

By: Dabarti 2008 Linux Editor

[How to Setup an Encrypted Filesystem](#)

Create an encrypted partition for your sensitive files.

People think about encrypting some or all their files for several reasons. Whether they can't depend on physical security to keep their files safe or they're carrying around a portable laptop with sensitive files and they're afraid of it being stolen or who knows for what other reasons. The encryption process will obscure certain information, making it unreadable without a special password or passcode. This article will explain how to setup an encrypted filesystem under Fedora Core Linux, using only Fedora tools. No external tools will require compiling and installing. After following this guide, your Linux system will have a new partition where you can move your sensitive files. This new partition will be encrypted at all times and reading the files in it won't be possible unless the proper password is used. Your current filesystem will be kept intact so don't worry about the possibility of damaging any files on your hard drive. It's not possible.

- First, load the loop blockdevice adaptor by executing the following command:
`modprobe cryptoloop && lsmod | grep cryptoloop`
If everything goes well, this command will list cryptoloop as a loaded kernel module.

- Next, you'll need to choose which algorithm to be used for encrypting the filesystem. To take a look at which algorithms are available on your system, run the command:
`modinfo /lib/modules/2.6.18-1/kernel/crypto/*`
Note that **2.6.18-1** is my current running kernel, which should differ from yours. If you don't know what's the current version of your kernel, you can find it out by running `uname -r`.

- Now you need to create a file block as your filesystem. Its size can vary depending on your needs but it shouldn't overtake the size of the current disk's free space. For this tutorial, I've created a 650MB file block so it could easily be burned onto a CDR. To create the 650MB file block, run the command:
`dd if=/dev/zero bs=1k count=665600 of=/root/secure`

- The next step consists of associating this file block with the encrypt type and setting a password for making it readable and finally creating the ext3 filesystem: (The last character from /dev/loop0 is a zero, not a big o).
NOTE: The first command will ask you for a password. This password will be used for mounting the encrypted filesystem at a later time so don't lose it!
`losetup -e serpent /dev/loop0 /root/securemkfs.ext3 /dev/loop0`

- It's now time to check the encrypted filesystem setup by mounting the partition:
`mkdir /mnt/securemount -t ext3 /dev/loop0 /mnt/secure`
If everything worked out fine, all files stored in "/mnt/secure" directory will be encrypted. After you've moved all the sensitive files to that partition, you should umount and disable it by using the commands:
`umount /mnt/securelosetup -d /dev/loop0sync`

- In order to mount the encrypted filesystem at a later time, run the following commands:
`losetup -e serpent /dev/loop0 /root/secure` (you will be asked for the encrypt password you've set earlier)
`mount -t ext3 /dev/loop0 /mnt/secure`

NOTE: If you enter the wrong encrypt password, the mount will fail and you will have to detach the file using `losetup -d /dev/loop0` and start over. To make things easier, you can make a couple of bash aliases in order to make mounting and unmounting the encrypted filesystem easier by adding these to the `/root/.bashrc` file:
`alias mountsecure='losetup -e serpent /dev/loop0 /root/secure; mount -t ext3 /dev/loop0 /mnt/secure'`
`alias umountsecure='umount /dev/loop0; losetup -d /dev/loop0; sync'`

To use your encrypted filesystem after adding the aliases, you'll only have to run the commands:
To mount it: `# mountsecure` (enter the encrypt password)
To umount and disable it: `# umountsecure`