Backing up for portables with Box Backup

SAFER BOX

The Box Backup network backup tool is a simple and secure solution for traveling clients. We'll show you how to get started with this open source tool. **BY KENNETH GEISSHIRT**

If your computer is a laptop, you probably carry it wherever you go. The hard drive is loaded with all your important data – including digital photos, email messages, notes, letters, and spreadsheets. Of course, your home directory is encrypted so no one can get access to your files if your laptop is stolen. Traditional backup software (such as Bacula, Tivoli, and Networker) initiate backups from the server, and the backup intervals are typically very regular. For example, the server might begin the backup at the same time every day.

This scheme does not work well with laptops. A laptop might not be online when the server initiates the backup. Moreover, the server might not even know the IP address of a roving mobile system.

A far better method for laptop backup is to let the client control the backup. Box Backup is such a solution. The client software sends the backup data to the server. The data is sent over an encrypted line using Secure Socket Layer (SSL), and the data is stored encrypted.

Installation

Box Backup is an open source tool with a BSD-style license. Most Linux distributions don't include the Box Backup utility by default. Unless you're using Debian, you'll probably need to compile the tool from source code. Box Backup is easy to install from source. You'll find the source code at the project website [1]. To compile Box Backup for Linux, you'll need SSL (libssl-dev) and Z compression (libzlib-dev).

The Berkeley database (libdb4.5-dev) and GNU Readline library (libreadline5-dev) are also helpful. The Berkeley database stores internal data structures on disk, so the daemons do not have to build up the data structures from scratch after a restart. The Readline library supports editing of command-line input.

```
$ tar xzf boxbackup-0.10.tgz
$ cd boxbackup-0.10/
$ ./configure ↲
--enable-gnu-readline
$ make
```

Box Backup is compiled into two separate parcels – one for the client and one for the server. The parcels are found in subdirectory *parcels*, each of which includes a simple installation script.

To install the client parcel:

```
$ cd parcels/boxbackup-0.10-↲
backup-client-linux-gnu/
$ sudo ./install-backup-client
```

To install the server parcel:

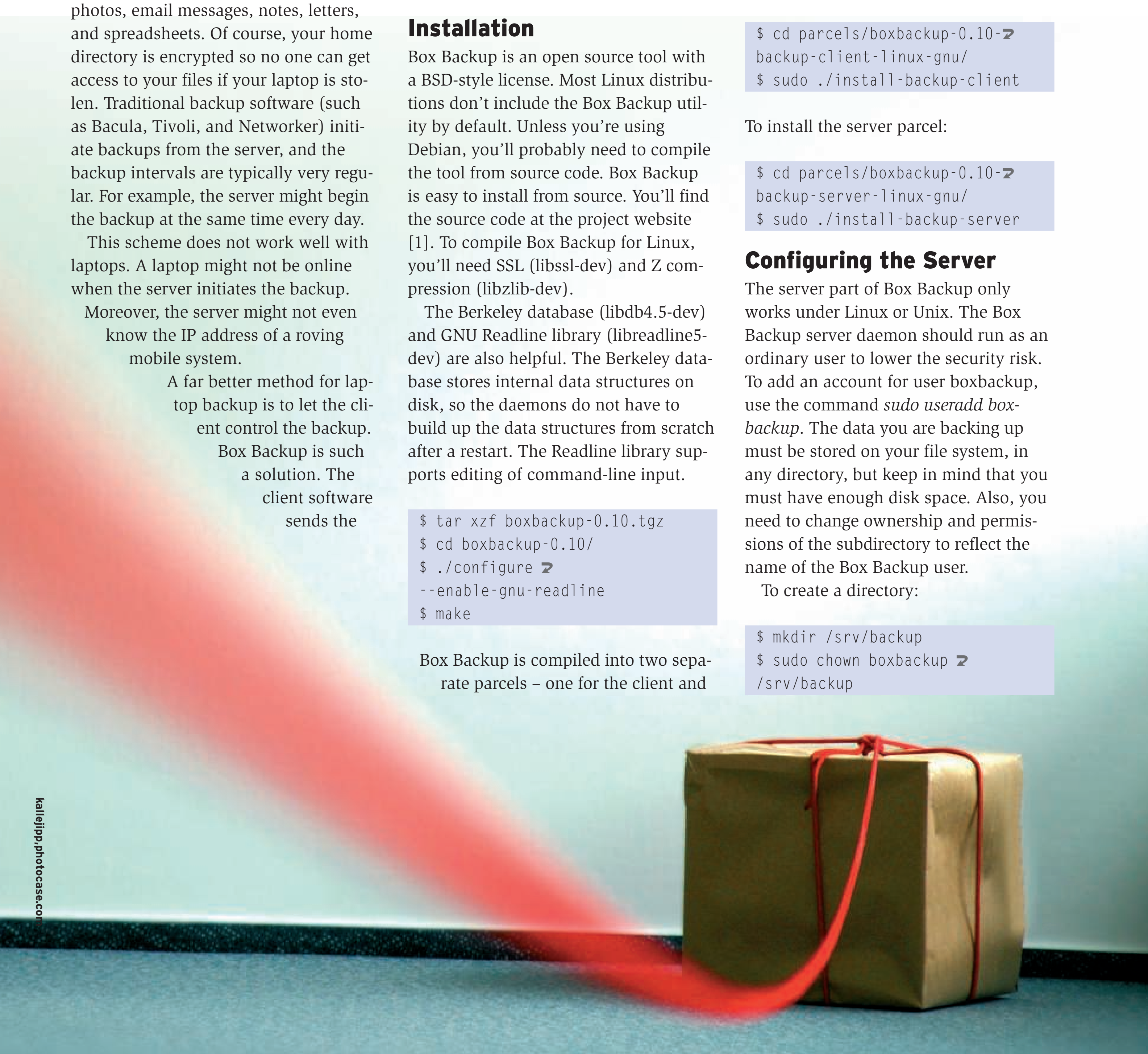
```
$ cd parcels/boxbackup-0.10-↲
backup-server-linux-gnu/
$ sudo ./install-backup-server
```

Configuring the Server

The server part of Box Backup only works under Linux or Unix. The Box Backup server daemon should run as an ordinary user to lower the security risk. To add an account for user boxbackup, use the command *sudo useradd boxbackup*. The data you are backing up must be stored on your file system, in any directory, but keep in mind that you must have enough disk space. Also, you need to change ownership and permissions of the subdirectory to reflect the name of the Box Backup user.

To create a directory:

```
$ mkdir /srv/backup
$ sudo chown boxbackup ↲
/srv/backup
```



kallejipp.photocase.com

Box Backup configuration is managed through configuration files. Through a collection of scripts included with the application, you can generate configuration files.

Box Backup can emulate RAID by spreading the data over any number of disks or volumes. The utility *raidfile-config* is used to generate configuration files for enabling or disabling this RAID feature. The utility *bbstored-config* generates the server configuration file. My server has the hostname *eagle.zigzak.net*, and I work with a block size of 8,192 bytes. The following commands generate the configuration files:

```
$ sudo raidfile-config ↩
/etc/boxbackup 8192 ↩
/srv/backup
$ sudo bbstored-config ↩
/etc/boxbackup ↩
eagle.zigzak.net ↩
boxbackup
```

The configuration files are stored in the directory */etc/boxbackup*. Permissions for the subdirectory must be fixed; the following two commands define permissions:

```
$ sudo chown ↩
-R boxbackup ↩
/etc/boxbackup
$ sudo chmod -R go-rw ↩
/etc/boxbackup
```

Secure Sockets Layer (SSL) is an essential part of Box Backup. Communication between the clients and the server is encrypted with SSL, and the backed up data is also encrypted with SSL. To make SSL work, you'll need to generate and sign a server certificate:

```
$ cd /etc/boxbackup
$ sudo bbstored-certs ↩
ca init
$ sudo bbstored-certs ↩
ca sign-server ↩
bbstored/↩
eagle.zigzak.net-csr.pem
```

The server is now ready.

Before you start the server, you might want to edit the configuration file */etc/boxbackup/bbstored.conf* and enable extended logging to the *syslog*. The command *sudo bbstored /etc/boxbackup/bbstored.conf* starts the server.

Configuring Client

The Box Backup client can run in either *snapshot* or *lazy* mode. Snapshot mode is similar to traditional backup software – the client takes a snapshot of the data and transfers it to the server.

Lazy mode is a bit smarter, and generally recommended. When running in lazy mode, the client transfers data to the server continuously. If a file has not changed after a given time interval (specified in the configuration file), the file is transferred to the server.

This timeout interval ensures that temporary files are not transferred.

Listing 1: Part of the Client Configuration

```
BackupLocations
{
        home-kneth
        {
                Path = /home/kneth
                ExcludeFilesRegex = *.(mp3|MP3)$
        }
}
```

Listing 2: Transferring and Signing Keys

```
(on client) sudo scp bbackupd/1001-csr.pem kneth@eagle.zigzak.net:/tmp
(on server) sudo cp /tmp/1001-csr.pem /etc/boxbackup/ca/clients
(on server) cd /etc/boxbackup
(on server) sudo bbstored-certs ca sign ca/clients/1001-csr.pem
(on server) sudo cp ca/roots/serverCA.pem /tmp
(on server) sudo cp ca/clients/1001-cert.pem /tmp
(on client) scp eagle.zigzak.net:/tmp/{serverCA,1001-cert}.pem .
(on client) sudo cp {serverCA,1001-cert}.pem /etc/boxbackup/bbackupd
```

To generate a configuration file, you can use the *bbackupd-config* utility. The parameters include mode (*lazy*), account number (*1001*), name of the server (*eagle.zigzak.net*), a directory for temporary storage (*/var/bbackupd*), and the directory to back up (*/home/kneth*):

```
$ sudo bbackupd-config ↩
/etc/boxbackup lazy 1001 \
  eagle.zigzak.net ↩
/var/bbackupd /home/kneth
```

Before you start the Box Backup client, you might want to edit the configuration file */etc/boxbackup/bbackupd.conf*, which has many options.

One interesting section is *Backup Locations*, where you can specify the areas of the filesystem you want to back up. Also, you can exclude files from the backup and control exclusion through regular expressions. Listing 1 gives an example from my computer, in which I exclude any MP3 files.

Start the Client

The client starts with the command *sudo bbackupd /etc/boxbackup/bbackupd.conf*. Once you have started the client daemon, you can control it with the *bbackupctl* utility, which executes one of three commands that are specified as command-line arguments. The *terminate* command halts the daemon, and the *reload* command reloads the configuration file. Finally, *sync* forces the client to synchronize (back up) its files with the server. This command is only usable in the snapshot mode.

Creating an Account

In Box Backup terminology, a client and an account are the same thing. To let the client send data to the server, you must create an account on the server. The utility *bbstoreaccounts* is the account administration tool.

Accounts are referenced with hexadecimal numbers. Each account has a soft and hard limit. A client is supposed to stay under the disk space defined by the soft limit, but for short periods of time, the client is allowed to use additional space as long as the usage does not exceed the hard limit.

To create a new account (*1001*) with a soft limit of 4GB (*4096M*) and a 10 percent higher hard limit (*4505M*), use the following command (the *0* refers to a configuration in which RAID is disabled):

```
$ sudo bbstoreaccounts -c ↩
/etc/boxbackup/bbstored.conf \
 create 1001 0 4096M 4505M
```

When you generate the configuration file for the client, you also generate a private SSL key.

To establish communication with the server, you must transfer and sign the key. Listing 2 shows the series of commands you must execute to transfer and sign the key.

The *bbstoreaccounts* utility provides many more options. For example, you can adjust the account limit as the system is running and delete accounts.

Key management can be tricky. If you forget one step or put the file in the wrong directory, the Box Backup daemons on either the client or the server will not start.

Verifying and Restoring Files

To verify and restore, use the general utility *bbackupquery*, which runs in either batch or interactive mode. The utility provides a number of commands such as *compare* for comparing (verifying) the files and *restore* for restoring a complete directory tree.

The *restore* command takes two arguments: the directory name in the backup and the directory name on the client.

```
sudo bbackupquery -c ↩
/etc/boxbackup/bbstored.conf \
  "compare -a" quit
sudo bbackupquery -c ↩
/etc/boxbackup/bbstored.conf \
 "restore /home/kneth/Projects ↩
/home/kneth/Projects.backup" ↩
quit
```

During a restore, the Box Backup client daemon (*bbackupd*) must not be running. If the client daemon is running, Box Backup will get confused, and all your data will be erased!

Summary

Unless you're a Debian user, you'll probably need to compile Box Backup. Fortunately, the release rate is so slow you probably won't have to recompile often.

When Box Backup is combined with encrypted home directories and a secure shell (for tunneling SMTP and IMAP), you can provide a safe and secure environment for your mobile laptop users. Although the server must run on Unix or Linux, a Windows version of the Box Backup client is available.

Because Box Backup pushes its data to the server, this innovative backup tool is a perfect solution for laptop users on networks with professional IT support. For some end-user environments, Box Backup might not be the best option because of the complexity associated with transferring and signing encryption keys at a command line. ■

INFO

[1] Box Backup project website: *http://www.boxbackup.org/*

THE AUTHOR

Kenneth Geisshirt is a freelance software developer and tech writer living in a suburb of Copenhagen, Denmark, with his wife and two children. He is a chemist by education and a geek by nature. He has been a free software user, developer, and advocate since the early 1990s.