Darwrap documentation

Contents

1 What is darwrap ?

Darwrap is a program to automate your backups: given a backup description in an XML file, it will automatically call the archiver, name the backup files, encrypt and sign them, create parity information, and send them to their destination.

Dar[1] is used to create the original archive. Darwrap has been designed to use destinations that are always reachable, like hard disks or remote servers. It has not been designed to do backup on e.g a cd-rom or a tape, but it might be possible to implement it anyway.

Darwrap also handles the rotation scheme of your backup. The tower of hanoi strategy is supported.

If your backup crashes in the middle, darwrap can resume it where you left it. This can really be useful when you want to send your backup to a remote place with an unreliable network connection.

2 Creating your configuration file

The first thing to do once you have installed darwrap is to create a backup configuration: a XML[2] file that describes how your backup should be done.

Each backup configuration file contains several backup profiles, each describing a specific type of backup, and global variables, applying to all backup in the configuration file.

2.1 A typical configuration file

To make it easier to follow, I will first show an example configuration file, and then explain the syntax and options as we go on.

```
<darwrapConfig>
  <global>
<spoolpath>/home/sebastien/tmp/backup.spool</spoolpath>
  </global>
  <!-- This is a comment which should be ignored by the parser -->
  <profile name="mount-passphrase-test">
<destination type="mountpoint">
```

[1]http://dar.linux.free.fr/
[2]http://en.wikipedia.org/wiki/XML

```
  <mountpoint>/media/disk/</mountpoint>
  <path>/media/disk/tmp/backup-test</path>
  <doumount>false</doumount>
</destination>
<source>
  /home/sebastien/rep/darwrap
</source>
<darOptions>
  <splitSize>100M</splitSize>
  <excludedFileList>
    <file>test</file>
    <source>/home/sebastien/rep/darwrap/test/backup.exclude</source>
  </excludedFileList>
</darOptions>
<cryptoOptions>
  <key>
    <passphrase>Hello World</passphrase>
  </key>
  <gpgOpts>--personal-digest-preferences SHA512</gpgOpts>
  <secring>/home/sebastien/rep/darwrap/test/test.secring</secring>
  <signring>/home/sebastien/rep/darwrap/test/test.keyring</signring>
  <signKey>5CB69F8D</signKey>
</cryptoOptions>
<parityOptions>
  <redundancy>5</redundancy>
</parityOptions>
<rotationScheme type="TowerOfHanoi">
  <narchives>10</narchives>
  <fullLevel>5</fullLevel>
</rotationScheme>
  </profile>
  <profile name="s3-public-test">
<destination type="s3">
  <bucket>john-smith</bucket>
  <location>backup</location>
  <keyfile>/home/sebastien/.awssecret</keyfile>
  <optStr>--insecure-aws --region=eu</optStr>
</destination>
<source>
  /home/sebastien/rep/darwrap
</source>
<rotationScheme type="TowerOfHanoi">
  <narchives>5</narchives>
  <fullLevel>4</fullLevel>
</rotationScheme>
<cryptoOptions>
  <key>
```

```
        <keyring>/home/sebastien/rep/darwrap/test/test.keyring</keyring>
        <keyId>5CB69F8D</keyId>
      </key>
      <gpgOpts>--personal-digest-preferences SHA512</gpgOpts>
      <secring>/home/sebastien/rep/darwrap/test/test.secring</secring>
      <signring>/home/sebastien/rep/darwrap/test/test.keyring</signring>
      <signKey>5CB69F8D</signKey>
    </cryptoOptions>
      </profile>
    </darwrapConfig>
```

As you can see, the content is enclosed into the `darwrapConfig` tag. Inside, you can find the global variable declaration between the `global` tags, and the description of one or more profile, enclosed in `profile` tags.

2.2 Global variables

Each variable is declared between a tag containing their name. There are two global variable, one of which *must* be declared.

2.2.1 spoolpath

Contains the *absolute* path to the backup spool path. The backup spool path is where all the temporary files of all your backups will go. It is used, among other things, to store index files keeping track of your backup's rotation, and also to store a copy of your backup files before they are transferred to their final destination. It is also used at restore time: this is where all the archives are downloaded.

This variable *must* be specified.

2.2.2 syspath

Contains the system path to use looking for executable. The syntax is the same as for the `PATH` bash variable.

The default is `/bin:/usr/bin:/usr/local/bin:/sbin:/usr/sbin:/usr/local/sbin`

2.3 Profile: basic structure

A profile describe a certain type of backup. Each profile *must* have a name, specified with the `name` attribute.

Each profile must contain several section:

- A destination (subsection 2.4) section, describing where to send the backup
- A source (subsection 2.5) section, describing what to backup
- A rotation scheme (subsection 2.6) section, describing how the backups should be rotated

Additionnaly, other sections can be specified:

- A dar options (subsection 2.7) section, describing options to pass to the dar command

- A crypto options (subsection 2.8) section, describing how to encrypt and sign the backup
- A parity options (subsection 2.9) section, describing how to encode parity information with the backup

2.4 The destination section

The destination section is enclosed in `destination` tags. It *must* have a type. Each type admits a different set of parameters.

2.4.1 The mountpoint type

If the destination type is "mountpoint", the backup files will be copied to an external drive that will be mounted using the mount command.

Inside the destination section, two variables are required: the mount point and the destination path. See the example configuration for how to specify these.

An optionnal variable, `doumount` can also be used to specify whether the device is unmounted once the backup has been done. If not specified, the device is unconditionally unmounted.

2.4.2 The S3 type

Darwrap supports backup to Amazon S3[3] . If the S3 type is specified, files will be copied to your S3 bucket, using tim kay's aws[4] program.

Three variables are required:

- The name of the bucket you want to put the files in
- The location of your files inside the bucket: this is simply prefixed to your file's path, separated with a slash
- The file containing S3's secret key

You can give additionnal option to `aws` using the optional `optStr` variable.

Note that the bucket will be created if it does not exist.

2.4.3 The rsync type

If this is specified, the backup files will be copied to a remote site using rsync[5] and the SSH[6] protocol.

Inside the destination section, four variables are required:

- `host`: The SSH server to send to, e.g `foo.example.com`
- `sshUser`: The SSH user to login as.

[3] `http://aws.amazon.com/s3/`

[4] `http://timkay.com/aws/`

[5] `http://rsync.samba.org/`

[6] `https://secure.wikimedia.org/wikipedia/en/wiki/Secure_Shell`

- `hostPath`: The directory on the server that will hold the data, e.g `/home/my-backup`. This can be given either as an absolute path, or relative to the SSH user's home directory.
- `sshConfigFile`: The absolute path to an SSH configuration file giving all the details on how to connect to the remote site. This must be in the format given by `man ssh_config`.

Two variables can be used to give additional options:

- `rsyncOpts`: specifies space-separated options to pass to `rsync`.
- `sshOpts`: specifies space-separated options to pass to `ssh`.

2.5 The source section

The source section contains only the folder to be backed up. It will be backed up recursively. The files to exclude are to be specified in the dar options (subsection 2.7) section.

2.6 The rotation scheme

2.6.1 What is a backup rotation scheme ?

A very simple backup strategy is to copy all your data to some external storage every day, overwriting the copy of the previous day in the process.

This has several shortcomings:

- If you have more space on your destination medium than the backup takes, you end up with unused capacity.
- While doing your copy, you are overwriting the data of the previous copy. If something happens during the backup, you have no complete copy to use to restore your backup.
- If a virus corrupts your data, so is your backup, which ends up being useless.

A better backup strategy would be to copy all your data to some external storage every day, but without overwriting the previous copy. Whenever you run out of space, just buy another disk.

While this corrects all the above problems, this has the obvious disadvantages of cost and storage space for all the disks you use.

The strategy used by darwrap is to sometimes do a full copy, that can be used alone for restoration, and a differential copy, which only contains the files that have changed with respect to another previous backup. To restore a backup, first restore the full copy, then all the differential copies.

Sometimes, you still need to remove some of the copies to make room for others (do you really care about a snapshot taken 101 days ago if you have one taken 100 days ago ?).

The strategy to decide the frequency of full and differential backup, as well as which are deleted to make room for others is called the rotation scheme.

2.6.2 The Tower of Hanoi rotation scheme

At the moment, darwrap only supports the tower of hanoi rotation scheme. Although it would not be difficult to add support for another rotation scheme, you should really look into that one to see if it fits your needs first.

The basic idea is to keep only the snapshots aged of e.g 32, 16, 8, 4, 2, 1 days. This way, if you store n snapshots, you can get back in time up to 2^{n-1} days.

Each snapshot is assigned a level from 1 to n. A snapshot at level m is overwritten every 2^{m-1} day. Snapshots of higher level take precedence, i.e on day 8, the snapshot at level 4 will be replaced rather than the one at level 3.

Full backups are always made at a higher level than differential backup. The exact level at which they are made is set by the `fullLevel` parameter. Of course, at first some full backup will have to be made at lower levels. Differential backups are made with respect to the latest backup with a higher level.

You should look at the code if you need more precision on the exact scheme used by darwrap.

What you should remember is that a rotation scheme is described by its type and by two parameters:

- `narchives`: the number of snapshots to use
- `fullLevel`: the level at which we do full backups, must be between 1 and `narchives - 1`. In case `narchives` is 1, it can only be 1.

2.6.3 How to choose the number of archives and the full level ?

The number of archives to choose depends on how much space you have available. I would *not* recommend setting it to 1, as it would break atomicity (see above). Any other value is reasonnable. You must know that wherever you set your full backup level, your destination can at some point contain a maximum of two full backups. You should experiment to know what the size of a full backup is for your system. Since archives are compressed, it will typically be less than the space you backup, but the exact percentage varies. Differential backups are much less expensive, but also depend on what you do with your system.

The full backup level is to be set very high if you do not want to have a lot of full backups stored on your destination media. The problem is that it will increase the time it takes to restore your backup, as a lot of differential backups will need to be downloaded and restored. You should also keep in mind that doing a full backup might increase the load on your system for a long time. The higher the backup level, the less frequent the full backup will be.

2.7 Dar options

2.7.1 Slice size

Darwrap can cut up the archive it creates into slices of a given size. They can be specified using the `splitSize` parameter. Cutting an archive into slices is especially useful if you want to transfer them over the network, or burn them to a CD or DVD.

2.7.2 Excluded file list

The list of files to exclude can be specified in two ways: first directly in darwrap's configuration file, using the `file` tag (one file per tag), or through an external file, containing the absolute path to each file, one file per line, specified using the `source` tag.

If a directory is specified, all its content will be excluded from the backup.

If you specify wildcard characters (e.g `*` or `?`) in your filepath, they will be expanded.

2.7.3 Additional options

The `optStr` tag can be used to specify a list of additional options to pass to darwrap.

2.8 Cryptography options

Darwrap offers you the opportunity to encrypt and sign your backups. I advise you to seriously consider that possibility. If you encrypt your backups, they will be protected if the medium containing them is stolen or compromised. If you sign them, you will be sure they haven't been altered while in transit, or while stored.

2.8.1 General options

The `gpgOpts` tag can be used to give additionnal, space-separated options to GPG. In the example, it is used to mandate the use of a strong digest algorithm for signing.

2.8.2 Signing options

If you want to sign your archives, you need to specify three parameters

- `secring`: the path to the keyring containing the private key to use for signing. To make automation possible, this key *should not be protected by a passphrase*
- `signring`: the path to the keyring containing the public key to use for signing
- `signKey`: ID of the key to use for signing

2.8.3 Encryption options

All encryption-related options are given through the `key` tag. Both asymmetric[7] and symmetric[8] encryption are supported. In short, if you use asymmetric encryption, different keys are used to encrypt and decrypt your backup, whereas if you use symmetric encryption, the same key is used.

I recommend you to use asymmetric encryption: this way you can backup an insecure machine using the public key, and keep the private key on a secure machine.

Darwrap uses GnuPG[9] for all cryptography tasks.

[7] `http://en.wikipedia.org/wiki/Asymmetric_encryption`
[8] `http://en.wikipedia.org/wiki/Symmetric-key_algorithm`
[9] `http://gnupg.org/`

Asymmetric encryption The two following parameters are used:

- `keyring`: the path to the GPG keyring containing the public key
- `keyId`: the ID of the public key to use

Symmetric encryption A passphrase should be specified. It can either be given directly in the configuration file using the `passphrase` tag, or it can be read from a file specified with the `passphraseFile` tag.

2.9 Parity options

Darwrap lets you add some redundancy to your backup, using parchive[10] . This can be useful if you are writing files to an unreliable medium, such as a CD, but it can also be used to check your file's integrity.

You can specify the percentage of redundancy you want using the `redundancy` tag.

3 Running darwrap

3.1 Creating a new snapshot

Now to run one of the backup profile you just created, just do:

```
$ darwrap s3-public-test /path/to/config.xml new
```

You might want to run it with root privillege depending on what you want to backup. The first argument is the name of the profile, the second is the path to the configuration file.

The command will take a backup snapshot in the spool path (subsubsection 2.2.1) you specified previously. Once finished, it will let a `status.xml` file behind, keeping track of the backup options that were used, and of the rotation status.

If you re-run the same command, another backup snapshot will be taken, possibly overwriting a different snapshot depending on your rotation scheme setting.

In short, you only have to schedule the above command to run e.g every day in your crontab[11] and you are (almost) done.

Note that if you change the destination or rotation scheme settings in your configuration file, darwrap will refuse to continue the same cycle and issue an error message: you will have to start everything from scratch. See below.

3.2 Resuming an interrupted backup

If your network connection is unreliable and you are sending your backup to a remote filesystem, darwrap might fail to transfer your files. Still they are still there, simply waiting to be transferred, no need to start the backup all over again. To resume the backup, just use the following command:

```
$ darwrap s3-public-test /path/to/config.xml continue
```

[10]`http://parchive.sourceforge.net/`

[11]`http://en.wikipedia.org/wiki/Cron`

This must be done once a backup started with a `new` command crashed. If you try to do a `new` on a backup that was already started, an error message will be issued. This is done so that the user is aware that something went wrong, and that his backup must be continued.

3.3 Resetting a backup cycle

If you have changed something in your configuration file, and darwrap does not want you to continue your backup cycle normally, or you simply want to start all over again, simply do:

```
$ darwrap s3-public-test /path/to/config.xml reset
```

This will set your status as if nothing ever happened, and you can start from scratch. Your old backup tapes will be removed once the new backup cycle has started.

If you did this accidentally, you can remove the status file and then issue a `new` or `continue`. Your old status file will be fetched back from your destination media. In that case, darwrap may need your private key if you used asymmetric encryption. Specify its keyring using the `--secring-path` option.

3.4 Getting darwrap's status

Once you have started a backup, you can get progression information by doing:

```
$ darwrap s3-public-test /path/to/config.xml status
```

This will tell you whether darwrap is doing a full or differential backup, and at what stage it is (creating the archive, crypting the files, sending the files...)

3.5 Preventing a backup from running

Running a backup can take a lot of system resources. If you are going to run some cpu-intensive process (like watching a movie) and do not want your backup to run during that time, you can use the `block` action. For example:

```
$ darwrap s3-public-test /path/to/config.xml block
```

Will prevent the `s3-public-test` profile from being run. If `new` or `continue` is now called with that profile, an error message will be printed. To enable the process to run again, use `unblock`, e.g:

```
$ darwrap s3-public-test /path/to/config.xml unblock
```

Note that you cannot block or unblock a profile while it is running.

3.6 Killing the current process

If a backup process is running and taking all your resources, you can simply stop it using the `kill` command. For example:

```
$ darwrap s3-public-test /path/to/config.xml kill
```

This will kill the process running the `s3-public-test` profile. You can then proceed to e.g block (subsection 3.5) the backup if you do not want it to run again for a while.

3.7 The darwrap-continued daemon

To automatically continue your unfinished backup, you can use the `darwrap-continued` utility. It simply tries to resume your backup e.g every ten minutes. To start it, run:

```
$ darwrap-continued profilelist 10
```

The first argument is a configuration file, whose content looks like this:

```
mount-passphrase-test /home/sebastien/now/darwrap/test/config.xml
neo-usb /home/sebastien/backup/config.xml
```

Each line contains a profile name, followed by a space and a path to the configuration file describing that profile.

The second argument is the time (in minute) between each continuation try. If the backup has finished correctly, nothing will be done.

Normally, only one profile is resumed at a time. If you want `darwrap-continued` to allow several resume processes at the same time, you can use the `--multiple-resume` switch.

4 Restoring your backup

4.1 The automatic way

Of course, darwrap can also restore your backups. Here is a quick reference on how to do it.

First, make sure that you have darwrap, your darwrap configuration file, and your various keys (gpg secret key, aws secret key...) available at the path indicated in your configuration file. This is the only data that darwrap will need in order to restore your backup. Things like the status file or the backup spool are not necessary but will somehow speed the process up.

Note that your configuration file has been saved on the backup destination. You can always download it from there if you have lost it.

4.1.1 Listing the archives

To see which archives are available (and at what time they have been created), use the `list` action, e.g:

```
$ darwrap s3-public-test /path/to/config.xml list
```

This will show you which archives you can restore from, whether it is a full or differential one, and the time at which each one has been created. This can help you choose from which date on you would like to restore.

4.1.2 Restoring your content

To restore your content, use the `restore` action. You can use several command line options with it:

- `--date=yyyy[MMddhhmm]`: tell darwrap you want to restore the content at the state it was no later than the date given. If this is not given, the most recent possible state will be chosen. The date format is: the year's 4 digits, followed (optionally) by the month's number two digits (from 01 to 12) followed (optionally) by the day's number two digits (from 01 to 31) followed (optionally) by the hour's two digits (from 00 to 23), followed (optionally) by the minute's two digits (from 00 to 59).

- `--dest-root=RESTOREDEST`: tell darwap where to restore your content. If this is not given, this will default to the source path in the configuration file.
- `--to-restore=RESTORESRC`: tell darwrap what to restore. This must be a path relative to the root of what you saved. For example, if you have made a backup with source `/home/john/` and you only want to restore `/home/john/.emacsrc`, you should use `--to-restore=.emacsrc`.
- `--dar-opts=DAROPTS`: specify additionnal options to pass to `dar`, separated by spaces. By default, no additionnal option is given. For example, if you do not want ownership to be restored (it cannot be done if you are not running as root), the `-O` option could come in handy.
- `--secring-path=SECPATH`: if you use assymmetric encryption, you must give the path to your secret keyring containing your private key. Darwrap will ask you once for the passphrase and then remember it.

As an example, let's say you have backed-up `/home/john/`, and you want to restore the directory `/home/john/important/pictures`. However, you do not want it to overwrite the existing one, so you want to restore it in `/tmp/pict/`. Since a virus corrupted your files on April 1st 2012, you want to restore files before that date. You should run the following command:

```
$ darwrap --date=20120331 --dest-root=/tmp/pict \
    --to-restore=important/pictures s3-public-test /path/to/config.xml \
    restore
```

4.1.3 Cleaning up

You can run `restore` several times. Darwap will keep the archives it already downloaded to speed up further restore. Once you are sure you have restored everything you need, you can cleanup the backup spool directory. To do that, run:

```
$ darwrap s3-public-test /path/to/config.xml restore-clean
```

4.2 The manual way

If for some reason you do not trust darwrap, or you don't have darwrap installed and you cannot download it, you always have the possibility to restore the dar archives manually. Here is how.

4.2.1 Restoring a few files

If you only want to restore a few files, you probably don't need to fetch all your backup: you can use `dar_manager` along with an index file, and only download the archives that you need.

First, fetch your backup index file from your media:

```
$ cp /removable/medium/index.profilename.path.date.dmd .
```

This file might have a gpg extension if you use encryption. In that case, decrypt it before carrying out the following steps. Also, check your archive's parity using par2verify.

Then do something like:

```
$ dar_manager -B index.profilename.path.date.dmd -r relative/path/to/file
```

to restore /relative/path/to/file . If dar asks you for archives, fetch them from your backup medium. Do not forget to always fetch the first slice (the one whose name ends with .1.dar)

4.2.2 Restoring a directory tree

If you want to restore a whole directory tree, you have no choice: you will need to download the full snapshot, and the relevant differential ones you made afterward.

First download all the slices of the full backup closest to the date of the state you want to restore. The date is indicated in the file name (five first numbers: year, month, day, hour, minutes; separated by dots)

Then download all the slices of the archive whose date is the closest to the state you want to restore. Note the level of that archive: the 6th number in the archive name.

Finally download all the archives whose date is between that of of the full backup and that of the archive you previously downloaded, and *whose level is higher than the one you noted.*

Then restore each archive, oldest first, using the following command:

```
$ dar -x \
  profilename.path.year.month.day.hour.minute.type.level.cycleday.archive \
  -R where/to/restore \
  -r -g directory/tree/to/restore
```

5 References

- Tower of Hanoi pattern for backup[12] is a good starting point on this rotation scheme. An algorithm is given for choosing the next tape that is used in darwrap.
- The Tao of Backup[13] gives a lot of very good backup advices in a humorous way.
- Reliable Linux backups[14] is a serious reference on how to do a proper backup of a unix-like system. It describes the problems that are to be solved and the best software to use. Dar is considered very good by the author.
- The official dar tutorial[15] is a very complete tutorial on dar and dar_manager.

[12] http://www.alvechurchdata.co.uk/softhanoi.htm
[13] http://www.taobackup.com/
[14] http://www.halfgaar.net/backing-up-unix
[15] http://dar.linux.free.fr/doc/Tutorial.html