



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'systemd-cryptsetup-generator.8' command

\$ man systemd-cryptsetup-generator.8

SYSTEMD-CRYPTSETUP-GENERATORsystemd-cryptsetup-genSYSTEMD-CRYPTSETUP-GENERATOR(8)

NAME

systemd-cryptsetup-generator - Unit generator for /etc/crypttab

SYNOPSIS

/usr/lib/systemd/system-generators/systemd-cryptsetup-generator

DESCRIPTION

systemd-cryptsetup-generator is a generator that translates /etc/crypttab into native systemd units early at boot and when configuration of the system manager is reloaded. This will create systemd-cryptsetup@.service(8) units as necessary. systemd-cryptsetup-generator implements systemd.generator(7).

KERNEL COMMAND LINE

systemd-cryptsetup-generator understands the following kernel command line parameters:

luks=, rd.luks=

Takes a boolean argument. Defaults to "yes". If "no", disables the generator entirely. rd.luks= is honored only in the initrd while luks= is honored by both the main system and in the initrd.

luks.crypttab=, rd.luks.crypttab=

Takes a boolean argument. Defaults to "yes". If "no", causes the generator to ignore any devices configured in /etc/crypttab (luks.uuid= will still work however). rd.luks.crypttab= is honored only in initrd while luks.crypttab= is honored by both the main

system and the initrd.

`luks.uuid=`, `rd.luks.uuid=`

Takes a LUKS superblock UUID as argument. This will activate the specified device as part of the boot process as if it was listed in `/etc/crypttab`. This option may be specified more than once in order to set up multiple devices. `rd.luks.uuid=` is honored only in the `initrd`, while `luks.uuid=` is honored by both the main system and the `initrd`.

If `/etc/crypttab` contains entries with the same UUID, then the name, keyfile and options specified there will be used. Otherwise, the device will have the name "luks-UUID".

If `/etc/crypttab` exists, only those UUIDs specified on the kernel command line will be activated in the `initrd` or the real root.

`luks.name=`, `rd.luks.name=`

Takes a LUKS super block UUID followed by an "=" and a name. This implies `rd.luks.uuid=` or `luks.uuid=` and will additionally make the LUKS device given by the UUID appear under the provided name.

This parameter is the analogue of the first `crypttab(5)` field `volume-name`.

`rd.luks.name=` is honored only in the `initrd`, while `luks.name=` is honored by both the main system and the `initrd`.

`luks.data=`, `rd.luks.data=`

Takes a LUKS super block UUID followed by an "=" and a block device specification for device hosting encrypted data.

For those entries specified with `rd.luks.uuid=` or `luks.uuid=`, the data device will be set to the one specified by `rd.luks.data=` or `luks.data=` of the corresponding UUID.

LUKS data device parameter is useful for specifying encrypted data devices with detached headers specified in `luks.options` entry containing "header=" argument. For example,

```
rd.luks.uuid=b40f1abf-2a53-400a-889a-2eccc27eaa40
```

```
rd.luks.options=b40f1abf-2a53-400a-889a-2eccc27eaa40=header=/path/to/luks.hdr
```

```
rd.luks.data=b40f1abf-2a53-400a-889a-2eccc27eaa40=/dev/sdx. Hence,
```

in this case, we will attempt to unlock LUKS device assembled from data device `"/dev/sdx"` and LUKS header (metadata) put in `"/path/to/luks.hdr"` file. This syntax is for now only supported on a per-device basis, i.e. you have to specify LUKS device UUID. This parameter is the analogue of the second `crypttab(5)` field `encrypted-device`.

`rd.luks.data=` is honored only in the `initrd`, while `luks.data=` is honored by both the main system and in the `initrd`.

`luks.key=`, `rd.luks.key=`

Takes a password file name as argument or a LUKS super block UUID followed by a `"="` and a password file name.

For those entries specified with `rd.luks.uuid=` or `luks.uuid=`, the password file will be set to the one specified by `rd.luks.key=` or `luks.key=` of the corresponding UUID, or the password file that was specified without a UUID.

It is also possible to specify an external device which should be mounted before we attempt to unlock the LUKS device.

`systemd-cryptsetup` will use password file stored on that device.

Device containing password file is specified by appending colon and a device identifier to the password file path. For example,

```
rd.luks.uuid=b40f1abf-2a53-400a-889a-2eccc27eaa40
```

```
rd.luks.key=b40f1abf-2a53-400a-889a-2eccc27eaa40=/keyfile:LABEL=keydev.
```

Hence, in this case, we will attempt to mount file system residing on the block device with label `"keydev"`. This syntax is for now only supported on a per-device basis, i.e. you have to specify LUKS device UUID.

This parameter is the analogue of the third `crypttab(5)` field `key-file`.

`rd.luks.key=` is honored only in the `initrd`, while `luks.key=` is honored by both the main system and in the `initrd`.

`luks.options=`, `rd.luks.options=`

Takes a LUKS super block UUID followed by an `"="` and a string of options separated by commas as argument. This will override the

options for the given UUID.

If only a list of options, without an UUID, is specified, they apply to any UUIDs not specified elsewhere, and without an entry in `/etc/crypttab`.

This parameter is the analogue of the fourth `crypttab(5)` field options.

It is possible to specify an external device which should be mounted before we attempt to unlock the LUKS device.

`systemd-cryptsetup` will assemble LUKS device by combining data device specified in `luks.data` with detached LUKS header found in "header=" argument. For example,

```
rd.luks.uuid=b40f1abf-2a53-400a-889a-2eccc27eaa40
```

```
rd.luks.options=b40f1abf-2a53-400a-889a-2eccc27eaa40=header=/luks.hdr:LABEL=hdrdev
```

```
rd.luks.data=b40f1abf-2a53-400a-889a-2eccc27eaa40=/dev/sdx. Hence,
```

in this case, we will attempt to mount file system residing on the block device with label "hdrdev", and look for "luks.hdr" on that file system. Said header will be used to unlock (decrypt) encrypted data stored on `/dev/sdx`. This syntax is for now only supported on a per-device basis, i.e. you have to specify LUKS device UUID.

`rd.luks.options=` is honored only by initial RAM disk (`initrd`) while

`luks.options=` is honored by both the main system and the `initrd`.

SEE ALSO

`systemd(1)`, `crypttab(5)`, `systemd-cryptsetup@.service(8)`, `systemd-cryptenroll(1)`, `cryptsetup(8)`, `systemd-fstab-generator(8)`

`systemd` 252

SYSTEMD-CRYPTSETUP-GENERATOR(8)