesamestreet 4096 Mar 31 10:24 elmo
drwx------ 3 homer   simpsons     4096 Mar 31 10:40 homer
drwxrwx--- 3 lisa    simpsons     4096 Apr  7 11:24 lisa
```

Figure 42: Listing the Files on C:

The chart below shows examples of other ways the **chmod** command can be used:

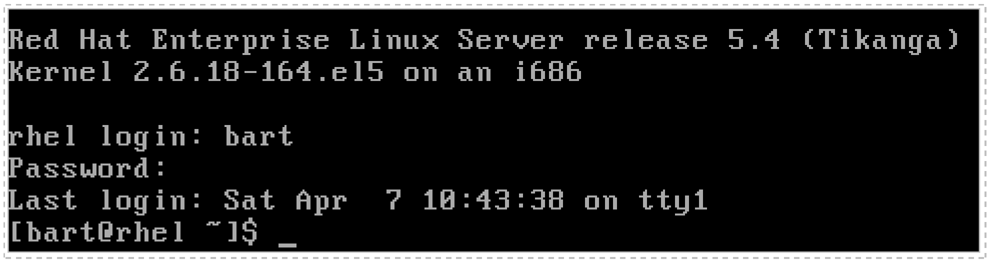
chmod command	Results
chmod u+rwx	Adds read, write, and execute permissions for the user
chmod u+rw	Adds read and write permissions for the user.
chmod o+r	Adds read permissions for others
chmod g-rwx	Removes read, write, and execute permissions for the group

14. To log out as lisa and end her session, type the following command:
    [lisa@rhel ~]$ **exit**

```
[lisa@rhel ~]$ exit
```

Figure 43: Typing exit

15. Log in as the user **bart** with the password of **boy**.



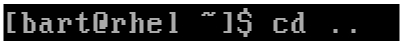
```
Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an i686

rhel login: bart
Password:
Last login: Sat Apr  7 10:43:38 on tty1
[bart@rhel ~]$ _
```

Figure 44: Logging in as bart

After a successful login, you will see the **[bart@rhel ~]** followed by the dollar sign.

16. Type the following command to go back one directory to the */home* directory:
[bart@rhel ~]$ **cd ..**

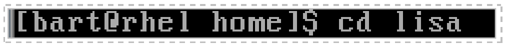


```
[bart@rhel ~]$ cd ..
```

Figure 45: Moving Back One Directory

Now that permissions are modified, bart should now be allowed to enter lisa's folder

17. Try to enter lisa's folder as bart by typing the following command:
[bart@rhel ~]$ **cd lisa**

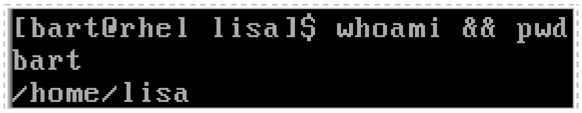


```
[bart@rhel home]$ cd lisa
```

Figure 46: Permission

18. To view the account you are logged in as and your present directory:
type:[bart@rhel ~]$ **whoami && pwd**

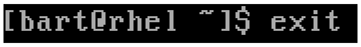
The "&&" between the commands allows commands to run sequentially without having to type them on separate lines.



```
[bart@rhel lisa]$ whoami && pwd
bart
/home/lisa
```

Figure 47: Displaying the user and present working directory

19. To log out as bart and end his session, type the following command:
[bart@rhel ~]$ **exit**



```
[bart@rhel ~]$ exit
```

Figure 48: Typing exit

3.2 Conclusion

With Linux, there are permissions for users, groups, and others on files on folders. Using the symbolic permissions, the owner of a file or folder can change those permissions. If permissions are added for a group, other users who are members of the group will be able to access files or folders to which they are granted permission.

3.3 Discussion Questions

1. What is the command to give the group read and write permissions for the lisa folder, within the home directory using symbolic permissions?
2. What is the command to give others read permissions for the lisa folder, within the home directory using symbolic permissions?
3. What is the command to take away the read permissions for group for the lisa folder, within the home directory using symbolic permissions?
4. What is the command to take away the read and execute permissions for the others for the lisa folder, within the home directory using symbolic permissions?

4 Absolute Permissions

There are other ways to assign permissions besides using the symbolic permissions. The use of absolute permissions is a different way to assign permissions to files and folders, which can provide the same results as using the symbolic permissions.

Absolute Permissions use a three-digit octal number is used to represent the permissions for owner, group, and other. Examples of absolute permissions are included below.

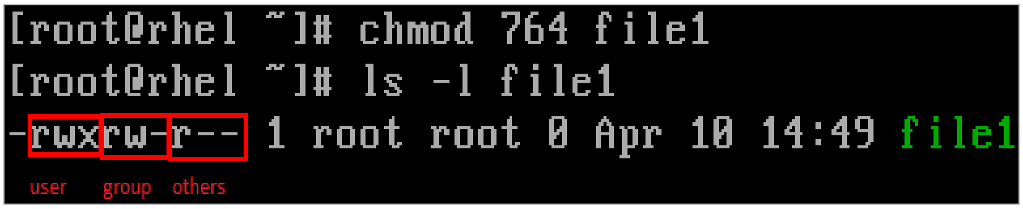
Number	Permissions
7	Read, Write, and Execute
6	Read and Write
5	Read and Execute
4	Read
3	Write and Execute
2	Write
1	Execute
0	None

By typing the following command, **chmod 764 file1**, these permissions will be assigned:

- The user will get Read, Write, and Execute permissions
- The group will get Read and Write
- Others will get Read Access

Breakdown of how **764** represents these permissions:

Digit	Binary Equivalent	Permission
7 (user)	111	1- Read 1- Write 1- Execute
6 (group)	110	1- Read 1-Write 0- No execute
4 (others)	100	1- Read 0- No write 0- No execute



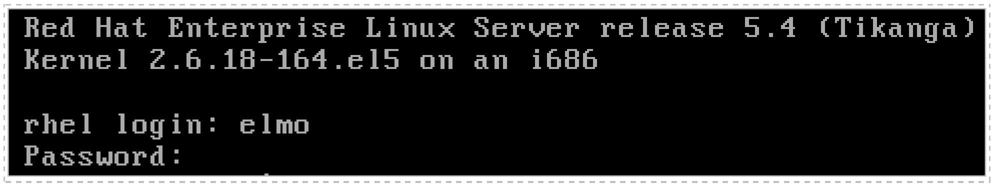
```
[root@rhel ~]# chmod 764 file1
[root@rhel ~]# ls -l file1
-rwxrw-r-- 1 root root 0 Apr 10 14:49 file1
```

Figure 49: Privileges fields for Users, Groups, and Others

4.1 Using Absolute Permissions

1. Log in as the user **elmo** with the password of **red**.

For security reasons, the password will not be displayed when you type it.

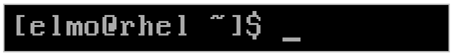


```
Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an i686

rhel login: elmo
Password:
```

Figure 50: Logging in as elmo

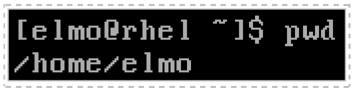
After a successful login, you will see the **[elmo@rhel ~]** prompt followed by the dollar sign.



```
[elmo@rhel ~]$ _
```

Figure 51: Regular users have a $ prompt

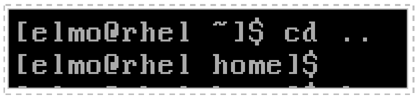
2. To view your present working directory or print your working directory, type: [elmo@rhel ~]$ **pwd**



```
[elmo@rhel ~]$ pwd
/home/elmo
```

Figure 52: The Present Working Directory of a Linux System

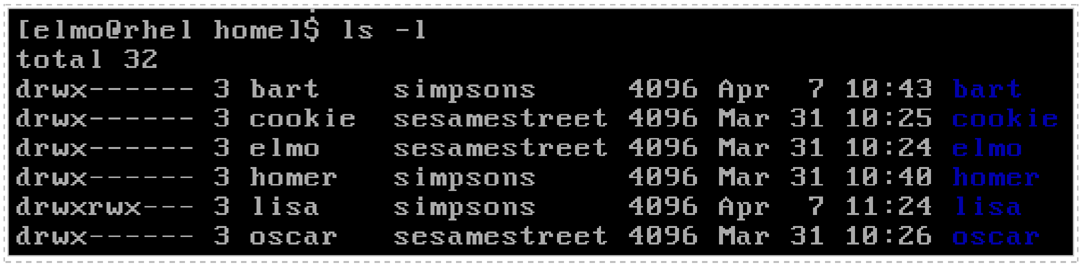
3. Type the following command to go back one directory to the */home* directory: [elmo@rhel ~]$ **cd ..**



```
[elmo@rhel ~]$ cd ..
[elmo@rhel home]$
```

Figure 53: Moving Back One Directory

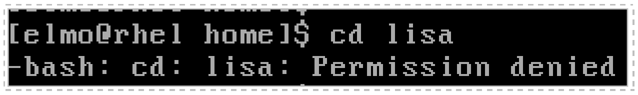
4. Type the following command to list all of the directories and their permissions: [elmo@rhel ~]$ **ls -l**



```
[elmo@rhel home]$ ls -l
total 32
drwx------ 3 bart     simpsons      4096 Apr  7 10:43 bart
drwx------ 3 cookie   sesamestreet  4096 Mar 31 10:25 cookie
drwx------ 3 elmo     sesamestreet  4096 Mar 31 10:24 elmo
drwx------ 3 homer    simpsons      4096 Mar 31 10:40 homer
drwxrwx--- 3 lisa     simpsons      4096 Apr  7 11:24 lisa
drwx------ 3 oscar    sesamestreet  4096 Mar 31 10:26 oscar
```

Figure 54: The Permissions of the Home Directory

5. Try to enter lisa's folder as elmo by typing the following command:
   [elmo@rhel ~]$ **cd lisa**

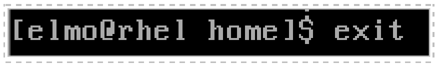


```
[elmo@rhel home]$ cd lisa
-bash: cd: lisa: Permission denied
```

Figure 55: Permission is Denied

Permission is denied because elmo is not in lisa's group.

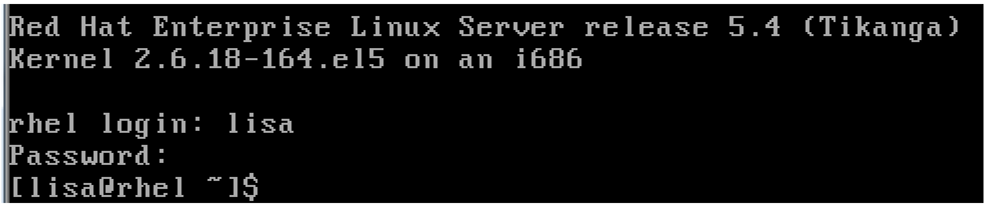
6. To log out as elmo and end his session, type the following command:
   [elmo@rhel ~]$ **exit**



```
[elmo@rhel home]$ exit
```

Figure 56: Typing exit

7. Log in as the user **lisa** with the password of **girl**.



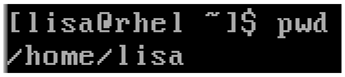
```
Red Hat Enterprise Linux Server release 5.4 (Tikanga)
Kernel 2.6.18-164.el5 on an i686

rhel login: lisa
Password:
[lisa@rhel ~]$
```

Figure 57: Logging in a lisa

After a successful login, you will see the **[lisa@rhel ~]** followed by the dollar sign.

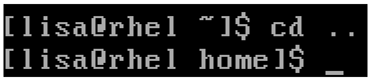
8. To view your present working directory or print your working directory, type:
   [lisa@rhel ~]$ **pwd**



```
[lisa@rhel ~]$ pwd
/home/lisa
```

Figure 58: The Present Working Directory of a Linux System

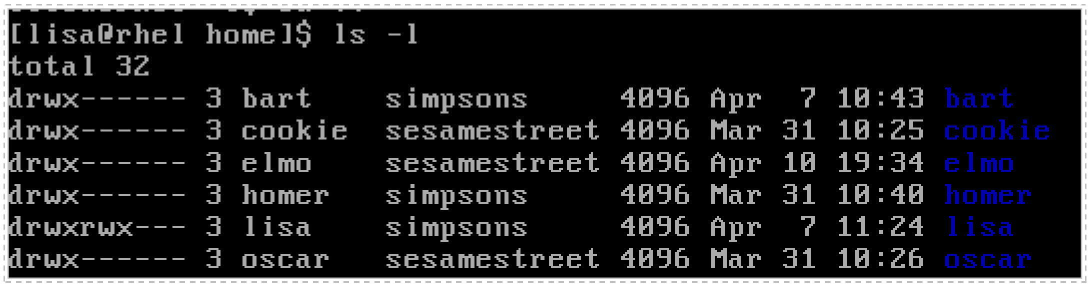
9. Type the following command to go back one directory to the */home* directory:
   [lisa@rhel ~]$ **cd ..**



```
[lisa@rhel ~]$ cd ..
[lisa@rhel home]$ _
```

Figure 59: Moving Back One Directory

10. Type the following command to list all of the directories and their permissions:
[lisa@rhel ~]$ **ls -l**



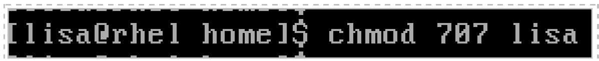
```
[lisa@rhel home]$ ls -l
total 32
drwx------ 3 bart    simpsons     4096 Apr  7 10:43 bart
drwx------ 3 cookie  sesamestreet 4096 Mar 31 10:25 cookie
drwx------ 3 elmo    sesamestreet 4096 Apr 10 19:34 elmo
drwx------ 3 homer   simpsons     4096 Mar 31 10:40 homer
drwxrwx--- 3 lisa    simpsons     4096 Apr  7 11:24 lisa
drwx------ 3 oscar   sesamestreet 4096 Mar 31 10:26 oscar
```

Figure 60: The Permissions of the Home Directory

The user elmo is denied from accessing the lisa folder. Lisa can grant access to others (everyone other than herself and the individuals within her group). You can give permissions to others to enter lisa's folder by using chmod with absolute permissions.

11. Type the following to add **read**, **write** and **execute** permissions for others:
[lisa@rhel ~]$ **chmod 707 lisa**

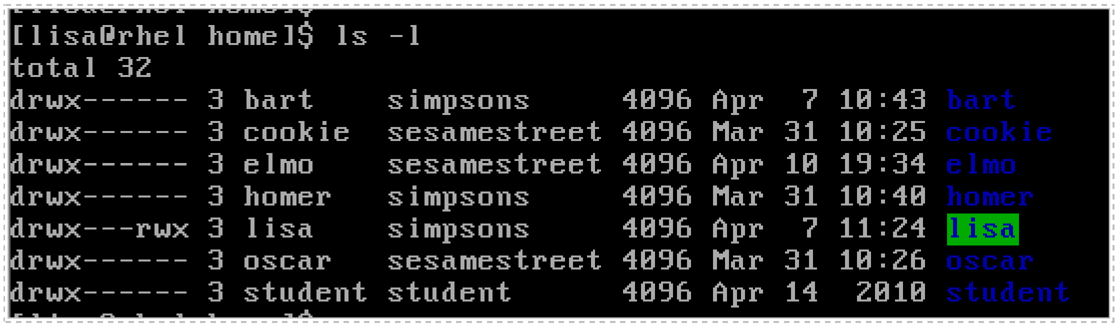
This command will also remove permissions for the group.



```
[lisa@rhel home]$ chmod 707 lisa
```

Figure 61: Changing the Permissions

12. Type the following command to list all of the directories and their permissions:
[lisa@rhel ~]$ **ls –l**



```
[lisa@rhel home]$ ls -l
total 32
drwx------ 3 bart    simpsons     4096 Apr  7 10:43 bart
drwx------ 3 cookie  sesamestreet 4096 Mar 31 10:25 cookie
drwx------ 3 elmo    sesamestreet 4096 Apr 10 19:34 elmo
drwx------ 3 homer   simpsons     4096 Mar 31 10:40 homer
drwx---rwx 3 lisa    simpsons     4096 Apr  7 11:24 lisa
drwx------ 3 oscar   sesamestreet 4096 Mar 31 10:26 oscar
drwx------ 3 student student      4096 Apr 14  2010 student
```

Figure 62: Listing the Files within /home directory

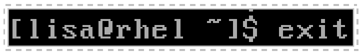
In this case, these permissions have been set for the lisa folder:

User	Read, Write, and Execute
Group	None
Owner	Read, Write, and Execute

In the chart below, there are other examples of how the **chmod** command can be used:

Command	Results
chmod 777	Gives read, write, and execute permissions for the user, group and others
chmod 000	Takes away read, write, and execute permissions for all accounts
chmod 440	Adds read permissions for user, group. No permissions for others
chmod 606	Gives read and write permissions for the user and others. None for group.

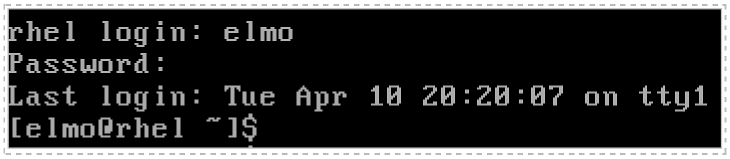
13. To log out as lisa and end her session, type the following command:
    [lisa@rhel ~]$ **exit**



```
[lisa@rhel ~]$ exit
```

Figure 63: Typing exit

14. Log in as the user **elmo** with the password of **red**.

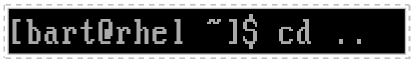


```
rhel login: elmo
Password:
Last login: Tue Apr 10 20:20:07 on tty1
[elmo@rhel ~]$
```

Figure 64: Logging in as elmo

After a successful login, you will see the **[elmo@rhel~]** followed by the dollar sign.

15. Type the following command to go back one directory to the */home* directory:
    [elmo@rhel ~]$ **cd ..**

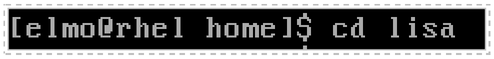


```
[bart@rhel ~]$ cd ..
```

Figure 65: Moving Back One Directory

Now that permissions are modified, elmo should now be allowed to enter lisa's folder:

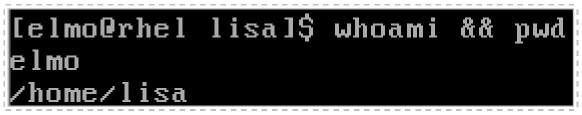
16. Try to enter lisa's folder as elmo by typing the following command::
    [elmo@rhel ~]$ **cd lisa**



```
[elmo@rhel home]$ cd lisa
```

Figure 66: Permission

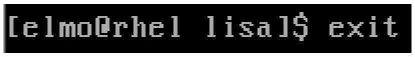
17. To view the account you are logged in as and your present directory, type:
    [elmo@rhel ~]$ **whoami && pwd**



```
[elmo@rhel lisa]$ whoami && pwd
elmo
/home/lisa
```

Figure 67: Displaying the user and present working directory

18. To log out as elmo and end his session, type the following command:
    [elmo@rhel ~]$ **exit**



```
[elmo@rhel lisa]$ exit
```

Figure 68: Typing exit

4.2 Conclusion

With Linux, there are permissions for users, groups, and others, which control access to files and folders. Using the absolute permissions, the owner of a file or folder can change those permissions. If permissions are added for a group, other users who are members of the group will be able to access files or folders to which they are granted permission.

4.3 Discussion Questions

1. What is the command to give the user, group, and others read and write permissions for the lisa folder, within the home directory using absolute permissions?
2. What is the command to give the user, group, and others read permissions for the lisa folder, within the home directory using absolute permissions?
3. What is the command to give read and execute permissions the user, group, and others for the lisa folder, within the home directory using absolute permissions?
4. What is the command to give read, write, and execute permissions the user, group, and others for the lisa folder, within the home directory using absolute permissions?

References

1. The passwd man page:
   http://unixhelp.ed.ac.uk/CGI/man-cgi?passwd

2. Understanding /etc/shadow file:
   http://www.cyberciti.biz/faq/understanding-etcshadow-file/

3. The chmod man page:
   http://ss64.com/bash/chmod.html

4. Linux: useradd - Linux man page:
   http://linux.die.net/man/8/useradd

5. Linux: groupadd - Linux man page:
   http://linux.die.net/man/8/groupadd