ADMINISTRACIÓN DE SISTEMAS Y SERVICIOS TELEMÁTICOS

APLICACIÓN DE DUMP

- Backup de Sistemas -

RUBÉN RODRIGUEZ ÁLVAREZ
RAFAEL REDONDO TEJEDOR

INDICE

1. INTRODUCCIÓN:

Las copias de seguridad del sistema son el principal mecanismo de recuperación, por no decir el único, que poseen los administradores para restaurar una máquina que por cualquier motivo (hackers, fallos de memoria, rotura de los sistemas de lectura/escritura en los dispositivos de almacenamiento,...) ha perdido datos. A los sistemas y dispositivos de backups no se les ha prestado la atención que merecen. Esto empieza a cambiar aunque suele ser después de que se hayan producido las primeras pérdidas. Hay que tener en cuenta que un simple fallo de alimentación puede provocar un desastre.

A los problemas de realizar backups se les suman otros adicionales, como la verificación del correcto funcionamiento, el mantenimiento, política de etiquetas y nombres, tamaño de los dispositivos... una simple verificación y muy eficaz sería restaurar la información y comprobar si se realizó correctamente. Algunos de estos casos se han tenido en cuenta en el desarrollo de la aplicación.

Debido a la filosofía de las copias de seguridad generalmente se realizan en dispositivos externos como otras máquinas o dispositivos de almacenamiento (cassettes, discos, memorias,...), aunque también se puede hacer de modo local en la propia máquina.

Por último, ¿qué almacenar? Obviamente debemos realizar copias de seguridad de los archivos que sean únicos a nuestro sistema; esto suele incluir directorios como /etc/, /usr/local/ o la ubicación de los directorios de usuario (dependiendo del Unix utilizado, /export/home/, /users/, /home/...). Por supuesto, realizar una copia de seguridad de directorios como /dev/ o /proc/ no tiene ninguna utilidad, de la misma forma que no la tiene realizar *backups* de directorios del sistema como /bin/ o /lib/: su contenido está almacenado en la distribución original del sistema operativo (por ejemplo, los CD-ROMs que utilizamos para instalarlo).

1.1 DISPOSITIVOS DE ALMACENAMIENTO:

Existen multitud de dispositivos diferentes donde almacenar nuestras copias de seguridad, desde un simple disco flexible hasta unidades de cinta de última generación. Evidentemente, cada uno tiene sus ventajas y sus inconvenientes, pero una cualidad que debe cumplir cualquiera de ellos es que sea estándar, casi por encima de su capacidad y velocidad de almacenamiento.

· Discos flexibles:

Es un medio muy barato y portable entre diferentes sistemas operativos, por contra su fiabilidad es muy baja. La capacidad de almacenamiento de los *floppies* es muy baja, de poco más de 1 MB por unidad; esto hace que sea casi imposible utilizarlos como medio de *backup* de grandes cantidades de datos, restringiendo su uso a ficheros individuales.

· Cintas magnéticas :

Las cintas magnéticas han sido durante años (y siguen siendo en la actualidad) el dispositivo de *backup* por excelencia. Las más antiguas, las cintas de nueve pistas poseen una capacidad entorno a 300 MB pero por el contrario tienen una velocidad muy baja, lo que las lleva al desuso en aplicaciones exigentes. Después de las cintas de 9 pistas aparecieron las cintas de 1/4 de pulgada, mucho más pequeñas en tamaño que las anteriores y con una capacidad máxima de varios *Gigabytes*; se trata de cintas más baratas que las de 9 pistas, pero también más lentas. El estándar más conocido es *QIC (Quarter Inch Cartridge*).

A finales de los ochenta aparece un nuevo modelo de cinta que relegó a las anteriores a un segundo plano y que se ha convertido en el medio más utilizado en la actualidad: se trata de las cintas de 8 mm. Están diseñadas en su origen para almacenar vídeo. Estas cintas son del tamaño de una *cassette* de audio y tienen una capacidad de hasta cinco *Gigabytes*, lo que las hace perfectas para la mayoría de sistemas: como toda la información a salvaguardar cabe en un mismo dispositivo, el operador puede introducir la cinta en la unidad del sistema, ejecutar un sencillo *shellscript*, y dejar que el *backup* se realice durante toda la noche.

Las cintas DAT, de 4 mm mejoraron la calidad de lectura y escritura y diseñadas ya en origen para almacenar datos. Son algo más pequeños que las cintas de 8 mm. pero con una capacidad similar, son el mejor sustituto de las cintas antiguas: son mucho más resistentes que éstas, y además relativamente baratas.

Sobre las citas cabe mencionar que existen innumerables formatos que salen cada año, y cada fabricante importante suele tener el suyo.

· CD-ROMs:

En la actualidad sólo se utilizan cintas magnéticas en equipos antiguos o a la hora de almacenar grandes cantidades de datos - del orden de *Gigabytes*. Utiliza harware muy barato y dispositivos de muy bajo coste y con una capacidad de almacenamiento suficiente para muchos sistemas.

DISPOSITIVO	FIABILIDAD	CAPACIDAD	COSTE / MB
Diskette	Baja	Baja	Alto
CD-ROM	Media	Media	Bajo
Disco Duro	Alta	Media/Alta	Medio
Cinta 8mm	Media	Alta	Medio
Cinta DAT	Alta	Alta	Medio

1.2 I NSTRUCCIONES: *dump*

Aunque muchos clones de Unix ofrecen sus propias herramientas para realizar copias de seguridad de todo tipo (por ejemplo, tenemos `mksysb` y `savevg/restvg` en AIX, `fbackup` y `frecover` en HP-UX, `bru` en IRIX, `fsphoto` en SCO Unix, `ufsdump/ufsrestore` en Solaris...), casi todas estas herramientas suelen presentar un grave problema a la hora de recuperar archivos: se trata de *software* propietario, por lo que si queremos restaurar total o parcialmente archivos almacenados con este tipo de programas, necesitamos el propio programa para hacerlo. Por este motivo, muchos administradores utilizan herramientas estándar para realizar las copias de seguridad de sus máquinas; estas herramientas suelen ser tan simples como un *shellscript* que se planifica para que automáticamente haga *backups* utilizando órdenes como `tar` o `cpio`, programas habituales en cualquier clon de Unix y que no presentan problemas de interoperabilidad entre diferentes operativos.

La herramienta clásica para realizar *backups* en entornos Unix es desde hace años `dump`, que vuelca sistemas de ficheros completos (particiones). `Restore` se utiliza para recuperar archivos de esas copias. Se trata de una utilidad disponible en la mayoría de los sistemas Unix, aunque pueden tener distintos nombres. Es muy potente y completamente compatibles entre sistemas Unix. normalmente se suelen automatizar las tareas de *backup* mediante *scripts* sencillos que se ejecutan de forma automática, que es el caso de la aplicación. Esto se verá más adelante. Básicamente existen dos tipos de *backu*p: completo cuando queremos almacenar todos los ficheros e incremental cuando queremos almacenar sólo aquellos ficheros que hayan cambiado desde la última vez. La sintaxis de *dump* es:

```
dump [0123456789ackMSu] [B records] [b blocksize] [d
density]
[e inode number] [f file] [F script] [h level] [L label]
[s feet] [T date] filesystem/directory
```

- 0123456789: indica el nivel de *backu*p: 0 completo; mayor que 0 incremental, salvándose sólo aquellos ficheros modificados respecto a *dump* anterior de nivel inferior
- a: fuerza a escribir hasta el final del soporte
- c: usar cinta “cartridge”
- k: usar autenticación *kerberos* para uso con servidor remoto
- M: para múltiples volúmenes (usando prefijo dado por f)
- S: estima, sin ejecutarlo, cuánto espacio es necesario para hacer el *backup*
- u: actualiza el fichero de monitorización */etc/dumpdates*
- B: indica el número de bloques *(record*s) del dispositivo
- b: indica el tamaño del bloque *(blocksize*) del dispositivo
- d: determina la densidad (density) del dispositivo
- e: excluye el inodo *inode* del *backup*
- f: hace el *backup* en fichero (tiene que ser un fichero asociado a un dispositivo)
- F: indica el *script* a usar para confirmar cambios de cinta
- h: configura el nivel a partir del que puede hacerse *backup*
- L: etiqueta a la que luego puede acceder *restore*
- s: estima la cantidad de cinta necesaria para una densidad particular *(fee*t), solicitando otra si se queda corta
- T: establecen momento a realizar *backup* en lugar de los dados en */etc/dumpdates*

El programa *dump* además almacena el historial de las copias realizadas sobre el fichero */var/lib/dumpdates.* Detalles de gestión y control de este y otros archivos se especificarán más adelante en el apartado de configuración, así como de los detalles de los parámetros más importantes. Algunos ejemplos son:

(user):~> ***dump 0Bbuf 120000 1000 /dev/rft0 /home***
(user):~> ***dump –0u -f ~user/file /home***

El primer caso hace una copia con los tamaños de bloque y tantos bloques como los especificados de la partición */home* al dispositivo */dev/rft0,* típico de las copias de seguridad en dispositivos externos previamente montados. El segundo caso almacena la copia en usuario *user* con el nombre es *file.* Las dos son de nivel 0.

Además la mayor parte de las versiones de `dump` permiten realizar copias de seguridad sobre máquinas remotas directamente desde línea de órdenes (`rdump`/`rrestore`) sin más que indicar el nombre de máquina precediendo al dispositivo donde se ha de realizar la copia. Aunque en esta aplicación se hará localmente y se harán copias con protocolos seguros. La diferencia es que habrá una copia en cada máquina en lugar de una remota. Para ello se utiliza el protocolo de conexión segura *ssh.* Otros protocolos del mismo tipo son SSL o TCFS.

SSH (*Secure Shell*) tiene como principal función permitir la conexión remota segura a sistemas a través de canales inseguros, aunque también se utiliza para la ejecución de órdenes en ese sistema remoto o transferir ficheros desde o hacia él de manera fiable. SSH está formado por un programa servidor, `sshd`, varios programas cliente (`ssh` y `scp` principalmente) y pequeñas aplicaciones para su configuración, como `ssh-add`, `ssh-keygen` o `ssh-agent`. El programa demonio (`sshd`) se ejecuta en la máquina contra la cual conectamos, mientras que los clientes se han de ejecutar evidentemente en el sistema desde el cual conectamos; así, podemos iniciar una sesión en la máquina remota con la siguiente sintaxis:

(user):~> ***ssh –l <nombre_usuario> <máquina>***
(user):~> ***ssh –l <nombre_dominio>***

El flag *–l* indica el nombre del usuario. También existe la posibilidad de ejecutar instrucciones, una vez conectados, y desconectarse al final. Para ello basta con insertar las instrucciones a continuación de las anteriores líneas de comandos. Pero para ello hay que introducir previamente el *password* del usuario. Para evitar esto se pueden generar un par de claves pública-privada con el comando ssh-keygen, con lo que se podrán borrar archivos remotamente. Esto se explicará más adelante. El otro programa cliente, *scp,* utilizado para transferir ficheros entre máquinas. Una instrucción simple es:

(user):~> ***scp-r <nom_dominio>:/ <directorio_origen> <directorio_destino>***

Con el flag –r se transfieren recursivamente todos los directorios dentro de la directorio cuenta del usuario especificado. Evidentemente también se pueden especificar archivos.

Si no se indica lo contrario con la opción `'-p'`, el cliente conecta al puerto 22 de la máquina servidora y verifica que esta máquina es realmente con la que se quiere conectar, intercambia las claves de cifrado entre sistemas (cifradas a su vez, para evitar que un atacante pueda obtener la información) y autentica utilizando `.rhosts` y `/etc/hosts.equiv`, RSA o claves de usuario; si todo es correcto, el servidor asigna una terminal virtual (generalmente) a la conexión y lanza un *shell* interactivo. Se puede ver con detalle este proceso utilizando la opción `'-v'` del cliente.

2. DESCRIPCIÓN DE LA APLICACIÓN:

Como se puede comprender, por el anterior apartado, la realización y gestión de sistemas de copias de seguridad no es tema sencillo porque involucra numerosos aspectos críticos de los sistemas operativos, como los son la gestión de nombres, permisos de escritura y lectura, gestión de claves de seguridad, protocolos de comunicación remota,... Este trabajo sólo refleja algunos aspectos de lo que suponen los principales métodos para realizar copias de seguridad y por su puesto existen numerosas variantes.

El objetivo propuesto es desarrollar una aplicación que sea capaz de realizar copias de seguridad de manera automática, en este caso se realizan de manera diaria aunque puede tener cualquier otra frecuencia, incluso puede ser variable dependiendo de la época de año. Se pretende que el *script* que activa el proceso de copia gestione además las etiquetas de las mismas, es decir, los nombres.

La herramienta para hacer las copias de seguridad, como ya se introdujo en el anterior apartado, es *dump* que permite varios niveles y control del historial de las copias realizadas. Así cada día de la semana se corresponde con un nivel, y cuando termine una semana completa se hará la copia de nivel 0. La aplicación se debe encargar de gestionar las copias semanales, borrando las más viejas si no cabe la nueva y borrando las diarias si se hace una semanal puesto que no tiene mucho sentido seguir conservándolas.

Las copias remotas se almacenan de manera local en la misma máquina. Esto puede tener un inconveniente y es que si se estropea la maquina se pierde toda información, la original y la copia. De esto se encarga la aplicación remota que almacena las copias en otra máquina. Por el contrario, este proceso, aunque más fiable, tiene el inconveniente de que es más complejo y puede tener problemas de seguridad puesto que las copias se van enviar fuera de la máquina local y alguien puede estar escuchando. Por ello las copias y borrados remotos se harán mediante *ssh* (*scp*) de manera segura utilizando el par de claves pública - privada.

Las aplicaciones además se deben encargar de hacer las notificaciones pertinentes si el sistema de copias falla, por ejemplo enviando un correo electrónico.

3. REQUISITOS

4. ARQUITECTURA DEL SERVICIO

El concepto de copia de seguridad es aplicable a cualquier distribución de Linux, y por lo tanto a Debian, que es la que se ha utilizado en el curso de Administración de Sistemas. Hoy día, las técnicas de backup son ampliamente utilizadas en sistemas configurados en red, donde existen tanto máquinas individuales como servidores de información. Como elemento conceptual que añaden las redes que utilizan técnicas de backup está el dispositivo de almacenamiento de datos. Dichos centros estarán conectados a servidores de backup que realizarán la tarea de copia remota de la información que se desee almacenar de las máquinas de la red. La posibilidad de realizar copias remotas y centralizadas de datos desde los servidores de backup facilita en gran medida el proceso de administración en estas grandes redes:

En primer lugar, porque el concepto de copia de seguridad remota dota al sistema de una gran robustez frente a la posible caída de máquinas individuales. Si por cualquier circunstancia fuese imposible acceder al disco de una máquina, la información que dicho terminal almacenase se habría perdido y sería imposible recuperarla. Resaltar en este punto la importancia del concepto de copia *centralizada* por el hecho de que, si la copia se depositase en el mismo disco del propio terminal, el problema de acceso sería el mismo y la información se habría perdido de igual manera.

En segundo lugar, porque la información se deposita en un único soporte físico, hecho que facilita notablemente el control y la gestión de estos dispositivos de almacenamiento. Se abre entonces la posibilidad de realizar copias globales y estructuradas como parte de la gestión de la red.

Como contraposición, el concepto de copia centralizada (un servidor de backup y un único dispositivo de almacenamiento) acarrea una mayor complejidad en la configuración del servidor de backup, así como un aumento en el tiempo dedicado a la realización de las copias. En el primer caso, porque se necesita un acceso remoto para cada una de las máquinas. En el segundo, porque una copia remota en un dispositivo externo es más lenta que una copia local en otro fichero del propio disco.

Para finalizar con el apartado de la arquitectura, vamos a explicar de qué forma intervienen los conceptos de copia remota y copia centralizada en la práctica realizada. En nuestro caso, la red ha estado constituida exclusivamente por dos elementos. Uno de ellos dispone de datos de los que se quiere hacer backup y el otro será el dispositivo de almacenamiento. Las características fundamentales son

I - No existe conceptualmente un servidor de backup que origina de forma centralizada
las copias y las lleva al dispositivo de almacenamiento.

II – El proceso se inicia en la máquina cuyos datos se desea guardar. Se ejecutará un

script (ver Configuración) que realiza una backup de la información en su propio disco. Posteriormente, la propia máquina hace una copia remota de dichos datos en el dispositivo de almacenamiento.

5. CONFIGURACIÓN

Por lo que respecta a la configuración del servicio, vamos a analizar una serie elementos que se han de tener en cuenta:

1- Fichero **/etc/fstab**:

Cuando Linux arranca, monta automáticamente los sistemas de ficheros especificados en el fichero /etc/fstab. Entre las informaciones que maneja este fichero están los dispositivos que contienen a los diferentes sistemas de ficheros, el directorio del sistema del que colgará cada sistema de ficheros concreto, los tipos de ficheros manejados o las opciones del montaje. Finalmente, aparece lo que se denomina *dump flag*. Este flag especifica si el comando dump creará una copia del sistema de ficheros concreto. Por lo tanto, deberá activarse este flag en aquel sistema de ficheros del que se desee hacer copia de seguridad.

2- Variable **partition**

Como puede observarse en el código del script **rcpsecurity**, aparece al principio una variable denominada *partition*. Dicha variable se utilizará para especificar de qué se desea hacer copia de seguridad. En nuestras pruebas hemos utilizado el directorio **home**. La única precaución que debe tenerse es mantener la concordancia entre el valor de *partition* y el flag en el archivo */etc/fstab*. Incidimos de nuevo en el hecho de que es necesaria la activación manual previa del flag del directorio *home*.

3- Directorio **/etc/cron.daily**

Este fichero contiene scripts que se ejecutan de forma automática y son fundamentales en el contexto de las copias de seguridad. En el directorio *cron.daily* se almacenan los scripts que se ejecutan automáticamente de forma diaria. De igual manera, en *cron.weekly* se ubicarán los ficheros que se deban ejecutar semanalmente. El objeto de manejar el directorio de ejecutables diarios es precisamente el situar allí nuestro script **rcpsecurity**. Mediante la ejecución del programa **date** del sistema se averiguará el día de la semana y se realizará una copia con unas características concretas. De esta forma, conceptualmente se realizan copias diarias y semanales con un único script que se ejecuta de forma automática diariamente.

6. FUNCIONAMIENTO

En primer lugar se presenta el código del script /etc/cron.daily/rcpsecurity:

```
#!/bin/sh

# Names
partition=home
daily=d.
weekly=w.
named=$partition$daily
namew=$partition$weekly
user=backupuser
domain=m1

# Variables
daycopy=7
day=`date +%w`   # day of the week (Sunday: 7)
tam_buffer=10    # maximum buffer size for weeks

direc=/var/cpsecurity    # directory for local backup

# Routine that makes the directory if it doesn't exist
if [ ! -d $direc ]; then
  mkdir -p $direc
  chmod 700 $direc
fi;

direc=$direc/$partition    # backup directory

if [ ! -d $direc ]; then
    mkdir -p $direc
    chmod 700 $direc
    ssh $user@$domain mkdir ~$user/$partition
    day=$daycopy
    num=0
    cont=0
else
    num=`ls -lrt $direc/$namew* | tail -1 | cut -f2 -d'.'`
    cont=`echo $num + 1 | bc`
    if [ "$cont" = "$tam_buffer" ]; then
       cont=0
    fi;
fi;
if [ "$day" = "$daycopy" ]; then
     rm -f $direc/$named*
     ssh $user@$domain rm -f ~$user/$partition/$named*
     name=$namew$cont
     day=0
else
     name=$named$day
fi;

while [ `dump $day -u -f $direc/$name /$partition` ]
do
    while [ ! -r $direc/$namew$cont ]
    do
      cont=`echo $cont + 1 | bc`
```

```
      if [ "$cont" = "$tam_buffer" ]; then
         cont=0
      fi;
    done;

    if [ "$cont" = "$num" ]; then
       echo enviar correo
       exit 0
    else
       rm $direc/$namew$cont
       ssh $user@$domain rm ~$user/$partition/$namew$cont
       cont=`echo $cont +1 | bc`
    fi;

done;
scp $direc/$name $user@$domain:~$user/$partition/$name
exit 1
```

En la primera linea se define la shell que ejecutará el script. A continuación, se define una serie de variables que representarán cadenas de caracteres que serán posteriormente utilizadas para referenciar directorios y ficheros. Se definen también la variable *día* (día de la semana), *daycopy* (el día que se hará la copia semanal) y *tam_buffer* (máximo número de copias semanales que se almacenarán).

Por lo que respecta a la política de nombres, pueden hacerse los siguientes comentarios:

- El fichero de almacenamiento tendrá la copia del directorio indicado por la variable *partition*. Al final del nombre, se añadirán los sufijos *d* o *w* para indicar que los backups son diarios o semanales respectivamente. Finalmente, se añadirá un punto seguido de un número entre 1 y 6 para los diarios y 0 y 9 para los semanales. Este valor indica el día de la semana en los diarios y la antigüedad en los semanales.

- Dentro del directorio destino escogido para ubicar las copias, éstas estarán a su vez dentro de un directorio con igual nombre que el directorio original *partition*.
  Si, por ejemplo, hacemos la primera copia semanal del home y queremos que quede en el directorio “dir”, quedará grabada como */dir/home/homew.0*

La rutina que sigue tiene por objeto crear el directorio especificado en *direc* en caso de no existir todavía. Esto sucederá la primera vez que se ejecute el script para la primera copia. En nuestro caso, la copia local se deposita en */var/cpsecurity*.

La siguiente rutina es muy parecida a la anterior, pero ahora se pregunta por la existencia del directorio /direc/partition. El objetivo es crear el directorio *partition* dentro del directorio destino para guardar las copias. Éste se creará la primera vez que se haga una copia de un directorio *partition* concreto. Además, este mismo proceso se lleva a cabo en la máquina remota de almacenaje. Para ello, debemos realizar un acceso remoto a dicha máquina con la instrucción **ssh**

(ver Descripción). Finalmente, además de inicializar variables, hacemos **day=$daycopy**. Con ello, hacemos que la primera vez que se acceda se haga una copia semanal de nivel 0.

En caso de que ambos directorios hayan sido creados con anterioridad, se almacena el número de la copia semanal que corresponderá hacer la próxima vez (si excedemos el número máximo de copias semanales, le damos nuevamente el valor ".0", sobreescribiendo la más antigua).

Lo siguiente que se hace es borrar todas las copias diarias (de Lunes a Sábado), en caso de que corresponda hacer una semanal (el Domingo no tendrá copia diaria porque la semanal es de nivel 0). El borrado se hará tanto localmente como en el dispositivo remoto con **ssh**. Finalmente, se realiza la copia con la instrucción **dump**. El nivel de la copia (ver Descripción) vendrá definido por la variable *day*, que indica el día de la semana.

Con objeto de optimizar la memoria disponible se utiliza un algoritmo que, en caso de no poder ejecutar el dump por no haber espacio suficiente, va borrando las copias semanales de nivel 0 hasta que tenga espacio suficiente o se encuentre con la última. En el caso de que no haya espacio para realizar el dump teniendo en memoria exclusivamente la copia de la semana pasada (la copia más reciente), el sistema responderá enviando un **mail** de aviso de *falta de memoria*. En nuetra aplicación esto último no está implementado y lo único que hace el programa es un eco con el mensaje **enviar correo** y salir de la aplicación.

Finalmente, se realiza una copia del fichero qure se ha guardado localmente en el directorio remoto de la otra máquina que hace de dispositivo de almacenamiento

7. PRUEBAS

Para comprobar el correcto funcionamiento del script **rcpsecurity** el proceso es muy sencillo. Se trata de ignorar la parte de análisis del día actual y fijar uno de forma manual asignando un valor entre 1 y 7 a la variable *day* (1 para el Lunes y 7 para el Domingo). A continuación, se ejecuta la instrucción (deberá indicarse el path si no estamos en /etc/cron.daily):

(prompt) **./rcpsecurity**

De esta forma, podemos comprobar en cualquier momento que el programa hace lo que se pretende para un día concreto. Para asegurarnos de que las copias se realizan de forma correcta y se depositan en el directorio adecuado, no tenemos más que comprobar que en el terminal que hace de dispositivo de almacenamiento se han almacenado los datos.

8. GESTIÓN DIARIA

En toda máquina y, sobre todo, en configuraciones de red, el proceso de gestión diaria adquiere una importancia fundamental dentro del proceso de administración. Cuanto mayor envergadura tenga la red, mayor y más complejo será el trabajo de gestión. En nuestro caso, debe tenerse en cuenta lo siguiente:

1 - La realización de la copia no requiere intervención del usuario y se realiza de forma automática al estar ubicado el script en /etc/cron.daily.

2 - En caso de querer hacer backups de datos distintos, deberá ir modificándose de forma manual el valor de la variable partition (ver Configuración).

3 - El momento exacto en que se realiza la copia es importante a efectos de carga en la red. Para nuestro ejemplo sencillo, esto no ha supuesto un problema porque sólo hay una máquina que hace copias de seguridad. En redes reales formadas por muchas máquinas, el instante en que se realiza el backup deberá ser tal que afecte lo menos posible las condiciones de carga de la red. Una buena política sería realizar estas copias por la noche, que es cuando normalmente la carga de la red es menor. Para ello, debe el usuario en nuestro caso configurar la hora de ejecución del script **rcpsecurity** dentro de /ect/cron.daily.

4 - La elección del tipo y número de dispositivos físicos de almacenamiento que se emplearán. En nuestro caso concreto no se utilizará un dispositivo específico, sino que se usará la memoria de una de las máquinas. Por ello, la gestión será diferente al caso de tener discos o cintas.

Finalmente, hemos de definir qué datos vamos a almacenar como medida de seguridad. Naturalmente, hay datos que por su naturaleza son más importantes que otros. Por otra parte, hay datos que varían más rápidamente mientras que permanecen estáticos dentro del sistema. De todo esto se deduce que, además de delimitar qué datos requieren copias de seguridad, también hemos de saber cuál es nivel de backup más adecuado para unos datos concretos. Como ya hemos explicado, el nivel de backup se introduce como parámetro en la instrucción **dump** (ver Descripción).

9. PROBLEMAS ENCONTRADOS

Nuestra política de almacenamiento remota consiste en guardar las copias en el directorio o cuenta de un usuario exclusivamente creado para ello donde solo puede acceder el administrador. En un principio se trató de que la aplicación de copias remotas accediera a otra máquina e hiciera una copia de sus particiones, en este caso */home*. Funcionaría como un sistema remoto que obliga a otras máquinas a hacer sus backups. Esto no se pudo lograr ya que por medio de ssh se accedía a un usuario y este no tenía permisos de para ejecutar la instrucción *dump*. En el caso de la ejecución local se encontraron errores de interpretación con sintaxis de la *shell*. Después se optó por la decisión final explicada en los anteriores apartados, haciendo una doble copia (una de las cuáles queda en la máquina origen y no se usa). Posiblemente la solución de utilizar la opción remota de *dump* y el control de la existencia de las copias remotas por medio de *ssh* es más sencilla y no haría falta almacenar las copias en la máquina local.

10. REFERENCIAS

- Manual man de Linux (dump, restore y ssh).

- Transparencias del curso de Administración y Gestión de Sistemas Telemáticos disponibles en la página www.lab.dit.upm.es/~las

- http://www.oreilly.com/catalog/debian

  Chapter 4: *Issuing Linux Commands*
  Chapter 7: *Configuring and Administering Linux*
  Appendix B: *Principal Linux Files*
  Appendix E: *Linux Command Quick Reference*

- http://dump.sourceforge.net

- http://www.debian.org

Sobre dispositivos de almacenamiento:

- J.M. Menéndez, F.J. Casajús: " Tecnologías de Audio y Vídeo". ETSIT (UPM)

Sobre Unix:

- LIBRO DE UNIX