Backup

Track SA-E

AfNOG workshop

May 15, 2009

Cairo, Egypt

(Slides by Phil Regnauld)

Backup

- What is backup?
  - backup is part of a larger domain called data security:
    - integrity, protection: cryptography
    - availability, redundancy: mirroring / RAID
- Why Backup?
  - Software and Hardware failures are a common thing in the computer world. Any number of occurrences can cause loss of valuable data.

Backup

- Types of failures
  - Power failures (software/hardware failure)
  - Natural disasters (fire, flood)
  - Security incidents (theft)
  - Hardware Failures (disk crash)
  - User error (rm -rf)

Types of backups

``Do nothing'' is not a computer program, but it is the most widely used backup strategy. There are no initial costs. There is no backup schedule to follow. Just say no. If something happens to your data, grin and bear it!

If your time and your data is worth little to nothing, then ``Do nothing'' is the most suitable backup program for your computer. But beware, UNIX is a useful tool, you may find that within six months you have a collection of files that are valuable to you.

``Do nothing'' is the correct backup method for /usr/obj, /usr/src and other directory trees that can be exactly recreated by your computer – but if in doubt, BACK IT UP!

dd

- The lowest level type of backup
- Bit-for-bit copy
- For example:

```
dd if=/dev/ad0s1a of=/backup/slash
```

- Exact copy, but not efficient
  - if you only use 100 MB on a 1 GB partition, you still end up with a backup of 1 GB
    - compression helps, but you still spend time copying unused space
  - Best for doing system recovery

Dump

- The traditional UNIX® backup programs
  - dump and restore.
  - Works at inode level
  - Takes backups of entire filesystems, but only the used space
  - It is unable to backup only part of a file system
- It does not backup across mount points (directory tree that spans more than one file system)

- **Note**: If you use dump on your / partition, you would not back up /home, /usr or or any other mounted FS. You must explicitly run dump for each FS.

Dump

- Dump can backup to several media
  - local file
  - remote file
  - tape
- Dump can take incremental dumps
  - only files that have changed are backup up

Remote dump

- It is possible to run dump over ssh for a secure transport:

```
# /sbin/dump -0uan -f - /usr
    | gzip -2 | ssh
targetuser@targetmachine.example.com
dd of=/backups/dump-usr.gz
```

Tar

- tar(1) (Tape Archive) dates back to Version 6 of AT&T UNIX (circa 1975). tar operates in cooperation with the file system; tar writes files and directories to tape or to a file.

- Just like with dump, one can use ssh to backup across the network:

```
# tar -cfz - /
   | (ssh remote;
      cat >/backups/backup-0425.tgz)
```

Examples using tar

- Let's take a backup of /etc where most configuration files reside, and place it in /home/backups

```
# mkdir /home/backups
# tar -cvf /home/backups/etc.tar /etc
```

Note: The -c option to tar tells it to create an archive, -v specifies verbose output and -f specifies the file to be either written to or read from

- You'll see quite a lot of output as tar creates the archive at this point.

Examples using tar

- Now we check whether our archive has actually been created

```
# cd /home/backups
# ls
```

- This now show us a new file in this directory `etc.tar`
- If we now wanted to view the contents of this backup we can run

```
# tar -tvf etc.tar
```

Examples using tar

- This will show you the contents of the etc directory as you backed it up.

- To actually restore and and unpack the contents that were backup up previously:

```
# cd /home/backups
# tar -xvf etc.tar
```

Examples using tar

- Notice that the restore actually creates a new directory etc where you are located – not in /etc!

- This is because tar by default removes the leading '/' from the directories it is backup up in order not to overwrite the original files on your system when you choose to do a restore.(security consideration)

Rsync

- Another very powerful tool is rsync
  http://samba.anu.edu.au/rsync/

- Rsync is very efficient: it only transfers files that have changed, and for those files, only the *parts* of the files that have changed
  - This is very efficient for large trees with many files, some of them large
- Great for replicating a server off-site, or for doing quick backups for a migration.

Rsync

- Combined with the --link-dest option, it allows to do snapshot-like backups.
- --link-dest takes the newest backup, and makes links (which take 0 space) to the files that have not changed, and replicates those that have changed
- Allows for backup.0, backup.1, backup.2, backup.3, where backup.X is a COMPLETE copy of the replicated source, but the disk space used is ONLY the difference.

Rsync – example script

- On remote backup host:

```
# rm -rf /backups/etc.2
# mv /backups/etc.1 /backups/etc.2
# mv /backups/etc.0 /backups/etc.1
# mv /backups/etc /backups/etc.0
```

- On machine to be backed up:

```
# rsync -avHS \
   --link-dest=etc.0 \
   /etc/ host:/backups/etc/
```

- This will backup only changed files from /etc/ to host:/etc/. Unchanged files are linked from etc.0

Other tools

- Rdiff-backup
  http://www.nongnu.org/rdiff-backup/
- Unison
  http://www.cis.upenn.edu/~bcpierce/unison/
- Rnapshot
  http://www.rsnapshot.org/

Other possible Backup methods

- Disk duplication
  - Using the dd command mentioned earlier, it is possible to duplicate your entire disk block by block on another disk. However the source and destination disk should be identical in size or the destination must be bigger than the source.

- Another way of doing this is using RAID1 mirroring and hot swappable disks:
  - make sure the RAID volume is rebuilt (OK)
  - remove one of the two disks (call it “backup”)
  - replace “backup” with a fresh disk, let the RAID rebuild
  - take “backup” home

  Remember: RAID or mirroring is not backup. An “`rm -rf /`” on your RAID set works very well!

Other possible Backup methods

- Disk duplication (2)
  - instead of mirroring the two disks, make two filesystems, and use rsync to copy every night from disk 1 to disk 2
  - in case of user error (rm -rf), you can recover from disk 2, without having to pull the backup tapes out of the safe

  NOTE: IT DOES NOT HELP IF THE SERVER IS STOLEN OR THERE IS A FIRE, IF BOTH DISKS ARE IN THE MACHINE!

Networked backup systems

- There are a number of networked backup systems out there for backing up many servers to one or more backup servers, using tape drives or disk storage.

- In the Open Source world, two backup systems stand out:

  - AMANDA - http://www.amanda.org/
  - BACULA - http://www.bacula.org/

Amanda

- Advanced Maryland Automatic Network Disk Archiver
  - has been around for many years
  - networked backup
  - support incremental backups to disk, tape
  - can backup to a holding disk, flush to tape later
  - encrypted data flows and backup data
  - tape library / loader control and labelling
  - Windows backup through SMB only...

Bacula

- Written by the people who invented AutoCAD
  - impressive documentation (400- pages!), including a developer's guide and tutorial
  - support incremental backups to disk, tape
  - complete SQL backend (MySQL, PgSQL, SQLite)
  - encrypted data flows using TLS (standard!)
  - tape library / loader control and labelling
  - native Windows client
  - good documented scenarios for specific backup cases, including complete “bare metal” restore

Reminder: Backup security

1. Take the disks / tapes / CDs off site!
   -> it does not help if there is a fire or if tapes are stolen

2. Consider encrypting the data on the disks / tapes / CDs
   -> what happens if the tapes are stolen? what happens when you throw them out?