geekyshacklebolt.wordpress.com

How to encrypt USB drives with LUKS

4-5 minutes

Hello readers! Ever thought about the risk of loosing your USB drive having important data? You surely don't want others to get that data without your permission. Right? In this case, encrypting your USB device is a recommended way to keep a security layer. Keep reading for a simple tutorial to encrypt USB drives with LUKS.

What is LUKS?

The *Linux Unified Key Setup* or ***LUKS*** is a disk-encryption *specification* created by **Clemens Fruhwirth** and originally intended for GNU/Linux. Notice the word *specification*; instead of trying to implement something of its own, LUKS is a standard way of doing drive encryption across tools and distributions. The reference implementation for LUKS operates on GNU/Linux and is based on an enhanced version of `cryptsetup`, using `dm-crypt` as the disk encryption backend.

Starting with the tutorial step by step (I am using Ubuntu 18.04 Bionic Beaver)

1. See available filesystems

```
df -hl
```

2. Connect your USB

3. Find out the new connected device

```
df -hl  # in my case it was /dev/sdb1
```

4. Unmount the USB

```
umount /dev/sdb1
```

5. Wipe filesystem from the USB

> **Note:** check the drive **name/path** *twice* before you press enter for any of the commands below. A mistake, might destroy your primary drive, and there is *no way to recover the data*. **So, execute with caution.**

```
sudo wipefs -a /dev/sdb1
```

```
/dev/sdb1: 8 bytes were erased at offset
0x00000036 (vfat): 46 41 54 31 36 20 20 20
/dev/sdb1: 1 byte was erased at offset 0x00000000
(vfat): eb
/dev/sdb1: 2 bytes were erased at offset
0x000001fe (vfat): 55 aa
```

6. Create a LUKS partition

```
sudo cryptsetup luksFormat /dev/sdb1

WARNING!
========
```

```
This will overwrite data on /dev/sdb1
irrevocably.

Are you sure? (Type uppercase yes): YES
Enter passphrase:
Verify passphrase:
```

7. Open the encrypted drive

```
sudo cryptsetup luksOpen /dev/sdb1 reddrive
Enter passphrase for /dev/sdb1:

ls -l /dev/mapper/reddrive
lrwxrwxrwx 1 root root 7 Jul 26 13:32 /dev/mapper
/reddrive -> ../dm-0
```

8. Create a filesystem

I am going with EXT4, you may create any other filesystem as well.

```
sudo mkfs.ext4 /dev/mapper/reddrive -L reddrive
mke2fs 1.42.13 (17-May-2015)
Creating filesystem with 245500 4k blocks and
61440 inodes
Filesystem UUID: 23358260-1760-4b7b-bed5-
a2705045e650
Superblock backups stored on blocks:
32768, 98304, 163840, 229376

Allocating group tables: done
Writing inode tables: done
Creating journal (4096 blocks): done
```

```
Writing superblocks and filesystem accounting
information: done
```

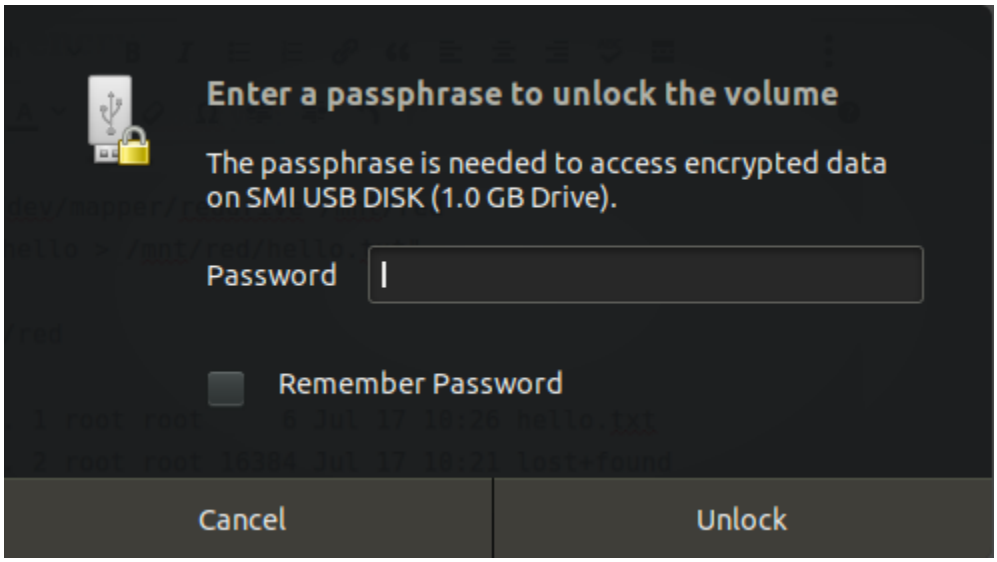
9. Using the encrypted USB

9.1: If you select to mount/unmount your encrypted USB using CLI:

```
sudo mount /dev/mapper/reddrive /mnt/red
su -c "echo hello > /mnt/red/hello.txt"
  Password:
  ls -l /mnt/red
  total 20
  -rw-rw-r--. 1 root root     6 Jul 17 10:26
hello.txt
  drwx------. 2 root root 16384 Jul 17 10:21
lost+found

sudo umount /mnt/red
sudo cryptsetup luksClose reddrive
```

9.2: If you just use GUI to use the encrypted USB as I do then a similar dialog will appear:



Just give your passphrase, save your data in it and eject safely. As simple as that!

Resources

- https://kushaldas.in/posts/encrypting-drives-with-luks.html
- https://miguelmenendez.pro/en/blog/2014/10/encrypt-usb-storage-device-linux-unified-key-setup-luks/

Conclusion

LUKS is wonderful, I recommend using it not just to keep your sensitive data secured but also in general.

Hope you are going to make use of LUKS and suggest your friends as well.

Thanks for reading!
See you in the next post 🙂