

29 July 2007

By: Marius Nestor, Linux Editor



Secure your data
www.securecomputing.net.au

Encrypted Ubuntu 7.04

How safe can you be?

Did you ever live with the fear that somebody may break into your system one day and steal your files? Well, those days are over, because you can now have an entire encrypted operating system. For this setup, we used a freshly installed Ubuntu 7.04 with up-to-date software, nothing else installed. But the following guide is supposed to work with your actual Ubuntu 7.04 installation (no reinstall needed). Beware though: if you don't have the

partitions setup like it's shown below, this **will NOT work**. I will NOT be held responsible for any data loss on your hard drive if this process will NOT work for you, so you have been warned: **TRY THIS AT YOUR OWN RISK!** Things needed:- Ubuntu LiveCD- cryptsetup software Here is how your partitions should look like: [CODE=0]/dev/sda1 -> /boot (about 150-200 MB, mine is 150 MB)/dev/sda2 -> swap (double as your computer RAM, mine is 2 GB because I have 1 GB of RAM)/dev/sda3 -> root (/) (should be more than 5 GB, mine is 35 GB)[CODE=1] **WARNING: I have a SATA drive, therefore my partitions are named sda. If you have an IDE drive, then you have to replace sda with hda in the guide.**

STEP 1 – Boot from the LiveCD Insert the Ubuntu 7.04 LiveCD into your optical drive and reboot your computer in order to boot from the CD. When the CD has loaded, open up a terminal (*Applications -> Accessories -> Terminal*) and become root by typing: [CODE=0] sudo su [CODE=1] You will be permanently root from now on (that means you will not have to type *sudo* anymore, until you exit this session). STEP 2 – Prepare the environment and backup the data Let's prepare the system for the encryption process by loading some necessary modules into the kernel. Type, or copy / paste the following lines in the terminal window: [CODE=0] modprobe aes modprobe dm-crypt modprobe dm-mod modprobe sha256 [CODE=1] Go to *System -> Administration -> Software Sources*, check the "Community-maintained Open Source software (universe)" and "Software restricted by copyright or legal issues (multiverse)" options, then click the "Close" button and when you'll be asked to reload the information about software sources, click the "Reload" button. Wait until the Software Source window disappears and then type in the terminal window: [CODE=0] apt-get install cryptsetup [CODE=1] Then let's backup the existing data by creating some temporary folders: [CODE=0] cd /mnt mkdir boot root tmp [CODE=1] Then mount the existing partitions to the newly created folders: [CODE=0] mount /dev/sda1 boot /dev/sda3 root [CODE=1] And now backup the data: [CODE=0] mkdir tmp/root cp -axv root/* tmp/root [CODE=1] This last code will output a lot of text (the files that are being copied), so wait until it stops. It takes about 6-7 minutes (depending on the number of files). STEP 3 – Encrypt the filesystem Good, now that the backup has finished, unmount the drive with: [CODE=0] umount root [CODE=1] And encrypt the filesystem with the following command: [CODE=0] cryptsetup -c aes-cbc-essiv:sha256 -y -s 256 luksFormat /dev/sda3 [CODE=1] **WARNING: All the data will be permanently erased!** Type YES when asked, and enter a strong password (twice). Then type: [CODE=0] cryptsetup luksOpen /dev/sda3 rootmkfs.ext3 /dev/mapper/root mount /dev/mapper/root root [CODE=1] Now let's copy back the data from the temporary folder to the newly created encrypted root partition: [CODE=0] cp -axv tmp/root/* root [CODE=1] Same as above, it will output a lot of text, so wait until it finishes and remove the temporary folder: [CODE=0] rm -rf tmp/root [CODE=1] STEP 4 – Final adjustments The filesystem is encrypted now, but it will not work until you do some final adjustments. Type, or copy / paste the following lines: [CODE=0] mkdir root/boot mount /dev/sda1 root/boot mount /dev/mapper/root root [CODE=1] At this moment, you are "virtually" in your root partition, and you can make modifications to it. Let's begin by installing the cryptsetup software: [CODE=0] apt-get update apt-get install cryptsetup [CODE=1] Then let's add the

necessary kernel modules to the `/etc/initramfs-tools/modules` file, so that they can be loaded at boot time:

```
[CODE=0]nano etc/initramfs-tools/modules[CODE=1]
```

And add the following lines to the end of the file:

```
[CODE=0]aesdm-cryptdm-modsha256[CODE=1]
```

Hit CTRL+O to and then ENTER to save the file. Hit CTRL+X to close the nano editor. You must adjust the `/etc/fstab` file to mount the correct encrypted root partition:

```
[CODE=0]nano etc/fstab[CODE=1]
```

And change the line that looks like this (the UUID is just an example... yours will be different):

```
[CODE=0]# /dev/sda3
UUID=4565t675-6c67-56hg-hg7j-67g5jk00b562 / ext3 defaults,errors=remount-ro 0 1
[CODE=1]
```

To look like this one:

```
[CODE=0]
/dev/mapper/root / ext3 defaults,errors=remount-ro 0 1[CODE=1]
```

So basically, you just replace `(# /dev/sda3 UUID=4565t675-6c67-56hg-hg7j-67g5jk00b562)` with `(/dev/mapper/root)`. Hit CTRL+O and then ENTER to save the file. Hit CTRL+X to close the nano editor. Now you must edit the `/etc/crypttab` file:

```
[CODE=0]nano etc/crypttab[CODE=1]
```

And add the following line at the end of the file:

```
[CODE=0]root /dev/sda3 none
luks,retry=1,cipher=aes-cbc-essiv:sha256[CODE=1]
```

Hit CTRL+O and then ENTER to save the file. Hit CTRL+X to close the nano editor. And now you have to edit the `/boot/grub/menu.lst` file:

```
[CODE=0]nano boot/grub/menu.lst[CODE=1]
```

Search the line that looks like this (the UUID is just an example... yours will be different):

```
[CODE=0]# kopt=root=
UUID=4565t675-6c67-56hg-hg7j-67g5jk00b562 ro[CODE=1]
```

And change it to look like this:

```
[CODE=0]# kopt=root=/dev/mapper/root ro[CODE=1]
```

Hit CTRL+O and then ENTER to save the file. Hit CTRL+X to close the nano editor. Update GRUB with the following command:

```
[CODE=0]update-grub[CODE=1]
```

And check the `/boot/grub/menu.lst` file to see if the entries changed like this:

```
[CODE=0]title          Ubuntu, kernel 2.6.20-16-genericroot          (hd0,0)
kernel          /vmlinuz-2.6.20-16-generic root=/dev/mapper/root ro quiet splash vga=775initrd
/initrd.img-2.6.20-16-genericquietsavedefault[CODE=1]
```

As you can see, I have an extra option at the end of the `kernel` line: `vga=775`. You are not supposed to have or add this option! Just make sure that you have `root=/dev/mapper/root` option. If so, then you can update `initramfs` with the following command:

```
[CODE=0]update-initramfs -u All[CODE=1]
```

WARNING: If you see an error message about "libdevmapper", just ignore it and continue with the guide. Exit the chrooted environment and reboot the system with:

```
[CODE=0]exitreboot[CODE=1]
```

When the system starts, you will see the Ubuntu boot splash, which will disappear after a few seconds and all you'll be able to see is a blinking line on the top left side of your monitor. Now you should type the password you've setup when you encrypted the filesystem and hit ENTER. You will notice that (if you typed the password correctly), the system continues to boot. That's it folks, your whole Ubuntu 7.04 is now fully encrypted!