Ejercicios Dia 2: Taller CEDIA

1 de Marzo, 2004

Preparamos por el dia

Primero, login usando el interfaz grafico.

Ahora, apreta "alt-ctrl-F2" - Esto te da un terminal de texto, pero todovia estamos corriendo en el nivel de inicializacion 5.

Login como "root"

Teclea "init 3"

Ahora vamos a arreglar los PCs para que tengan una direccion de IP estatica, que tu usuario puede usar comandos priveligiados, y para que usan el teclado de Español. Primero, cambiamos la direccion de IP permanente a ser estatica:

```
cd /etc/sysconfig/network-scripts/
vi ifcfg-eth0
```

En este archivo va a especificar los datos sobre el dispositvio eth0 y vamos a apagar el uso de DHCP para recibir direcciones de IP:

```
USERCTL=no
PEERDNS=yes
DEVICE=eth0
BOOTPROTO=none
ONBOOT=yes
TYPE=Ethernet
IPADDR=192.188.58.xx
NETMASK=255.255.255.224
GATEWAY=192.188.58.65
NETWORK=192.188.58.64
BROADCAST=192.188.58.95
```

Donde dice "xx" usa la direccion por tu maquina que te hemos dado en antes. El rango es 192.188.68.66-94 disponible a la sala de nuestro taller.

Ahora daremos privilegios a tu usuario para que se puede correr comandos privilegiados sin tener que ser root.

```
vi /etc/sudoers
```

Y busca la linea que dice:

```
# User privilege specification
```

```
root    ALL=(ALL) ALL
```

Abajo la entrada por root agrega:

usuario ALL=(ALL) ALL

Graba el archivo, y sale de la cuenta root usando "logout" - Entra como tu usuario y prueba si puedes usar "su" en un terminal:

```
su (contraseña de root)
```

Y, si tu session de terminal deberia cambiar por ser una de root.

Finalmente, para que tiene un teclado que funciona bien en Español hacemos lo siguiente:

```
vi /etc/sysconfig/keyboard
```

En este archive cambia la linea que dice 'KEYTABLE="us"' a 'KEYTABLE="es"' - Graba el archivo (:wq), y luego:

```
service keytable on
```

Ahora deberia tener un teclado funcionando bien en Español.

1.) Las particiones

Primero, login como tu usuario.

Ahora, en un terminal/shell veamos las particiones. Tipea:

```
df
df -h
```

Que diferenica viste entre "df" y "df -h". Y, donde se vea cuanto swap tienes?

```
cat /etc/swaps
top ("q" para salir)
```

Se puede imaginar que el comando "top" usa el interfaz /proc/swaps por algo de los datos que muestra. Hablaremos de /proc mas adelante.

Si quieres ver el estado de los particiones hay dos herramientas graficas por esto. Primero, como vi, la herramienta mas comun, pero un poco mas dificl para usar es "fdisk" - Para usarla tipea:

```
sudo fdisk
```

Para ver el estado de tus particiones apreta "p".

El comando fdisk, y luego cfdisk, son comandos privilegiados que tienen que estar corrido como root. Un comando un poco mas facil de usar es cfdisk. Tipea:

```
sudo cfdisk
```

Anotaste que no tuviste que usar la contraseña de root la segunda vez que usaste "sudo"?

2.) Montando sistemas de archivo

Primero vea que hay en /etc/fstab (y, para que usamos "cat" y no "more" o "less"?).

```
cat /etc/fstab
```

Ahora, pasa algunos minutos leyendo sobre las cosas asociadas con /etc/fstab:

```
man fstab
man mount
man fs
```

Ahora, vamos a usar el poder de Linux/Unix para crear un sistema de archivo y montarlo como queremos. Primero vamos a crear un imagen de ISO (ISO9660 que se puede usar para quemar un compact). Por ser caso, esto es un ejercicio un poco "inventado". Usamos el comando "mkisofs" por esto. Se lea "man mkisofs" va a ver que es bastante involcrado:

```
cd /usr/src
mkisofs -o /tmp/kernel-2.4.20-8.iso -v -l -L
-allow-lowercase linux-2.4.20-8
```

Se demora un poco en terminar el comando. Cuando termina vamos a crear un lugar para montar el nuevo sistema de archivo y verlo:

```
sudo mkdir /mnt/kernel
ls /mnt/kernel (no hay nada)
mount -o loop /tmp/kernel-2.4.20-8.iso
/mnt/kernel
```

Ahora, vaya a /mnt/kernel y vea que hay. Existe todo los archivos y directorios que estaban (y todovia estan) en /usr/src/linux-2.4.20-8, pero estan en /mnt/kernel, pero realmente estan dentro el archivo /tmp/kernel-2.4.20-8.iso, que realmente representa un sistema de archivos ISO9660 que es un estandar por los compact discs...

Que pasa si quieres que tu sistema siempre tiene disponsible los archivos de que se representa /tmp/kernel-2.4.20-8.iso? Si creas que esto no pasa, vea nuestro servidor por el clase - tenemos los tres compacts de Red Hat disponible usando este "truco" de "mount -o loop fn.iso /directorio"

Bueno, para hacer que siempre estan disponible tienes que poner una entrada en el archivo de /etc/fstab que describe como montar el archivo /tmp/kernel-2.4.20-8.iso cada vez que se inicializa tu maquina. El formata de /etc/fstab parece un poco complicado, pero hay un truco - vea adentra /etc/mtab... Ahi tiene el formato exactamente como necesitas por /etc/fstab. Entonces, haz esto:

```
cat /etc/mtab
```

Y, copia el texto:

/tmp/kernel-2.4.20-8.iso /mnt/kernel iso9660 rw,loop=/dev/loop0 0 0

y ponerlo *al fin* del archivo /etc/fstab. No se pone arriba porque es necesario tener montado las particiones que tiene el punto de montar (/mnt/kernel) *antes* que se trata de montar el archivo de ISO.

Ahora, si quieres asegurar que el formato es correcto corre el comando:

```
mount -a
```

Que monte todo que hay en /etc/fstab, y que revisa el formato. Usa el comando "df" o "df -h" para ver como se vea.

3.) Permisos y los archivos*

*Referencia: Shah, Steve, "Linux Administration: A Beginner's Guide", 2nd. ed., Osborne press, New York, NY.

Si veas los archivos en un directorio usando "ls -al" va a ver los permisos que tiene cada archivo y directorio. Aqui es un muestra:

```
drwxrwxr-x    3 hervey   hervey       4096 Feb 25 09:49 directorio
-rwxr--r--   12 hervey   hervey       4096 Feb 16 05:02 archivo2
```

La columna a la izquierda es que es importante. Se puede verla asi:

```
Tipo Usuario Grupo Mundo links  dueno  grupo  tamaño fecha  hora  nombre
d    rwx     rwx   r-x   3      hervey hervey 4096   Feb 25 09:49 directorio
-    rwx     r     r     12     hervey hervey 4096   Feb 16 05:02 archivo2
```

Entonces, el directorio tiene acceso r (leer), w (escribir), x (ejecutar) al usuario, grupo, y al mundo tiene leer (r) y ejecutar (x). El archivo "archivo2" tiene un poco meno acceso con solo leer (r) permitido al mundo (todo lo demas usuarios en el sistema) y leer al grupo "hervey" tambien.

Para cambiar los permisos se usa el comando "chmod". Chmod usa un sistema de base ocho (octals) para configurar permisos, o se puede usar una forma especificando una columna a la vez.

fLos permisos tienen valores asi:

Letra	Permiso	Valor
R	Leer	4
W	Escribir	2
X	Ejecutar	1

Entonces, se puede dar permiso a un archivo usando la suma de las columnas de permiso que quiere por cada tipo de permision. Aqui es un ejemplo:

Letra	Permiso	Valor
---	Ninguno	0
r--	Leer solo	4
rw-	Leer y escribir	6
rwx	Leer, escribir, y ejecutar	7
r-x	Leer y ejecutar	5
--x	Ejecutar	1

Esto es una columna, entonces, para dar todo las combinaciones se tiene una tabla asi:

Permiso	Equivalente numerico	Descripcion
-rw-------	600	Dueno tiene el permiso leer y escribir.
-rw-r--r--	644	Dueno tiene el permiso leer y escribir. Grupo y mundo tiene permiso leer.
-rw-rw-rw-	666	Todos (dueno, grupo, mundo) tiene permiso de leer y escribir (peligroso).
-rwx------	700	Dueno tiene permiso leer, escribir, ejecutar.
-rwxr-xr-x	755	Dueno tiene permiso leer, escribir, ejecutar. Resto del mundo tiene permiso leer y ejecutar. (tipico por paginas de web, o 644).
-rwxrwxrwx	777	Todos tienen todo acceso (leer, escribir, ejecutar).
-rwx--x--x	711	Dueno tiene permiso de leer, escribir, ejecutar. Grupo y mundo tiene permiso de ejecutar.
drwx------	700	Solo el dueno tiene acceso a este directorio. Todo los directorios requiere que "x" esta puesto.
drwxr-xr-x	755	Dueno tiene todo acceso al directorio. Lo demas se puede ver los contenidos.
drwx--x--x	711	Todos pueden leer los archivos en el directorio, pero tienen que saber el nombre porque no es ejecutable.

Entonces, ahora practicamos poniendo los permisos. Como usuario normal haz lo siguiente:

```
cd (que hace el comando "cd" asi?)
echo "archivo de prueba" > leer.txt
chmod 444 leer.txt
```

A pesar que el archivo no tiene permiso de escribir por el dueno, el dueno todovia puede cambiar el permiso al archivo para que es posible escribirlo:

```
chmod 744 leer.txt
```

O, se puede hacer esto usando este forma de chmod:

```
chmod u+w leer.txt
```

Las formas de chmod, para agregar permisos, si no usa los numeros son:

chmod u+r, chmod u+w, chmod u+x
chmod g+r, chmod g+w, chmod g+x
chmod a+r, chmod a+w, chmod a+x

Nota que "a+r" es por el mundo. El "a" es por "all" o "todos" en Ingles.

Ahora, cambia el archivo para que el dueno del archivo no se puede leerlo, pero se puede escribir al archivo...

```
chmod u-r leer.txt
```

O, se puede hacer algo como:

```
chmod 344 leer.txt
```

Y, probablemente fijaste que se puede usar el "-" para sacar permisos de un archivo, tambien.

Finalmente, el concepto que cuando se ejecuta un archivo, normalemente, se lo ejecuta con los permisos de la persona que haz el comando. Por ejemplo, si el usuario "angelo" tipea "netstat", el programa "netstat" corre con sus privilegios. Pero, si quieres que netstat siempre corre con los permisos del dueno, o el grupo a que pertenece el archivo se puede configurar el bit de "SetUID" o de "SetGID". Uno se lo hace con el comando "chmod" - y, por ser caso, ojo con este - puede ser una mala idea por la seguridad.

Para hacer esto agrega un "4" al octal de chmod para poner el SetUID, y un "2" para poner el SetGID.

Como ejemplo se podria hacer:

```
chmod 4755 /bin/netstat
```

Naturalmente tendrias que ser root para hacer esto, o usar el comando "sudo".

Y, para poner el SetGID seria:

```
chmod 2755 /bin/netstat
```

Despues que hice el comando "sudo chmod 4755 /bin/netstat" los permisos de archivo se vea asi:

```
-rwsr-xr-x    1 root     root         85240 Feb 11  2003 /bin/netstat
```

Nota el "s" en la columna de dueno.

Y, despues que el comando "sudo chmod 2755 /bin/nestat" se vea asi:

```
-rwxr-sr-x    1 root     root         85240 Feb 11  2003 /bin/netstat
```

4.) Usando /etc/hosts

Si mira a /etc/hosts (cat /etc/hosts) veas que el nombre "localhost" y "localhost.localdomain" esta conectado a la direccion 127.0.0.1. Esto es un convenio de como funciona servicios de redes y IP. En nuestra sala de taller tenemos un servidor con la direccion 192.188.58.126, pero no tiene un nombre. Entonces, para acedirlo (por ejemplo, para cargar un archvio de RPM de Red Hat) tiene que tipear todo la direccion. Podemos usar el archivo /etc/hosts para hacer nuestras vidas un poco mas simple durante la semana. Llamamos el servidor "noc" (centro de operaciones de redes), y, entonces en el archivo /etc/hosts agrega una linea al fin del archivo que se vea asi:

```
192.188.58.126  noc     noc
```

Nota que no es necesario tener un dominio en el nombre. Entonces ahora en vez de tipear:

lynx 192.188.58.126

para ver el servidor de web en el "noc" se puede tipear:

lynx noc

no mas...

5.) Configurando el Kernel

Si tienes instalado el fuente por el kernel (es un paquete opcional durante la instalacion de Red Hat, y, ya esta instalado), se puede cambiar como esta configurado tu kernel. Por ejemplo, si queria tener apoyo por el sistema de archivos "ntfs", podrias agregar el apoyo por esto en el archivo de configuracion por el kernel, recompilar el kernel, poner el imagen de kernel y la mapa del kernel en /boot, configurar grub (el boot loader), y reinicializar tu sistema, eligir el nuevo kernel que aparece en el menu de grub, y vea si funciona que agregaste al kernel... Facil, no?

Un punto *clave* a todo esto; es muy largo y dificil de elegir todo las opciones por un kernel - es casi un arte. Entonces, deberias usar el archivo de configuracion que ya existe por tu kernel, y construye tu configuracion usando esto, porque probablemente vas a hacer solo un cambio en vez de cientos...

El archivo de config por nuestro kernel es:

/usr/src/linux-2.4/configs/kernel-2.4.20-i386.config

Pero, para usarlo cuando se recompila el kernel tiene que copiar el archivo al directorio /usr/src/linux-2.4 y tiene que nombrarlo ".config" - si existe este archivo en el directorio /usr/src/linux-2.4, entonces, cuando corres las herramientas para configurar un kernel nuevo, se van a usar el archivo .config las elecciones de menu que haya.

Los pasos cortos para hacer esto son (mas o menos):

```
cd /usr/src/kernel-2.4
cp configs/kernel-2.4.20-i386.config .config
make menuconfig o make xconfig
make dep
make clean
make bzImage
make modules
cp /usr/src/linux-2.4/arch/i386/boot/bzImage
/boot/vmlinuz-2.4.20-8-ntfs
vi /boot/grub/grub.conf
```

Y, agrega un entrada por el nuevo kernel en grub para que sale en el menu de grub como un eleccion cuando reinicializes tu PC. Tiene que correr grub y agrega un archivo de "initrd" usando el comando en grub:

```
initrd vmlinuz-2.4.20-8-ntfs
```

Nota que cuando compilamos usamos un imagen de "bzImage" en vez de "zImage" por el kernel. Se carga un poco mas lento (muy minimo con un pentium y arriba), pero es necesario con el kernel de Red hat 9 para que esta bastante chico despues que compilacion.

En make menuconfig se encuentra la opcion por apoyo de "ntfs" en el menu de "File systems" y el submenu "NTFS file system support (read only)".

Como se puede ver el proceso es largo. Recomiendo leer un FAQ sobre compilar el kernel desde un lugar como http://www.kernel.org/.

6.) Modulos

Esto es corto, pero para que sepas, se puede manipular bien los modulos que estan cargado en memoria para apoyar el hardware que tiene tu maquina. Por ejemplo, para ver que esta cargado:

lsmod

Y, si queria eliminar un modulo de memoria se usa "rmmod". Si estas experimentando con modulos y un hardware nuevo se puede usar "insmod".

Primero mira que hay en tu archivo /etc/modulos.conf, y despues lea sobre este archivo (man modulos.conf).

Hervey Allen
Marzo 2004