

Materia: Sistemas Operativos
Profesor: Dr. Moisés Alencastre

Gilberto Silva González
A01018723

Práctica Final

Paquetes RPMs que no deben estar instalados:

Hice un Shell para buscar paquetes, y comprobar que la mayoría de los paquetes fueron desinstalados a excepción de algunos paquetes.

Todo lo relacionado con el mail

```
#smtp  
rpm -qa | grep -i smtp
```

```
#sendmail  
echo -Sendmail:  
rpm -qa | grep -i sendmail  
echo -----
```

FTp, telnet, rdate, rsh, talk, finger, etc

```
#ftp  
rpm -qa | grep -i ftp
```

```
#telnet  
rpm -qa | grep -i telnet
```

```
#rdate  
rpm -qa | grep -i rdate
```

```
#rlogin  
rpm -qa | grep -i rlogin
```

```
#rsh  
rpm -qa | grep -i rsh
```

```
#talk  
rpm -qa | grep -i talk
```

```
#finger  
rpm -qa | grep -i finger
```

```
#rsync  
rpm -qa | grep -i rsync
```

Lo relacionado con Samba

```
#samba  
rpm -qa | grep -i samba
```

Java de Linux, gcj, jakarta

```
#gcj  
rpm -qa | grep -i gcj
```

```
#jakarta  
rpm -qa | grep -i jakarta
```

Nfs, Isdn, nis, ypbind, rpc

```
#NFS  
rpm -qa | grep -i nfs
```

```
#ISDN  
rpm -qa | grep -i isdn
```

```
#NIS  
echo -NIS:  
rpm -qa | grep -i nis  
echo -----
```

```
#ypbind  
rpm -qa | grep -i ypbind
```

```
#RPC  
echo -RPCs:  
rpm -qa | grep -i rpc  
echo -----
```

```
#vino  
rpm -qa | grep -i vino
```

```
#vinager  
rpm -qa | grep -i vinager
```

```
#sudo  
echo -Sudo:  
rpm -qa | grep -i sudo  
echo -----
```

```
#cups  
echo -Cups:  
rpm -qa | grep -i cups
```

echo -----

#openconnect
rpm -qa | grep -i openconnect

#vpn
rpm -qa | grep -i vpn

#pptp
rpm -qa | grep -i pptp

#raid
rpm -qa | grep -i raid

#torrent
rpm -qa | grep -i torrent

#transmission
rpm -qa | grep -i transmission

#msn
rpm -qa | grep -i msn

#msn
rpm -qa | grep -i messenger

#empathy
rpm -qa | grep -i empathy

#evolution
rpm -qa | grep -i evolution

#packagekit
echo -PackageKit:
rpm -qa | grep -i packagekit
echo -----

Bluetooth

#bluetooth
echo -Bluetoth:
rpm -qa | grep -i bluetooth
echo -----

Servidores
rpm -qa | grep -i news

Servicios

En un shell puse los servicios que deben ser levantados, el servidor web (puerto 80 y 443), el segundo servidor web (puerto 8080) y el secure shell (puerto 22). Todos estos están deshabilitados para garantizar mayor protección, el único puerto activo es el 68, que se encarga de manejar el protocolo DHCP, el cual nos garantiza el acceso a la red inalámbrica.

Shell:

```
#!/bin/bash
```

```
#Inicia el primer servidor web, puerto 80
service httpd start
#Inicia el segundo servidor web, compilado a mano, puerto 8080
/usr/local/bin/apachectl start
#Inicia el servicio del secure secure shell
service sshd start
```

Para que estos puertos estén abiertos y puedan ser accedidos desde afuera, se necesita meterse a la interfaz gráfica del firewall, y en la pestaña de “Trusted Services” palomear el puerto 80 y el 443, además en la pestaña “Other Ports” añadir manualmente el puerto 8080 y palomearlo, de esta manera ya se puede acceder desde afuera.

Para que el firewall tenga que filtrar el puerto 3306 (mysql), se ingresó el siguiente comando:
-> iptables -A INPUT -p tcp -s 0/0 -d 0/0 --dport 3306 -i wlan0 -j REJECT

Y para que filtrara el puerto 68:
-> iptables -A INPUT -p udp -s 0/0 -d 0/0 --dport 68 -i wlan0 -j REJECT

Para enlistar lo del firewall, se usa:
-> iptables -list
O también se puede leer directamente desde el archivo:
-> cat /etc/sysconfig/iptables

Para guardar la configuración en memoria:
iptables-save > /etc/sysconfig/iptables

Actualizar Firefox

Paquetes que se necesitan para la actualización a la versión 20.0:

```
-> nspr.4.9.5-2.fc17.x86_64.rpm
-> nspr-devel.4.9.5-2.fc17.x86_64.rpm
-> nss.3.14.3-1.fc17.x86_64.rpm
-> nss-softokn3.14.3-1.fc17.x86_64.rpm
-> nss-softokn-freelb.3.14.3-1.fc17.x86_64.rpm
-> nss-tools.3.14.3-1.fc17.x86_64.rpm
-> nss-sysinit.3.14.3-1.fc17.x86_64.rpm
-> nss-util.3.14.3-1.fc17.x86_64.rpm
-> xulrunner.10.0-1.fc17.x86_64.rpm
```

Comando (siendo root):

```
rpm -Uvh nspr.4.9.5-2.fc17.x86_64.rpm nspr-devel.4.9.5-2.fc17.x86_64.rpm nss.3.14.3-1.fc17.x86_64.rpm nss-softokn3.14.3-1.fc17.x86_64.rpm nss-softokn-freelb.3.14.3-1.fc17.x86_64.rpm nss-tools.3.14.3-1.fc17.x86_64.rpm nss-sysinit.3.14.3-1.fc17.x86_64.rpm nss-util.3.14.3-1.fc17.x86_64.rpm xulrunner.10.0-1.fc17.x86_64.rpm
```

Kernel

Para buscar una versión más reciente del kernel, ir a la siguiente página:

<https://www.kernel.org/>

Descargar el código fuente, para compilar el código fuente:

```
-> make clean  
-> ./configure  
-> make  
-> make install
```

Si no hubo errores en la descarga, borrar la versión anterior del kernel para ya sólo tener la nueva.

Instalar chrome:

Paquetes que se necesitan:

```
-> google-chrome-stable-24.0.1312.70-181759.x86_64  
-> xorg-x11-drv-openchrome-0.2.905-5.fc17.x86_64
```

Comando (siendo root):

```
rpm -ivh google-chrome-stable-24.0.1312.70-181759.x86_64 xorg-x11-drv-openchrome-0.2.905-5.fc17.x86_64
```

Web

Se creó un nuevo grupo y un usuario nuevo perteneciente a ese grupo (todo como root):

```
-> groupadd -g 1001 Developer  
-> useradd Pruebas -g 1001 -p P1r2u3b4a5.  
-> chgrp 1001 /html
```

Y el resultado de un `ls -l` en el directorio `/var/www`, debe ser el siguiente:

```
...  
drwxrwxr-x. 2 root Developer 4096 May  7 11:36 html  
...
```

Para comprobar que está funcionando el servidor web, se debe ingresar a las siguientes direcciones e el navegador:

```
-> http://localhost  
-> https://localhost
```

*Ojo: Tener los servicios 80 y 443 levantados para un correcto funcionamiento.

Crear un hola mundo desde html y php desde este nuevo usuario:

su – Pruebas

```
cd /var/www/html
```

Ejemplo de un “Hola Mundo” en formato HTML:

```
vi HolaMundo.html
```

```
<html>
  <title>
  </title>
</head>

<body>
  <font color="#ff0000">
    <b>
      Hola Mundo
    </b>
  </font>
</body>
</html>
```

Ejemplo de un “Hola Mundo en php:

```
vi HolaMundo.php
```

```
<?php print "Hola Mundo" ?>
```

Ejemplo de un php info:

```
vi info.php
```

```
<?php phpinfo(); ?>
```

Para probarlos respectivamente en el servidor:

<http://localhost/HolaMundo.html>

<http://localhost/HolaMundo.php>

<http://localhost/info.php>

SELinux

Para ver todo lo que permite SELinux, ingresamos el siguiente comando:

```
-> sestatus -b | grep ' on'
```

Hay que bajar el cgi y el bluetooth porque son inseguros:

```
setsebool httpd_enable_cgi off
```

```
setsebool xguest_use_bluetooth off
```

Ahora su estado ya aparece como “off”, brindándonos mayor seguridad.

Variables de entorno

Para editar las variables de entorno, uno se tiene que meter al siguiente archivo de configuración:

```
-> vi ~/.bashrc
```

Una vez dentro se deben poner manualmente los exports de las variables:

```
-> export JAVA_HOME=/usr/java/latest
```

```
-> export PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:$JAVA_HOME/bin
-> export CLASSPATH=$JAVA_HOME/jre/lib/rt.jar:.
-> export LD_LIBRARY_PATH=/usr/autodesk/mudbox/lib:/lib64:/usr/lib64:/usr/local/lib
    64/cuda/lib64:/lib:/usr/lib:/usr/local/lib:/usr/local/cuda/lib
```

BootLoader

El archivo de configuración del bootloader, se encuentra en: /boot/grub2.
Para modificarlo (siendo root):

```
-> vi /boot/grub2/grub.cfg
set timeout = 8
```

Ahora ya le cambie el timeout por default.

Java y plugins

Para hacer funcionar el java, se necesitan establecer las siguientes dos ligas simbólicas, para ello, se seguirá el siguiente procedimiento (como root):

```
-> cd /usr/java/default
-> ln -s /usr/java/latest/jre/lib/amd64/libjavaplugin_jni.so
    /usr/lib64/mozilla/plugins/libjavaplugin_jni.so
-> ln -s /usr/java/latest/jre/lib/amd64/libnprp2.so /usr/lib64/mozilla/plugins/libnprp2.so
```

Así pues, queda de la siguiente manera en la carpeta /usr/lib64/mozilla/plugins/ :

```
->ls -l
total 388
lrwxrwxrwx. 1 root root  41 Feb 13 11:48 libflashplayer.so -> /usr/lib64/flash-
plugin/libflashplayer.so
lrwxrwxrwx. 1 root root  51 Mar  6 11:42 libjavaplugin_jni.so ->
/usr/java/latest/jre/lib/amd64/libjavaplugin_jni.so
-rwxr-xr-x. 1 root root 6056 Apr 24 2012 librhythmbox-itms-detection-plugin.so
-rwxr-xr-x. 1 root root 106608 May  8 2012 libtotem-cone-plugin.so
-rwxr-xr-x. 1 root root 111312 May  8 2012 libtotem-gmp-plugin.so
-rwxr-xr-x. 1 root root 73472 May  8 2012 libtotem-mully-plugin.so
-rwxr-xr-x. 1 root root 86256 May  8 2012 libtotem-narrowospace-plugin.so
```

De esta manera los plugins de java ya pueden funcionar, y ya se pueden ver applets en internet. Para comprobarlo, se puede ver en la siguiente página, con algunos ejemplos:

-> <http://mainline.brynmawr.edu/Courses/cs110/spring2002/Applets/Examples.html>

MySQL

La primera vez que se entra:

```
-> mysql -u root -p
```

Ingresar el siguiente comando:

```
-> use mysql
```

```
-> Update user SET Password=PASSWORD('C0mpl1 c4d02') WHERE user='root';
```

```
-> service mysqld restart
```

Para importar una base de datos:

```
-> mysql -u root -p < /home/gilicrack/Documents/daai-nomina.sql
```

```
mysql> DELETE FROM mysql.user
```

```
-> WHERE User='user_name' and Host='host_name';
```

```
mysql> FLUSH PRIVILEGES; #para guardar los cambios
```

```
-> GRANT select, insert ON mysql.* TO daddy@localhost IDENTIFIED BY 'G1i2l3b4e5r6t7o8';
```

insert into user values

[illegible]

```
update user set password=PASSWORD('passwd') where User='daddy';
```

flush privileges;

Compilar y ejecutar en java:

-> javac "nombre_del_archivo.java"

Si no hay error:

-> java "nombre de la clase"

Compilar y ejecutar en c:

```
-> gcc "nombre_del_archivo".c
```

Si no hay error, ejecutar:

-> ./a.out

Compilar y ejecutar en c++:

-> g++ "nombre_del_archivo".cpp

Si no hay error, ejecutar:

-> ./a.out

Se hace por medio de las bitácoras:

Las dos más importantes del sistema se encuentra son:

-> tail -f /var/log/messages

-> tail -f /var/log/secure

Las del servidor web:

-> tail -f /var/log/httpd/access_log

-> tail -f /var/log/httpd/error_log

-> tail -f /var/log/httpd/ssl_access_log

-> tail -f /var/log/httpd/ssl_error_log

-> tail -f /var/log/httpd/ssl_request_log

La de la base de datos (mysql)

-> urtail -f /var/log/mysql.log

Shell en AIX

#!/#Servicios activos

lssrc -a | awk '/active/ {print \$1}'

#Imprime unicamente los números de procesos de root

ps -fea | awk '/root/ {print \$2}'

#Imprime la version de AIX

uname -a | awk '{print \$4 "." \$3}'

Shells en bash

-> Shell 1:

#!/bin/bash

#Shell 1 en bash (User IDs)

#El delimitador son dos puntos. Imprime la columna 3 del archivo /etc/passwd

El parámetro -n hace que no se salte línea al imprimir

awk -F':' -n '{printf("%s ", \$3)}' /etc/passwd >> /tmp/backup

-> Shell 2:

#!/bin/bash

#Ejecuta el comando disk usage, -k(kilobytes), en el kernel que tengo

#Tan solo se imprime la primer columna

du -k /boot/vmlinuz-3.7.3-101.fc17.x86_64 | awk '{print \$1}' >> /tmp/backup

-> Shell 3:

#!/bin/bash

Imprime los .ko sin su extension

#Se enlistan los modulos relacionados con IPv6, se imprime sólo la primer columna, y se le pasa a un comando tr (translate) que traduce el .ko a un espacio

```
ls /lib/modules/3.7.3-101.fc17.x86_64/kernel/net/ipv6 | awk '{ print $1}' | tr ".ko" " " >> /tmp/backup
```

-> Shell 4:

```
#!/bin/bash
```

```
## Mac Address de los access points
```

```
#escanea y se lo pasa al archivo escaneo
```

```
iwlist wlan0 scan > escaneo
```

```
#Donde esta la palabra Address, imprime la columna 5, sin saltarse de renglón y
```

```
#con separador de dos puntos
```

```
awk '/Address:/ {printf("%s:", $5)}' escaneo >> /tmp/backup
```