Account Management

Introduction and/or Background
Account management is a case where Windows beats Unix/Linux hands down. Microsoft learned what was missing when they created the OS. Linux functions into two pieces; Users and Groups. Lets grasp the concepts -

Users

useradd - create an account. Can be a normal user or a system user. Format:
useradd <user_name> is the simplest and the tool will perform account creation using system wide default values. There are some useful options -

e - Sets the expiration date of the account.
f - The number of days an account can remain inactive before permanent deletion.
G - The list of groups to add this account to at creation.
k - Passes the contents of the skel directory into the user's home directory.
m - Linux to create a home directory for the user.
p - Request password creation.

Example:

useradd -e 2020-12-30 -k -p smythe1

This will set the account to expire at the end of this year. On creation certain files will be copied from the /skel directory. A password will be prompted for.

The downside? useradd is a low level account creation tool. In Debian the account will be created but unless a HOMEDIR is specified none will be created. Many distributions overcome this limitation with

adduser <user_name>

Common options:

–home DIR
Use DIR as the user's home directory, rather than the default specified by the

configuration file. If the directory does not exist, it is created and skeleton files are copied.
–shell SHELL
Use SHELL as the user's login shell, rather than the default specified by the configuration file.
–ingroup GROUP
Add the new user to GROUP instead of a usergroup or the default group defined by USERS_GID in the configuration file. This affects the users primary group. To add additional groups, see the add_extra_groups option.
–no-create-home
Do not create the home directory, even if it doesn't exist.

Example:

adduser smythe1

userdel

Yes, it deletes a user account. Example:

userdel <user_name>

By now you know there are more options available. Two most common:

f - force deletion of the account with prejudice. Even if the user is logged in they are kicked off and their account data destroyed along with their home directory.
r - The more subtle way to eliminate an account. Like f option but it does not purge the user off. It will wait till the user logs off to prevent further logins.

Use the f option only if you have to. For consistency, the userdel has a corollary deluser, to compliment the adduser command.

deluser

Yes, it deletes a user account.

Example:

deluser <user_name>

usermod
This changes an attribute of the user account previously created. See the useradd for the list of options as they are the same.

Example:

usermod -e 2025-12-30 smythe

This action changed the expiry date of the account. usermod also accepts the -p password option. However it is not recommended for use since if it is part of a script the new password could be intercepted.

passwd
Creates or modifies a password for a particular user. Remember that useradd and usermod can create an account but not apply a password if desired. passwd can be used to add a password to the account. Sadly there is a file with the same name as this command. Make sure you know the context of the conversation when using either.

Here is another anomaly. If you are in an account and issue the passwd command you changed that user's password! You were of course thinking you were in your own account. As a GAP issue a whoami and look at the result before issuing the passwd command.

Most commonly used options:

- d - remove a password from an account
- e - immediately expire an account
- l - lock the password for the account preventing login.
- n - minimum days between password changes

Example:

passwd -l smythe

Groups
Group commands parallel those in the User commands.

groupadd

Adds a user to a group. As a general rule whenever a user account is created they are also given their own group name as the user account name. Most common options used -

- o - nonunique. Useful when overlaying multiple names to a single group.
- r - system only group. Generally used by programs and processes.

Example:

groupadd -r Apache4

groupdel

Eliminate a group. No options with this command.

Example:

groupdel Apache4

groupmod
Changes the attributes of an existing group. Most common used options:

- n - change the name of an existing group to a new name.
- o - modify a non unique group name, or permit one.

Example:

groupmod -o Apache4

Restrictions
Limit user command access
First, create a symlink called rbash from Bash as shown below. The following commands should be run as a root user.

ln -s /bin/bash /bin/rbash

Next, create a user called “ostechnix” with rbash as his/her default login shell.

useradd ostechnix -s /bin/rbash

Set password to the new user.

passwd ostechnix

Create a bin directory inside the home folder of the new user.

mkdir /home/ostechnix/bin

Now, we need to specify which commands the user can run. Here, I am going to let the

user run only “ls”, “mkdir”, and “ping” commands. You can assign any commands of your choice. To do so, run the following commands:

```
# ln -s /bin/ls /home/ostechnix/bin/ls
# ln -s /bin/mkdir /home/ostechnix/bin/mkdir
# ln -s /bin/ping /home/ostechnix/bin/ping
```

Log out of your current account and login as ostechnix. Attempt to ping 8.8.8.8. You should receive an error at the command prompt. Try the ls and mkdir commands as well. Proceed to delete the ostechnix account. The above technique is typically how production Unix/Linux systems are handled.

Permissions

The ch* commands permit one to change permissions, aka ownership, of assets in the system. Be they files, directories or links to other objects in the system. Keep in mind that the object to be changed has to exist in the system for these commands to take effect.

chown

chown [options]... [OWNER][:[GROUP]] file

- -c, –changes like verbose but report only when a change is made
- -f, –silent, –quiet suppress most error messages
- -v, –verbose output a diagnostic for every file processed

chmod

chmod [options] mode[,mode] file1 [file2 ...]

The following are some of the usual options used with chmod:

- -f, –silent, –quiet: Suppresses most error messages
- -v, –verbose: Outputs a diagnostic for every file processed
- -c, –changes: Like verbose but reports only when a change is made
- -R, –recursive: Change files and directories recursively
- –help: Displays help and exits
- –version: Outputs version information and exits

chgrp

chgrp [options]... GROUP FILE

- -c, –changes like verbose but report only when a change is made
- -f, –silent, –quiet suppress most error messages
- -v, –verbose output a diagnostic for every file processed

Special bits

Setuid

When the setuid bit is used, the behavior described above it's modified so that when an executable is launched, it does not run with the privileges of the user who launched it, but with that of the file owner instead. So, for example, if an executable has the setuid bit set on it, and it's owned by root, when launched by a normal user, it will run with root privileges. It should be clear why this represents a potential security risk, if not used correctly.

An example of an executable with the setuid permission set is passwd, the utility we can use to change our login password.
We can verify that by using the ls command:

ls -l /bin/passwd

Which results in the folling output -

-rwsr-xr-x. 1 root root 27768 Feb 11 2017 /bin/passwd

How to identify the setuid bit? As you surely have noticed looking at the output of the command above, the setuid bit is represented by an s in place of the x of the executable bit. *The s implies that the executable bit is set, otherwise you would see a capital S.* This happens when the setuid or setgid bits are set, but the executable bit is not, showing the user an inconsistency: the setuid and setgit bits have no effect if the executable bit is not set. The setuid bit has no effect on directories.

Setgid

Unlike the setuid bit, the setgid bit has an effect on both files and directories. In the first case, the file which has the setgid bit set, when executed, instead of running with the privileges of the group of the user who started it, runs with those of the group which owns the file: in other words, the group ID of the process will be the same of that of the file.

When used on a directory, instead, the setgid bit alters the standard behavior so that the group of the files created inside said directory, will not be that of the user who created them, but that of the parent directory itself. This is often used to ease the sharing of files (files will be modifiable by all the users that are part of said group). Just like the setuid, the setgid bit can easily be spotted (in this case on a test directory):

ls -ld test

Which results in -

drwxrwsr-x. 2 egdoc egdoc 4096 Nov 1 17:25 test

This time the s is present in place of the executable bit on the group sector.

Sticky Bit
The sticky bit works in a different way: while it has no effect on files, when used on a directory, all the files inside the directory will be modifiable only by their owners. A typical case in which it is used, involves the /tmp directory. Typically this directory is writable by all users on the system, so to make impossible for one user to delete the files of another one, the sticky bit is set:

$ ls -ld /tmp

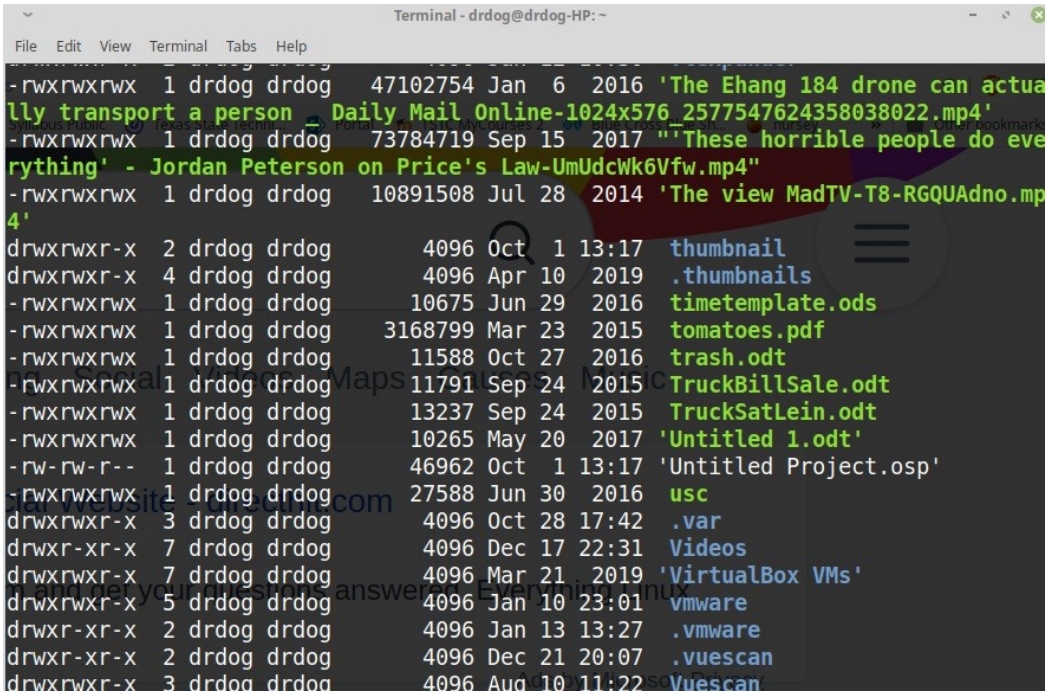
Which results in -

drwxrwxrwt. 14 root root 300 Nov 1 16:48 /tmp

In this case the owner, the group, and all other users, have full permissions on the directory (read, write and execute). The sticky bit is identifiable by a t which is reported where normally the executable x bit is shown, in the "other" section. Again, a lowercase t implies that the executable bit is also present, otherwise you would see a capital T.

Rights management overall
Unlike the Microsoft implementation of user management (distribution lists, security lists and domain rights inheritance) Unix/Linux depends on the above tools at the server level. Externally Unix/Linux depends on services like Kerberos, LDAP and Pam for those high order functions that Windows provides internally.

ls display layout showing permissions:



Objectives

In this project/lab the student will:

- Gain familiarity with account management tools

Equipment/Supplies Needed

- As specified in Lab 0.0.1.

Procedure

Perform the steps in this lab in the order they are presented to you. Answer all questions and record the requested information. Use the Linux Virtual Machine to perform lab activities as directed. Unless otherwise stated, all tasks done as a non-root user. If root access is needed use the sudo command.

Assignment

Lab Submissions Proof: Provide screenshots as indicated in the lab; upload your proof to Canvas for grading.

Launch Debian.

Part 1 - Users

Include adduser to the system -

1. `sudo apt-get install adduser`

Follow the prompts to complete. Attempt to add a user,

2. su -
3. adduser smythe1

Now confirm that the account has been created:

4. tail /etc/passwd

Can you confirm that the account has been created? How? Record a screenshot. Place that image in a Word or Writer document. What is his user ID number?

Let's become smythe1, try the following:

5. whoami
6. su - smythe1

Do the following:

7. whoami
8. passwd

What name was returned? Record a screenshot. Place that image in a Word or Writer document.

Perform:

9. exit
10. whoami

What account are you in now?

Close terminal, and reboot and log out of your account. Attempt to log in under Smythe account using the above password. Did it work? Record a screenshot. Place that image in a Word or Writer document.

Logout of smythe1 and back into your own at this point.

Part two – Groups

Let's consider Groups for a moment. Try:

10. less /etc/passwd

By default a user get their own group id. Can you find that group ID for Smythe in the above file? What is it? Place your answer in the Word or Writer document.

Have you considered an unusual event happening? Take a thought, place your observation in a Word or Writer document.

So smythe1 has his own ID. Take a look at this file,

11. less /etc/group

Can you find smythe1? Look for the news group entry. Note the group ID. So let's add smythe1 to it,

12. sudo usermod -a -G news smythe1

Note: we are not using a group command but the user commands to do this. Can you verify Smythe is in the group? Record a screenshot. Place that image in a Word or Writer document.

So lets create a group,

13. sudo groupadd test

Run:

14. less /etc/group

test should be at the bottom of the list. Are there any members for test? Add the Smythe account to the test group. Record a screenshot. Place that image in a Word or Writer document.

Part Three - Cleanup

Finally let's clean up. Execute:

15. sudo groupdel test

Remove smythe1 from news:

16. sudo deluser smythe1 news

Finally remove account smythe1:

17. sudo userdel smythe1

Execute:

18. cat /etc/passwd

Record a screenshot. Place that image in a Word or Writer document.

Rubric

Checklist/Single Point Mastery

Concerns Working Towards Proficiency	Criteria Standards for This Competency	Accomplished Evidence of Mastering Competency
	Criteria #1: Record screenshot of Part 1, step 4 (10 points)	
	Criteria #2: Record answer of Part 1, step 4 (10 points)	
	Criteria #3: Record screenshot in Part 1, step 7 (10 points)	
	Criteria #4: Record screenshot in Part 1, step 9 (20 points)	
	Criteria #5: Record screenshot of Part 2, step 10 (10 points)	
	Criteria #6: Answer to question in Part 2, step 12 (10 points)	
	Criteria #7: Record screenshot in Part 2, step 14 (10 points)	
	Criteria #8: Record screenshot in Part 2, step 18 (20 points)	