computercorrect.com

Auto-mounting a VeraCrypt volume under Ubuntu / Debian Linux

by Andrew D. Anderson · Published 2018-07-17 · Updated 2020-05-19

4-5 minutes

Problem:

You have a VeraCrypt volume you'd like to mount at boot under Ubuntu. You'd like to do this automatically – without providing credentials every time the disk actually mounts. The VeraCrypt GUI has options for mounting volumes at launch, but it still prompts for a password to finish mounting the volume. Forgoing the password via the GUI, even at startup, isn't straightforward…

Solution:

crypttab

The solution can be summed up in a word: *crypttab* – incidentally, this is also the name of the file you'll need to edit. It's located here: `/etc/crypttab` (Right along side fstab, incidentally – which you'll also need to edit to get auto-mounting working.)

So, to make your VeraCrypt volume auto-mount you'll need to know what volume it is. If it's a file, you'll need to know where it's located. If it's a block device or partition – you'll need to know which it is.

You merely need to add a single line (per volume you wish to auto-mount) to your crypttab file. The general format is:

```
volume_name /path/to/volume password tcrypt-
veracrypt,tcrypt-keyfile=/path/to/keyfile
```

- The volume name is one you make up. (It gets added to /dev/mapper/ – and you'll use that fact in a minute.)
- The path to your volume can either be a file (/home /user/encrypted.volume) or a block device (/dev/sdx1) or a UUID.
- The password can be set as `none` – meaning the user will need to provide it at boot (only once, still better than at every login). Alternatively, it can be a path to a file containing only the password. If you aren't using a password and are only using one or more keyfiles – then set the password to `/dev/null`
- Since this is a VeraCrypt volume you're auto-mounting, you need to mark it as such with the `tcrypt-veracrypt` option. If you're using a keyfile, then you need to specify that and it's location `tcrypt-keyfile=/path/to/keyfile`(If you're not using keyfiles, then be sure not to copy-paste the option in… and delete the unnecessary comma, too!)

More options are available, but the basics should get you going in most cases. If you're looking for more details, then read these crypttab docs.

At this point, the encrypted volume will be auto-unencrypted at startup. Half the job is done. Now you just need to auto-mount it.

fstab

You'll update your fstab: `/etc/fstab` with a single line per volume. The general format will look like this:

```
/dev/mapper/volume_name /mnt/point auto
nosuid,nodev,nofail 0 0
```

- Remember that volume_name you chose earlier? Here's another place to use it.
- The mount point can be anywhere you'd like.
- You can specify the filesystem type or leave it as *auto*.
- The rest of the options shown here just ensures others can read and write to the mounted volume.

When you're done editing fstab, feel free to restart the system and enjoy your auto-mounting VeraCrypt volume

Reminders:

- **Auto-mounting a VeraCrypt volume is a great way to share an encrypted disk between multiple OSes (ie Linux and Windows) in a multi-boot setup.**
- Storing a password that you use for anything else as plain-text is ~~generally~~ a terrible idea.
- Keyfiles can be used across OSes and can be randomly generated from within VeraCrypt.
- If you will use a keyfile or a plain-text password file – be sure it is stored on an encrypted disk (other than the one you're unlocking, clearly). Consider encrypting the OS disk and keeping the keys for your VeraCrypt volume there.
- This volume is mounted outside of the VeraCrypt GUI – it will not

be listed there and you cannot dismount it from there.

- If there’s a problem with your crypttab and things hang on boot – wait several minutes. Your system should eventually skip over trying to mount the encrypted volume and allow you to log in.

Yep. It auto-mounts. Carry on.

Tip this:

.002 ETH.02 ETH.2 ETH