Installation de Slint avec des partitions cryptées

Installer Slint avec des partitions cryptées rend très difficile l'accès par des tiers non autorisées aux informations stockées sur ces partitions. Cet article propose une des manières les plus accessibles de la faire.

L'auteur de ces lignes est un simple utilisateur, non un expert en la matière. Documentez-vous auprès de sources plus compétentes si vos données à crypter sont cruciales. J'en propose quelques unes, mais votre esprit critique doit toujours s'exercer.

- Le cryptage d'une partition n'empêche pas l'accès à vos données localement ou par le réseau quand le système fonctionne.
- Le cryptage est inopérant pour un tiers pouvant accéder à la phrase de passe (*passphrase* en Anglais).
- Si vous oubliez la phrase de passe vous ne pourrez plus accéder à vos données. Elles seront définitivement perdues.
- Le risque de perdre des données ou au moins l'accès à celles-ci la suite d'une fausse manœuvre ou d'une défaillance d'un périphérique est élevé, et les conséquences peut-être plus importantes que celles d'un vol ou d'une divulgation. Si vous tenez à vos données, **sauvegardez-les** (cryptées, bien entendu). Si elles n'en valent pas la peine, leur cryptage est peut-être superflu ...
- Lire à ce sujet la section 6 Backup and Data Recovery de la FAQ de cryptsetup.

Il existe différentes méthodes de cryptage matériel et logiciel, décrites notamment dans cet article (en Anglais).

Nous vous proposons d'utiliser les logiciels et options suivants:

- le pilote intégré au noyau Linux dm-crypt. dm est l'acronyme de *Device Mapper* ou mappeur de périphérique. Sa fonction est d“associer au périphérique (ou à une de ses partitions) un nom de fichier par lequel on y accédera après avoir ouvert la communication en tapant la phrase de passe.
- L'utilitaire associé (en ligne de commande) cryptsetup, qui permet comme son nom l'indique de paramétrer le cryptage.
- L'option LUKS (pour Linux Unified Key Setup) de cryptsetup. En gros, cela consiste à crypter des partitions et non des périphériques blocs dans leur totalité. Pour en savoir plus, voir la spécification de LUKS.

Pourquoi ces choix ? Ils sont accessible aux non-spécialistes, couvrent les cas les plus courants pour une utilisation individuelle... Et *last but maybe not least*, tous les outils nécessaires sont inclus dans Slint et son installateur 😉

Quoi: le schéma proposé

Quoi crypter dépend de l'utilisation prévue de votre système, du schéma de partitionnement et des menaces dont vous souhaitez vous protéger.

Ce n'est pas une décision à prendre à la légère. Prenez votre temps d'y réfléchir, consultez la FAQ de cryptsetup, demandez si nécessaire de l'aide sur la liste de diffusion de dm-crypt.

Nous proposons ici un schéma simple, qui peut convenir pour un utilisateur individuel (ordinateur de bureau ou ordinateur mobile).

Pour les besoins de l'exposé nous supposerons que la machine est doté de deux unités de stockage: un SSD (mais ce pourrait aussi bien être un disque dur) sur lequel Slint sera installé et un disque dur hébergeant des données volumineuses mais à accès relativement peu fréquent (photos, vidéos, musique archives diverses).

Nous supposons aussi que la mémoire vive est suffisante pour se passer de partition d'échange et que le répertoire /home est sur la partition racine / et non sur une partition séparée. En résumé, dans l'hypothèse d'installation de Slint sur une machine neuve sans autre système, nous créerons trois partitions:

- Une partition de petite taille (100M) sur le SSD pour l'amorçage du système, non cryptée.
- Une partition cryptée occupant l'espace restant sur le SSD
- Une partition cryptée occupant tout l'espace disponible sur le disque dur.

Ce schéma est proposé à titre d'exemple et a l'avantage de la simplicité mais n'en faites pas un modèle: il doit vous permettre de comprendre la méthode, à vous ensuite de l'adapter à votre situation.

Les étapes de l'installation

tout d'abord, si ce n'est déjà fait, familiarisez vous avec le processus d'installation, le cryptage des partitions s'insérant dans ce processus. Et lisez tout ce qui suit jusqu'au bout avant mise en pratique.

De façon schématique :

1. Démarrez l'installateur et identifiez-vous comme “root”.
2. Procédez au partitionnement du SSD et du disque dur avec l'outil `cfdisk`.
3. Cryptez la partition système du SSD avec `cryptsetup` puis “ouvrez” cette partition en la nommant pour que l'installateur puisse y accéder. Faites de même pour la partition du disque dur.
4. Démarrez l'installation proprement dite en tapant `setup`, et sélectionnez TARGET pour formater les partitions, en prenant soin de désigner les partitions cryptées par les noms attribués à l'étape précédente.
5. Installez les paquets logiciels (aucune particularité à signaler dans cette phase).
6. Procédez à toutes les étapes de configuration comme d'habitude en portant une attention particulière au le choix et à la configuration de l'amorceur (elilo ou lilo selon que vous aurez démarré l'installation en mode EFI ou BIOS).
7. A la fin du processus il vous sera proposé de redémarrer votre système. Refusez (choisissez **Non**) car l'amorçage du système n'est pas encore possible.
8. Référencez les partitions cryptées dans la table des partitions cryptées du système en cours d'installation, puis créez et installez un `initrd` permettant d'ouvrir les partitions cryptées durant l'amorçage du système.
9. Modifiez le chargeur initial et changez de noyau Linux, puis redémarrez.

Comment: le détail des étapes clef

Dans cette partie nous nous référons aux étapes numérotées ci-avant.

Étape 2: partitionnement

Tout d'abord tapez la commande:

```
lsblk -o model,name,size
```

Cela vous permettra d'identifier correctement les périphériques blocs.

Dans ce qui suit, nous supposerons que sda désigne le SSD ou disque dur sur lequel le système sera installé, sdb le disque dur destiné à contenir les archives. Sinon, adaptez les commandes qui suivent en conséquence.

Tout d'abord, partitionnez le disque système:

```
cfdisk /dev/sda
```

Choisissez une table de partition `gpt` si votre machine a démarré en mode UEFI sinon `dos`.

Créez deux partitions:

1. La première de taille 100M, de type Système EFI (table gpt) ou Linux (table dos).
2. La seconde occupant l'espace restant, de type Linux.

Puis, partitionnez le disque d'archives:

```
cfdisk /dev/sdb
```

Créez une seule partition (occupant tout l'espace disponible, a moins que vous préfériez garder de la place pour une ou plusieurs partition(s) non cryptées), de type Linux.

Étape 3 : cryptage des partitions

Tout d'abord, effacez le contenu de la partition système en y écrivant des 0 avec la commande suivante:

```
cat /dev/zero > <partition système>
```

Par exemple si cette partition est nommée sda2 (vérifiez avec `lsblk`) :

```
cat /dev/zero > /dev/sda2
```

Ce processus d'effacement peut prendre **beaucoup** de temps, patientez. Heureusement, il n'est à effectuer qu'une seule fois pour toute la durée de vie de la partition. Quand il sera terminé vous

verrez le message suivant:

```
cat: write error: no space left on device
```

Cela signifie simplement que le processus s'est arrêté car il n'y as plus de place où écrire sur la partition. C'est normal, ne vous en inquiétez pas.

Ensuite, tapez la commande suivante pour crypter la partition système et enregistrer la phrase de passe:

```
cryptsetup -y luksFormat /dev/sda2
```

Le programme vous alertera (en Anglais) sure le fait que cela effacera tout le contenu de la partition et vous demandera de confirmer en tapant YES (yes an capitales). Faites-le.

Il vus demandera ensuite d'entrer phrase de passe. Faites le, après avoir soigneusement choisi une phrase (plusieurs mots séparés par des espaces, assez longue). Incluez des fotesdo'rtografffe pour rendre le déchiffrage plus difficile.

La FAQ de cryptsetup conseille de n'utiliser que des caractères de la table ASCII pour écrire la phrase de passe, pour éviter qu'un éventuel changement de (plan de) clavier vous fasse taper autre chose qu'attendu. En tous cas, vous prendrez soin d'indiquer la langue du clavier (utilisée lors de l'installation) lors de la création de l'initrd, après installation des paquets. Nous y reviendrons.

L'option -y est là pour vous demander de taper une deuxième fois la phrase de passe, pour vérification.

pour en savoir plus tapez “man cryptsetup” sans les guillemets

- Vérifiez bien le nom de la partition système, car son contenu sera irrémédiablement effacé par la commande `cat`.
- N'oubliez la phase de passe, cela rendrait le contenu de la partition définitivement inaccessible.

Ensuite, “ouvrez” (rendez accessible) la partition cryptée tout en lui attribuant un nom. Dans l'exemple qui suit nous l'appellerons `root` (racine, en Anglais):

```
cryptsetup luksOpen /dev/sda2 root
```

Le programme vous demandera de nouveau de taper la phrase de passe (c'est plutôt rassurant, non?).

Pour la partition sur le disque dur les étapes sont semblables.Nous supposons qu'elle est initialement nommée `sdb1` (encore une fois, vérifiez avec `lsblk`) et que vous la renommez `archives` après cryptage :

```
cat /dev/zero > /dev/sdb1
cryptsetup -y luksFormat /dev/sdb1
cryptsetup luksOpen /dev/sda2 archives
```

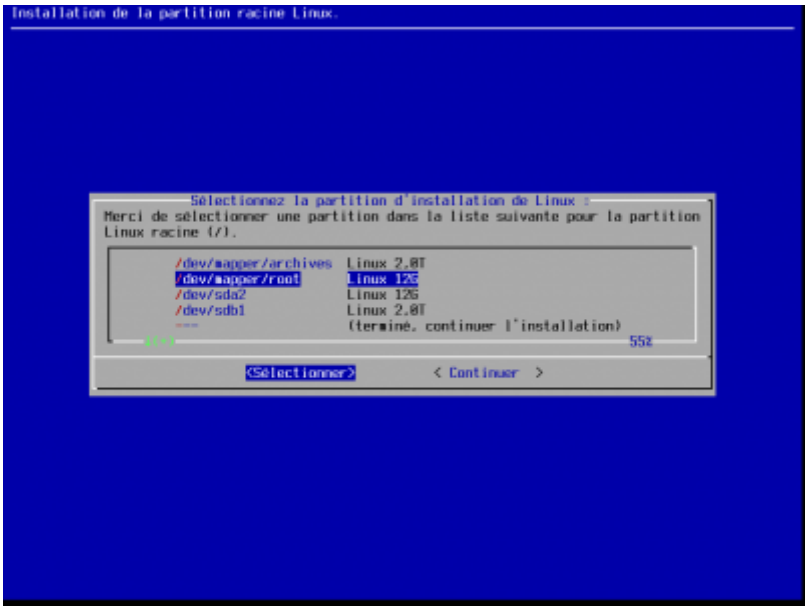
Utiliser la même phrase de passe pour toutes le partitions cryptées vous évite d'avoir à en mémoriser plusieurs.

Vous pouvez vérifier le résultat grâce à la commande `lsblk`.

Étape 4: formatage des partitions

Dans cette étape, l'important est de désigner chaque partition cryptée par le nom que vous venez de lui attribuer.

Dans l'exemple ci-dessous nous sélectionnerons `/dev/mapper/root` comme partition racine ou `/` puis `/dev/mapper/archives` en indiquant comme point de montage `/archives`.



Pour la partition de démarrage, deux cas se présentent:

- En mode UEFI, le système vous proposera de formater pour vous la partition EFI (/dev/sda1 dans notre exemple). Acceptez, c'est indispensable pour pouvoir amorcer le système après installation.
- En mode BIOS (dit aussi Legacy ou CSM pour Compatibility Support Module) formatez cette même partition avec comme point de montage /boot, C'est également indispensable.

Étape 6 : Choix et configuration de l'amorceur lors de la configuration.

Si vous avez démarré la machine en mode BIOS (ou Legacy, ou CSM), l'installateur vous proposera d'installer lilo. Acceptez. Quand on vous demandera où installer lilo, choisissez le MBR ou secteur d'amorçage principal.

Si vous avez démarré la machine en mode UEFI, la machine vous proposera de ne pas installer lilo mais d'installer plutôt elilo. Acceptez en choisissant “ne pas installer lilo”. Ensuite installez elilo et éventuellement une entrée pour Slint dans le menu d'amorçage du micrologiciel.

Notez bien qu'à ce stade le système ne peut en fait pas encore être amorcé car dans le cas d'une partition racine cryptée cela nécessite un `initrd`, comme nous allons le voir. Cependant, l'installateur aura déjà créé le fichier de configuration de l'amorceur. Nous devrons le modifier par la suite, mais cela nous donnera une base de travail.

Étape 8 : référencement des partitions cryptées et création d'un initrd

Tout d'abord, il faut enregistrer dans la table des partitions cryptées le lien entre le nom initial de la partition et celui que nous lui avons attribué après cryptage. Cette table sera nommée /etc/crypttab dans le système installée mais comme sa partition racine est pour le moment montée sur /mnt, le chemin vers ce fichier est /mnt/etc/crypttab. Il faut que ce fichier /mnt/etc/crypttab contienne les deux lignes suivantes:

```
root       /dev/sda2
archives   /dev/sdb1
```

Vous pouvez écrire ces lignes avec l'éditeur de texte nano, d'utilisaation facile, en le démarrant avec cette commande :

```
nano /mnt/etc/crypttab
```

N'oubliez pas d'enregistrer avant de quitter nano !

nano se commande au clavier. Les combinaison de touches associées aux commandes sont indiqués en bas de l'écran, sachant que le symbole ^ (accent circonflexe) représente la touche Ctrl. Par exemple la combinaison de touches pour quitter l'éditeur est ^X, c'est à dire Ctrl+X (maintenir Ctrl et taper X).

Ou bien vous pouvez alimenter directement ce fichier avec les commande suivantes, à taper telles quelles :

```
echo "root       /dev/sda2" >> /mnt/etc/crypttab
echo "archives   /dev/sdb1" >> /mnt/etc/crypttab
```

Dans tous les cas, vérifiez le contenu du fichier après édition grâce à cette commande :

```
cat /mnt/etc/crypttab
```

Vous allez maintenant crér un `initrd`, c'est à dire une archive compressée contenant un mini système Linux qui sera chargé en mémoire vive lors de l'amorçage su système, ainsi que le noyau. Cet initrd contiendra notamment des composants permettant d'ouvrir les partitions cryptées durant la séquence de démarrage du système.

Il faut tout d'abord passer de l'installateur au système en cours d'installation, grâce à la commande suivante :

```
chroot /mnt
```

Dorénavant, les commandes tapées s'appliqueront directement au système installé, notamment la création de l'initrd.

Cette création se fait grâce à la commande `mkinitrd` en indiquant comme arguments les composants que l'initrd doit inclure.

Pour éviter les erreurs, le mieux est de commencer par la commande suivante, dont le résultat sera

(après aménagement) la commande à taper pour créer l'initrd. Dans l'exemple suivant, nous indiquons la commande initiale et son résultat:

```
sh /usr/share/mkinitrd/mkinitrd/mkinitrd_command_generator.sh -r
mkinitrd -c -k 4.4.38 -f ext4 -r root -m jbd2:mbcache:ext4 -C /dev/sda2 -u -
o /boot/initrd.gz
```

voici quelques explications sur les arguments et options de la deuxième ligne (à taper ensuite avec des modifications que nous indiquerons) :

- -c (pour “clear”) permet de repartir de zéro dans la création de l'initrd au cas ou la commande mkinitrd aurait déjà été utilisée.
- -k (pour “kernel” ou noyau) indique le version du noyau à considérer, ici 4.4.38.
- -f (pour “fichier”) indique le système de fichier de la partition racine, ici ext4.
- -r (pour “root” ou racine) indique le nom de la partition racine du système à démarrer ici root (nom après cryptage)
- -m (pour “modules”) indique les modules du noyau à inclure dans l'initrd.
- -C (pour “cryptée”) indique les partitions cryptées à décrypter lors du démarrage, ici /dev/sda2 (au moment de monter /dev/sdb1 l'initrd aura déjà “rendu la main” au sytème installé sur le SSD).
- -u inclut udev (pour la gestion des périphériques) dans l'initrd.
- -o (pour “output”) indique le chemin vers le fichier initrd à écrire.

A la commande proposée par le script mkinitrd_command_generator.sh il faut ajouter l'option -L, qui indique d'ajouter à l'initrd les composants permettant le “mappage” entre noms de partitions avent et après cryptage, par exemple de faire la relation entre `/dev/sda2` et `root`.

Il faut aussi lui ajouter l'option -l (pour “langue”) qui inclura le code langue du plan de clavier utilisé pour entrer la phase de passe, pour que celle-ci soit reconnue en utilisant la même séquence de pressions de touches quand on vous la demandera au cours du démarrage du système.

Dans le case d'une installation en Français le plan de clavier utilisé pour taper la phrase de passe est `fr`, à moins que vous l'ayez changé avant *avant* de la taper (donc, avant de taper `setup`). Vous pouvez afficher le plan de clavier en cours avec la commande suivante:

```
cat /var/log/setup/tmp/SeTkeymap
```

Si le clavier *fr* a été utilisé, `fr.map` devrait être affiché. Dans ce cas on inclura dans la commande de création de l'initrd `-l fr`.

Si la commande affiche un autre résultat, remplacez *fr* par ce qui précède *.map seulement si* vous avez changé le plan de clavier *avant* de taper les phrase de passe.

Pour abréger une longue histoire, voici la commande à taper dans l'exemple. Bien entendu, il faut l'adapter au résultat donné par mkinitrd_command_generator.sh, avec un plan de clavier (argument de -l) différent le cas échéant.

```
mkinitrd -c -k 4.4.38 -f ext4 -r root -m jbd2:mbcache:ext4 -C /dev/sda2 -u -
o /boot/initrd.gz -L -l fr
```

Relisez soigneusement la commande avant d'appuyer sur [Entrée]

Si tout va bien un message s'affichera, se terminant par:

```
/boot/initrd.gz created.
Be sure to run lilo egain if you use it.
```

Avant de ré-exécuter éventuellement (c'est à dire, dans le cas d'un démarrage en mode BIOS et non UEFI) lilo, il faut modifier le fichier /etc/lilo.conf, comme indiqué ci-après.

Modification du chargeur initial et changement du noyau linux

Avec des partitions cryptées l'amorçage du système nécessite:

1. Un noyau réduit au minimum, désigné ici comme *générique* (generic en Anglais), qu nous nommerons vmlinuz-generic.
2. L'initrd que nous avons créé, nommé initrd.gz.
3. Un fichier de configuration qui les référence, nommé elilo.conf ou lilo.conf selon le cas.

La localisation de ces éléments diffère selon le chargeur d'amorçage utilisé:

elilo	**lilo**
/boot/efi/EFI/Slint/vmlinuz-generic	/boot/vmlinuz-generic
/boot/efi/EFI/Slint/initrd.gz	/boot/initrd.gz
/boot/efi/EFI/Slint/elilo.conf	/etc/lilo.conf

Le fichier de configuration existe déjà mais est à modifier.

D'autre part, dans le cas de elilo il faut copier l'initrd que nous avons créé dans /boot dans /boot/efi/EFI/Slint, supprimer le noyau déjà installé dans /boot/efi/EFI/Slint, nommé vmlinuz, et y copier le noyau vmlinuz-generic que l'installateur a placé dans /boot.

Pour plus de clarté, nous distinguerons les modifications à effectuer selon le chargeur initial utilisé, en rappelant qu'il s'agit de lilo pour une démarrage en mode BIOS (aussi appelé Legacy ou CSM), de elilo pour un démarrage en mode UEFI,

Modifications si lilo est utilisé

das ce cas, seule la modification de /etc/lilo.conf est nécessaire. Vous pouvez utiliser l'éditeur *vi* si vous le connaissez mais le plus simple est d'utiliser *nano*. Chargez d'abord le fichier /etc/lilo.conf grâce à cette commande:

```
nano /etc/lilo.conf
```

Ceci affichera son contenu.

Les modifications à effectuer, toutes concernant la fin du fichier, sont résumées ci-après

Ligne initiale	**Modification**
image = /boot/vmlinuz	

root = /dev/mapper/root	*supprimer cette ligne*
read-only	*insérer une nouvelle ligne au-dessous avec:* initrd = /boot/initrd.gz

Après ces modifications, la fin du fichier devrait être:

```
image = /boot/vmlinuz-generic
  label = Linux
  read-only
  initrd = /boot/initrd.gz
# Linux bootable partition config ends
```

La ligne commençant par “root =” n'est pas nécessaire car le nom de la partition racine est mémorisé dans l'initrd.

Vérifiez, sauvegardez, quittez, puis revérifiez en tapant:

```
cat /etc/lilo.conf
```

Puis, tapez:

```
lilo -t -v
```

Ici, -t est pour “test” et -v pour “verbeux” Si tout va bien (si le message en retour comporte éventuellement des “Warning” mais pas de “Fatal”), tapez: lilo Le message en retour devrait inclure cette ligne:

```
Added Linux + *
```

Sinon, corrigez le fichier /etc/lilo.conf avant de recommencer.

Vous êtes prêt à redémarrer la machine: tapez `exit` puis `reboot`.

Modifications si elilo est utilisé

Dans ce cas il faut supprimer le noyau que l'installateur a copié dans /boot/efi/EFI/Slint/ et le remplacer par le noyau linuz-generic, et copier l'initrd dans le même répertoire.

les commandes suivantes permettent d'effectuer ces modifications:

```
rm /boot/efi/EFI/Slint/vmlinuz
cp /boot/vmlinuz-generic /boot/efi/EFI/Slint
cp /boot/initrd.gz /boot/efi/EFI/Slint
```

Vous pouvez vérifier le résultat avec la commande suivante “ls -1 /boot/efi/EFI/Slint” qui devrait donner le résultat ci-dessous :

```
ls -1 /boot/efi/EFI/Slint
customization.msg
```

```
elilo.conf
elilo.efi
help.msg
initrd.msg
vmlinuz-generic
```

Ensuite, il faut modifier le fichier /boot/efi/EFI/Slint/elilo.conf pour lui donner le contenu suivant :

```
cat /boot/efi/EFI/Slint/
chooser=textmenu
prompt
timeout=200
#
# the files containing the text (with attributes) to display
#
message=message.msg
#
# files to load when the corresponding function key is pressed
#
f1=help.msg
f2=customization.msg
image=vmlinuz-generic
  label=Slint
  description="Start Slint 14.2"
  read-only
  initrd=initrd.gz
  append="vga=normal"
```

Pour ce faire, vous pouvez utiliser la commande:

```
nano /boot/efi/EFI/Slint/elilo.conf
```

Vérifiez bien après modification que vous obtenez exactement le résultat ci-dessus.

Dans ce cas, il n'y a pas de secteur d'amorçage à écrire, car le micrologiciel UEFI trouvera le fichier d'amorçage elilo.efi et tout le reste dans la partition /dev/sda1.

Vous pouvez donc redémarrer en tapant `exit`, puis `reboot`.