

12 June 2007

By: Marius Nestor, Linux Editor

Encrypted Filesystem in 5 Minutes

The ultimate protection for your files!



Lock
www.secure-it.com

Have you ever dreamed of having the ultimate protection for your computer? Of course you did, especially if you have sensitive files that you don't want anybody to see. Well, your dream can come true with the help of an encrypted filesystem. The encrypted filesystem is one that resides on an encrypted disk or partition. There are many methods to create such an encrypted filesystem, but today I am going to teach you an easy method to use an encrypted filesystem to protect your data. There are also a lot of tools to encrypt your filesystem, all free, but some of them have weaknesses. So, I've chosen the **dm-crypt** (device-mapper crypto target) which provides transparent encryption of block devices with the help of cryptoapi, that can be found in the new Linux 2.6 kernel. We will use a 256-bit AES (Advanced Encryption Standard) encryption, so make sure that your kernel has AES support loaded. Open a console and type:

```
[CODE=0]cat /proc/crypto[CODE=1]
```

I've got the following result:

```
~$ cat /proc/crypto
name      : md5driver      : md5-genericmodule      :
kernelpriority : 0refcnt      : 1type      : digestblocksize : 64digestsize : 16
```

Which is not good! So, if you get the same result, type the following code in order to activate the AES module:

```
[CODE=0]sudo modprobe aes[CODE=1]
```

Now if I type again `cat /proc/crypto` I'll get the following result:

```
~$ cat /proc/crypto
name      : aesdriver      : aes-genericmodule      :
aespriority : 100refcnt      : 1type      : cipherblocksize : 16min keysize : 16max
keysize : 32name      : md5driver      : md5-genericmodule      : kernelpriority : 0refcnt
: 1type      : digestblocksize : 64digestsize : 16
```

Which shows me that the AES module was successfully loaded. You have to install two more tools, *dmsetup* and *cryptsetup*:

```
[CODE=0]sudo apt-get install dmsetup cryptsetup[CODE=1]
```

Now let's load the *dm-crypt* module:

```
[CODE=0]sudo modprobe dm-crypt[CODE=1]
```

To see if the device-mapper has recognized the *dm-crypt* module and added *crypt* as an available target, type the following code:

```
[CODE=0]sudo dmsetup targets[CODE=1]
```

I've got the following result:

```
~$ sudo dmsetup targets
crypt v1.3.0striped v1.0.2linear v1.0.2error v1.0.1
```

Which shows me that *crypt* was added to available targets and I can continue with the encryption process. You need to setup a block device and mount it as an encrypted logical volume. First, let's create a logical volume with *cryptsetup* and bind the block device to it: **MAKE SURE THE PARTITION IS UNMOUNTED AND EMPTY BEFORE YOU TYPE THE FOLLOWING**

```
[CODE=0]sudo cryptsetup -y create securedata /dev/sdb4[CODE=1]
```

In the example above, I've chosen the *securedata* name for the logical volume, but you can choose whatever name you want. And */dev/sdb4* is the partition I've chosen to encrypt, so check first with `sudo fdisk -l` the partition you want to encrypt. You will be asked for a passphrase (twice) so be careful what you type (don't forget it, or you will lose everything on the encrypted partition).

```
~$ sudo cryptsetup -y create securedata /dev/sdb4
Enter passphrase:
Verify passphrase:
```

Then you should verify if the logical volume was created, so type the following code:

```
[CODE=0]sudo dmsetup ls[CODE=1]
```

I've got the following result:

```
~$ sudo dmsetup ls
sesecuredata (254, 0)
```

Now, if you type:

```
[CODE=0]ls -l /dev/mapper[CODE=1]
```

you will see that *device-mapper* created a virtual block device under */dev/mapper*, which is transparently encrypted:

```
~$ ls -l /dev/mapper/total
Ocrw-rw---- 1 root root 10, 61 2007-06-12 16:47 controlbrw-rw---- 1 root disk 254, 0 2007-06-12 17:07 securedata
```

Create an ext3 filesystem on the virtual block device:

```
[CODE=0]sudo mkfs.ext3 /dev/mapper/securedata[CODE=1]
```

Create a mount point under */mnt*:

```
[CODE=0]sudo mkdir /mnt/securedata[CODE=1]
```

Mount the virtual block device:

```
[CODE=0]sudo mount -t ext3 /dev/mapper/securedata /mnt/securedata[CODE=1]
```

And finally, change the owner of the mount point, so you can have full access to the encrypted partition with your username:

[CODE=0]sudo chown yourusername /mnt/securedata[CODE=1]Congratulations! Now you have an encrypted partition to store all your sensitive files. All the data you write to */mnt/securedata* will be transparently encrypted before being written to hard drive, and the whole content will be decrypted on the fly every time you read it. To automatically mount this partition every time you boot-up your PC, add the following line in the */etc/fstab* file:
[CODE=0]/dev/mapper/securedata /mnt/securedata ext3 noauto,noatime 0 0[CODE=1]For easy access to the encrypted partition, you can create a shortcut on the desktop. If you have KDE, right click on the desktop, go to *Create new -> Link to Location (URL)*, enter a desired name for the shortcut (e.g. My Secure Data), add the location, which is obviously */mnt/securedata* and click OK. Now you should have a new icon on the desktop, called *My Secure Data*. Right click on it, go to Properties, click on the question mark icon and select a pretty icon for your shortcut. I'll get back soon with another guide on how to have an encrypted Ubuntu operating system. Until then, test this one. Enjoy!