# /etc/crypttab

informations statiques sur les systèmes de fichier chiffrés

Ce fichier contient des informations sur les systèmes de fichiers chiffrés. crypttab est seulement lu par les programmes (cryptdisks_start et cryptdisks_stop), et non en écriture. C'est à l'administrateur système de créer et maintenir ce fichier. Chaque système de fichier est décris sur un ligne sépartée ; les champs sur chaque ligne sont séparés par des espaces. L'ordre des enregistrements dans crypttab est importante parce qu'il est lu séquentiellement.

Le premier champ, target, décris le nom du périphérique mappé. Il doit être un nom de fichier dans composant répertoire. Un périphérique mappé sera créé sous /dev/mapper/target.

Le second champ, source-device, décris soit le périphérique block spéciale, ou le fichier qui contient les données chiffrées. Au lieu de donner le périphérique explicitement, l'UUID est supporté également.

Le troisième champ, Key-file, décris le fichier à utiliser comme clé pour déchiffrer les données. Noter que le fichier de clé sera utilisé comme passphrase ; la passphrase ne doit pas être suivie par le caractèse newline. Peut églalement être un nom de périphérique.

Si le fichier de clé est la chaîne "none", une passphrase sera lue interactivement depuis la console. Dans ce cas, les options precheck, check, checkargs et tries peuvent être utiles.

Le quatrième champ, options, décris les options cryptsetup associées avec le processus de chiffrement. Au minimum, le champ devrait contenir soit la chaîne luks respectivement tcrypt ou les options cipher, hash, et size. Les options sont au format key=value. Noter que les 4 champs sont obligatoires.

# OPTIONS

**cipher=<cipher>** Algorithme de chiffrement (ignoré pour LUKS et TCRYPT).

**size=<size>** Taille de la clé de chiffrement (ignoré pour LUKS et TCRYPT)

**hash=<hash>** Algorithme de hashage (ignoré pour LUKS et TCRYPT)

**offset=<offset>** Offset de début (ignoré pour LUKS et TCRYPT)

**skip=<skip>** Secteur du début à sauter (ignoré pour LUKS et TCRYPT)

**verify** Vérifie le mot de passe

**readonly** Le périphérique est lecture seule

**discard** Permet l'utilisation des requêtes discards (TRIM) pour le périphérique. déconseillé pour des raisons de sécurité.

**luks** Utilise le périphérique avec les extensions LUKS

**tcrypt** Utilese le périphérique avec les extensions TCRYPT

**veracrypt** Utilise l'extension VeraCrypt de TCRYPT.

**swap** Lance mkswap sur le périphérique créé

**tmp=<tmpfs>** Lance mkfs avec le type tmpfs spécifié dans le périphérique créé. Défaut : ext4

**precheck=<precheck>** Vérifie le contenu du périphérique source par un programme. Si la vérification échoue, le périphérique n'est pas créé.. cryptdisks/cryptroot recherche un programme dans /lib/cryptsetup/checks par défaut. preckeck n'est pas invoqué par les périphériques LUKS

**check=<check>** Vérifie le contenu du périphérique cible par un programme. Si la vérification échoue, le périphérique n'est pas créé.. cryptdisks/cryptroot recherche un programme dans /lib/cryptsetup/checks par défaut.

**checkargs=<arguments>** Donne les arguments spécifiés au programme de vérification.

**tries=<num>** L'entrée de la passphrase est tentée num fois en cas d'erreur. Défaut=3. 1 ne retente pas, et 0 demande la passphrase jusqu'à ce quelle soit correct.

**initramfs** Processus hook initramfs traitant le périphérique root, tout périphérique avec cette option sont traités dans initramfs.

**noearly** Les scripts init sont invoqués 2 fois durant le processus de boot. Une fois avant lvm et raid, et une fois après.

**noauto** Ignore entièremenet le périphérique au processus de boot.

**loud** mode verbeu. Affiche une alerte si un périphérique n'existe pas.

**quiet** mode silencieu.

**keyscript=<path>** L'exécutable utilisé avec le fichier de clé et la sortie est utilisée comme clé.

**keyslot=<slot>** Slot de la clé (LUKS uniquement)

**header=<path>** fichier d'en-tête détaché (ignoré pour les périphériques dm-crypt plain)

**tcrypthidden** Utilise l'en-tête TCRYPT caché.

# checkscripts

**blkid** Vérifie les systèmes de fichier inconnus. Supporte un type comme argument via checkargs : "" réussis si un fs valide est trouvé, "none" réussis si aucun fs valide n'est trouvé, "ext4" (xfs, swap, crypto_LUKS, etc) réussis si un fs ext4 est trouvé.

**un_blkid** Vérifie les système de fichier connus.

# Exemples

Périphérique d'échange chiffré :
**cswap /dev/sda6 /dev/urandom cipher=aes-xts-plain64,size=256,hash=sha1,swap**
Disque LUKS chiffré avec mot de passe intéractif, identifié par son UUID :
**cdisk0 UUID=12345678-9abc-def012345-6789abcdef01 none luks**
disque TCRYPT chiffré avec mot de passe interactif :
**tdisk0 /dev/sr0 none tcrypt**
Disque ext4 chiffré avec mot de passe interactif, retente 5 fois :
**cdisk1 /dev/sda2 none cipher=aes-xts-plain64,size=256,hash=sha1,checkargs=ext4,tries=5**
Utiliser un script de vérification spécifié, sans retentative :
**cdisk2 /dev/sdc1 none cipher=aes-xts-plain64,size=256,hash=sha1,check=customscript,tries=1**
Utiliser Twofish et un hash RIPEMD-160 :
**cdisk3 /dev/sda3 none cipher=twofish,size=256,hash=ripemd160**

# Variables d'environnement

**CRYPTDISKS_ENABLE** à yes, lance les initscripts cryptdisks au démarrage.

**CRYPTDISKS_MOUNT** Spécifie les points de montage qui sont montés avant d'invoquer cryptdisks. Prend les points de montage configuré dans /etc/fstab comme arguments, séparés par un espace.

**CRYPTDISKS_CHECK** Spécifie le checkscript par défaut à lancer avec le périphérique cible après avoir invoqué cryptdisks

**CRYPTDISKS_PRECHECK** Spécifie le checkscript par défaut à lancer avec le périphérique dm-crypt, avant d'invoquer cryptdisks