ELEMENTOS DE ADMINISTRACIÓN

BITÁCORAS DEL SISTEMA

La seguridad y administración de un sistema operativo tiene en las bitácoras un gran aliado, ya que en ellas se registran los eventos que ocurren el sistema operativo, es decir eventos que el administrador pasaría inadvertidos sin el respaldo de las bitácoras.

Para una buena administración de bitácoras es necesario conocer 3 cosas:

1. Conocer el propósito de cada bitácora.
2. Conocer el formato en el que se presenta la información.
3. Conocer los ataques propios de cada servicio.

Las bitácoras en el caso de Linux están dentro del directorio /var/log/, para otros sistemas Unix se encuentran en /var/adm/, el propósito de las bitácoras encontradas mas comúnmente es se muestra a continuación:

SYSLOG

El demonio syslogd (Syslog Daemon) se lanza automáticamente al arrancar un sistema Unix, y es el encargado de guardar informes sobre el funcionamiento de la máquina. Recibe mensajes de las diferentes partes del sistema (Kernel, programas...) y los envía y/o almacena en diferentes localizaciones, tanto locales como remotas, siguiendo un criterio definido en el fichero de configuración /etc/syslog.conf, donde especificamos las reglas a seguir para gestionar el almacenamiento de mensajes del sistema.

Cada programa puede generar un mensaje de syslog, el cual consta de 4 partes:

Nombre del programa
Facility (servicio o facilidad)
Prioridad
Mensaje de bitácora.

Es posible configurar syslogd para que cada tipo de mensaje sea depositado en uno o varios archivos. Por regla general, el archivo /var/log/messages contiene los logs de la mayoría de los eventos.

Programa	Mensaje	Significado
halt	halted by <usuario>	El <usuario> uso el comando halt

login	ROOT LOGIN REFUSED ON <tty> [FROM<host>]	Intento ingresar como root, esto no es una acción segura.
login	REPEATED LOGIN FAILURES ON <tty> [FROM<host>] <usuario>	Alguien intento ingresar y fallo mas de 5 veces
reboot	Rebooted by <usuario>	El <usuario> uso el comando reboot
su	BAD su <usuario> on <tty>	El <usuario> uso el comando su y no tecleo el password correcto
shutdown	Reboot, halt o shutdown by <usuario>	El <usuario> uso el comando halt, shutdown o reboot

Mensajes típicos de información:

Programa	Mensaje	Significado
date	Data set by <usuario>	<usuario> cambio la fecha
login	ROOT LOGIN REFUSED ON <tty> [FROM<host>]	Intento ingresar como root, esto no es una acción segura.
su	BAD su <usuario> on <tty>	Su para cambiar a Super Usuario
getty	<tty>	Abrir <tty>

Bitácoras básicas

messages

Este archivo contiene bastante información, por lo que debemos buscar sucesos inusuales, aquí podemos ver todos los mensajes que el sistema mando a la consola, por ejemplo, cuando el usuario hace el login, los TCP_WRAPPERS mandan aquí la información, el cortafuegos de paquetes IP también la manda aquí, etc.

Para observar los primeros 10 mensajes contenidos en este archivo podemos ejecutar el siguiente comando: `more messages | head`, esto despliega una lista actualizada similar la siguiente:

```
Dec   4   09:07:19   localhost   syslogd   1.4-0:   restart.
Dec   4   09:07:19   localhost   syslogd   1.4-0:   restart.
Dec   4   09:07:20   localhost   syslogd   1.4-0:   restart.
```

```
Dec   4   09:07:20   localhost   syslogd   1.4-0:   restart.
Dec   4   09:07:20   localhost   syslogd   1.4-0:   restart.
Dec   4   09:07:20   localhost   syslogd   1.4-0:   restart.
dic 4 10:31:10 localhost gdm[798]: gdm_child_action: Master
rebooting...
dic 4 10:31:11 localhost rc: Stopping keytable: succeeded
Dec 4 10:31:11 localhost Font Server[767]: terminating
dic 4 10:31:12 localhost xfs: xfs shutdown succeeded
```

xferlog

Si el sistema comprometido tiene servicio FTP, este archivo contiene el loggeo de todos los procesos del FTP. Podemos examinar que tipo de herramientas han subido y que archivos han bajado de nuestro servidor.

Su formato normal es: fecha, hora, host remoto, tamaño de archivo, nombre de archivo, tipo de transferencia, banderas de acción especial, modo de acceso username, nombre de servicio, método de autenticación y uid.

wtmp

Cada vez que un usuario entra al servidor, sale del mismo, la máquina resetea, este archivo es modificado. Este archivo al igual que el anterior esta en binario por lo que tendremos que usar alguna herramienta especial para ver el contenido de este archivo. Este archivo contiene la información en formato usuario, hora de conexión, e IP origen del usuario, por lo que podemos averiguar de donde provino el agresor (decir que aunque contemos con esta información puede que haya sido falseada por el agresor utilizando alguna técnica para ocultar su IP original o haya borrado su entrada).

El formato es el siguiente:

```
bash-2.03# last -f wtmp|more

root   pts/0   :0   Fri   Dec   6   15:25   still   logged   in
root    :0    Fri    Dec    6    15:25    still    logged    in
reboot  system  boot  2.4.2-2  Fri  Dec  6  15:24  (00:01)
root   pts/0   :0   Thu   Dec   5   16:49   -   18:42   (01:52)
root   pts/1   :0   Thu   Dec   5   16:40   -   16:44   (00:04)
root   pts/0   :0   Thu   Dec   5   16:29   -   16:49   (00:20)
root   pts/0   :0   Thu   Dec   5   16:29   -   16:29   (00:00)
root    :0    Thu    Dec    5    16:19    -    down    (02:22)
reboot  system  boot  2.4.2-2  Thu  Dec  5  16:19  (02:22)
root    tty1    Thu    Dec    5    16:15    -    crash    (00:03)
reboot  system  boot  2.4.2-2  Thu  Dec  5  16:13  (02:28)
wtmp begins Wed Dec 4 10:31:10 2002
```

Como podemos ver se uso el comando last -f para poder ver el contenido.

secure

Algunos sistemas Unix loggean mensajes al archivo secure, ya que utilizan algún software de seguridad para ello, como el TCP Wrapper, ejemplo:

```
Dec 4 10:31:12 localhost sshd[642]: Received signal 15;
terminating.
Dec 5 16:13:42 localhost sshd[660]: Server listening on
0.0.0.0 port 22.
Dec 5 16:13:42 localhost sshd[660]: Generating 768 bit RSA
key.
Dec 5 16:13:43 localhost sshd[660]: RSA key generation
complete.
Dec 5 16:19:34 localhost sshd[679]: Server listening on
0.0.0.0 port 22.
Dec 5 16:19:34 localhost sshd[679]: Generating 768 bit RSA
key.
Dec 5 16:19:35 localhost sshd[679]: RSA key generation
complete.
```

Nota: *El formato es muy similar al de la bitácora messages.*

lastlog

Registra los últimos logeos al sistema. Es posible mirar el contenido de este archivo mediante el comando lastlog.

```
Username Port From Latest
root pts/1 :0 vie dic 6 15:25:58 -0500 2002
bin ** Never logged in**
daemon **Never logged in**
adm **Never logged in**
lp **Never logged in**
sync **Never logged in**
shutdown **Never logged in**
halt **Never logged in**
mail **Never logged in**
```

maillog

Guarda por lo general entradas de cada conexión pop/imap (nombre de usuario y logout), y el encabezamiento de cada uno de los correos que entran o salen del sistema (de quien, a quien, msgid, estado, etc.). Es posible ver su contenido mediante el comando more maillog. Por ejemplo:

```
bash-2.03# more maillog
```

Dec 5 16:13:46 localhost sendmail[696]: alias database /etc/aliases rebuilt by root
Dec 5 16:13:46 localhost sendmail[696]: /etc/aliases: 40 aliases, longest 10 bytes, 395 bytes total
Dec 5 16:13:46 localhost sendmail[706]: starting daemon (8.11.2): SMTP+queueing@01:00:00
Dec 5 16:19:38 localhost sendmail[715]: alias database /etc/aliases rebuilt by root
Dec 5 16:19:38 localhost sendmail[715]: /etc/aliases: 40 aliases, longest 10 bytes, 395 bytes total
Dec 5 16:19:39 localhost sendmail[725]: starting daemon (8.11.2): SMTP+queueing@01:00:00
Dec 6 15:24:56 localhost sendmail[696]: alias database /etc/aliases rebuilt by root
Dec 6 15:24:56 localhost sendmail[696]: /etc/aliases: 40 aliases, longest 10 bytes, 395 bytes total
Dec 6 15:24:56 localhost sendmail[706]: starting daemon (8.11.2): SMTP+queueing@01:00:00

access_log

Para el caso del demonio httpd, las bitácoras se guardan en access_log, un registro típico de esta bitácora contiene: Nombre de la computadora remota, login, nombre de usuario, fecha y hora, comandos que ejecutó (ej. get), número con el código de estatus y número de bytes que transfirió.

dmesg

Imprime los mensajes desplegados por el "kernel" al inicio.

```
DMESG(8)                    DMESG(8)



  NAME
  dmesg - print or control the kernel ring buffer

  SYNOPSIS
  dmesg [ -c ] [ -n level ] [ -s bufsize ]

  DESCRIPTION
  dmesg is used to examine or control the kernel ring
  buffer.
```

Al arranque, la salida de dmesg viene del inicio del núcleo, mostrando los dispositivos que ha encontrado y si ha sido capaz de configurarlos

(fuera de la configuración del usuario). Este registro es también disponible en el archivo /var/log/dmesg. Ejemplo: more dmesg

```
Linux version 2.4.2-2 (root@porky.devel.redhat.com) (gcc
version 2.96 20000731 (Red Hat Linux 7.1 2.96-79)) #1 Sun
Apr 8 20:41:30 EDT 2001BIOS-provided physical RAM map:
BIOS-e820: 0000000000009fc00 @ 0000000000000000 (usable)
BIOS-e820: 0000000000000400 @ 0000000000009fc00 (reserved)
BIOS-e820: 0000000000020000 @ 00000000000e0000 (reserved)
BIOS-e820: 000000000fdc0000 @ 0000000000100000 (usable)
BIOS-e820: 0000000000018000 @ 000000000fee0000 (ACPI data)
BIOS-e820: 0000000000008000 @ 000000000fef8000 (ACPI NVS)
BIOS-e820: 0000000000020000 @ 000000000fec0000 (ACPI data)
BIOS-e820: 0000000000080000 @ 00000000ffb80000 (reserved)
```

Nota: *Es importante tener una idea de aquellos mensajes y patrones de error o alerta que guardan las bitácoras, pues el revisar periódicamente estos archivos no tiene valor ni sustento si no se conoce que tipos de ataques se puede sufrir y como son registrados.*

LA GESTIÓN DE PROCESOS

Un proceso es cualquier instrucción o programa que en ese momento se está ejecutando en nuestro sistema. Todo proceso tiene un PID (Process Identifier), es decir un número que lo identifica y lo diferencia de todos los demás. Una característica importante, es que todo proceso tiene un estado: corriendo, durmiendo, zombie o parado.

Entre los procesos se diferencian los que se están ejecutando en 1º o 2º plano. Los que se ejecutan en 1º plano son los que interactúan con el usuario en ese momento, mientras que los procesos en segundo plano se ejecutan pero están ocultos, y muy posiblemente el usuario no tenga constancia de que se esté ejecutando.

Sólo puede haber un proceso en primer plano por consola. Eso es una limitante cuando no se está trabajando en un entorno gráfico. Para ejecutar entonces, varios comandos, lo que se puede hacer es ejecutar los comandos en segundo plano, para ello, sólo hay que añadir & al final del comando. Por ejemplo:

```
$>ls -R / > /dev/null &
```

En el anterior ejemplo se listan todos los ficheros de todos los directorios del sistema, se envía la salida a /dev/null para que su salida no moleste. El caracter & manda el proceso a un segundo plano.

job: muestra todos los procesos que se están ejecutando en un segundo plano.

fg: sirve para mandar un proceso a un primer plano.

Los procesos pueden ser suspendidos. Un proceso suspendido es aquel que no se esta ejecutando actualmente, sino que esta temporalmente parado sin consumo de memoria y CPU. Después de suspender una tarea, puede indicar a la misma que continúe, en primer plano o en segundo, según necesite. Retomar una tarea suspendida no cambia en nada el estado de la misma, la tarea continuará ejecutándose justo donde se dejó.

El comando bg se utiliza cuando se tienen procesos suspendidos y se quiere volver a ponerlos en marcha.

El comando kill

El comando kill permite interactuar con cualquier proceso mandando señales (signal). Cuando se ejecuta kill pid se manda la señal de terminar el proceso. Es posible usar cualquier otro tipo de señal, para ello se utilizará **kill signal pid**. Se puede conseguir una lista de señales usando `kill -l`. Una señal útil para algunas ocasiones es **-9**, esta señal fuerza a terminar cualquier proceso. Como su nombre lo indica se está matando el proceso.

También se puede utilizar el comando `killall` con el que podemos mandar señales a un proceso utilizando el nombre, en vez del PID.

El comando ps

Permite mostrar todos los procesos que están corriendo en nuestro sistema. Ejemplo:

$ps -aux

Los parámetros **aux** permiten ver todos los procesos que se están ejecutando. El parámetro **a** muestra lo que se está ejecutando en las tty conocidas, el parámetro **x** añade los procesos que no se conoce la tty en la que se está ejecutando y **u** muestra los usuarios que están ejecutando esos procesos.

La columna USER informa el usuario está ejecutando el proceso, "PID" es el número del proceso, "%CPU" es el porcentaje de CPU que está utilizando al igual que "%MEM" es el porcentaje de memoria. También incluye la cantidad de memoria en kilobytes que ha utilizado dicho proceso, en la columna "RSS". La columna "TTY" muestra la consola desde la que se está ejecutando. "Stat" muestra el estado del

proceso, S "durmiendo", R "corriendo", T "parado", Z "zombie". La columna "START" muestra la hora a la que empezó el proceso, la columna "TIME" muestra el tiempo de CPU que ha usado el proceso desde que inició y "COMMAND" muestra el nombre del comando que se está ejecutando.

El comando top

Es una utilidad que permite la monitorización de los procesos de la CPU. También muestra el estado de la memoria. Es una mezcla del comando ps, free y uptime.

SEGURIDAD

Descripción del archivo /etc/passwd

En un sistema Unix habitual cada usuario posee un nombre de entrada al sistema o login y una clave o password; ambos datos se almacenan generalmente en el archivo /etc/passwd. Este archivo contiene una línea por usuario (aunque hay entradas que no corresponden a usuarios reales, como veremos a continuación) donde se indica la información necesaria para que los usuarios puedan conectar al sistema y trabajar en él, separando los diferentes campos mediante ':'. La forma de una línea en este archivo es la siguiente:

```
usuario:contraseña:user-id:group-
id:comentario:directorio:shell
```

usuario	Nombre con el que puede identificarse un usuario
contraseña	En versiones anteriores, en este campo se almacenaba la contraseña de acceso del usuario, en forma encriptada. Actualmente contiene un carácter "x", y la contraseña y sus características están en el archivo /etc/shadow.
user-id	Número asignado al usuario para identificarlo internamente. Este número es utilizado por el sistema para efectuar los controles de propiedad de archivos y subdirectorios. Los números más bajos quedan reservados para los usuarios con privilegios de administrador.
group-id	Número que identifica el grupo al que se asocia ese usuario. Debe tener una correspondencia con el archivo /etc/group. Afectará a los permisos de acceso a archivos a nivel de grupo.

comentario	Normalmente contiene información sobre el propietario del login, como puede ser su nombre completo, su dirección, etc.
directorio	Ruta completa del directorio HOME, en el que sitúa al usuario cuando tanto el login como la contraseña son correctos.
shell	Nombre completo del proceso que se arrancará automáticamente para el usuario cuando se identifica. Aunque puede ser cualquier programa ejecutable, normalmente será un intérprete de comandos (shell).

Por ejemplo, podemos encontrar entradas parecidas a la siguiente:

toni:LEgPNjqSCHCg:1000:100:Antonio
Villalon,,,:/export/home/toni:/bin/sh

Elección de la clave

Es importante insistir en que la clave no se ha de escoger a la ligera. Normalmente todos los sistemas incluyen métodos para evitar que los usuarios elijan claves sencillas de averiguar. Si un intruso logra descargar el archivo de claves de nuestro sistema a través de una cuenta de invitado o por cualquier otro método, puede aplicarle un averiguador de contraseñas (password cracker) para descifrar las claves de los usuarios. Normalmente estos programas funcionan a base de un diccionario de palabras. Éstas se codifican usando cualquier tipo de cifrado y se comparan con la clave de está en el fichero de claves. Si coincide, se ha averiguado la clave y se le muestra al usuario. Para los sistemas Unix, los programas de este tipo más conocidos son Crack y Jhon the ripper. Podemos utilizar estos programas nosotros mismos como medida para prevenir el uso de claves fácilmente averiguables.

Lo mejor es elegir una clave que sea un mnemónico de alguna frase y que no se limite solamente a caracteres y números, conteniendo algún tipo de signo de puntuación y caracteres en minúsculas y mayúsculas.

Averiguar una clave, teniendo el archivo de claves, es solo cuestión de tiempo. Los ataques por hardware funcionan del mismo modo que los ataques por diccionario, pero probándose millones de combinaciones de caracteres.

En caso de intrusión en su sistema, los archivos más buscados son el /etc/password (el cuál ha de tener permiso de lectura para todos los usuarios si se quiere que el sistema funcione correctamente) y /etc/shadow.

Contraseñas aceptables

La principal forma de evitar este tipo de ataque es utilizar passwords que no sean palabras de los archivos diccionario típicos: combinaciones de minúsculas y mayúsculas, números mezclados con texto, símbolos como &, $ o %, etc. Por supuesto, hemos de huir de claves simples como internet o beatles, nombres propios, combinaciones débiles como Pepito1 o qwerty, nombres de lugares, actores, etc. Se han realizado numerosos estudios sobre cómo evitar este tipo de passwords en los usuarios y también se han diseñado potentes herramientas para lograrlo, como Npasswd o Passwd+. Es bastante recomendable instalar alguna de ellas para `obligar' a los usuarios a utilizar contraseñas aceptables (muchos sistemas Unixs ya las traen incorporadas), pero no conviene confiar toda la seguridad de el sistema a estos programas. Como norma, cualquier administrador deberá ejecutar con cierta periodicidad algún programa adivinador, tipo Crack, para comprobar que los usuarios no han elegidos contraseñas débiles (a pesar del uso de passwd o Passwd+): se puede tratar de claves generadas antes de instalar estas utilidades o incluso de claves asignadas por el propio root que no han pasado por el control de estos programas.

Shadow Password

Como se ha visto un atacante puede leer el archivo /etc/passwd, el cual debe tener permisos de lectura para todos los usuarios si se quiere que el sistema funcione correctamente.

Un método cada día más utilizado para proteger las contraseñas de los usuarios es el denominado Shadow Password u oscurecimiento de contraseñas. La idea básica de este mecanismo es impedir que los usuarios sin privilegios puedan leer el archivo donde se almacenan las claves cifradas; como se ha visto el archivo /etc/passwd tiene que tener permiso de lectura para todo el mundo si queremos que el sistema funcione correctamente. En equipos con oscurecimiento de contraseñas este archivo sigue siendo legible para todos los usuarios, pero a diferencia del mecanismo tradicional, las claves cifradas no se guardan en él, sino en el archivo /etc/shadow, que sólo el root puede leer.

En el campo correspondiente a la clave cifrada de /etc/passwd no aparece ésta, sino un símbolo que indica a determinados programas

(como /bin/login) que han de buscar las claves en /etc/shadow, generalmente una x:

toni:x:1000:100:Antonio Villalon,,,:/export/home/toni:/bin/sh

El aspecto de /etc/shadow es en cierta forma similar al de /etc/passwd, existe una línea por cada usuario del sistema, en la que se almacena su login y su clave cifrada. Sin embargo, el resto de campos de este archivo son diferentes; corresponden a información que permite implementar otro mecanismo para proteger las claves de los usuarios, el envejecimiento de contraseñas, del que se hablará continuación:

toni:LEgPN8jqSCHCg:10322:0:99999:7:::

Desde hace un par de años, la gran mayoría de Sistemas Unix incorporan este mecanismo; si al instalar el sistema operativo las claves aparecen almacenadas en /etc/passwd se puede comprobar si existe la orden pwconv, que convierte un sistema clásico a uno oscurecido. Si no es así, o si utilizamos un Unix antiguo que no posee el mecanismo de Shadow Password, es muy conveniente que se consiga el paquete que lo implementa (seguramente se tratará de un archivo shadow.tar.gz que podemos encontrar en multitud de servidores) y lo instalemos en el equipo. Permitir que todos los usuarios lean las claves cifradas ha representado durante años, y sigue representando, uno de los mayores problemas de seguridad de Unix; además, una de las actividades preferidas de piratas novatos es intercambiar archivos de claves de los sistemas a los que acceden y crackearlos, con lo que es suficiente que una persona lea nuestro archivo para tener en poco tiempo una colonia de intrusos en nuestro sistema. Shadow Password.

Envejecimiento de contraseñas

En casi todas las implementaciones de Shadow Password actuales se suele incluir la implementación para otro mecanismo de protección de las claves denominado envejecimiento de contraseñas (Aging Password). La idea básica de este mecanismo es proteger los passwords de los usuarios dándoles un determinado periodo de vida: una contraseña sólo va a ser válida durante un cierto tiempo, pasado el cual expirará y el usuario deberá cambiarla.

Realmente, el envejecimiento previene más que problemas con las claves problemas con la transmisión de éstas por la red: cuando conectamos mediante mecanismos como telnet, ftp o rlogin a un

sistema Unix, cualquier equipo entre el nuestro y el servidor puede leer los paquetes que se envían por la red, incluyendo aquellos que contienen el nombre de usuario y la contraseña; de esta forma, un atacante situado en un computador intermedio puede obtener muy fácilmente el login y password. Si la clave capturada es válida indefinidamente, esa persona tiene un acceso asegurado al servidor en el momento que quiera; sin embargo, si la clave tiene un periodo de vida, el atacante sólo podrá utilizarla antes de que el sistema obligue a cambiarla.

TCP Wrappers

Se debe deshabilitar cualquier servicio al exterior que no sea estrictamente necesario. Un usuario convencional con conexión a Internet a través de una cuenta con un proveedor no necesita siquiera Sendmail. Basta con que configure su cliente de correo proporcionando los servidores indicados por su proveedor.

Normalmente, todos los servicios que se inician en forma separada (es decir, no a través de inetd) disponen de mecanismos par restringir el acceso en función de la dirección de origen del cliente. Estas restricciones se indican en los archivos de configuración de los propios programas. Para aquellos programas que no incorporan estas capacidades, se puede utilizar unas utilidades llamadas envoltorios TCP, más conocidas como TCP Wrappers. Estos se interponen entre inetd y el servidor en cuestión, envolviendo a este y filtrando las peticiones en función del origen de las mismas. El tcp wrapper que se incluye en todas las distribuciones Linux es tcpd. Cuando este inetd recibe una petición para un servicio determinado, llama a tcpd. Ésta registra información si es necesario y realiza algunas comprobaciones. Si el servicio esta permitido para la dirección de origen, tcpd lanza el servicio en cuestión. Una de las ventajas de TCPD frente a otros sistemas de restricciones por dirección de origen es que comparte una misma configuración para todos ellos. Esta configuración se basa en los archivos /etc/hosts.allow y /etc/hosts.deny. En ellos se indican los servicios permitidos y denegados respectivamente.

Por ejemplo, para un usuario en casa que se conecte a Internet ocasionalmente a través de un proveedor, el archivo /etc/hosts.deny debería contener:

ALL: ALL

Esto significa que se deniegan todas las peticiones a cualquier servicio controlado por tcpd originadas desde cualquier dirección. Sin embargo, puede ser interesante tener servicios activados y que se

pueda acceder a ellos desde el propio equipo. Para ello editamos el archivo /etc/hosts.allow para que contenga:

ALL: 127

Indicando que se admite cualquier petición originada desde la máquina local (recordemos que se emplea el interfaz de loopback para distintas funciones en el sistema y que tiene la dirección 127.0.0.1 y máscara de red 127.0.0.0)

Conclusión: tcpd funciona únicamente con los servicios iniciados desde inetd y alguno que otro más. Para servicios que se inicien de manera independiente habrá que estudiar la configuración de los mismos para ver si soporta este tipo de reglas.

RESPALDOS

Los datos almacenados en el computador son lo más importante en éste, y pueden perderse en segundos debido a un comando mal introducido, un virus o un programa con un sistema de instalación defectuoso.

Primero se debe analizar en que lugares (directorios) se realizan los cambios al configurar y personalizar el Sistema Operativo Linux y qué se guarda en cada uno de esos lugares. Básicamente son dos los lugares más importantes:

El directorio de usuario, por ejemplo: /home/usuario y el directorio del supersusuario /root, en ellos están almacenadas el 90% de las personalizaciones y configuraciones particulares que se realizan a los programas que se usan.

El directorio /etc: en el cuál se sitúan el 100% de las configuraciones que se realizan al sistema, o propiamente dicho a su funcionamiento. Por ejemplo las unidades y sus puntos de montaje (archivo fstab), el nivel de inicio con o sin interfase gráfica (inittab), el mensaje del sistema a cada usuario que se logea (motd), los servicios que se quiere ejecutar (en /init.d, dentro del respectivo runlevel), comandos personalizados de inicio (rc.local), el nombre del sistema (hostname), módulos del kernel que se cargan para activar hardware como placas de sonido y de red (conf.modules), etc.

Para hacer un backup de la configuración de un sistema Linux es necesario guardar en un lugar seguro los directorios nombrados con todo su contenido (subdirectorios y carpetas, más detalles sobre esto luego).

Los directorios a copiar (o backupear) son entonces:

/etc
/home
/root

Es necesario copiarlos por completo, incluyendo los directorios y archivos ocultos (los que tienen el punto delante del nombre), para hacer la copia es recomendable usar el Midnight Commander (mc) con la opción de copia que aparece por defecto al presionar F5 (Copiar), pues de esta manera se conservan permisos y propietarios de los directorios y archivos evitando futuros problemas de acceso a ellos.

Restaurando la configuración

El backup se realiza con el obvio propósito de usarlo para tener una instalación nueva, por lo general funcionando con un nivel alto de personalizaciones y configuraciones particulares, minutos luego de finalizada la instalación o actualización, o en un tiempo mucho menor de lo que tomaría resetear desde cero toda la configuración que necesitamos o que usamos habitualmente.

Existen dos clases de respaldos los completos y los incrementales. Un comando relativamente simple para la creación de respaldos es el: tar.

Desde el punto de vista del administrador, el sistema de archivo debe respaldarse de acuerdo con algún proceso automatizado, de preferencia cuando el sistema no se encuentre en uso, y con la menor intervención posible de los operadores. Además debe tener un plan de respaldo que satisfaga sus necesidades y que haga posible la restauración de copias recientes de archivos, utilizando una combinación de respaldos completos e incrementales.

Un respaldo completo es el que contiene todos los archivos del sistema. Y el respaldo incrementado es el que contiene archivos que han cambiado desde el último respaldo. Estos pueden realizarse a diferentes niveles:

Nivel 0 Respaldo completo.

Nivel 1 Incrementado con respecto al último respaldo completo.

Nivel 2 Incrementado con respecto al último respaldo del nivel 1.

TAR, GZIP y unos disquetes.

tar es perfectamente utilizable para la realización de copias de seguridad y una opción potente de tar es la posibilidad de trabajar con múltiple volúmenes. Lo que quiere decir que si optamos por guardar los datos en simples disquetes, en el momento que el primer disco está lleno, tar para y pide el siguiente "volumen" (disquete).

El comando:

```
$ tar cvfzM /dev/fd0 /etc
```

Hará una copia de seguridad de /etc utilizando el disquete /dev/fd0. La opción M de tar es la que permite que la copia de seguridad sea copia multi-volumen, la c indica la creación de un archivo, y la f especifica el archivo destino, en este caso es la unidad de disco y v verbose.

Con el comando **`tee /root/indice`** se crear un listado de los archivos copiados y se dirigen a /root/indice.

Para recuperar el contenido del disquete usaremos el comando:

```
$ tar xvfzM /dev/fd0
```

Este método puede ser utilizado también si se tiene una unidad de cinta (/dev/rmt0) conectada al sistema, o una unidad zip que es tratada como un disco duro mas.

REFERENCIAS BIBLIOGRÁFICAS

Seguridad:

http://www.linuxlots.com/~barreiro/seguridad.html
http://www.desarrolloweb.com/articulos/804.php
http://temu.tco.plaza.cl/mg_tec/documentos/linux/seguridad_basica_ii.htm
http://www.geocities.com/petsburgh/haven/2095/first.html

Respaldos:

http://www.planetalinux.com.ar/article.php?aid=42
http://www.htmlweb.net/linux/redes/redes_linux_7.html
http://estigia.fi-b.unam.mx/linux/prared11.html

Mensajes:

http://software.uaemex.mx/documentos/investigacion/logs.htm
http://lucas.hispalinux.es/Manuales-LuCAS/GSAL/gsal-19991128-htm/ficheroslog.htm

http://www.osmosislatina.com/unix/comandos.htm
http://openbsd.appli.se/faq/es/faq2.html