

Chapter 13

231011 Kevin Antonio Andrade Lopez

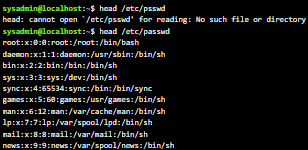


universidad tecnologica de aguascalientes

13.2

In this exercise you’ll learn about user’s accounts and the archives and the commands who displays the users info.

User and system accounts are defined in the /etc/passwd and /etc/shadow files. See the first ten lines of the /etc/passwd file:

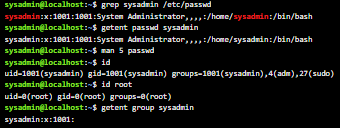


Note that this file contains a colon-delimited database of all user and system accounts available on this system.

User accounts are assigned to users to allow them access to the operating system. The sysadmin account that you used to log in to the system is a normal user account.

The root account is a special user account that has virtually unlimited access and control over the system. Sometimes called a “superuser” account.

System accounts are used by the operating system or the process services that run on it. By not having these files running as root, the system is kept more secure by limiting the damage that a composite service account could cause. System accounts are not used directly by regular users.



Note: In this case we do not have any network account, so the displayed result is like viewing the /etc/passwd file

You can see the documentation for the fields in the /etc/passwd file with the following command:

man 5 passwd

Remember that while viewing a man page, you can press the Enter key to move forward line by line, Space page by page, and q to exit.

The /etc/passwd file has the following fields delimited by colons:

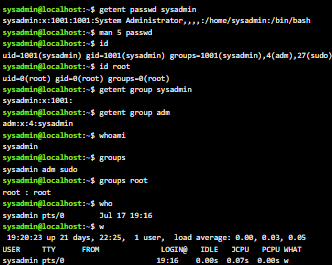
account:password:UID:GID:GECOS:directory:shell

Breakdown of these fields:

* The account field is the username.
* This field was originally used to store the password, however the password is now typically stored in the etc/shadow file. The x serves as an indicator.
* UID is the user identification number.
* GID is the user's primary group identification number.
* GECOS is the field normally used to store the user's full name.
* The directory is the user's home directory, where they are placed after first logging into the system.
* The shell field defines the shell such as /bin/bash for a normal user, or /sbin/nologin for an account using the system or a service.

The output of the commands shows your user identity in the form of a number and a name: uid=1001(sysadmin). It also shows your main group identity, ggid=1001(sysadmin), and all the groups you belong to groups=1001(sysadmin), 4(adm) and 27(sudo).. In this case, your user account It only belongs to three groups.

The /etc/group file, along with /etc/passwd, determines your group membership. Your default primary group is determined by matching your GID located in /etc/passwd with the GID defined for a group in /etc/group. Any secondary group memberships are defined in the /etc/group file.



Groups are used to control access to files.

By default any new file you create will be owned by your primary group. To have all new files you create owned by one of your child groups, use the newgrp command. The basic format of this command is newgrpgroup\_name. You have to be a member of a group to use the newgrp command. The id command will then display that group\_name as your primary group.

To change the group ownership of one of the existing files that your user owns, you can use the chgrp command. The basic format of this command is chgrpgroup\_name file\_name.

You must be a member of a group to change a group property of files to that group, and you must own the file.

The output of the who command has four columns:

The username: The first column shows the name of the user (sysadmin in the example above).

The terminal: The second column shows an identifier for a type of the terminal (tty in the example above). Each terminal has a name that is used by the superuser to control processes.

Date/Time: The third column is the date and time the session was started (Apr 11 14:32 in the first line of output in the example above).

Host: Although there is no output for the fourth column in this case, it can be a name or IP address of a local or remote host. The following forms indicate local logins: (:#) or (:#.#). Otherwise, a remote host can be displayed by name (if resolvable) or by IP address.

An entry for each user with their username, tty name, host, login time, idle time, JCPU (CPU time used by background jobs), PCPU (CPU time used by process current) and what is being executed on the current command line.

13.4 The root user

In this task, you will learn how to access the root user account in a couple of different ways.

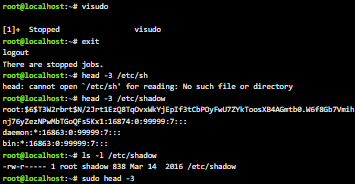
The root user account is special in Linux in that it has a virtually unlimited amount of control and access to the system. It is not actually the name of the user, but the UID with a value of zero that grants this authority.

In general, it is not considered good practice to log into the system as the root user. This is especially true in the graphical environment.

To access the root user account, the su or sudo commands are typically used.

The su command is usually used to switch users to start a new shell as another user; The default is the root user. The su command is normally used when you need to execute a series of commands as the root user.

The sudo command is typically used to run a single command as the root user by prefixing that command with sudo. The sudo command must be configured by the root user before a normal user can use it.



To configure the sudo command for regular users, use the visudo command. This requires root access as the visudo command fails if run by regular users.

The dash after the su command is a shorthand for the -l option, which makes the login performed by su a full login by running login scripts belonging to the root user.

Without -, or -l, the su command will change your user identity, but it will not run the new user's login scripts. This can cause some problems, especially when switching to the root account. In most cases when you change users, you want to perform a full login so that you can initialize the environment variables, functions, and aliases that are normally initialized for a user.

Try looking at the first few lines of the /etc/shadow file, a file that contains users' encrypted passwords and expiration information:

head -3 /etc/shadow

Note that the permissions on the /etc/shadow file indicate that only members of the shadow group have permission to view the file:

ls -l /etc/shadow

Note that the root user can view any file. This is because the root account has special privileges that transcend regular file permissions.

Use the sudo command to view the first lines of the /etc/shadow file:

sudo head -3 /etc/shadow

(provides the password for the sysadmin user: netlab123)

sysadmin@localhost:~$ sudo head -3 /etc/shadow

[sudo] password for sysadmin:

root:$6$T3W2rbrt$N/2Jrt1EzQ8TqOvxWkYjEpIf3tCbPOyFwU7ZYkToosXB4AGmtb0.W6f8Gb7Vmihnj76yZezNPwMbTGoQFs5Kx1:16874:0:99999:7:::

daemon:\*:16863:0:99999:7:::

bin:\*:16863:0:99999:7:::

sysadmin@localhost:~$

Important note: The password you provided was for your sysadmin account, not the root account. Once sudo is configured for your account, you do not need to know the root password to run sudo commands as the root user.

