ENCRYPTION

Logical volumes can be encrypted using `dmcrypt`. Encryption can be done in different ways, specially with LVM. `ceph-volume` is somewhat opinionated with the way it sets up encryption with logical volumes so that the process is consistent and robust.

In this case, `ceph-volume lvm` follows these constraints:

- only LUKS (version 1) is used
- Logical Volumes are encrypted, while their underlying PVs (physical volumes) aren't
- Non-LVM devices like partitions are also encrypted with the same OSD key

LUKS

There are currently two versions of LUKS, 1 and 2. Version 2 is a bit easier to implement but not widely available in all distros Ceph supports. LUKS 1 is not going to be deprecated in favor of LUKS 2, so in order to have as wide support as possible, `ceph-volume` uses LUKS version 1.

> **Note:** Version 1 of LUKS is just referenced as "LUKS" whereas version 2 is referred to as LUKS2

LUKS ON LVM

Encryption is done on top of existing logical volumes (unlike encrypting the physical device). Any single logical volume can be encrypted while other volumes can remain unencrypted. This method also allows for flexible logical volume setups, since encryption will happen once the LV is created.

WORKFLOW

When setting up the OSD, a secret key will be created, that will be passed along to the monitor in JSON format as stdin to prevent the key from being captured in the logs.

The JSON payload looks something like:

```
{
    "cephx_secret": CEPHX_SECRET,
    "dmcrypt_key": DMCRYPT_KEY,
    "cephx_lockbox_secret": LOCKBOX_SECRET,
}
```

The naming convention for the keys is **strict**, and they are named like that for the hardcoded (legacy) names ceph-disk used.

- `cephx_secret` : The cephx key used to authenticate
- `dmcrypt_key` : The secret (or private) key to unlock encrypted devices
- `cephx_lockbox_secret` : The authentication key used to retrieve the `dmcrypt_key`. It is named *lockbox* because ceph-disk used to have an unencrypted partition named after it, used to store public keys and other OSD metadata.

The naming convention is strict because Monitors supported the naming convention by ceph-disk, which used these key names. In order to keep compatibility and prevent ceph-disk from breaking, ceph-volume will use the same naming convention *although they don't make sense for the new encryption workflow*.

After the common steps of setting up the OSD during the prepare stage, either with filestore or bluestore, the logical volume is left ready to be activated, regardless of the state of the device (encrypted or decrypted).

At activation time, the logical volume will get decrypted and the OSD started once the process completes correctly.

Summary of the encryption workflow for creating a new OSD:

1. OSD is created, both lockbox and dmcrypt keys are created, and sent along with JSON to the monitors, indicating an encrypted OSD.
2. All complementary devices (like journal, db, or wal) get created and encrypted with the same OSD key. Key is stored in the LVM metadata of the OSD
3. Activation continues by ensuring devices are mounted, retrieving the dmcrypt secret key from the monitors and

decrypting before the OSD gets started.