Backup−With−MSDOS mini−HOWTO

Table of Contents

Backup−With−MSDOS mini−HOWTO.....1
- Christopher Neufeld, neufeld@physics.utoronto.ca.....1
- 1. Preface/Introduction.....1
- 2. The technique.....1
- 3. Notes.....1
- 4. Copyright.....1
- 1. Preface/Introduction.....1
- 2. The technique.....1
- 3. Notes.....7
- 4. Copyright.....8

Backup−With−MSDOS mini−HOWTO

Christopher Neufeld, `neufeld@physics.utoronto.ca`

v, 5 August 1997

This HOWTO describes how to use a Linux−compatible tape drive installed on an MS−DOS machine to back up the filesystem of a Linux machine.

---

1. Preface/Introduction

2. The technique

3. Notes

4. Copyright

---

1. Preface/Introduction

Earlier I posed the question on the Net, how does one back up a Linux machine to a Colorado Jumbo 250 tape drive on an MS−DOS machine. From the email I received, it seems that this is a frequently pondered problem. Now that I've figured it out, I'm posting the method. If anybody wants to massage this into a HOWTO document, let me know. I should thank Jim Nance (jlnance@isscad.com) for pointing out that an MS−DOS machine need not always be an MS−DOS machine. This technique should also work for any other tape drive supported by the ftape module, and for SCSI tape drives with suitable obvious changes (i.e. substituting /dev/st0 for /dev/ftape).

The criteria I set were that the resulting setup should be as secure as possible and should be fairly simple, and take up little or no space on the MS−DOS machine's hard drive. It should also be capable of recovering from the worst system corruptions, up to and including the theft of the hard disk, requiring a restore to a bare Linux file system. The technique described here uses no hard drive space on the MS−DOS machine, though it requires that that machine be assigned an IP#. You will need three formatted, blank 1.44MB diskettes.

---

2. The technique

Throughout this description I will refer to two machines as ``msdos'' and ``linux''. ``msdos'' is the name of the machine which has the tape drive and is usually running MS−DOS. ``linux'' is the Linux machine whose disk you are trying to back up or restore to the tape drive. For simplicity I will refer to the first machine as ``msdos'' even when it is booted into and running Linux. Further, all path names in this document should be considered to be relative to the Linux machine with the Search−And−Rescue (SAR) disks mounted somewhere on the system. That means that the file `/etc/passwd` is the password file for your Linux machine's hard drive, while, for instance, `/tape144/etc/passwd` is the corresponding file on the floppy disk.

I am using Karel Kubat's backup scripts, version 1.03, available at

> `ftp://sunsite.enc.edu:/pub/Linux/system/Backup/backup−1.03.tar.gz`

Throughout this document I will refer to these simply as ``the backup scripts''. You do not have to use these scripts for your own backups to tape. I like these scripts as they use afio to form an uncompressed archive of compressed files, rather than a compressed archive of uncompressed files. The former is much safer if there is a media read error during the restore. I understand that Karel is no longer supporting backup, and now has produced 'tob', or tape oriented backup. While I haven't tried the new package myself, it cannot make a significant difference to the procedure outlined here.

First of all, obtain the ftape module. It is a part of all modern kernels, but if you are using an older kernel you can find the module at:

> `ftp://sunsite.unc.edu/pub/Linux/kernel/tapes/ftape−2.05.tar.gz`

Next, get a Slackware boot disk (I got the net disk, but it doesn't make much difference) and the `tape144` root disk, and put the images onto 3"1/2 floppies.

The `ftape` module will only work if it is installed in the kernel which was running when you compiled it. I could not get it to work with the `ftape.o` module on the `tape144` root disk, I think because that module has been stripped of symbols and won't install. So, you now have to make a new kernel with network and ftape support, and if you're running an old kernel, a new `ftape.o`. Read the directions which ship with the `ftape` archive for directions at this stage. Remember that the kernel you compile must support the Ethernet cards on both the Linux machine and the MS−DOS machine.

Copy the newly created kernel image over top of the one on the Net boot disk. Use `/bin/cp`, do not create a boot disk with the ``dd'' command as you would to create a bootable kernel image. Write protect the boot disk, and label it: SAR#1.

Now, mount the `tape144` root disk. I'll assume that the mount point is `/tape144`, to avoid confusion in file names. We need to free some space on it, so delete the following files:

```
/tape144/bin/dialog
/tape144/bin/elvis
/tape144/bin/vi
/tape144/boot/ftape.o
```

Now, create a new file:

```
/tape144/etc/exports
```

which contains the following line:

```
/mnt    msdos(ro)
```

Where ``msdos'' should be replaced with the name or IP# of the MS−DOS machine which has the tape drive installed.

Next, so that you don't have to rely on a name server, add lines to the file `/tape144/etc/hosts` with the names and IP numbers of the Linux and MS−DOS machines. For instance, mine contains the following two lines:

```
128.100.75.114  caliban.physics.utoronto.ca caliban caliban.physics
128.100.75.111  ariel.physics.utoronto.ca ariel ariel.physics
```

Now, there's some sort of problem with the inetd configuration. We have to put the full path name of the rsh daemon in it. Change line 19 of `/tape144/etc/inetd.conf` to read:

```
shell   stream  tcp     nowait  root    /usr/etc/tcpd   /usr/etc/in.rshd
```

Add local net routing information to `/tape144/etc/rc.d/rc.inet1` to enable the MS−DOS machine to use the network. The format of this depends on your network configuration, you can just copy the appropriate format out of your Linux `/etc/rc.d/rc.inet1`. For my network, the lines that have to be added are:

```
/etc/ifconfig eth0 128.100.75.111 broadcast 128.100.75.0 netmask 255.255.255.0
/etc/route add −net 128.100.75.0 netmask 255.255.255.0
```

The IP# in the ifconfig entry is that of the MS−DOS machine.

Now, copy this file into `/tape144/etc/rc.d/rc.inet1−l`, and change the IP# in the new file to reflect that of the Linux machine rather than the MS−DOS machine.

Next, clip out lines 3 to 11 of `/tape144/etc/rc.local`. That's an if statement which executes the `rc.inet*` files. We don't want this to happen during the bootup.

Create a new file: `/tape144/root/.rhosts` containing the line:

```
linux root
```

where, again, ``` ``linux" ``` is replaced with the full machine name (including domain) or the IP# of the Linux machine.

Fill in the password field in `/tape144/etc/passwd` for the root login to keep people from logging onto the MS−DOS machine while you're doing the backup. You can do this by copying the corresponding field from your Linux machine's `/etc/passwd file`.

Copy /usr/bin/rsh into /tape144/usr/bin.

Copy the following files from `/usr/etc` into `/tape144/usr/etc`:

```
in.rshd
rpc.mountd
rpc.nfsd
rpc.portmap
services
tcpd
```

Create a new script, `/tape144/bin/tapesetup`, which consists of the following: (change ``` ``linux" ``` to reflect your Linux machine name).

```
#! /bin/sh

/bin/sh /etc/rc.d/rc.inet1
/bin/sh /etc/rc.d/rc.inet2

/bin/mount linux:/nfs /mnt
/bin/insmod /mnt/ftape.o
```

Note that newer kernels will not require the insmod line.

Next, create another new script, `/tape144/bin/msdosset`, as follows: (change ``` ``linux ```" to reflect your Linux machine name).

```
#! /bin/sh

/bin/sh /etc/rc.d/rc.inet1
/bin/sh /etc/rc.d/rc.inet2

mount linux:/mnt /mnt
/bin/insmod /mnt/ftape.o
```

As above, newer kernels will not require the insmod line.

Create a readable file, `/tape144/root/notes`, which contains this helpful information for use in full recovery:

```
For a full recovery to a trashed hard disk,
boot the Linux machine with the SAR disks #1 and #2
then type the following:

   /bin/sh /etc/rc.d/rc.inet1−l
   /bin/sh /etc/rc.d/rc.inet2

   /usr/etc/rpc.portmap
   /usr/etc/rpc.mountd
   /usr/etc/rpc.nfsd

Next, insert SAR disk #3 and type:

   mount /dev/fd0 /mnt

Create a new mount point, with:

   mkdir /mnt2

and mount your Linux hard disk partition on this point.
You may have to reformat the partition first, if so,
follow the directions in the Linux Installation HOWTO.
The SAR disks contain all the files necessary to do the reformat.

Finally, use disks SAR#1 and SAR#2 to boot up
the MS−DOS machine and run the /bin/msdosset script on that machine.
It will take about a minute to run that script because it is getting
an NSF file from a floppy drive, so be patient.
Now, recover the tape to /mnt2 on the Linux machine.
```

If you are using the backup scripts you will need to copy 'afio' into the `/tape144/local/bin` subdirectory. It is not necessary to have the rest of the backup script files on the recovery disks, an archive can be recovered using only 'afio' and 'gzip'.

I was unable to use the backup scripts as they come shipped. The tape archive appears to build cleanly, but it is unrecoverable. I found that removing the block size and conversion statements fixed it. Here is the patch to the ``netbackup'' script. Apply this patch to the Linux machine's hard disk copy of 'netbackup' as well as to the copy on the SAR disks.

```
*** netbackup.orig      Mon Jan  9 17:22:32 1995
−−− netbackup   Mon Jan  9 17:23:25 1995
***************
*** 35,41 ****
                  "'mknod", devname, "p'");
      exec ("su −", USERNAME, "−c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "of=" REMOTE_DEVICE, "obs=20k", "conv=sync\"",
                        "<", devname,
                "'&"
         );
−−− 35,41 −−−−
                  "'mknod", devname, "p'");
      exec ("su −", USERNAME, "−c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "of=" REMOTE_DEVICE, "\"",
                        "<", devname,
                "'&"
         );
***************
*** 50,56 ****
                  "'mknod", devname, "p'");
      exec ("su", USERNAME, "−c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "if=" REMOTE_DEVICE, "ibs=20k", "conv=sync\"",
                        ">", devname,
                "'&"
         );
−−− 50,56 −−−−
                  "'mknod", devname, "p'");
      exec ("su", USERNAME, "−c",
                  "'rsh ", REMOTE_HOST,
!                       "\"dd", "if=" REMOTE_DEVICE, "\"",
                        ">", devname,
                "'&"
         );
```

You have now finished your SAR disk #2. Write protect it.

Next, mount a clean, formatted disk (create it with `fdformat` and `mkfs`). Copy the `ftape.o` file onto it, and label it SAR#3. For some reason things go badly if you write protect this disk, so leave it write−enabled.

On the Linux machine, create a new directory for NFS file serving. I made a directory:

```
/nfs
```

Put the `ftape.o` (unstripped, about 500+ kB) into this subdirectory. Create an entry in your Linux's exports file `/etc/exports`:

```
/nfs    msdos(ro)
```

Note that all files in your NFS directory and it's subdirectories are not secure. Somebody else could boot the MS−DOS machine into Linux with his own boot disks and mount this directory, so be certain that you don't put anything sensitive in your NFS subdirectory.

Restart your NFS daemons, `rpc.mountd` and `rpc.nfsd`. They don't seem to take kindly to a `SIGHUP` restart, so kill them and reinvoke them. If you're not activating these daemons in your `/etc/rc.d/rc.inet2` you might want to do so now.

OK, now we're all set to back up and recover. To make a full backup from the Linux machine, boot the MS−DOS machine with SAR#1. When prompted for the second disk, load SAR#2. Log in as root, and execute the script: `/bin/tapesetup`. Log out of the MS−DOS machine. If you're using the backup scripts, the netbackup command will now work. You can also use the ``` ``−f msdos:/dev/ftape'' ``` switch on GNU `tar`, `cpio`, or `mt`, and make your backup this way. If you have a backup program, which is only capable of writing to a local file, do the following. Assume that the backup program is called ``` ``localbackup'' ``` and writes to the file represented by its command line argument:

```
mknod /tmp/tapepipe p
rsh msdos dd of=/dev/ftape < /tmp/tapepipe &
localbackup /tmp/tapepipe
```

when it's done, delete `/tmp/tapepipe`.

Recovering to a live Linux machine: the `netbackup` script, `tar`, `cpio`, and so on will all work without special actions on the part of the operator. If you have a local recovery program which recovers from a file, do this:

```
mknod /tmp/tapepipe p
rsh −n msdos dd if=/dev/ftape >> /tmp/tapepipe &
localrecovery /tmp/tapepipe
```

and delete `/tmp/tapepipe` when you're done.

Notice that I'm using '`rsh`' to the root user on the MS−DOS machine. This works with a correct `.rhosts` entry. The configuration on the '`tape144`' disk allows `rsh` to root, but does not allow `telnet` or `rlogin` to root, logins are restricted to the console. This is good for security.

If you are worried about a root `.rhost` file, you can create a new user on SAR#2, ``` ``tapeuser'' ```, with permissions to operate the tape drive but not the disks (create a new group and put tapeuser in that group, then `chown` and `chmod` the files `/dev/rft*` and `/dev/nrft*`). Your backup program then has to know to `rsh` to that username rather than to root. Of course, now there must be an `.rhosts` file in `~tapeuser` on SAR#2. For my own use, I have chosen this course, rather than a root `.rhosts`.

Finally, the directions for a complete recovery to a trashed hard disk. This assumes that the Linux partition is completely unrecoverable. If necessary, reformat that partition as described in the Linux Installation HOWTO. Boot the Linux machine from SAR disk #1. When prompted, insert disk #2. Now, follow the directions in the file `/root/notes` (this was `/tape144/root/notes` when it was mounted on your Linux machine). Once both machines have been booted up, run the recovery routine you need. If you are running the backup scripts you can do it as follows:

1. change directory to the mount point of the hard disk partition which you will be recovering.
2. if any mounted volumes are on the backup, and you want to recover them, create the mount points within the hard disk partition and mount the volumes.
3. Enter the command:

   ```
   rsh −n msdos dd if=/dev/ftape | afio −i −v −Z −c 1024 −
   ```

or

```
rsh −n −l tapeuser msdos dd if=/dev/ftape | afio −i −v −Z −c 1024 −
```

or

```
mknod /tmp/backpipe p
rsh −n msdos dd if=/dev/ftape >> /tmp/backpipe &
afio −i −v −Z −c 1024 /tmp/backpipe
```

This reads the tape on the remote machine, writing the result to stdout, where `afio` picks it up. The '`−i`' switch tells it to recover the files relative to the current working directory (which is the root of the hard disk partition). '`−v`' is verbose, listing the files as they are recovered. '`−Z`' tells afio that this is an archive of individually compressed files. '`−c 1024`' tells it to use a 5 MB streaming buffer to avoid a lot of tape rewinding.

---

3. Notes

The commands listed in the `/tape144/root/notes` file could be run from a script. When I tried, I got rpc setup errors. I suspect it was just that the commands were run too quickly, and the portmapper hadn't properly installed itself. I found that typing the sequence in manually worked fine, so I've recommended that.

I think this setup is secure. Note that somebody can still get access to all your files if they go to the tape drive and pull the tape out before you get there, then then read the tape themselves. People with very sensitive data might consider encrypting the stream from the archiver. Archive to standard output and pipe the output to the encrypter, and redirect the output of the encrypter to append to the named pipe `/tmp/tapepipe` as described above. Note that errors in the recovery process will result in all files after that point being unrecoverable, as the entire archive is now a single DES−encrypted stream. It is possible to use options on afio to send each file in the archive first through gzip, then into an encryption program like des, but note that this compressing first does provide a fair amount of known plaintext for determined code breakers to work with, so a better approach might be to skip the gzip step and simply encrypt it with des, at the expense of significantly more tape area. Needless to say, DES encrypted files don't compress.

The `rc.inet1` directions I've included will allow only communication with the local network, not the rest of the world through a gateway.

During a full recovery to a blank hard disk the SAR disk #3 provides `ftape.o` to the MS−DOS machine through NFS. This is because some old versions of the `ftape` module can't control some tape drives when there is a disk mounted in the floppy drive. With newer kernels, the entire NFS stuff can be omitted.

This is very important. *****TEST***** the SAR recovery procedure. I did, but don't leave anything to chance. Make sure that you can recover at least one file from your tape to the Linux machine using only the SAR disks (i.e. without mounting the hard disk). If you can't reboot the Linux machine without inconveniencing a lot of users, change the setup information on the SAR disks to assign the ``` ``linux ``` '' identity to another MS−DOS machine and then boot the two MS−DOS machines into Linux to make sure everything works. Then, change the ``` ``linux ``` '' identity back again so that you have usable SAR disks.

---

4. Copyright

Copyright Jan 10, 1995 by Christopher Neufeld (neufeld@physics.utoronto.ca)

Modified Feb 6, 1996.

Modified Aug 5, 1997.
