Encrypted Virtual Filesystems with BestCrypt

Locked Up Data

There is no fun in having your laptop stolen, but thanks to encrypted filesytems at least you can rest assured that the thief will not be able to get access to your personal data. BY CARSTEN SCHNOBER

Whether you want to avoid personal data being sniffed on a multi-user system, or simply take precautions in case you lose the storage media, an encrypted virtual filesystem with transparent access to the encrypted data seems the most convenient solution. The solution – BestCrypt.

Own Kernel Modules

BestCrypt [1] is available in both Linux and Windows versions. As the Linux version of BestCrypt provides its own kernel modules, there is no need to re-compile as of version 2.4.3.

BestCrypt is free for a 30-day trial; an individual license with 12 month update support is available for $49.95 US. Jetico supplies the source code allowing users to revise the program code.

The source code needs to be compiled first. This assumes you have installed **kernel header** files.

You will need *root* privileges to install the *bctool*, the accompanying toolset, the manpage and kernel modules using *make install*. This command also creates the */dev/bcrypt0* through */dev/bcrypt15* **block devices**. To uninstall, you can launch the *./uninstall.sh* script.

Encrypted Containers

Launch BestCrypt with:

```
/etc/init.d/bcrypt start
```

To terminate the program, replace the *start* argument with *stop*. BestCrypt will launch when you reboot, if your kernel allows you to load kernel modules.

BestCrypt stores the filesystem in a file, the so-called container, which can be mounted just like any other device, data containers can be created by non-privileged users. Admins who install BestCrypt, should be aware that users can hide some of their data from *root*.

BestCrypt uses the *Gost, Blowfish, Twofish, Cast, IDEA*, and *DES* encryption algorithms with variable key lengths. The following example uses the Rijndael algorithm [2]. To use a different algorithm, look for the *-a* option in the manpage (*man bctool*).

```
bcnew -s 10M -a rijndael -d
"My Files " private
```

will create a new 10 MByte container called *private* that uses Rijndael encryption and goes by the name of *My Files*. The filesize following the *-s* option can be specified in MBytes (suffix *M*) or KBytes (suffix *k*). Choose this option carefully, as it cannot be changed later. BestCrypt will prompt you for a password with at least six characters. You still need a filesystem where you can store your files for the container. If Windows computers will be accessing the container, you should opt for *-t msdos* or *-t vfat*, but otherwise a Linux filesystem is preferable:

```
bcformat -t ext2 private
```

will format the container in *private* using Ext2FS. The default filesystemis FAT16.

The encrypted filesystem needs to be mounted in a directory, before you can store data in it:

```
mkdir crypted
bcmount private crypted
```

The encrypted filesystem becomes invisible when you enter:

```
bcumount crypted
```

to unmount the filesystem from its current mountpoint, *crypted*.

Bunches of Options

BestCrypt containers not only provide more convenient access than encrypted loop devices, they offer more options.

For example, *bcinfo private* displays the description of the *private* container, as entered when defining the container. *bcpasswd private* allows you to change the password and *bctool add_passwd private* adds an additional valid password, which can be deleted again using *bctool del_passwd private*.

bcreencrypt private -a blowfish will allow you to change the encryption algorithm at a later stage. *fsck* can be used to check the integrity of the container filesystem, just like a normal Ext2 filesystem, however, the *bcfsck private* command is required to call the tool.

BestCrypt will only protect your data until you mount the filesystem. Once mounted, users will still need to specify access privileges to prevent other accessing the encrypted filesystem.

The security of encrypted filesystems is further compromised by the fact that residual data on the swap partition or the */tmp* directory may allow bypassing of the encryption mechanism. ■

GLOSSARY

Kernel headers: *Header files typically make other programs or program segments, such as functions, data types, variables, etc. visible to a program. Thus, the kernel header files contain descriptions of every single kernel function that other programs, particularly the kernel modules, should be able to access.*

Block device: *Block orientated devices exchange data blocks rather than single characters with the operating system (in contrast to character devices). Block devices (these include hard disks (partitions) for example) are easily identified by the fact that the ls -l command writes a "b" at the beginning of the line when displaying the file that represents the device:*

```
brw-r----- 1 root operator 3,
1 Feb 17  2000 hda1
```

INFO

[1] *http://www.jetico.com/*

[2] *http://www.esat.kuleuven.ac.be/~rijmen/rijndael/*