Respaldos multinivel robustos y simples utilizando `rsync`

Gunnar Wolf — gwolf@gwolf.org
http://www.gwolf.org/seguridad/munin

Instituto de Investigaciones Económicas, UNAM
Desarrollador del proyecto Debian

III Encuentro Internacional de Seguridad Informática
7-9 de octubre 2009, Manizales, Colombia

Contents

¿Cómo hacemos nuestros respaldos?

Un esquema muy poderoso, muy profesional...

- Cruzo los dedos y espero que no pase nada
  - Nunca me ha pasado nada desde que me corrieron del trabajo anterior
  - Ya aprendí, y ahora soy un buen administrador - ¡Soy invulnerable!
  - ¡Chin! Ya me volvieron a despedir :-/
- Realizo respaldos periódicos utilizando *fulanator*
  - Nunca he tenido que probar los mecanismos de recuperación
  - Funciona de maravilla en la plataforma que uso actualmente
  - Tiene una interfaz muy bonita

¿Cómo hacemos nuestros respaldos?

Un esquema muy poderoso, muy profesional...

- Cruzo los dedos y espero que no pase nada
  - Nunca me ha pasado nada desde que me corrieron del trabajo anterior
  - Ya aprendí, y ahora soy un buen administrador - ¡Soy invulnerable!
  - ¡Chin! Ya me volvieron a despedir :-/
- Realizo respaldos periódicos utilizando *fulanator*
  - Nunca he tenido que probar los mecanismos de recuperación
  - Funciona de maravilla en la plataforma que uso actualmente
  - Tiene una interfaz muy bonita

¿Cómo hacemos nuestros respaldos?

Un esquema muy poderoso, muy profesional...

- Cruzo los dedos y espero que no pase nada
  - Nunca me ha pasado nada desde que me corrieron del trabajo anterior
  - Ya aprendí, y ahora soy un buen administrador - ¡Soy invulnerable!
  - ¡Chin! Ya me volvieron a despedir :-/
- Realizo respaldos periódicos utilizando *fulanator*
  - Nunca he tenido que probar los mecanismos de recuperación
  - Funciona de maravilla en la plataforma que uso actualmente
  - Tiene una interfaz muy bonita

Respaldos en cinta

La cinta es el medio favorito para respaldos masivos

- Es compacta, puedo guardar terabytes en un cajón
- Es relativamente barata
- Unidades caras, medio físico medianamente accesible (del órden de MX$300 (US$20) pesos por cinta, para cientos de GB)
- Si necesito mayor volúmen, puedo comprar un robot multicintas (aunque son tremendamente caros y voluminosos)

Respaldos en cinta

Sin embargo, no puedo recomendar su uso más que como un medio auxiliar en un esquema de respaldos más completo

- La recuperación de archivos es muy lenta
  - La cinta es un medio de acceso secuencial
  - Requiere el uso de un programa específico, no es *montable* (directamente utilizable) desde el sistema
- Las cintas tienen una vida útil corta — En el mejor de los casos, 10 usos, y aún eso es demasiado arriesgado
- El fallo de un bit en el medio magnético puede destruir el respaldo entero
  - Muchas veces usamos compresión para aprovechar mejor el espacio
  - Si usamos paquetería de respaldos, muchas veces no podemos evitar el uso de compresión o el algoritmo a emplear

Discos ópticos

Mucha gente ha preferido el uso de medios ópticos (CD, DVD) para realizar respaldos

- Tanto medio como unidades muy baratos y confiables, aunque de baja capacidad — CDs, 700MB; DVDs 4GB o 9GB; Blu Ray 25GB(mas compresión)
- Medio rápido y fácil de comprobar para la recuperación: Si no utilizo compresión, puedo montar el disco como una unidad más del sistema.
- Incluso puedo usar discos con compresión directamente desde algunos sistemas operativos
- El medio reescribible es relativamente duradero
- El proceso de grabación nos reporta si hay defectos importantes en el medio
- Existen robots cambiadores (aunque en mi experiencia, no son muy confiables)

Principal problema con el uso de medios removibles

Los medios removibles son suceptibles a un fallo muy común:

Híííííííjole. . .

¡Se me olvidó meter la cinta/el disco de hoy!

¡Yo creí que sí estaba haciendo el respaldo, pero el cable está zafado desde hace meses!

(. . . y yo que no leo las bitácoras)

Principal problema con el uso de medios removibles

Los medios removibles son suceptibles a un fallo muy común:

Híííííííííjole. . .

¡Se me olvidó meter la cinta/el disco de hoy!

¡Yo creí que sí estaba haciendo el respaldo, pero el cable está zafado desde hace meses!

(. . . y yo que no leo las bitácoras)

Principal problema con el uso de medios removibles

Los medios removibles son suceptibles a un fallo muy común:

Híííííííííjole. . .

¡Se me olvidó meter la cinta/el disco de hoy!

¡Yo creí que sí estaba haciendo el respaldo, pero el cable está zafado desde hace meses!

(. . . y yo que no leo las bitácoras)

Principal problema con el uso de medios removibles

Los medios removibles son suceptibles a un fallo muy común:

Híííííííííjole. . .

¡Se me olvidó meter la cinta/el disco de hoy!

¡Yo creí que sí estaba haciendo el respaldo, pero el cable está zafado desde hace meses!

(. . . y yo que no leo las bitácoras)

Principal problema con el uso de medios removibles

Los medios removibles son suceptibles a un fallo muy común:

Híííííííííjole. . .

¡Se me olvidó meter la cinta/el disco de hoy!

¡Yo creí que sí estaba haciendo el respaldo, pero el cable está zafado desde hace meses!

(. . . y yo que no leo las bitácoras)

Respaldos a disco

Confiando en la capacidad de recordar sus tareas a tiempo, le encargo a mi servidor el llevar a cabo periódicamente los respaldos a disco

- El espacio en disco es relativamente barato
- Vale más mi tranquilidad que un disco duro adicional de 1TB
- Al hacer los respaldos en horarios de menor actividad, no sobrecargo la red en horas hábiles
- Es muy fácil verificar los respaldos y recuperar la información
- Es muy fácil programar los respaldos para las horas de menor actividad

Respaldos a disco

...Pero tampoco estos son todo lo que necesitamos

- ¡Me *cepillé* el respaldo junto con los datos!
  - Claro está, ninguna máquina debe ser destinatario de su propio respaldo
- No tenemos protección ante desastres mayores
  - Se incendió el cuarto de servidores, y...
  - Ahora bien... De los que hacen respaldo a cinta o medios ópticos, ¿quién los guarda a más de 50 metros del servidor?
  - No tengo un respaldo histórico — Fundamental para datos sujetos a auditoría

Respaldos a disco

...Pero tampoco estos son todo lo que necesitamos

- Un intruso que penetra a un servidor tiene acceso a los datos de los demás.
- **No usemos** el esquema de «respaldo de A en B, respaldo de B en A». Debemos designar un host específico para los respaldos.
  - Aislado de la red externa (y, de ser posible, de la red interna)
  - Físicamente alejado (recuerden los incendios/temblores, incluso robos de equipo)
  - Consideremos el manejo de cifrado

No existe un claro ganador

- No hay un esquema perfecto
- Debemos combinar estas soluciones según nos resulte más conveniente para nuestras necesidades específicas

Contents

Conociendo el sistema de archivos

- Para planear nuestro esquema de respaldos, resulta muy importante los detalles básicos de la interfaz que define a los sistemas de archivos en Unix
- Se llama genéricamente *sistema tipo Unix* a un sistema que implementa las interfaces POSIX
- POSIX especifica varios supuestos que cualquier sistema que cumple con su especificación debe cumplir para permitir una correcta interoperabilidad y permitiendo la fácil portabilidad de sistemas y de conceptos.

No me estoy refiriendo a la marca registrada Unix — Me refiero explícitamente a todos los sistemas *tipo Unix*

Los datos, los archivos y los inodos

En todos los sistemas de archivo nativos a cualquier sistema tipo Unix:

- Los directorios incluyen únicamente apuntadores a los datos
- Cada archivo se representa mediante un *inodo*, que indica la localización física de un conjunto de datos en una partición
- Cada inodo puede tener más de un nombre, y estar en más de un punto del árbol de directorios. Esto se llama una *liga dura* (detalles respecto las ligas más adelante)
- Podemos eliminar a un archivo de su directorio, y los demás lugares que apunten al inodo se mantienen sin modificación
- Claro está, sólo puedo crear ligas duras a archivos que estén dentro de la misma partición

Tipos de liga

En Unix tenemos más de un tipo de ligas - es importante comprenderlas para poder avanzar al siguiente punto.

Ligas duras

- Basadas en los principios que definimos en el punto anterior
- Cada uno de los archivos es equivalente, no hay uno maestro
- Deben estar en la misma partición
- Son poco utilizadas en la administración diaria
- No podemos hacer ligas duras a directorios — Esto podría crear ciclos en la estructura del árbol

Ligas duras — Un ejemplo

```
gwolf@lactop:/tmp$ cat > mis_datos.txt
Estos son mis datos, y los quiero. Me son muy importantes.
gwolf@lactop:/tmp$ ls -l mis_datos.txt
-rw-r--r-- 1 gwolf gwolf 77 2009-08-08 12:52 mis_datos.txt

gwolf@lactop:/tmp$ ln mis_datos.txt tambien_aqui.txt
gwolf@lactop:/tmp$ ls -l mis_datos.txt tambien_aqui.txt
-rw-r--r-- 2 gwolf gwolf 77 2009-08-08 12:52 mis_datos.txt
-rw-r--r-- 2 gwolf gwolf 77 2009-08-08 12:52 tambien_aqui.txt

gwolf@lactop:/tmp$ cat >> tambien_aqui.txt
Mis datos crecen!
gwolf@lactop:/tmp$ cat mis_datos.txt
Estos son mis datos, y los quiero. Me son muy importantes.
Mis datos crecen!

gwolf@lactop:/tmp$ rm mis_datos.txt
gwolf@lactop:/tmp$ ls -l tambien_aqui.txt
-rw-r--r-- 1 gwolf gwolf 95 2009-08-08 12:54 tambien_aqui.txt
gwolf@lactop:/tmp$ cat tambien_aqui.txt
Estos son mis datos, y los quiero. Me son muy importantes.
Mis datos crecen!
```

Tipos de liga

Ligas simbólicas

- Son sencillamente apuntadores a un inodo
- Una entrada en el directorio que apunta a otra entrada del directorio
- Pueden estar en cualquier punto del sistema de archivos (incluso en sistemas de archivos sin representación física)
- Si elimino a la copia maestra, elimino la información, y todas las ligas a ella quedan rotas (*colgantes*)

Ligas simbólicas — Un ejemplo

```
gwolf@lactop:/tmp$ cat > datos.txt
Estos son otros datos, y aunque no parezca, los quiero también.
gwolf@lactop:/tmp$ ls -l datos.txt
-rw-r--r-- 1 gwolf gwolf 64 2009-08-08 13:04 datos.txt

gwolf@lactop:/tmp$ ln -s datos.txt tambien.txt
gwolf@lactop:/tmp$ ls -l datos.txt tambien.txt
-rw-r--r-- 1 gwolf gwolf 64 2009-08-08 13:04 datos.txt
lrwxrwxrwx 1 gwolf gwolf 13 2009-08-08 13:05 tambien.txt -> datos.txt

gwolf@lactop:/tmp$ cat >> tambien.txt
Mis datos siguen creciendo!
gwolf@lactop:/tmp$ cat datos.txt
Estos son otros datos, y aunque no parezca, los quiero también.
Mis datos siguen creciendo!

gwolf@lactop:/tmp$ rm datos.txt
gwolf@lactop:/tmp$ ls -l tambien.txt
lrwxrwxrwx 1 gwolf gwolf 13 2009-08-08 13:05 tambien.txt -> datos.txt
gwolf@lactop:/tmp$ cat tambien.txt
cat: tambien.txt: No such file or directory
gwolf@lactop:/tmp$
```

El gran integrador: cp

El comando de copia en Unix, cp, nos simplifica la creación de ligas cuando lo requiramos — Citando a `man cp` (del cp de GNU):

- **-H** Follow command-line symbolic links
- **-l** Link files instead of copying
- **-L** Always follow symbolic links
- **-P** Never follow symbolic links
- **-p** Preserve the specified attributes (default: mode, ownership, timestamps), if possible additional attributes: links, all
- **-s** make symbolic links instead of copying

Y si bien es muy poco frecuente utilizar cp de esta manera a diario, resulta fundamental para el esquema global que veremos más adelante

¿Y Windows?

- En este apartado, estamos hablando únicamente de sistemas tipo Unix.
- En Windows no existen las ligas duras.
- Windows puede participar en el esquema de respaldos que proponemos únicamente como cliente
- Sin embargo... Todos sabemos que hablar de seguridad y Windows es por sí sólo ridículo ;-)
- Así que no se preocupen. Ni el mejor de los respaldos ayuda.

¿Y Windows?

- En este apartado, estamos hablando únicamente de sistemas tipo Unix.
- En Windows no existen las ligas duras.
- Windows puede participar en el esquema de respaldos que proponemos únicamente como cliente
- Sin embargo... Todos sabemos que hablar de seguridad y Windows es por sí sólo ridículo ;-)
- Así que no se preocupen. Ni el mejor de los respaldos ayuda.

Contents

¿Qué es eso?

- `ssh` (Secure Shell) es el principal protocolo (y así se designa también a los programas que lo implementan) que permite la administración de equipos o ejecución de comandos remotamente, empleando un *canal cifrado*
- El primer uso que cualquiera da a ssh es el de modo interactivo — Para permitirme administrar al servidor
- Pero aquí lo usaremos sencillamente para establecer un canal cifrado, seguro y confiable para transmitir nuestra información

¿Para qué?

- Para llevar a cabo un respaldo remoto, requerimos un método para permitir a nuestros servidores acceso al servidor de respaldos
- El acceso –en el caso de nuestra implementación– tiene que ser con privilegios de superusuario, pues de otro modo no se podrían reflejar todos los metadatos de los archivos
- El acceso con privilegios de superusuario significa que otorgamos acceso completo, absoluto e irrestricto al sistema , y debe ser manejado con sumo cuidado, limitándolo tanto como sea posible.
- Para *acabarla de amolar*, en mi Instituto no contamos con un servidor dedicado a los respaldos (¡y los realizamos en mi computadora personal!)

Generación de nuestro par de llaves

Vamos a generar las llaves para que los sistemas puedan identificarse mutuamente de manera automática.
En el servidor *del* cual crearemos respaldos (nuestro servidor actual):

```
root@server:~# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id\_dsa):
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id\_dsa.
Your public key has been saved in /root/.ssh/id\_dsa.pub.
The key fingerprint is:
30:bd:01:a0:8a:12:1f:4a:75:09:5f:bd:6c:d9:10:65
```

Generación de nuestro par de llaves

Importante — Contrario a toda recomendación que hayan visto: Las instrucciones de la lámina anterior van a crear una llave para la autenticación *sin contraseña*

- El respaldo va a ser automático
- Poner una contraseña en un script es peor que no tener dicha contraseña — Nada peor que un *falso sentido de seguridad*
- Como sea: ¡Estén conscientes del riesgo! Un atacante puede aprovechar esta llave para eliminar todos nuestros respaldos, o tener acceso a información muy sensible

Envío de la llave al servidor de respaldos

- Permitimos acceso directo como administrador en nuestro servidor de respaldos. en `/etc/ssh/sshd_config`: `PermitRootLogin yes`
- Enviamos la llave al servidor de respaldos, y comprobamos que funcione correctamente entrando — ya no debe solicitarnos contraseña.

```
root@server:~# ssh-copy-id root@respaldos
Password:
Now try logging into the machine, with "ssh 'localhost'", and check in:
.ssh/authorized_keys
to make sure we haven't added extra keys that you weren't expecting.
root@server:~# ssh root@respaldos
Linux respaldos 2.6.26-1-vserver-amd64 #1 Sat Jan 10 19:46:42 UTC 2009
root@respaldos:~#
```

¿Y para servidores Windows?

- Es igual
- En Windows hay varios clientes disponibles de ssh
- Incluyendo varios de intercambio de archivos (scp, sftp)
- Pero lo que mejor sirve para nuestros fines, existe también un port basado en el OpenSSH que todos queremos y amamos (http://sshwindows.sourceforge.net/), y hay varias otras alternativas disponibles (http://www.openssh.com/windows.html)
- Todo lo mencionado en esta sección aplica exactamente igual en Windows

Contents

¿Y si no tengo un servidor dedicado?

Si no contamos con un equipo para destinar a servidor de respaldos y no nos gusta tener llaves sin contraseña, no todo está perdido. Podemos construir un entorno protegido y casi autónomo dentro de nuestro servidor: un *chroot*

- Chroot (*Ch*ange *root*) es una facilidad que provee todo sistema tipo Unix a través de la llamada al sistema del mismo nombre
- Aísla a un proceso y sus hijos del resto del sistema, cambiando el directorio raíz
- Oculta todo lo que queda fuera del directorio especificado
- **Pero no debe ser visto como un mecanismo de seguridad**, sólo como una cómoda facilidad de Unix

Viviendo en un chroot

- Puedo hacer una instalación de un sistema operativo completa dentro de este directorio
- Puede ser incluso una distribución diferente del sistema anfitrión, siempre y cuando funcione con el mismo núcleo
- Dentro de mi entorno *chrooteado* puedo correr ssh
- Aislando la sesión entrante del resto del sistema
- Los usuarios del sistema huésped son independientes de los del sistema anfitrión
- No es necesario que dé `PermitRootLogin` en mi sistema anfitrión
- Incluso puedo correrlo a determinada hora y cerrarlo una vez recibida la conexión de los respaldos diarios

Contextos de seguridad: `vserver`

Como ya mencionamos, un `chroot` no debe ser visto como suficiente infraestructura desde el punto de vista de seguridad. El proyecto vserver extiende las ideas de un chroot, agregando:

- Todos los procesos llevan un *contexto de seguridad*. Un proceso no puede interactuar directamente (IPC, señales, ni siquiera verlos en `/proc`) con uno de un contexto diferente.
- Puedo especificar límites de consumo de recursos (memoria, disco) a los *servidores virtuales*.
- Los servidores virtuales no pueden crear *nodos de dispositivo* (archivos a través de los cuales se pueda acceder a dispositivos físicos). *Sí pueden utilizarlos* si éstos son creados en su área desde el sistema huésped.
- Hay varias maneras de configurar la red, pero típicamente es montando un firewall en el sistema huésped

Creando un vserver

```
root@anfitrion:~# vserver respaldos build -m debootstrap --rootdir \
    /var/lib/vhosts/respaldos -- -d lenny
(...)
root@anfitrion:~# vserver respaldos start
Starting enhanced syslogd: rsyslogd.
root@anfitrion:~# vserver respaldos enter
root@respaldos:/# ls /dev/
core  full     log   ptmx  ram     shm     stdin   tty      xconsole
fd    initctl  null  pts   random  stderr  stdout  urandom  zero
root@respaldos:/# exit
root@anfitrion:~# vserver-stat
CTX   PROC    VSZ    RSS  userTIME   sysTIME    UPTIME NAME
40009    4  26.9M     3M   0m00s84   0m00s15   1m06h21 respaldos
root@anfitrion:~# vserver respaldos stop
Stopping enhanced syslogd: rsyslogd.
Asking all remaining processes to terminate...done.
All processes ended within 1 seconds....done.
root@anfitrion:~# vserver respaldos delete
Are you sure you want to delete the vserver pruebita (y/N) y
```

Contents

¿Qué es rsync?

Rsync es un protocolo y un programa para la sincronización de depósitos de archivos binarios. De su página de manual:

- support for copying links, devices, owners, groups, and permissions
- exclude and exclude-from options similar to GNU tar
- a CVS exclude mode for ignoring the same files that CVS would ignore
- can use any transparent remote shell, including ssh or rsh
- does not require root privileges
- pipelining of file transfers to minimize latency costs
- support for anonymous or authenticated rsync daemons (ideal for mirroring)

Nosotros **no** utilizaremos el demonio propio de `rsync` — Nos conectaremos, como vimos, a través de `ssh`.

¿Cómo lo invocaremos?

```
rsync -q -a -l -H -o -g -D -t -e ssh --delete-excluded /bin /boot /dev
/etc /home /lib /opt /root /sbin /srv /usr /var --exclude
/var/spool/squid root@respaldos:/respaldo/servidor
```

¡¿Huh?!

- -q Operación silenciosa (sólo muestra los errores, si es que hay)
- -a Modo de archivo/espejo
- -l Mantiene las ligas simbólicas como tales
- -H Mantiene las ligas duras como tales
- -o Conserva la información del dueño de cada archivo
- -g Conserva la información del grupo de cada archivo
- -D Respalda los árboles de dispositivos Unix

¿Cómo lo invocaremos?

```
rsync -q -a -l -H -o -g -D -t -e ssh --delete-excluded /bin /boot /dev
/etc /home /lib /opt /root /sbin /srv /usr /var --exclude
/var/spool/squid root@respaldos:/respaldo/servidor
```

¡¿Huh?!

- -q Operación silenciosa (sólo muestra los errores, si es que hay)
- -a Modo de archivo/espejo
- -l Mantiene las ligas simbólicas como tales
- -H Mantiene las ligas duras como tales
- -o Conserva la información del dueño de cada archivo
- -g Conserva la información del grupo de cada archivo
- -D Respalda los árboles de dispositivos Unix

¿Cómo lo invocaremos?

```
rsync -q -a -l -H -o -g -D -t -e ssh --delete-excluded /bin /boot /dev
/etc /home /lib /opt /root /sbin /srv /usr /var --exclude
/var/spool/squid root@respaldos:/respaldo/servidor
```

¡¿Huh?!

-e ssh Utiliza un shell remoto en vez del protocolo rsync

directorios Los directorios a respaldar

-exclude *directorios* Los directorios a excluir del respaldo (p.ej. áreas temporales, caché, etc.

–delete-excluded Si existe en el destino un directorio marcado como excluído, eliminarlo

root@respaldos:/respaldo/servidor Usuario, servidor y ruta destino para enviar los datos

¡Ahorre ancho de banda! ¡Use rsync!

- Rsync es un protocolo muy eficiente para comparar grandes conjuntos de datos, evitando enviar repeticiones
- Los archivos que no se han modificado no se tienen que reenviar
- Los archivos grandes que sí han sido modificados se separan en bloques, y sólo se envían los bloques modificados
- Los demás bloques son comprobados por un algoritmo de checksum de 128 bits
- Rsync es altamente eficiente al preparar la lista de cambios a enviar — No es descabellado utilizarlo para realizar respaldos remotos, incluso con conexiones aptas para un pequeño negocio (DSL)

La eficiencia de rsync

- En un firewall: 750Kb de 1GB (0.7 %)

```
firewall# rsync -v -a -l -H -o -g -D -t  -e ssh --delete /bin \
/boot /dev /etc /home /lib /opt /root /sbin /srv /usr /var \
--exclude /dev root@172.16.10.1:/home/backups/firewall
sending incremental file list
    (...)
sent 759884 bytes  received 15153 bytes  516691.33 bytes/sec
total size is 1030176379  speedup is 1329.20
```

- En un servidor de archivos: 2.2GB de 575GB (0.5 %)

```
fileserver# rsync -v -a -l -H -o -g -D -t  -e ssh --delete /bin \
/boot /dev /etc /home /lib /opt /root /sbin /srv /usr /var \
--exclude /dev root@172.16.10.1:/home/backups/fileserver
sending incremental file list
    (...)
sent 2412866255 bytes  received 3915958 bytes  1244801.55 bytes/sec
total size is 618204347784  speedup is 255.80
```

¿Rsync y Windows?

- No quieres que tu servidor de respaldos sea Windows.
- No, en serio. No quieres.
- Ok, ¿necesitas razones? La principal es porque Windows no puede manejar ligas duras — y a continuación veremos su importancia...
- Sin embargo, sí, existe rsync como cliente y como servidor para Windows
- La mayor parte de los documentos en la red apuntan a que lo mejor es instalar rsync utilizando Cygwin
- Cygwin es una implementación del API completo de Unix
- Es muy grande, sin embargo, y es recomendable evitarlo si no es indispensable.
- http://www.gaztronics.net/rsync.php menciona una implementación nativa

¿Rsync y Windows?

- No quieres que tu servidor de respaldos sea Windows.
- No, en serio. No quieres.
- Ok, ¿necesitas razones? La principal es porque Windows no puede manejar ligas duras — y a continuación veremos su importancia...
- Sin embargo, sí, existe rsync como cliente y como servidor para Windows
- La mayor parte de los documentos en la red apuntan a que lo mejor es instalar rsync utilizando Cygwin
- Cygwin es una implementación del API completo de Unix
- Es muy grande, sin embargo, y es recomendable evitarlo si no es indispensable.
- http://www.gaztronics.net/rsync.php menciona una implementación nativa

¿Rsync y Windows?

- No quieres que tu servidor de respaldos sea Windows.
- No, en serio. No quieres.
- Ok, ¿necesitas razones? La principal es porque Windows no puede manejar ligas duras — y a continuación veremos su importancia...
- Sin embargo, sí, existe rsync como cliente y como servidor para Windows
- La mayor parte de los documentos en la red apuntan a que lo mejor es instalar rsync utilizando Cygwin
- Cygwin es una implementación del API completo de Unix
- Es muy grande, sin embargo, y es recomendable evitarlo si no es indispensable.
- http://www.gaztronics.net/rsync.php menciona una implementación nativa

Contents

Esquema general

- Respaldo periódico utilizando rsync
- Comunicación entre los servidores y el equipo de respaldos vía ssh con intercambio de llaves
- Antes de iniciar el respaldo, copiamos el respaldo actual en el servidor de respaldos
- La copia se hace con un simple `cp -al`, evitando ocupar espacio adicional con los archivos que no han sido modificados
- El único espacio adicional que se ocupa es la duplicación del árbol de directorio (y, claro, las diferencias entre los varios árboles)

Recuperando un archivo del respaldo

Llega un usuario, y me pide...

- Necesito recuperar el archivo llamado `mi_archivo`

```
root@servidor:~# scp \
    respaldos:/respaldo/servidor/home/usuario/mi_archivo \
    /home/usuario/mi_archivo_recuperado
```

- No recuerdo el nombre del archivo...Pero era un PDF

```
root@servidor:~# ssh respaldos
root@respaldos:~# cd /respaldo/servidor/home/usuario
root@respaldos:~# ls *.pdf
  (...)
root@respaldos:~# logout
root@servidor:~# scp \
    respaldos:/respaldo/servidor/home/usuario/archivo.pdf \
    /home/usuario/
```

Recuperando un archivo del respaldo

Llega un usuario, y me pide...

- Necesito recuperar el archivo llamado `mi_archivo`

```
root@servidor:~# scp \
    respaldos:/respaldo/servidor/home/usuario/mi_archivo \
    /home/usuario/mi_archivo_recuperado
```

- No recuerdo el nombre del archivo...Pero era un PDF

```
root@servidor:~# ssh respaldos
root@respaldos:~# cd /respaldo/servidor/home/usuario
root@respaldos:~# ls *.pdf
  (...)
root@respaldos:~# logout
root@servidor:~# scp \
    respaldos:/respaldo/servidor/home/usuario/archivo.pdf \
    /home/usuario/
```

Recuperando un archivo del respaldo

- Ninguno de los que me muestras me suena... Era un archivo relacionado con la agroindustria.

```
root@servidor:~# ssh respaldos
root@respaldos:~# cd /respaldo/servidor/home/usuario
root@respaldos:~# grep -li agro *.pdf
bimbo.pdf   campaña_maiz.pdf    monsanto.pdf
root@respaldos:~# logout
root@servidor:~# scp \
    respaldos:/respaldo/servidor/home/usuario/campaña_maiz.pdf \
    /home/usuario/
```

En fin, su uso es como el uso de un sistema de archivos normal Unix.
Sin embargo. . . ¡Podemos llegar más allá!

Recuperando un archivo del respaldo

- Ninguno de los que me muestras me suena... Era un archivo relacionado con la agroindustria.

```
root@servidor:~# ssh respaldos
root@respaldos:~# cd /respaldo/servidor/home/usuario
root@respaldos:~# grep -li agro *.pdf
bimbo.pdf   campaña_maiz.pdf    monsanto.pdf
root@respaldos:~# logout
root@servidor:~# scp \
    respaldos:/respaldo/servidor/home/usuario/campaña_maiz.pdf \
    /home/usuario/
```

En fin, su uso es como el uso de un sistema de archivos normal Unix.
Sin embargo... ¡Podemos llegar más allá!

Auxiliando al administrador del sistema

- Es un archivo que borré hace cuatro días

```
root@respaldos:~# cd /respaldo/servidor
root@respaldos:servidor# diff -ur <(find /home/usuario -type f) \
    <(cd ../servidor.4; find /home/usuario -type f)
```

- ¿Qué configuraciones cambié de ayer a hoy?

```
root@respaldos:~# cd /respaldo/
root@respaldos:/respaldo# diff -ur servidor.1/etc servidor.2/etc
```

- ¡Un exploit en /var/tmp/.bash_history! ¡Tengo que encontrar la puerta que utilizaron en la bitácora! ¿Hace cuántos días habrán entrado?

```
root@respaldos:/respaldo# ls -dl servidor*/var/tmp/.bash_history
```

En fin... Como todo en la administración: La imaginación es el límite

Auxiliando al administrador del sistema

- Es un archivo que borré hace cuatro días

```
root@respaldos:~# cd /respaldo/servidor
root@respaldos:servidor# diff -ur <(find /home/usuario -type f) \
    <(cd ../servidor.4; find /home/usuario -type f)
```

- ¿Qué configuraciones cambié de ayer a hoy?

```
root@respaldos:~# cd /respaldo/
root@respaldos:/respaldo# diff -ur servidor.1/etc servidor.2/etc
```

- ¡Un exploit en /var/tmp/.bash_history! ¡Tengo que encontrar la puerta que utilizaron en la bitácora! ¿Hace cuántos días habrán entrado?

```
root@respaldos:/respaldo# ls -dl servidor*/var/tmp/.bash_history
```

En fin... Como todo en la administración: La imaginación es el límite

Auxiliando al administrador del sistema

- Es un archivo que borré hace cuatro días

```
root@respaldos:~# cd /respaldo/servidor
root@respaldos:servidor# diff -ur <(find /home/usuario -type f) \
    <(cd ../servidor.4; find /home/usuario -type f)
```

- ¿Qué configuraciones cambié de ayer a hoy?

```
root@respaldos:~# cd /respaldo/
root@respaldos:/respaldo# diff -ur servidor.1/etc servidor.2/etc
```

- ¡Un exploit en /var/tmp/.bash_history! ¡Tengo que encontrar la puerta que utilizaron en la bitácora! ¿Hace cuántos días habrán entrado?

```
root@respaldos:/respaldo# ls -dl servidor*/var/tmp/.bash_history
```

En fin... Como todo en la administración: La imaginación es el límite

Auxiliando al administrador del sistema

- Es un archivo que borré hace cuatro días

```
root@respaldos:~# cd /respaldo/servidor
root@respaldos:servidor# diff -ur <(find /home/usuario -type f) \
    <(cd ../servidor.4; find /home/usuario -type f)
```

- ¿Qué configuraciones cambié de ayer a hoy?

```
root@respaldos:~# cd /respaldo/
root@respaldos:/respaldo# diff -ur servidor.1/etc servidor.2/etc
```

- ¡Un exploit en /var/tmp/.bash_history! ¡Tengo que encontrar la puerta que utilizaron en la bitácora! ¿Hace cuántos días habrán entrado?

```
root@respaldos:/respaldo# ls -dl servidor*/var/tmp/.bash_history
```

En fin... Como todo en la administración: La imaginación es el límite

¿Y la implementación?

- La razón por la que pasamos por toda la explicación técnica previa es mostrar que... no requerimos de un software complejo diseñado a la medida
- Las facilidades básicas de Unix *nos brindan todo lo necesario* para tener un sistema profesional de respaldos listo para ser utilizado
- La filosofía básica de Unix nos permite sumar pedacitos para lograr esquemas tan completos como el aquí mencionado –comentarios y variables incluídos– en 73 líneas de código
- El script, tal como lo corro en mis servidores de producción, está disponible en el mismo URL que esta presentación (ver siguiente lámina).

¡Muchas gracias!

¿Dudas?

gwolf@gwolf.org
http://www.gwolf.org/seguridad/respaldos_con_rsync/

Muchas gracias a Mike Rubel, pues la idea e implementación original es suya. Su artículo:
http://www.mikerubel.org/computers/rsync_snapshots/