Insider Tips: Backups

# ON THE SAFE SIDE

Data always seems to get lost at exactly the wrong moment, but the right backup strategy can help you restore those missing files.

**BY MARC ANDRÉ SELIG**

Because the causes and requirements surrounding data loss can be so vastly different, a number of different solutions have emerged over the years. All of these solutions have benefits and drawbacks. In this month's Admin Workshhop, I'll describe some common backup tools and techniques.

## Backup Alternatives

Magnetic tape was once the most common backup medium, and tape backups are still popular for networks with large quantities of data. Tapes are fairly cheap despite their high capacity, with speed being the major disadvantage. A tape backup solution in connection with a jukebox is ideal for fully automated backups. This said, tape drives are often too expensive for SOHO environments.

CD, DVD, flash memory, as well as internal and external hard disk backups, are now more common. In larger environments, admins might use a NAS (Network Attached Storage) system to add hard disk capacity.

Just as there are different backup media, there are also different backup strategies. In most cases, admins opt for incremental backups, which only store the changes that have occurred since the last full or incremental backup. This approach saves space on the backup media, which improves cost-efficiency and makes the incremental backup a popular option.

The big disadvantage of incremental backups is that restoring lost or damaged files takes more time than with a full backup. Additionally, admins may need to change the backup media if they do not have a jukebox-style solution. The third variant, the differential backup always stores the changes since the previous full backup. Figure 1 illustrates the three approaches.

## Off-line, On-line, Hot

The choice of a backup method may depend upon the circumstances in which the data will be restored. If the file a user needs is located on a tape in a cupboard, the act of restoring the file will require time and labor.

In contrast to this, on-line "hot" backups reside on media that support 24x7 (and automated) access. This method saves time, and often money. however, hot backups only protect against hardware damage. They are no protection against user or admin errors, which will propagate onto the backup medium soon after the user or admin stores the data in question. This is why most admins do not consider hot backup an alternative to conventional backup.

## Formats

Admins disagree on the pros and cons of using single files or more complex

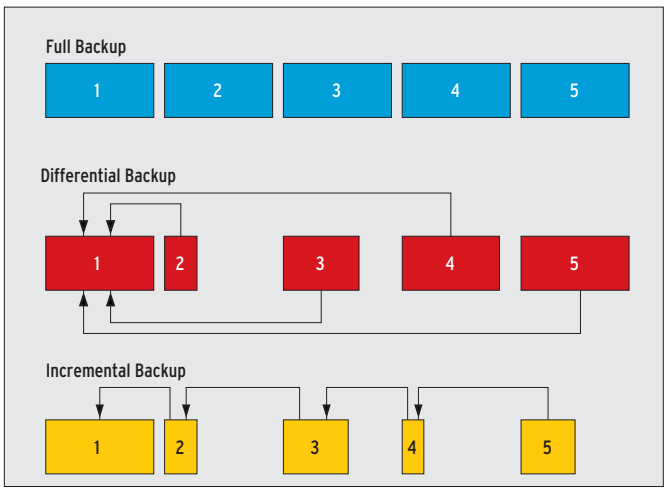


**Figure 1: A full backup saves all files, a differential backup saves all those files modified since the last full backup, and an incremental backup saves all those files modified since the last incremental or full backup.**

archives that contain a structured backup set (or all the files saved on a backup medium) along with metadata and a checksum.

Single files can be restored more quickly, and if the backup medium has a local fault, the fault only affects a single file, whereas a damaged archive means losing multiple files.

Archive files have some features that individual file backups can't give you. For example, they store owner data, access privileges and timestamps along with the file content. You can even back up special devices from the */dev/* directory. Additionally, magnetic tapes are not well-suited to storing masses of individual files.

Some programs, including *tar* and *cpio*, attempt to find the golden mean. If a *cpio* file is damaged, the damage is restricted to the files stored at the faulty location. The program resyncs at the next file end marker, and any files that follow the marker are easily restored.

When discussing the single file vs. archive issue, you also need to take compression and encryption into consideration. The resynchronization feature for *cpio* files only works for uncompressed backups. If a read error prevents you from decompressing the archive, *cpio* is powerless to help you.

The popular *gzip* tool quits in the case of read errors and is thus a bad choice for backups.(*zcat* will at least decompress the archive up to the point where the read error occurs.) The alternative, *bzip2*, compresses and decompresses files in blocks of 900 kbytes at the most. If a read error occurs, you may only lose a single block; any preceding and following blocks are likely to be unaffected.

Admins face a similar dilemma with encrypted data. Most stream ciphers that backup programs use will prevent any access to the archive in case of error. One approach might be to compress or encrypt every single file in the archive individually. The *afio* tool is a possible alternative to *cpio* as it can handle separate encryption of archived files.

## CD Backup

A tape backup solution such as Amanda (see Box 1) scales down well but is still perhaps better suited to larger environments. Private users or small businesses might be just as happy with a simple CD or DVD based backup. In comparison to magnetic tapes, CDs and DVDs are extremely cheap and have a longer lifecycle.

Listing 1 gives you a simple backup script, which calls *gpg* to encrypt the backup data and stores a simple MD5 checksum to boot. If a CD goes astray, at least you do not need to worry about unauthorized access to your data. You could modify this script to support flash media or an external hard disk.

## The Right Approach

A backup system is only as good as the data on the medium. And this data is not necessarily what the backup program intended to write. So it makes sense to check your backups for readability and accuracy at regular intervals.

Also, you should make sure that users are capable of restoring their own data. There is nothing more annoying than needing to restore a backup that someone else set up a long time ago, and not being able to do so because the engineer who set up the system is not available.

The case of total data loss leads to a whole bunch of additional issues. As the operating system itself may not be available, a rescue system makes sense. The rescue system should boot from CD or an external disk and allow the admin to restore the full set of data from there. Of course, this kind of solution requires planning and practice. ■

### Listing 1: Simple Backup Script

```
01 #!/bin/sh
02
03 [ `id -u` -eq 0 ] || ( echo
   'Must be root to write a
   CD/DVD!' && exit )
04
05 TODAY=`date +%Y%m%d.%H%M`
06 MYKEY='0x598342d9'
07
08 umask 022
09 mkdir -p /tmp/root/backup-$TODAY
10
11 cd /
12 tar cf - etc home usr/local | \
13  gpg -v --homedir $HOME/.gnupg
   -e -r $MYKEY | \
14  tee /tmp/root/backup-$TODAY/
   backup-$TODAY.tar.gpg | \
15  md5sum -b >/tmp/root/backup-
   $TODAY/backup-$TODAY.tar.gpg.md5
16
17 cd /tmp/root
18 mkisofs -r -pad -o backup.iso
   backup-$TODAY
19 cdrecord -v -eject -multi
   dev=0,0,0 -driveropts=
   burnproof -speed=24 -pad
   backup.iso
20
21 rm -rf backup-$TODAY backup.iso
```

### Example: Tape Backup

Tapes are a popular and widespread backup medium. Tapes often have isolated read errors, which can be avoided by more sophisticated software tools. What makes things worse is the fact that many kernel drivers need pre-formatted blocks for tape devices. In other words, not every tape device makes a good target for *tar cpf*.

The easiest approach is to use a ready-to-run backup system such as Amanda [2], which can collect data from an (almost) unlimited number of hosts and write backups to tape. Amanda supports a variety of Unix systems, and there are even clients for Microsoft Windows [3].

The system is based on a client/server model. You need to install an Amanda client on each host where you want Amanda to collect backup data. Of course, the client will need read privileges for any data destined for the Amanda server. The master then uses UDP to transmit requests to the clients which respond by sending their backups via TCP. Amanda can use either *dump* or *tar* to create the archive files.

Amanda has a sophisticated approach to scheduling individual backups. The master program references information about the tape pool and the configured full backup intervals to specify full and incremental backup jobs. This means that each host is backed up as often as possible, and at least as often as configured. Amanda then fills the gaps on the tapes with incremental backups.

### INFO

[1] Afio: *http://directory.fsf.org/sysadmin/backup/afio.html*

[2] Amanda: *http://www.amanda.org*

[3] Amanda client for Windows: *http://sourceforge.net/projects/amanda-win32/*