

IE-0117 Programacion bajo plataformas abiertas

Laboratorio 2: Usuarios, permisos y credenciales en GNU/Linux

M. Sc. Ricardo Román Brenes - `ricardo.roman@ucr.ac.cr`

I-2018

Tabla de contenidos

1. Enunciado	1
1.1. Usuarios, grupos y permisos	1
1.2. Instalacion de paquetes y programas desde otros repositorios	2
1.3. Cron, crontab, rsync, ssh	2
1.4. Instalacion de programas desde codigo fuente	3
1.5. Capturas de pantalla	3

1. Enunciado

Entregue un archivo comprimido que incluya un directorio llamado informe con los archivos necesarios para generar el PDF del informe. El informe debe contener la documentacion necesaria que demuestre que se llego a una solucion satisfactorio de los problemas resueltos

1.1. Usuarios, grupos y permisos

1. Cambie de usuario a root. `sudo -s`
2. Cree un usuario nuevo llamado labo2, con la contraseña labo2.
3. Cambie de usuario a labo2. Cree un directorio llamado PruebasPermisos en el \$HOME de este usuario.
4. Dentro de este directorio nuevo, cree un archivo llamado README, que su contenido sea una única linea que diga: Realizando pruebas de permisos.

5. Con un solo comando, cambie los permisos del directorio PruebasPermisos y todos los archivos que contiene, de manera tal que solamente el usuario dueño del directorio, así como el grupo al que pertenece, tengan permisos de lectura, escritura y ejecución. Los demás no tendrán permisos de nada.
6. Cambie a su usuario original. Trate de escribir al directorio PruebasPermisos y al archivo README contenido. Documente el resultado.
7. Cambie de usuario a root nuevamente.
8. Cree un grupo nuevo llamado grupolabo2, agregue el usuario labo2 y su propio usuario a este grupo.
9. Cierre la sesión gráfica de Linux y vuelva a iniciarla.
10. Vuelva a iniciar sesión como su usuario, y cambie en terminal el usuario a labo2.
11. Cambie recursivamente el grupo de pertenencia del directorio PruebasPermisos al grupo grupolabo2.
12. Cierre la sesión del usuario labo2 y vuelva a su propio usuario. Trate de escribir al directorio PruebasPermisos y al archivo README. Documente el resultado.
13. Explique con sus propias palabras el procedimiento que se realizó anteriormente. ¿Por qué antes su usuario no podía escribir en el archivo? ¿Por qué ahora sí puede? En principio no era posible modificar con mi usuario el archivo README ni el directorio PruebasPermisos ya que solo el usuario labo2 tenía los permisos para hacerlo. Luego al crear un grupo grupolabo2 e incluir mi usuario y el usuario labo2 en este grupo, y modificar los permisos para que quienes pertenezcan al grupo puedan modificarlo, fue posible hacer modificaciones desde mi usuario.

1.2. Instalación de paquetes y programas desde otros repositorios

1. Siga el tutorial de instalación de Spotify en Linux que se encuentra en el siguiente enlace: <https://www.spotify.com/cr/download/linux/>. Documente en su informe el porqué se realizó cada uno de los pasos.

1.3. Cron, crontab, rsync, ssh

1. Realice revisión bibliográfica de qué es cron. Cron es un daemon que se ejecuta desde el mismo instante en el que arranca el sistema. Compruebe si existe alguna tarea para ser ejecutada de acuerdo a la hora configurada en el sistema. En función de la distribución, se inicia utilizando las carpetas /etc/rc.d/ o /etc/init.d y a cada minuto comprueba los ficheros /etc/crontab en busca de posibles ejecuciones [1]
2. Escriba un breve resumen de cuál es el formato del archivo crontab y cómo se utiliza. Crontab es un archivo de texto que posee una lista con todos los scripts a ejecutar. Es posible modificarlo con el comando crontab -e.
La estructura del comando es la siguiente:
min(0-59) hr(0-23) día(0-31) mes(1-12) día de la semana(0-6) comando a ejecutar
3. Investigue sobre el uso del comando rsync, y su uso a través de ssh. Puede utilizar este enlace como guía <https://www.digitalocean.com/community/tutorials/how-to-copy-files-with-rsync-over-ssh>

4. Cree llaves ssh para su usuario de manera que no le pida contrasena a la hora de conectarse la computadora de su companero.
5. Ejecute un comando de rsync sobre ssh que permita respaldar su \$HOME en algún directorio del \$HOME del usuario de su companero.
6. Programe su computadora para realizar el ejercicio el punto anterior todos los miercoles a las 3:00am.

1.4. Instalacion de programas desde codigo fuente

1. Visite el siguiente enlace: <https://ffmpeg.org/download.html>.
2. Descargue el fichero comprimido ffmpeg-3.4.2.tar.bz2.
3. Proceda a instalar el programa a traves del codigo fuente descargado. Nota: recuerde revisar el README o el INSTALL para conocer las dependencias e instalarlas antes de iniciar la configuracion del paquete y la compilacion. Si no lo hace el script de configure o la compilacion podrian fallar.

1.5. Capturas de pantalla

```
miguel@miguepc:~$ sudo -s
[sudo] password for miguel:
root@miguepc:~# adduser labo2
Adding user `labo2' ...
Adding new group `labo2' (1001) ...
Adding new user `labo2' (1001) with group `labo2' ...
Creating home directory `/home/labo2' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for labo2
Enter the new value, or press ENTER for the default
  Full Name []:
    Room Number []:
    Work Phone []:
    Home Phone []:
    Other []:
Is the information correct? [Y/n] y
root@miguepc:~# su labo2
labo2@miguepc:/home/miguel$ mkdir ~/PruebasPermisos
labo2@miguepc:/home/miguel$ touch ~/PruebasPermisos/README
labo2@miguepc:/home/miguel$ nano PruebasPermisos/README
```

Figura 1: comandos de enunciado del 1 al 4

```
labo2@miguepc:/home/miguel$ nano ~/PruebasPermisos/README
labo2@miguepc:/home/miguel$ chmod -R 770 ~/PruebasPermisos
labo2@miguepc:/home/miguel$ su miguel
Password:
miguel@miguepc:~$ touch ../labo2/PruebasPermisos/prueba
touch: cannot touch '../labo2/PruebasPermisos/prueba': Permission denied
miguel@miguepc:~$ cat ../labo2/PruebasPermisos/README
cat: ../labo2/PruebasPermisos/README: Permission denied
miguel@miguepc:~$ sudo -s
[sudo] password for miguel:
root@miguepc:~# addgroup grupolabo2
Adding group `grupolabo2' (GID 1002) ...
Done.
root@miguepc:~# usermod -a -G grupolabo2 labo2
root@miguepc:~# usermod -a -G grupolabo2 miguel
root@miguepc:~# pkill -KILL -U miguel
```

Figura 2: comandos de enunciado del 5 al 9

```

miguel@miguepc:~$ su labo2
Password:
labo2@miguepc:/home/miguel$ pwd
/home/miguel
labo2@miguepc:/home/miguel$ cd ../labo2
labo2@miguepc:~$ chgrp -R grupolabo2 PruebasPermisos
labo2@miguepc:~$ su miguel
Password:
miguel@miguepc:/home/Labo2$ pwd
/home/Labo2
miguel@miguepc:/home/Labo2$ touch PruebasPermisos/prueba
miguel@miguepc:/home/Labo2$ cat PruebasPermisos/README
Realizando pruebas de permisos
miguel@miguepc:/home/Labo2$

```

Figura 3: comandos de enunciado del 10 al 12

```

miguel@miguepc:~/Desktop/gitlabos/A33805labo2/informe$ sudo apt install snapd
[sudo] password for miguel:
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following NEW packages will be installed:
snapd
0 upgraded, 1 newly installed, 0 to remove and 321 not upgraded.
Need to get 12,4 MB of archives.
After this operation, 57,8 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 snapd amd64 2.34.2+18.04 [12,4 MB]
Fetched 12,4 MB in 47s (264 kB/s)
Selecting previously unselected package snapd.
(Reading database ... 404070 files and directories currently installed.)
Preparing to unpack .../snapd 2.34.2+18.04 amd64.deb ...
Unpacking snapd (2.34.2+18.04) ...
Setting up snapd (2.34.2+18.04) ...
Created symlink /etc/systemd/system/multi-user.target.wants/snapd.service → /lib/systemd/system/snapd.service.
Created symlink /etc/systemd/system/multi-user.target.wants/snapd-core-fixup.service → /lib/systemd/system/snapd-core-fixup.service.

```

Figura 4: Instalando snapd

```

Created symlink /etc/systemd/system/snapd.service → /lib/systemd/system/snapd.service.
Created symlink /etc/systemd/system/sockets.target.wants/snapd.socket → /lib/systemd/system/snapd.socket.
Created symlink /etc/systemd/system/final.target.wants/snapd.service → /lib/systemd/system/snapd.service.
Processing triggers for man-db (2.8.3-2) ...
snapd.service is a disabled or a static unit, not starting it.
Processing triggers for man-db (2.8.3-2) ...
miguel@miguepc:~/Desktop/gitlabos/A33805labo2/informe$ snap install spotify
2018-09-15T16:03:15-06:00 INFO Waiting for restart...
spotify 1.0.89.313.g34a58dea-5 from 'spotify' installed
miguel@miguepc:~/Desktop/gitlabos/A33805labo2/informe$

```

Figura 5: Instalando spotify

```

ie0217@miguel:~$ rsync -avz -e "ssh -o StrictHostKeyChecking=no -o UserKnownHostsFile=/dev/null" --progress ./id_rsa_text.txt jp@192.168.13.177:/home
Warning: Permanently added '192.168.13.177' (ECDSA) to the list of known hosts.
sending incremental file list
id_rsa_text.txt
 395 100% 0.00KB/s 0:00:00 (xfr#1, to-chk=0/1)
rsync: mkstemp "/home/.id_rsa_text.txt.hDIRjp" failed: Permission denied (13)

sent 443 bytes received 117 bytes 65.88 bytes/sec
total size is 395 speedup is 0.71
rsync error: some files/attrs were not transferred (see previous errors) (code 2) at main.c(1196) [sender=3.1.2]

```

Figura 6: Uso de ssh para transferir un fichero a otra computadora

```

# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
45 19 11 * * rsync -avz -e "ssh -o StrictHostKeyChecking=no -o UserKnownHostsFil
e=/dev/null" --progress ./id_rsa_text.txt jp@192.168.13.177:/home/jp

```

Figura 7: Uso de crontab y ssh para respaldar periodicamente un fichero a otra computadora

```

le0217@miguel:~$ wget https://ffmpeg.org/releases/ffmpeg-4.0.2.tar.bz2
--2018-09-11 19:48:25-- https://ffmpeg.org/releases/ffmpeg-4.0.2.tar.bz2
Resolving ffmpeg.org (ffmpeg.org)... 79.124.17.100
Connecting to ffmpeg.org (ffmpeg.org)|79.124.17.100|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10424065 (9.9M) [application/x-bzip2]
Saving to: 'ffmpeg-4.0.2.tar.bz2'

ffmpeg-4.0.2.tar.bz 100%[=====] 9.94M 1.11MB/s in 17s
2018-09-11 19:48:43 (584 KB/s) - 'ffmpeg-4.0.2.tar.bz2' saved [10424065/10424065]

```

Figura 8: Instalacion de programas desde codigo fuente

Referencias

- [1] Adrian Crespo. Como utilizar cron y crontab en linux para progrmar tareas. 2017.