# Full Disk Encryption sous Arch Linux (busybox)

Ce document est une méthode que je concidère déprécié. je préconise maintenant Full Disk Encryption sous Arch Linux (systemd)

## Introduction

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

AVANTAGE	INCONVENNIANT
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	

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 :

<b>4</b>	<b></b>	.+	<b>. </b>
ESP partition:  Partition SWAP	Partition Boot:   	Partition Root:	1
/boot/efi  [SWAP] 	/boot   	I/ I	I
  /dev/store/cryptoswap    UUID= <uuid-vol-swap></uuid-vol-swap>	  UUID= <uuid-vol-boot>  </uuid-vol-boot>	<pre> /dev/mapper/cryptoroot  UUID=<uuid-vol-root> -+</uuid-vol-root></pre>	+
/dev/sda1  /dev/sda4  UUID= <uuid-part-esp>  UUID=<uuid-part-swap>  non-chiffré  chiffré</uuid-part-swap></uuid-part-esp>	/dev/sda2    UUID= <uuid-part-boot>    chiffré  </uuid-part-boot>	/dev/sda3  UUID= <uuid-part-root>  chiffré</uuid-part-root>	

+-----+

### **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/sdal /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/ -0 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/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
}</pre>
```

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<sup>1)</sup>:

```
# 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-b00t></uuid-part-b00t>	/crypto_keyfile.bin
cryptoroot	UUID= <uuid-part-r00t></uuid-part-r00t>	/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> 0 1</uuid-part-esp>	/boot/efi	vfat	defaults
UUID= <uuid-vol-boot> 0 1</uuid-vol-boot>	/boot	ext4	defaults
UUID= <uuid-vol-root> 0 1</uuid-vol-root>	/	ext4	defaults
UUID= <uuid-vol-swap> 0 2</uuid-vol-swap>	none	swap	defaults

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?rev=1569507703

Last update: 2019/09/26 14:21

