Chiffrage transparent des données sous Linux

Gilles BERNARD

25/01/2012 Ed 2.0

Préambule

L'informatique est devenue extrêmement nomade. Si vous lisez cet article, il y a de fortes chances pour que vous possédiez un Laptop ou un NetBook sous Linux et que vous l'emportez souvent par monts et par vaux. Maintenant que se passerait-il si vous veniez à perdre votre portable ou qu'on vous le volait. Au delà de la perte directe financière liée au vol ou à la perte, il y a le risque d'exploitation criminelle de vos données : données professionnelles, login/mot de passe à tous vos sites, photographies, identifiants bancaires...

Introduction

Cet article décrit comment mettre en place un niveau de sécurité modéré pour les données de votre appareil. Il s'agit de chiffrer l'intégralité de vos données à l'exclusion du système d'exploitation. Ainsi, on gagne en sécurité sur les données sans perdre en performance sur le système. Par niveau de sécurité modéré, j'entends qu'il suffit amplement pour protéger vos données contre l'utilisation frauduleuse par un individu possédant des compétences en informatiques. Il s'agit donc d'aller plus loin que la simple protection par mot de passe qui contrôle l'accès à votre espace de travail (voir un éventuel mot de passe du BIOS) mais aucunement l'accès au données. Un simple boot sur un média tiers ou l'extraction et la connexion du disque dur sur un autre ordinateur permettent d'accéder à toutes vos données sans le moindre effort. C'est à la portée d'un utilisateur averti.

Dit plus techniquement, je vais décrire comment chiffrer l'intégralité de la partition /home de votre système d'exploitation Linux. Les pré-requis sont donc :

- Avoir une installation Linux quelconque (testé ici sous Mageia 1) ;
- Avoir une partition libre de taille suffisante pour stocker tout votre répertoire `/home`.

Une fois l'installation terminée, votre appareil vous demandera le mot de passe de décryptage pendant la séquence de boot. Ensuite tout se passera de manière transparente pour vous. A ceci prêt

que l'accès à vos données pour un tiers ne connaissant pas votre mot de passe ne pourra se faire que par la casse du cryptage. Ce qui nécessite, temps, puissance de calcul et energie.

Pour la suite de cet article `/dev/sda3` représente la partition que vous allez chiffrer pour stocker votre espace /home. Cette partition sera intégralement effacée.

Création de la partition chiffrée

La première étape dans la mise en place du chiffrement est d'initialiser la partition avec des données aléatoires. L'objet de cette étape est de rendre encore plus difficile la casse du cryptage par une méthode « brute force ». Cette étape est relativement longue et dépend de la taille de votre partition et de la puissance de votre machine. L'étape est facultative si vous n'êtes pas totalement paranoïaque. Notez que les données de la partition `/dev/sda3` vont être irrémédiablement perdues.

La commande à exécuter (**avec beaucoup de précautions sur la syntaxe**) est :

```
[root@pc ~]# dd if=/dev/urandom of=/dev/sda3 bs=1024k
```

Pour vous donner un ordre de grandeur, le débit de sortie de `/dev/urandom` sur un Atom N330 est de 1.6Mo/s, soit dix bonnes minutes par giga octets.

Maintenant on va mettre en place le chiffrage a proprement parler. Dans cet exemple, je laisse les paramètres par défaut pour le cryptage qui sont largement suffisant pour le niveau de sécurité que je recherche. Vous pouvez consulter la documentation de `cryptsetup` pour mettre en place un cryptage plus fort.

```
[root@pc ~]# cryptsetup luksFormat /dev/sda3


WARNING!

========

Cette action écrasera définitivement les données sur
/dev/sda3.


Are you sure? (Type uppercase yes): YES

Entrez la phrase secrète LUKS : *****

Verify passphrase: *****
```

La partition est maintenant cryptée, on peut l'ouvrir pour l'utiliser. Dans la suite, le système sera modifié pour que l'opération soit complétement transparente est réalisée pendant la séquence de boot.

```
[root@pc ~]# cryptsetup luksOpen /dev/sda3 chome
```

La commande suivante permet de vérifier la présence du périphérique spécial dans `/dev/mapper` :

```
[root@pc ~]# ls –l /dev/mapper/chome
```

```
lrwxrwxrwx 1 root root 7 nov.  30 08:13 /dev/mapper/chome
-> ../dm-0
```

La partition étant ouverte, on va pouvoir la formater et la monter. N'importe quel système de fichier supporté par le noyau linux peut être utilisé. Dans l'exemple suivant, le formatage se fait en `ext4`.

```
[root@pc ~]# mkfs.ext4 /dev/mapper/chome
```

Recopie des données et mise en place de /home

A ce stade vous avez une partition cryptée prête a être utilisée. Il vous reste à déplacer le contenu de votre répertoire actuel `/home` vers la partition cryptée. Pour ce faire, je vous conseil de passer en mode utilisateur simple (`init 1`) ou au moins à quitter le mode graphique (`init 3`) et à vous logger directement en tant que `root` pour que le répertoire `/home` ne soit pas utilisé et verrouillé par un processus utilisateur.

Passage en mode text et login en tant que `root` :

```
[root@pc ~]# init 3
```

Renommage de l'ancien répertoire `/home` :

```
[root@pc ~]# mv /home /home.tmp
```

Création du point de montage pour le `/home` crypté :

```
[root@pc ~]# mkdir /home
```

Montage de la partition cryptée sur `/home` :

```
[root@pc ~]# mount /dev/mapper/chome /home
```

Recopie des données de l'ancien répertoire `/home`

```
[root@pc ~]# cp -rp /home.tmp/* /home
```

Effacement de l'ancien répertoire (si tout s'est bien passé) :

```
[root@pc ~]# rm -rf /home.tmp
```

Retour au mode graphique :

```
[root@pc ~]# init 5
```

A ce stage, la partition cryptée est bien montée sur le répertoire /home. Le passage en mode graphique doit donc se faire sans encombre. Il va nous rester maintenant à faire en sorte que le montage se fasse automatiquement. C'est l'objet du chapitre suivant.

Configuration du système

La mise en place de la configuration pour l'établissement automatique de la partition cryptée au boot nécessite d'éditer ou de créer le fichier `</etc/crypttab>` et d'ajouter la ligne suivante :

```
chome /dev/sda3 none luks
```

De même, éditez `</etc/fstab>` et ajoutez :

```
/dev/mapper/chome /home ext4 acl,noatime 2 2
```

La configuration est terminée. Le fichier `</etc/crypttab>` sera automatiquement interprété au démarrage du système. Les périphériques qu'ils contient seront démarrés dans le processus de boot

et le mot de passe vous sera automatiquement demandé.

Le fichier </etc/fstab> indique au système que le répertoire /home est un point de montage et que le montage doit être fait automatiquement au démarrage. Le montage intervient après le démarrage du périphérique crypté.

Gestion des clés de chiffrage

Le chiffrage avec Luks vous permet d'avoir plusieurs clés pour accéder à vos données et de gérer ces dernières. Vous pouvez ainsi ajouter de nouvelles clés, en modifier d'anciennes et en supprimer des obsolètes.

Ajout d'une clé :

```
[root@pc ~]# cryptsetup luksAddKey /dev/loop2
Entrez une phrase de passe : *une_clé_valide*
Entrez une nouvelle phrase secrète pour l'emplacement de clé : *une_nouvelle_clé*
Verify passphrase: *une_nouvelle_clé*
```

Modification d'une clé existante :

```
[root@pc ~]# cryptsetup luksChangeKey /dev/loop2
Enter LUKS passphrase to be changed: *une_nouvelle_clé*
Enter new LUKS passphrase: *une_nouvelle_clé_changée*
Verify passphrase: *une_nouvelle_clé_changée*
```

La clé « *une_nouvelle_clé* » a été changée en « *une_nouvelle_clé_changée* ». Dans l'absolu, il est fortement souhaitable de changer régulièrement le mot de passe de chiffrage en utilisant cette commande.

Suppression d'une clé :

```
[root@pc ~]# cryptsetup luksRemoveKey /dev/loop2
Entrez la phrase secrète LUKS à effacer : *une_nouvelle_clé_changée*
```

A noter qu'il n'existe pas de commande pour lister les clés actives.

Conclusion

Vous pouvez maintenant redémarrer votre système. Pendant la séquence de boot, le mot de passe de chiffrage du disque /home vous sera demandé. Tant que ce mot de passe n'a pas été fourni, vos données ne peuvent être exploitée.

Important : une défaillance du disque peut être fatale à toutes vos données. Si un secteur défectueux empêche le montage, les données sont illisibles.

Les menaces résiduelles sont :

- Les données sont librement accessibles lorsque le mot de passe a été saisi ;
- La partition système n'est pas chiffrée, sont donc accessibles :

    - Les différents paramètres de configuration et notamment les accès aux réseaux WiFi (ex : `/etc/wpa_supplicant.conf`) ;
    - Les fichiers temporaires `/tmp` et `/var/tmp` ;
    - Les fichiers de log `/var/log` ;
- La partition de `swap` n'est pas chiffrée et peut donc être exploitée pour obtenir d'éventuelles informations ;

Malgré tout, le niveau de sécurité de votre configuration est largement plus élevé.