

Access Control Lab

Copyrights 2016-2017 Frank Xu, Bowie State University.
The lab manual is developed based on the post
http://www.sis.pitt.edu/lersais/education/labs/access_control.php
Comments and suggestions can be sent to wxu@bowiestate.edu

Objectives

The objective of the exercises presented here is to familiarize the students with the access control features available in UNIX-based system.

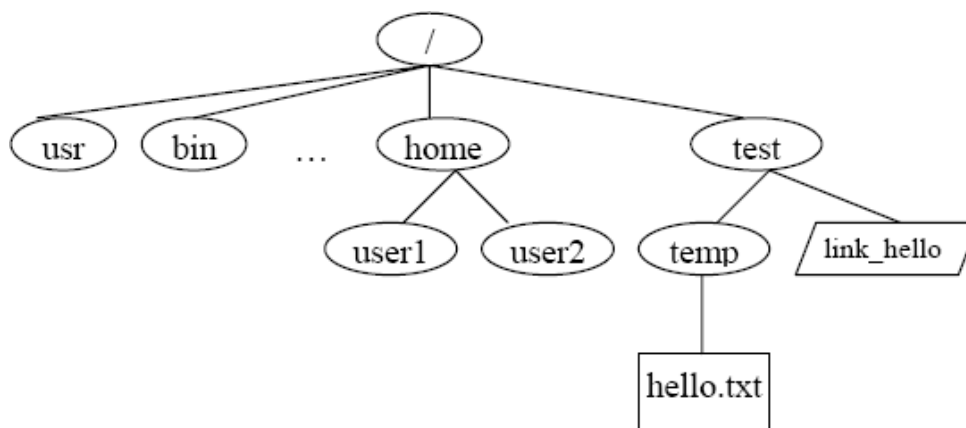
Lab Environment

We have created two accounts in the VM. The usernames and passwords are listed in the following:

1. User ID: root, Password: seedubuntu.
 - o Note: Ubuntu does not allow root to login directly from the login window. You have to login as a normal user, and then use the command **su** to login to the root account.
2. User ID: seed, Password: dees

Unix File Hierarchy

3. The Unix file system is organized as a hierarchy with the root (/) directory at the highest level. Each directory may contain subdirectories and files. Typically, some of the directories that may occur under the root are usr, bin, sbin, home, var, boot, dev, etc. In Figure , user1 and user2 are sub-directories under home. hello.txt is a plain-text file and link_hello is a linking file that points to hello.txt. In order to access the file /test/temp/hello.txt, the system begins its search from the root(/) folder and then to test and temp folders consecutively and then finally the file hello.txt.



Ownership and Permissions

- Ownership of files in UNIX can be viewed in one of three ways: owner (creator), group or others. Using this simple notion of ownership access to files can be controlled by associating unique user ID (UID) and group ID (GID) with twelve permission bits for each file as shown below.

Permission Bits											
Extra			Owner			Group			Others		
su	sg	t	r	w	x	r	w	x	r	w	x

- Typically these bits are divided into three sets of three bits and three extra bits as shown in table below. r, w and x bits stand for read, write and execute bits for each of the owner, group and others permissions. su, sg and t stand for set_user_id, set_group_id and sticky bits. These 4 sets of bits are often represented in their octal digits. For example, "100 111 101 101" is represented as "4755." When the su bit is set, whosoever executes the file, the UID of the process will be the owner of the file.

Unix Lab Procedures

- Setting up File Structure and User Space. The objective of this exercise is to setup the file hierarchy structure and the users that are required for the exercises in this section. The su command is used to switch users.
 - Login as root
 - Use useradd command to create two new users user1 and user2 as follows:

- useradd user1 -m -g users
- useradd user2 -m -g users

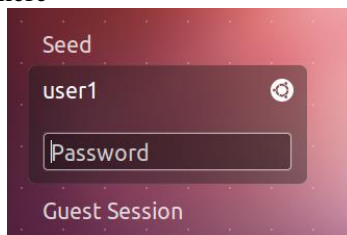
```
[09/16/2016 08:08] root@ubuntu:/home/seed# useradd user1 -m -g users
[09/16/2016 08:08] root@ubuntu:/home/seed# ls /home
seed user1
```

- Change password (I use 123456).

- passwd user1 123456

```
[09/16/2016 08:28] root@ubuntu:/home/seed# passwd user1
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
[09/16/2016 08:28] root@ubuntu:/home/seed#
```

- You can login as user1 here



- Check user information with the id command. Note the uid, gid for each output.

- id user1
- id user2
- id

```
[09/16/2016 07:28] root@ubuntu:/home/seed# id user1
uid=1001(user1) gid=100(users) groups=100(users)
[09/16/2016 07:29] root@ubuntu:/home/seed# id user2
uid=1002(user2) gid=100(users) groups=100(users)
[09/16/2016 07:29] root@ubuntu:/home/seed# id
uid=0(root) gid=0(root) groups=0(root)
[09/16/2016 07:29] root@ubuntu:/home/seed# █
```

```
[09/16/2016 09:41] root@ubuntu:/home/seed# id user1
uid=1001(user1) gid=100(users) groups=100(users)
[09/16/2016 09:41] root@ubuntu:/home/seed# cat /etc/passwd | grep user1
user1:x:1001:100::/home/user1:/bin/sh
[09/16/2016 09:42] root@ubuntu:/home/seed# █
```

d) Create a directory structure

- mkdir /test
- mkdir /test/temp

```
[09/16/2016 09:49] root@ubuntu:/home/seed# ls /test -la
total 12
drwxr-xr-x  3 root root 4096 Sep 16 09:46 .
drwxr-xr-x 24 root root 4096 Sep 16 09:46 ..
drwxr-xr-x  2 root root 4096 Sep 16 09:46 temp
```

e) Switch user roles as user1 and then back to root using the su command

- whoami
- su user1
- su OR su root

f) Create a new file as root user and change group ownership as well as user ownership of the file.

- touch /home/user2/HelloWorld
- ls -l /home/user2/HelloWorld (observe owner and group)

```
[09/16/2016 09:53] root@ubuntu:/home/seed# touch /home/user2/HelloWorld
[09/16/2016 09:55] root@ubuntu:/home/seed# ls -l /home/user2/HelloWorld
-rw-r--r-- 1 root root 0 Sep 16 09:55 /home/user2/HelloWorld
```

- chgrp users /home/user2/HelloWorld

```
[09/16/2016 10:09] root@ubuntu:/home/seed# chgrp users /home/user2/HelloWorld
[09/16/2016 10:09] root@ubuntu:/home/seed# ls -l /home/user2/HelloWorld
-rw-r--r-- 1 root users 0 Sep 16 09:55 /home/user2/HelloWorld
```

- chown user2:users /home/users/HelloWorld
- ls -l /home/user2/HelloWorld (observe owner and group)

```
[09/16/2016 10:09] root@ubuntu:/home/seed# chown user2:users /home/user2/HelloWorld
[09/16/2016 17:08] root@ubuntu:/home/seed# ls -l /home/user2/HelloWorld
-rw-r--r-- 1 user2 users 0 Sep 16 09:55 /home/user2/HelloWorld
```

- Notes:

```
chown owner-user file
chown owner-user:owner-group file
chown owner-user:owner-group directory
chown options owner-user:owner-group file
```

3. Questions.

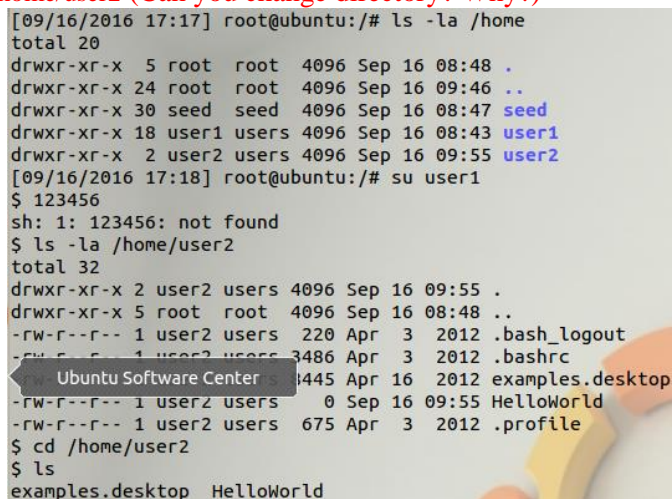
- a) Explain what chgrp and chown do?
- b) What do -g and -m options mean?

4. Differences in File and Folder Permissions. The objective of the following exercises would be to see the differences in file and folder permissions. The chmod command will be used to change file

and directory permission to demonstrate the slight differences in permissions for files and directories.

a) Observe the result of ls and cd commands

- cd /
- ls -l
- ls -la /home
- Switch to user1 using su user1
- ls -la /home/user2 (Can you list directory? Why?)
- cd /home/user2 (Can you change directory? Why?)



```
[09/16/2016 17:17] root@ubuntu:/# ls -la /home
total 20
drwxr-xr-x  5 root  root  4096 Sep 16 08:48 .
drwxr-xr-x 24 root  root  4096 Sep 16 09:46 ..
drwxr-xr-x 30 seed  seed  4096 Sep 16 08:47 seed
drwxr-xr-x 18 user1 users 4096 Sep 16 08:43 user1
drwxr-xr-x  2 user2 users 4096 Sep 16 09:55 user2
[09/16/2016 17:18] root@ubuntu:/# su user1
$ 123456
sh: 1: 123456: not found
$ ls -la /home/user2
total 32
drwxr-xr-x  2 user2 users 4096 Sep 16 09:55 .
drwxr-xr-x  5 root  root  4096 Sep 16 08:48 ..
-rw-r--r--  1 user2 users  220 Apr  3 2012 .bash_logout
-rw-r--r--  1 user2 users 3486 Apr  3 2012 .bashrc
-rw-r--r--  1 user2 users 1445 Apr 16 2012 examples.desktop
-rw-r--r--  1 user2 users   0 Sep 16 09:55 HelloWorld
-rw-r--r--  1 user2 users  675 Apr  3 2012 .profile
$ cd /home/user2
$ ls
examples.desktop HelloWorld
```

b) Change directory permissions of user2 directory and try again as user1.

- su root
- chmod 740 /home/user2
- Repeat steps written in RED in the previous example (Can you list or change directory of user2 if you are the user 1? Why)

```
[09/16/2016 17:44] root@ubuntu:/home/user2# chmod 740 /home/user2
[09/16/2016 17:45] root@ubuntu:/home/user2# ls -la /home/user2
total 32
drwxr----- 2 user2 users 4096 Sep 16 09:55 .
drwxr-xr-x 5 root root 4096 Sep 16 08:48 ..
-rw-r--r-- 1 user2 users 220 Apr 3 2012 .bash_logout
-rw-r--r-- 1 user2 users 3486 Apr 3 2012 .bashrc
-rw-r--r-- 1 user2 users 8445 Apr 16 2012 examples.desktop
-rw-r--r-- 1 user2 users 0 Sep 16 09:55 HelloWorld
-rw-r--r-- 1 user2 users 675 Apr 3 2012 .profile
[09/16/2016 17:45] root@ubuntu:/home/user2# su user1
$ ls -la /home/user2
ls: cannot access /home/user2/HelloWorld: Permission denied
ls: cannot access /home/user2/.bashrc: Permission denied
ls: cannot access /home/user2/.bash_logout: Permission denied
ls: cannot access /home/user2/..: Permission denied
ls: cannot access /home/user2/.profile: Permission denied
ls: cannot access /home/user2/examples.desktop: Permission denied
ls: cannot access /home/user2/..: Permission denied
total 0
d???????? ? ? ? ? ? ? ? ?
d???????? ? ? ? ? ? ? ? ?
-???????? ? ? ? ? ? ? ? ?
-???????? ? ? ? ? ? ? ? ?
-???????? ? ? ? ? ? ? ? ?
-???????? ? ? ? ? ? ? ? ?
-???????? ? ? ? ? ? ? ? ?
$
```

- su root
- chmod 750 /home/user2
- Repeat steps **RED** in the previous example (Can you list or change directory? Why?)

```
[09/16/2016 17:56] root@ubuntu:/home# su user1
$ ls /home/user2
examples.desktop HelloWorld
$ cd /home/user2
$
```

- touch /home/user2/hello12.txt (Can you create new file? Why?)

```
$ cd /home/user2
$ touch /home/user2/hello2.txt
touch: cannot touch '/home/user2/hello2.txt': Permission denied
```

- su root
- chmod 770 /home/user2
- su user1
- touch /home/user2/hello12.txt (Can you create new file?)
- ls -l /home/user2

5. Question. What are the directory permissions for user1, user2 and test directories?
6. Alternative Syntax for chmod Command. You are expected to learn both the ways to use chmod. The access permissions for the file hello.txt is to set the su bit only, allow all access permissions to owner, read and execute rights to the group and only read rights to others. In other words the 12 bit permission required on the file hello.txt is as follows: "100 111 101 100." This can be achieved in several ways using chmod command:
 - c) chmod 4754 hello.txt
 - d) chmod u+srwx g+rx o+r hello.txt
 - e) chmod u=srwx, g=rx, o=r hello.txt
7. (optional) New Text Files and Linking Files. Unix supports two kinds of link files--a hard link and a symbolic link. A hard link is a file with the actual address space of some ordinary file's data

blocks. A symbolic link is just a reference to another file. It contains the pathname to some other file.

- f) In the /test/temp/ directory, as root user, create a new text file ("hello") and fill it with some text using touch, pico, vi etc.
- g) Create a link link_hello in the test folder pointing to hello.txt in the temp folder (refer to file structure in introduction)
 - cd /
 - ln -s /test/temp/hello /test/link_hello
 - Is there any difference in file permissions of link_hello and hello?
 - cat /test/link_hello What is the output?
8. (optional) Default file permissions and Group Access Control. Whenever a new file is created using C program, default permissions can be assigned to it. UNIX system allows the user to filter out unwanted permissions by default. This default setting can be set by the user using the umask command. It is a system call that is also recognized by the shell. The command takes the permissions set during file creation and performs a bitwise AND to the bitwise negation of mask value. Some common umask values are 077 (only user has permissions), 022 (only owner can write), 002 (only owner and group members can write), etc.
 - a) In a terminal window, make sure you are a root user. If not the root user, then switch back to root user (use your password to switch).
 - b) Use umask command to check the current mask permission and assign a new mask.
 - umask
 - What is the current mask? How is it interpreted? (try umask -S or the man pages)
 - cd /test
 - touch testmask1
 - ls al
 - What are the permissions of the file testmask1
 - umask 0077
 - touch testmask2
 - Now what are the permissions of the file testmask2
 - c) What is the effect of setting mask value to 0000?
9. (optional) Setuid Bit, Setgid Bit and Sticky Bit. As explained in the background above, the highest three bits of the permission bits represent the setuid bit, setgid bit and the sticky bit. If the setuid bit is set then the uid will always be set to the owner of the file during execution. If the setuid bit is not set then the uid will be the user who executes the process. Similarly, if the setgid bit is set then the gid will be set to the group that owns the file during execution. If the setgid bit is not set then the gid will be the group that executes the process. The sticky bit is set to keep processes in the main memory. In the following exercise, the objective is to demonstrate how processes are affected when the setuid bit is set. The exercise must be begun with root privileges.
 - a) which touch
 - b) ls -l /bin/touch
 - c) chmod 4755 /bin/touch
 - d) ls -l /bin/touch
 - e) ls -l /home/user2
 - f) chmod 700 /home/user2/HelloWorld
 - g) ls -l /home/user2 (observe timestamp and permissions)
 - h) su user1
 - i) touch /home/user2/HelloWorld
 - j) ls -l /home/user2 (observe timestamp)

- k) `su root`
 - l) `chmod 0755 /bin/touch`
 - m) `su user1`
 - n) `touch /home/user2/HelloWorld`
10. (optional) Question. Why is permission denied?
11. Restore the System. After the series of exercises, it is most essential that the system is restored to its normal state so that other students may undertake the exercises again. Below are the series of commands that are expected to restore the system to its original form.
- a) `su root`
 - b) `umask 0022`
 - c) `chmod 0755 /bin/touch`
 - d) `userdel user1`
 - e) `userdel user2`
 - f) `rm -rf /home/user1`
 - g) `rm -rf /home/user2`
 - h) `rm -rf /test`
 - i) `rm -rf /home/test/`

Reference

http://www.sis.pitt.edu/lersais/education/labs/access_control.php
http://www.tutorialspoint.com/unix_commands/adduser.htm

Extra Tutorial: **useradd**

NAME

useradd - create a new user or update default new user information

SYNOPSIS

Tag	Description
useradd [<i>options</i>] <i>LOGIN</i>	
useradd -D	
useradd -D [<i>options</i>]	

DESCRIPTION

When invoked without the **-D** option, the **useradd** command creates a new user account using the values specified on the command line and the default values from the system. Depending on command line options, the **useradd** command will update system files and may also create the new user's home directory and copy initial files. The version provided with Red Hat Linux will create a group for each user added to the system by default.

OPTIONS

The options which apply to the **useradd** command are:

Tag	Description
-c, --comment <i>COMMENT</i>	
	Any text string. It is generally a short description of the login, and is currently used as the field for the user's full name.
-b, --base-dir <i>BASE_DIR</i>	
	The default base directory for the system if -d dir is not specified. <i>BASE_DIR</i> is concatenated with the account name to define the home directory. If the -m option is not used, <i>BASE_DIR</i> must exist.
-d, --home <i>HOME_DIR</i>	
	The new user will be created using <i>HOME_DIR</i> as the value for the user's login directory. The default is to append the <i>LOGIN</i> name to <i>BASE_DIR</i> and use that as the login directory name. The directory <i>HOME_DIR</i> does not have to exist but will not be created if it is missing.

-e, --expiredate <i>EXPIRE_DATE</i>	
	The date on which the user account will be disabled. The date is specified in the format <i>YYYY-MM-DD</i> .
-f, --inactive <i>INACTIVE</i>	
	The number of days after a password expires until the account is permanently disabled. A value of 0 disables the account as soon as the password has expired, and a value of -1 disables the feature. The default value is -1.
-g, --gid <i>GROUP</i>	
	The group name or number of the user's initial login group. The group name must exist. A group number must refer to an already existing group. <i>/etc/default/useradd</i> .
-G, --groups <i>GROUP1[,GROUP2,...[,GROUPN]]</i>	
	A list of supplementary groups which the user is also a member of. Each group is separated from the next by a comma, with no intervening whitespace. The groups are subject to the same restrictions as the group given with the -g option. The default is for the user to belong only to the initial group.
-h, --help	
	Display help message and exit.
-M	
	The user's home directory will not be created, even if the system wide settings from <i>/etc/login.defs</i> is to create home dirs.
-m, --create-home	
	The user's home directory will be created if it does not exist. The files contained in <i>SKEL_DIR</i> will be copied to the home directory if the -k option is used, otherwise the files contained in <i>/etc/skel</i> will be used instead. Any directories contained in <i>SKEL_DIR</i> or <i>/etc/skel</i> will be created in the user's home directory as well. The -k option is only valid in conjunction with the -m option. The default is to not create the directory and to not copy any files.
-l	
	Do not add the user to the last login log file. This is an option added by Red Hat.
-n	
	A group having the same name as the user being added to the system will be created by default. This option will turn off this Red Hat Linux specific behavior. When this option is used, users by default will be placed in whatever group is specified in <i>/etc/default/useradd</i> . If no default group is defined, group 1 will be used.

-K, --key <i>KEY=VALUE</i>	
	<p>Overrides /etc/login.defs defaults (UID_MIN, UID_MAX, UMASK, PASS_MAX_DAYS and others).</p> <p>Example: -K PASS_MAX_DAYS=-1 can be used when creating system account to turn off password ageing, even though system account has no password at all. Multiple -K options can be specified, e.g.: -KUID_MIN=100 -K UID_MAX=499</p> <p>Note: -K UID_MIN=10,UID_MAX=499 doesn't work yet.</p>
-o, --non-unique	
	Allow the creation of a user account with a duplicate (non-unique) UID.
-p, --password <i>PASSWORD</i>	
	The encrypted password, as returned by crypt(3) . The default is to disable the account.
-r	This flag is used to create a system account. That is, a user with a UID lower than the value of UID_MIN defined in /etc/login.defs and whose password does not expire. Note that useradd will not create a home directory for such an user, regardless of the default setting in /etc/login.defs. You have to specify -m option if you want a home directory for a system account to be created. This is an option added by Red Hat
-s, --shell <i>SHELL</i>	
	The name of the user's login shell. The default is to leave this field blank, which causes the system to select the default login shell.
-u, --uid <i>UID</i>	
	The numerical value of the user's ID. This value must be unique, unless the -o option is used. The value must be non-negative. The default is to use the smallest ID value greater than 999 and greater than every other user. Values between 0 and 999 are typically reserved for system accounts.
-Z, --selinux-user <i>SEUSER</i>	
	The SELinux user for the user's login. The default is to leave this field blank, which causes the system to select the default SELinux user.

1.