How to encrypt a partition with DM-Crypt LUKS on Linux

Article Number: 294 | Rating: Unrated | Last Updated: Fri, Jul 13, 2018 10:34 AM

How to encrypt a partition with DM-Crypt LUKS on Linux

DM-Crypt is transparent drive encryption that is kernel module and part of the device mapper framework for mapping physical block device onto higher-level virtual block devices, it uses cryptographic routines from the kernel's crypto api. Long story short, device mapping encryption provided by the kernel "linux" crypto api.

Make sure that you have **at least** one partition with no data in it. If you don't have any partitions available, use parted, gparted or whatever program you like to shrink some of your existing partitions and create a new one.

I'll use partition called /dev/sda3, and our first task will be to overwrite that partition 3 times with random data, that's enough to protect you against forensic investigation. It took me nearly 30 minutes for 20 GB partition to be overwritten 3 times.

```
shred --verbose --random-source=/dev/urandom --iterations=3 /dev/sda3
```

Create cryptographic device mapper device in LUKS encryption mode:

```
cryptsetup --verbose --cipher aes-xts-plain64 --key-size 512 --hash sha512 --iter-time 5000 --use-random luksFormat /dev/sda3
```

You'll be asked the following question:

```
WARNING!

========

This will overwrite data on /dev/sda3 irrevocably.



Are you sure? (Type uppercase yes): YES

Enter passphrase:

Verify passphrase:

Command successful
```

Unlock the partition, here **"root"** is device mapper name, think of it as label.

```
cryptsetup open --type luks /dev/sda3 root
```

We have to create filesystem in order to write encrypted data that would be accessible through the device mapper name (label).

```
mkfs.ext4 /dev/mapper/root
```

Mount the device and transfer all of your data:

```
mount -t ext4 /dev/mapper/root /mnt
```

Unmount and close the device once you are done:

```
umount /mnt
```

```
cryptsetup close root
```

Last but not least, clear the copy and cache buffers:

```
sysctl --write vm.drop_caches=3
```

That was it, simple and straightforward encryption. From now on all you have to do is: unlock, mount, transfer data, unmount and close the device.

If you have couple hours to spare and experiment, feel free to read those pages:

link 1, link 2, link 3, link 4, link 5, link 6, link 7

Protect your **/boot** partition if you want full disk encryption. Everything is written in great details how to do it in the above links.

Post edit: The things get even better as I just learnt that it is possible to burn LUKS encrypted CD and DVD discs.

Instead using drive partition, we will create a file via **dd** and the kernel's random number generator **/dev/urandom** that will fill the initial file with fake entropy.

Create 500MB file that will be used as file system within a single file.

```
dd if=/dev/urandom of=encrypted.volume bs=1MB count=500
```

Just replace the first command in this post **(shred)** with the **dd** one and type the rest commands as is.

Now you can be sure that no one will get past your data that it is burn within the single file which is entire file system in LUKS encryption, just make sure to unmount and close **encrypted.volume** before burning it to the disc.

Posted - Fri, Jul 13, 2018 10:34 AM. This article has been viewed 8308 times.

Online URL: http://kb.ictbanking.net/article.php?id=294