Apuntes sobre Ubuntu.

davidfm

Algunas notas sobre cosas que me han llamado la atención de **Ubuntu** por si le sirve a alguien.

DISCOVER Y HOTPLUG

Ubuntu realiza en cada reinicio una detección del hardware instalado mediante hotplug y su posterior configuración con la herramienta discover de Progeny. Esta herramienta consulta una base de datos de dispositivos contra los buses del sistema buscando dispositivos conectados que anuncian sus características. La herramienta funciona bien y configura el hardware para que podamos usarlo aunque no hay soporte para todos los dispositivos ni los buses ya que sólo el hardware más nuevo permite detectar así los dispositivos.

Discover realiza detección de los buses pci,usb,ide,scsi,pcmcia,isa,parrallel y serial y podemos elegir que no se escanee un determinado bus con directivas disable

*Código:

```
disable pcmcia
```

A parte de los buses del sistema podemos elegir que dispositivos detectaremos en el inicio cambiando la línea donde intentará detectar todos

*Código:

```
boot all
```

por los tipos de dispositivos buscados

*Código:

```
boot bridge cdrom disk scsi ethernet ide sound usb video
```

Si no vamos a conectar nuevo hardware de un determinado tipo o bus podemos elegir desactivar alguno de los tipos de detección de discover y cargar los módulos necesarios directamente en el inicio añadiéndolos en /etc/modules. Podemos ejecutar lsmod para verlos

*Código:

```
lsmod
(salida simplificada)
8139too
snd_cmipci
snd_pcm_oss
snd_mixer_oss
snd_pcm
snd_page_alloc
snd_opl3_lib
snd_timer
snd_hwdep
snd_mpu401_uar
t snd_rawmidi
```

```
snd_seq_device
snd
```

y desactivar la detección de ese tipo de hardware o bus, en este caso ethernet y sound.

*Código:

```
boot bridge cdrom disk scsi ide usb video
```

SUDO

En **ubuntu** el usuario *root* no está pensando para convertirse en él en la consola. En su lugar las tareas administrativas está pensando para que que se hagan con el comando sudo. Este comando, permite a un usuario ejecutar programas con privilegios de otro usuario, por defecto root y es particularmente útil cuando varias personas deben administrar la máquina, ya que permite no tener que darle los máximos privilegios a todos y guardar logs de los comandos por si hacen alguna trastada. La sintaxis básica de sudo es ésta(los corchetes indican que el argumento es opcional)

*Código:

```
sudo [-u usuario] comando
```

tras introducir la contraseña de nuestro usuario, se ejecutará el comando y se guardará una estamación de fecha(un ticket) para nuestro usuario que permitirá que ejecute comandos durante un tiempo determinado sin tener que ingresar la contraseña. sudo permite también restringir que usuarios y que programas se pueden ejecutar con sudo configurando el archivo /etc/sudoers

Opciones de sudo

Podemos configurar algunas opciones de sudo en la sección Defaults para todos los usuarios o creando subsecciones Defaults:nombreusuario para definir opciones por defecto para ese usuario. Por ejemplo si queremos que sudo nos pida la contraseña cada vez lo ejecutemos podemos desactivar la opción timestamp_timeout(las opciones se desactivan con el carácter exclamación !) Sobre todo es importante desactivar en la sección Defaults las opciones root_sudo y requiretty

*Código:

```
Defaults !lecture,tty_tickets,!root_sudo,requiretty
```

con estas opciones desactivadas evitamos que cualquier usuario que tenga acceso a sudo puede conseguir una shell con privilegios de root ejecutando

*Código:

```
sudo sudo /bin/sh
```

y que alguien que haya hecho login en el sistema remótamente puedo usar el comando sudo(si teneis que logearos con ssh o similar y necesitais usar sudo no añadais la opción requiretty aunque desde luego no lo veo muy recomendable). Si desactivais root_sudo tened en cuenta que la consola root terminal de gnome no funcionará. De todas formas con esto no se puede evitar que alguien ejecute

*Código:

```
sudo chroot /
```

para obtener una shell como root, por lo que es necesario sólo permitir algunos comandos de uso frecuente.

También podemos especificar una opción logfile para guardar un registro de todos los comandos que se ejecutan con sudo

*Código:

```
Defaults logfile=/var/log/sudolog
```

Configuración en ubuntu

En la instalación de **ubuntu**, creamos un usuario sin privilegios que por defecto tiene permitido ejecutar cualquier comando con sudo, por lo que es el usuario con máximos privilegios en el que nos podemos convertir(es decir, que sabemos la contraseña). Si usamos este usuario para tareas que pueden ser peligrosas como leer correo, navegar,es decir las tareas normales de cada día si alguien consiguiese ejecutar comandos en nuestro sistema(localmente o remótamente), para conseguir ser root sólo tendría que usar el comando sudo para ejecutar lo que el quisiese. Esto choca totalmente con la política de usar el pc como usuario no privilegiado para evitar los problemas que tiene la gente que usa windows como administrador(o tiene un W98 Smile y para remediarlo podemos hacer la configuración por defecto de ubuntu bastante segura si simplemente creamos otro usuario con adduser o useradd que será el que usemos en nuestras tareas diarias. A este usuario, podemos permitirle realizar determinadas cosas como root(montar dispositivos con mount, ejecutar kppp,etc) y dejar el otro usuario para tareas administrativas que no haya más remedio que hacerlas con todos los privilegios. De este modo, podemos restringir los usuarios que tengan acceso a sudo en el archivo /etc/sudoers y restringir los comandos que podrán ejecutar como root. Las entradas de /etc/sudoers para configurar los comandos que usará un usuario tiene esta pinta

*Código:

```
UsuarioNuestroDeCadaDia Host= comandos directorios
```

Con una línea como esta

*Código:

```
omicron ALL= /bin/mount /mnt/
```

Especificaríamos que el usuario omicron puede ejecutar desde cualquier máquina(ALL) el comando /bin/mount y los comandos que se encuentren en el directorio /mnt como root. Si configurais el directorio /usr/sbin(o el directorio donde esté chroot), recordad que estais permitiendo que consigan una shell root.

Eso sí, no olvidéis tener un usuario con todos los permisos por si alguna vez tenéis que modificar algo que se sale de lo que tengáis configurado en /etc/sudoers

*Código:

```
hurd ALL=(ALL) ALL
```

Sudo puede dar más problemas de los que soluciona, pero por ejemplo permite que varios administradores realicen tareas en el sistema sin tener que darles la contraseña de root y que los usuarios ejecuten tareas comunes sin tener que cambiar de usuario.

PAQUETES EN UBUNTU

Ubuntu, como **debian**, ofrece paquetes precompilados con binarios y las fuentes por si queremos configurarlos e instalarlos nosotros mismos. Los dividen en ramas main(los que forman el sistema base), Warty(los paquetes para instalar por los usuarios), restricted(con software no soportado directamente y a veces comercial o con licencias no compatibles) y security(que honestamente, todavía no sé que es Smile Por lo general son un poco más conservadores que gentoo en cuanto a las versiones de los paquetes y bastante más en cuanto al número de éstos, aunque hacen algunas excepciones como el kernel 2.6.8.1 y el Gnome 2.6.8 Si algún paquete no lo encontrais al hacer(la opción -s es para simular la instalación)

*Código:

```
apt-get -s install nombrepaquete
```

Podeis buscar en los repositiorios universe de Ubuntu descomentado las opciones en /etc/apt/sources/list

*Código:

```
deb http://archive.ubuntu.com/ubuntu/ warty main restricted universe
deb-src http://archive.ubuntu.com/ubuntu/ warty main restricted  universe
```

y probar con

*Código:

```
apt-get source nombrepaquete
```

aunque hay que tener en cuenta que el repositorio universe no contiene paquetes especialmente preparados para ubuntu como los otros, si no que tendréis que configurarlo vosotros mismos desde las fuentes. Después de bajar el paquete del repositorio universe volved a comentarlo en /etc/apt/sources.list porque si no se os dará problemas con los catálogos de paquetes

Runlevels y Scripts de inicio

Los niveles de ejecución son estados por los que pasa el sistema operativo y tienen asociados scrips de inicio que realizan tareas asociadas a ese nivel. El 4 por ejemplo, corresponde a linux en modo texto y el 5 a linux en modo gráfico. Para pasar de un runlevel a otro podemos usar el comando telinit especificando el runlevel al que queremos hacer la transición, por ejemplo, para ejecutar linux en el modo gráfico ejecutaríamos

*Código:

```
telinit 5
```

Los scripts de inicio se encuentran en carpetas en /etc/rc.d/init.d y para seleccionar que scripts son necesarios podemos añadir o eliminar los enlaces simbólicos que hay en /etc/rc*.d(donde * es el número de nivel de ejecución de linux) y que es lo que indica cuales se deben cargar y cuando. Por ejemplo:

*Código:

```
/etc/rc4.d:
S10sysklogd S20alsa S20hal S25mdadm S99rmnologin
S11klogd S20apmd S20inetd S89atd S99stop-bootlogd
S14ppp S20cupsys S20makedev S89cron
S18portmap S20dbus-1 S20powernowd S99acpi-support
S20acpid S20fam S20rsync S99gdm
```

La nomenclatura indica el tipo del script(S para iniciar, K para parar) y el número el orden en el que se cargará. Si queremos eliminar por ejemplo los scripts que se encargan del sistema pcmcia(porque no lo usamos por ejemplo), ejecutamos

*Código:

```
rm -f /etc/rc*.d/*pcimcia*
```

Creado durante el 2004