Full Disk Encryption sous Arch Linux (busybox)

Ce document est une méthode que je considère déprécié. je préconise maintenant Full Disk Encryption sous Arch Linux (systemd). Cette nouvelle méthode est plus simple et plus rapide.

Introduction

Cette documentation permet d'installer Arch Linux avec du chiffrement intégrale du disque dur en mode EFI.

AVANTAGE	**INCONVENIENT**
Chiffrement intégrale, même le /boot	La clé de chiffrement stocké à la racine
Utilisation de l'EFI	Lent au démarrage
Une seule passphrase à taper au boot	Gros problème de compatibilité entre le layout et les clavier USB
L'“hibernation” du système est fonctionnelle et chiffrée	

Cette solution repose sur un chiffrement par passphrase de la partition **/boot**, et du chiffrement des autres partitions par clé de chiffrement stocké dans l'image **INITRAMFS** (qui est stocké dans la partition **/boot**)

Voici l'agencement des partitions utilisé dans cette documentation. La seule réel contrainte sont les deux partitions **/boot** et **/boot/efi** :

```
+----------------------+-----------------------+-----------------------+----
------------------+
|ESP partition:        |Partition Boot:        |Partition Root:
|Partition SWAP         |
|                      |                       |                        |
|
|/boot/efi             |/boot                  |/
|[SWAP]                 |
|                      |                       |                        |
|
|                      |/dev/mapper/cryptoboot |/dev/mapper/cryptoroot
|/dev/store/cryptoswap  |
|                      |UUID=<UUID-VOL-BOOT>   |UUID=<UUID-VOL-ROOT>
|UUID=<UUID-VOL-SWAP>   |
|                     +----------------------+-----------------------+----
------------------+
|/dev/sda1             |/dev/sda2              |/dev/sda3
|/dev/sda4              |
|UUID=<UUID-PART-ESP>  |UUID=<UUID-PART-BOOT>  |UUID=<UUID-PART-ROOT>
|UUID=<UUID-PART-SWAP>  |
```

```
|non-chiffré           |chiffré                |chiffré
|chiffré                |
+----------------------+-----------------------+----------------------+----
------------------+
```

Installation

Boot en mode EFI

Il vous faut boot en mode EFI, pour vérifier, le dossier **/sys/firmware/efi/efivars/** doit exister

Changer le Layout du clavier

Vu qu'en France nous avons un clavier AZERTY, il faut le changer pour se simplifier la vie :

```
# loadkeys fr
```

Se connecter à Internet

Là je vous renvoie vers la documentation officiel, ce sujet est bien trop exhaustif

Partitionnement des disques

Pour cela, je vous renvoie vers de la documentation de fdisk

Il faut ensuite mettre le système de fichier FAT32 sur la partition **/boot/efi**

```
# mkfs.fat -F32 /dev/sda1
```

puis on va chiffré la partition **/boot** avec une PassPhrase :

```
# cryptsetup luksFormat /dev/sda2
```

Ensuite on va créer une clé de chiffrement pour les autres partitions:

```
# dd if=/dev/urandom of=crypto_keyfile.bin bs=512 count=4
```

Puis on chiffre les autres partitions ainsi que la partition boot avec cette dites clé

```
# cryptsetup luksAddKey /dev/sda2 crypto_keyfile.bin
# cryptsetup luksFormat /dev/sda3 crypto_keyfile.bin
# cryptsetup luksFormat /dev/sda4 crypto_keyfile.bin
```

Maintenant on va les déchiffrés :

```
# cryptsetup open /dev/sda2 cryptboot
# cryptsetup open /dev/sda3 cryptroot --key-file=crypto_keyfile.bin
# cryptsetup open /dev/sda4 cryptswap --key-file=crypto_keyfile.bin
```

puis de les formater :

```
# mkfs.ext4 /dev/mapper/cryptboot
# mkfs.ext4 /dev/mapper/cryptroot
# mkswap /dev/mapper/cryptswap
```

puis on les montes :

```
# mount /dev/mapper/cryptroot /mnt/
# mkdir /mnt/boot
# mount /dev/mapper/cryptboot /mnt/boot/
# mkdir /mnt/boot/efi
# mount /dev/sda1 /mnt/
```

Installation de Arch Linux

On va commencer par choisir notre miroir pour les dépots Arch en modifiant le fichier **/etc/pacman.d/mirrorlist** en sélectionnant le pays de son choix

On va ensuite mettre a jour les clés des packet

```
# pacman -Sy archlinux-keyring
```

et enfin on installe Arch en lancant :

```
# pacstrap /mnt base base-devel
```

On y copie la clé de chiffrement :

```
# cp crypto_keyfile /mnt
```

puis on se chroot pour faire la configuration:

```
# arch-chroot /mnt
```

Configuration de Arch

On commence par configurer le time zone :

```
# ln -sf /usr/share/zoneinfo/Europe/Paris /etc/localtime
# hwclock --systohc
```

Puis on génère le langage français :

```
# echo "fr_FR.UTF-8 UTF-8" > /etc/locale.gen && locale-gen && echo
"LANG=fr_FR.UTF-8" > /etc/locale.conf && echo "KEYMAP=fr" >
/etc/vconsole.conf
```

Puis on installe les paquets utiles :

```
# pacman -Syu efibootmgr grub
```

Ensuite on configure le nom d'hôte dans le fichier **/etc/hostname** :

```
Arch-0000
```

ainsi que le fichier **/etc/hosts** :

```
127.0.0.1   localhost.localdomain    localhost
::1     localhost.localdomain    localhost
127.0.1.1   Arch-0000.localdomain    Arch-0000
```

Configuration du boot

On modifie le fichier **/etc/default/grub**, il faut modifié cet ligne comme tel:

```
GRUB_CMDLINE_LINUX_DEFAULT="cryptdevice=UUID=<UUID-PART-SDA3>:cryptroot
resume=UUID=<UUID-VOL-CRYPTOSWAP>"
```

ainsi que la ligne suivante:

```
GRUB_TERMINAL_INPUT=at_keyboard
```

et pour finir décommente la ligne :

```
GRUB_ENABLE_CRYPTODISK=y
```

on ajoute dans le fichier /etc/grub.d/40_custom :

```
insmod keylayouts
keymap /boot/grub/fr.gkb
```

et on copie le fichier

fr.gkb

dans le dossier **/boot/grub/**

Ensuite on génère la config grub :

```
# grub-mkconfig -o /boot/grub/grub.cfg
# grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-
id=LINUX --recheck
```

Puis régénère le RAMDISK EFI de grub pour qu'il ajoute le clavier :

```
# grub-mkstandalone -d /usr/lib/grub/x86_64-efi/ -O x86_64-efi --
modules="part_gpt part_msdos crypto cryptodisk luks disk diskfilter" -o
"/boot/efi/EFI/LINUX/grubx64.efi" "boot/grub/grub.cfg=/boot/grub/grub.cfg"
"boot/grub/fr.gkb=/boot/grub/fr.gkb"
```

Maintenant on va générer l'image **initramfs** pour le déchiffrement, pour cela il faut aller modifier le fichier **/etc/mkinitcpio.conf** en modifiant le champ **HOOKS** de cet manière :

```
HOOKS=(base udev autodetect modconf keyboard keymap block encrypt openswap
resume filesystems fsck)
```

ainsi que le champ **FILES** comme ceci :

```
FILES=(/crypto_keyfile.bin)
```

Ensuite on va créer le HOOK openswap, donc on va créer le fichier **/etc/initcpio/install/openswap** comme ceci :

openswap

```
build ()
{
add_runscript
}
help ()
{
cat<<HELPEOF
This opens the swap encrypted partition /dev/sda3 in
/dev/mapper/cryptswap
HELPEOF
}
```

ainsi que le fichier **/etc/initcpio/hooks/openswap**

openswap

```
run_hook ()
{
cryptsetup open --key-file=/crypto_keyfile.bin /dev/disk/by-uuid/<UUID-
PART-SWAP> cryptswap
rm -f /crypto_keyfile.bin
}
```

puis on exécute la commande[1)] :

```
# sed -i 's/rm -f ${ckeyfile}//g' /usr/lib/initcpio/hooks/encrypt
```

puis on génère l'image :

```
# mkinitcpio -p linux
```

puis on créer le fichier **/etc/crypttab** en ajoutant les volumes suivants :

```
cryptoboot     UUID=<UUID-PART-BOOT>     /crypto_keyfile.bin
cryptoroot     UUID=<UUID-PART-ROOT>     /crypto_keyfile.bin
```

à noté que le déchiffrement de la partition de la partition **swap** à déjà été réalisé durant le HOOK de l'**initramfs**, d'où son absence dans le fichier.

Et pour finir le fichier **/etc/fstab** :

```
UUID=<UUID-PART-ESP>          /boot/efi          vfat        defaults
0 1
UUID=<UUID-VOL-BOOT>          /boot              ext4        defaults
0 1
UUID=<UUID-VOL-ROOT>          /                  ext4        defaults
0 1
UUID=<UUID-VOL-SWAP>          none               swap        defaults
0 2
```

[1)]
Cette commande (et la suivante) devra peut-être être retapé en cas de mise a jour de initcpio. Elle sera nécessaire si vous avez l'erreur “failed to open key file” lors du lancement du hook “openswap” au démarrage

From:
https://wiki.virtit.fr/ - **VirtIT**

Permanent link:
https://wiki.virtit.fr/doku.php/kb:linux:donnees:full_disk_encryption_sous_arch_linux_busybox

Last update: **2019/09/26 15:01**