cryptmount

Monter/démonter/configurer un système de fichier chiffré

cryptmount permet de monter ou démonter un système de fichier chiffré dans nécessiter de privilèges root, et l'assister root dans la création de nouveau système de fichiers chiffrés. Une fois la configuration initiale du système de fichier par l'administrateur système, l'utilisateur doit seulement fournir le mot de passe de déchiffrement pour que cryptmount configure automatiquement device-mapper et les cibles loopback avant de monter le système

cryptmount a été écris en réponse aux différences entre la nouvelle infrastructure device-mapper de linux 2.6 et l'ancien cryptoloop qui permettait aux utilisateurs standards d'accéder aux systèmes de fichiers directement via mount.

OPTIONS

-a, –all Agit sur toutes les cible disponible, ex : pour monter toutes les cibles

-m, –mount Monte la cible spécifiée. L'utilisateur devra fournir un mot de passe pour débloquer la clé de déchiffrement pour le système de fichier.

-S, –status Indique si la cible est montée ou non

-l, –list Liste toutes les cibles disponible, incluant des informations sur le système de fichier et le point de montage.

-c, –change-password Change le mot de passe qui protège la clé de déchiffrement pour un système de fichier donné.

-g, –generate-key size Génère une clé de déchiffrement opur un nouveau système de fichier, avec la taille spécifiée en octets

-e, –reuse-key existing-target Définis une clé de déchiffrement pour un nouveau système de fichier, en utilisant une clé existante pour un autre système de fichier. Uniquement disponible pour root

-f, –config-fd num Lit la configuration des cible depuis le descripteur de fichier spécifié au lieu du fichier de configuration par défaut. Uniquement disponible pour root

-m, –passwd-fd num Lit les mots de passe depuis le descripteur de fichier spécifié au lieu du terminal. Chaque mot de passe est lu une seule fois.

-p, –prepare Prépare les périphériques device-mapper et loopback nécessaire pour accéder à la cible, mais ne la monte pas. Utilisé par root pour installer un système de fichier sur un périphérique chiffré

-r, –release Relache tous les périphériques device-mapper et loopback associés avec un cible particulière. Uniquement disponible pour root.

-s, –swapon Active le paging et swaping pour la cible. Uniquement disponible pour root.

-x, –swapoff Désactive le paging et swaping pour la cible. Uniquement disponible pour root.

-k, –key-managers Liste tous les formats disponibles pour protéger les clés d'accès au système de fichier

-B, –system-boot Configure toutes les cibles marquées "bootaction". Généralement utilisé pour monter automatiquement les systèmes de fichiers chiffré. Uniquement disponible pour root.

-Q, –system-shutdown Stop toutes les cibles marquées bootaction. L'opposé de -B. Uniquement disponible pour root.

-n, –safetynet Tente de stopper toute cible mounté qui devrait normalement être arrêtée avec –unmount ou –swapoff. Uniquement disponible pour root, et prévu exclusivemeent durant l'extinction du système.

Codes de sortie

1 Option de ligne de commande non-reconnue

2 Nom de cible non-reconnue

3 Erreur en exécutant le programme helper

100 Privilèges insuffisants

101 Erreur de sécurité à l'installation

Exemple d'utilisation

Pour pouvoir créer un nouveau système de fichier chiffré par cryptmount, utiliser le programme cryptmount-setup qui peut être utilisé par le superuser pour le configurer interactivement.

Alternativement, une configuration manuelle permet un contrôle avancé. Avant de le faire, il faut s'assurer que le kernel supporte /dev/loop et /dev/mapper (modprobe -a loop dm-crypt).

Ensuite, supposons que l'on souhaite définir un nouveau système de fichier chiffré qui aura comme nom "opaque". Si on a une partition libre, disons /dev/hdb63, on peut donc l'utiliser directement pour stocker le système de fichier chiffré. Alternativement, si l'on souhaite stocker le système de fichier chiffré dans un fichier ordinaire, on doit créer un fichier avec par exemple :
dd if=/dev/zero of=/home/opaque.fs bs=1M count=512
Puis remplacer toutes les occurences de /dev/hdb63 avec /home/opaque.fs.
Premièrement, il faut ajouter une entrée dans /etc/cryptmount/cmtab, qui décris le chiffrement qui sera utilisé pour protéger le système de fichier lui-même et la clé d'accès, comme suit :

```
opaque {
dev=/dev/hdb63 dir=/home/crypt
fstype=ext2 mountoptions=defaults cipher=twofish
keyfile=/etc/cryptmount/opaque.key
keyformat=builtin
}
```

Ici, on utilise l'algorithme twofish pour chiffrer le système de fichier lui-même, avec le gestionnaire de clé embarqué utilisé pour protéger la clé de déchiffrement (dans /etc/cryptmount/opaque.key) qui sera utilisé pour chiffrer le système de fichier lui-même, on peut exécuter, en root :

Cela va générer une clé 32 octets (256bits), qui est connu pour être supportée par twofish, et la stocke dans une forme chiffrée après avoir demandé le mot de passe.

Si on exécute, en root :
cryptmount –prepare opaque
Le mot de passe sera demandé, qui va permettre à cryptmount de définir une cible device-mapper (/dev/mapper/opaque). On peut désormais utiliser les outils standard pour créer le système de fichier dans /dev/mapper/opaque :
mke2fs /dev/mapper/opaque
après avoir exécuté :
cryptmount –release opaque
mkdir /home/crypt
Le système de fichier chiffré est prêt. Généralement, les utilisateurs peuvent le monter avec :
cryptmount -m opaque
ou
cryptmount opaque
et le démontent avec
cryptmount -u opaque
cryptmount conserve un enregistrement sur quel utilisateur a monté chaque système de fichier pour pouvoir fournir un mécanisme de blockage pour s'assurer que seul le même utilisateur (ou root) peut le démonter.

Changement de mot de passe

Une fois un système de fichier utilisé, on peut souhaiter changer le mot de passe d'accès, par exemple :
cryptmount –change-password opaque

Système de fichier chiffré avec LUKS

cryptmount peut être utilisé pour fournir un accès simple à aux systèmes de fichier chiffrés compatibles avec LUKS.

Pour accéder à une partition LUKS existante, une entrée a besoin d'être créé dans /etc/cryptmount/cmtab. Par exemple, si la partition /dev/hdb62 est utilisé pour contenir un système de fichier ext3 LUKS, une entrée sous la forme :

```
LUKS {
keyformat=luks
dev=/dev/hdb62 keyfile=/dev/hdb62
dir=/home/luks-dir fstype=ext3
}
```

Que permet de monter via cryptmount sous /home/luks-dir en exécutant :
cryptmount LUKS
cryptmount va également permettre à un utilisateur qui connaît un des mots de passe d'accè de changer leur mot de passe via
cryptmount –change-password LUKS
cryptmount fournis également un support basic pour créer un nouveau système de fichier chiffré LUKS, qui peut être placé dans des fichiers ordinaires aussi bien que des partitions disque, via –generate-key. Cependant, pour exploiter toutes les fonctionnalité de LUKS, tel qu'ajouter plusieurs mots de passes, il faut utiliser cryptsetup.

Il est fortement recommandé de ne pas tenter d'utiliser le support LUKS en combinaison avec les fonctionnalités de cryptmount pour stocker plusieurs systèmes de fichiers chiffrés dans une simple partition disque ou un fichier ordinaire. Cela est dû à la supposition dans le design de cryptsetup-luks que la clé LUKS est toujours stockée au début de la partition.