systemd-cryptsetup

- Description
- Credentials
- See Also

systemd

systemd-cryptsetup

8

systemd-cryptsetup

systemd-cryptsetup@.service

Full disk decryption logic

systemd-cryptsetup

OPTIONS

attach

VOLUME

SOURCE-DEVICE

KEY-FILE

CRYPTTAB-OPTIONS

systemd-cryptsetup

OPTIONS

detach

VOLUME

`systemd-cryptsetup@.service`

`system-systemd\x2dcryptsetup.slice`

Description

`systemd-cryptsetup` is used to set up (with `attach`) and tear down (with `detach`) access to an encrypted block device. It is primarily used via `systemd-cryptsetup@.service` during early boot, but may also be called manually. The positional arguments `VOLUME`, `SOURCE-DEVICE`, `KEY-FILE`, and `CRYPTTAB-OPTIONS` have the same meaning as the fields in `crypttab(5)`.

`systemd-cryptsetup@.service` is a service responsible for providing access to encrypted block devices. It is instantiated for each device that requires decryption.

`systemd-cryptsetup@.service` instances are part of the `system-systemd\x2dcryptsetup.slice` slice, which is destroyed only very late in the shutdown procedure. This allows the encrypted devices to remain up until filesystems have been unmounted.

`systemd-cryptsetup@.service` will ask for hard disk passwords via the password agent logic, in order to query the user for the password using the right mechanism at boot and during runtime.

At early boot and when the system manager configuration is reloaded, `/etc/crypttab` is translated into `systemd-cryptsetup@.service` units by `systemd-cryptsetup-generator(8)`.

In order to unlock a volume a password or binary key is required. `systemd-cryptsetup@.service` tries to acquire a suitable password or binary key via the following mechanisms, tried in order:

1. If a key file is explicitly configured (via the third column in `/etc/crypttab`), a key read from it is used. If a PKCS#11 token, FIDO2 token or TPM2 device is configured (using the `pkcs11-uri=`, `fido2-device=`, `tpm2-device=` options) the key is decrypted before use.
2. If no key file is configured explicitly this way, a key file is automatically loaded from `/etc/cryptsetup-keys.d/volume.key` and `/run/cryptsetup-keys.d/volume.key`, if present. Here too, if a PKCS#11/FIDO2/TPM2 token/device is configured, any key found this way is decrypted before use.
3. If the `try-empty-password` option is specified then unlocking the volume with an empty password is attempted.
4. If the `password-cache=` option is set to `yes` or `read-only`, the kernel keyring is then checked for a suitable cached password from previous attempts.
5. Finally, the user is queried for a password, possibly multiple times, unless the `headless` option is set.

If no suitable key may be acquired via any of the mechanisms describes above, volume activation fails.

Credentials

`systemd-cryptsetup` supports the service credentials logic as implemented by `ImportCredential=`/`LoadCredential=`/`SetCredential=` (see `systemd.exec(5)` for details). The following credentials are used by `systemd-crypsetup@root.service` (generated by `systemd-gpt-auto-generator`) when passed in:

`cryptsetup.passphrase`
This credential specifies the passphrase of the LUKS volume.

`cryptsetup.tpm2-pin`
This credential specifies the TPM pin.

`cryptsetup.fido2-pin`
This credential specifies the FIDO2 token pin.

`cryptsetup.pkcs11-pin`
This credential specifies the PKCS11 token pin.

`cryptsetup.luks2-pin`
This credential specifies the pin requested by generic LUKS2 token modules.

See Also

`systemd(1)`, `systemd-cryptsetup-generator(8)`, `crypttab(5)`, `systemd-cryptenroll(1)`, `cryptsetup(8)`, TPM2 PCR Measurements Made by systemd

---

□□□ #1
□ □□□ □□□ 2026-10-31 14:59:37 UTC
□ □□□ □□□ 2026-10-31 14:59:37 UTC