Démarrer via le terminal initramfs

> L'invite de commande `initramfs>` correspond à un shell de secours fourni par l'initramfs (Initial RAM Filesystem) lorsque le processus de démarrage normal échoue.

Dans notre cas, cela peut arriver si notre configuration est erroné, alors au démarrage le serveur proposera une interface semblable à ceci :

image.png

Pas de panique, ce tuto permet de débloquer cette situation afin de faire démarrer le serveur.

Déverrouiller le disque principale

Tout d'abord, si on est dans l'invite de commande de l'initramfs, c'est que le disque principal n'a pas était déverrouillé via notre fichier clé présent dans la clé USB.

Donc la première étape est d'identifier le slot du disque principal ainsi que sa partition pour faire la commande `cryptsetup` pour le déverrouiller à la main.

Pour se faire la commande `blkid` fonctionne et va donner une résultat ressemblant à :

```
/dev/sdb1: UUID="868906c1-d1db-4431-aa76-44d7babb6798" BLOCK_SIZE="4096" TYPE="ext4"
PARTUUID="XXXXXXX-XX"
/dev/sda5: UUID="f6d7xxx-xxx-xxx-xxx-xxxx" TYPE="crypto_LUKS" PARTUUID="456azee1-05"
/dev/sda1: UUID="f831bxxxx-xxx-xxx-xxx-xxxxx" BLOCK_SIZE="1024" TYPE="ext2"
PARTUUID="456azee1-01"
```

Nous voyons bien notre clé USB (sdb1) ainsi que le disque dur (sda1) avec la partition chiffré (sda5). Nous pouvons remarquer que la partition UUID de notre HDD principal est la même pour sda5 et sda1.

Maintenant que nous connaissons les slots de nos différents périphériques nous avons 2 possibilités pour nous débloquer de cette situation :

- Option 1 : Monter la clé USB et utiliser le fichier clé pour déverrouiller la partition LUKS. Puis monter le système à la main.
  - Cela permettrai de tester que tout se passe bien durant la procédure de démarrage dans l'initramfs.
- Option 2 : Déverrouiller la partition LUKS avec le mot de passe puis de monter le système à la main
  - Si vous souhaitez directement modifier des fichiers dans votre systèmes (le fstab, modules, scripts, etc..)

Option 1 - Monter la clé USB et utiliser le fichier de la clé

Pour cette étape nous commençons par créer le dossier qui servira de point de montage. Dans le système de l'initramfs le dossier "mnt" n'existe pas.

```
mkdir /mnt
```

Maintenant on va monter la clé USB en utilisant la même commande utilisé dans le fichier de montage automatique :

```
mount -t ext4 /dev/sdb1 /mnt
```

> Techniquement si le montage ne fonctionne pas avec cette commande c'est que le problème vient probablement de cette étape.

Afin de vérifier que tout c'est bien passé vous êtes sensé voir votre fichier dans le dossier mnt.

```
ls /mnt
> keyfile
```

Si on a bien le fichier on va pouvoir utiliser la deuxième commande utilisé dans le fichier de montage automatique :

```
cryptsetup luksOpen /dev/sda5 sda5_crypt --key-file /mnt/keyfile
```

> Techniquement si le déverrouillage ne fonctionne pas avec cette commande c'est que le problème vient probablement de cette étape.

Maintenant que la partition LUKS est déverrouillé nous pouvons passer à l'étape de montage du système Linux.

Option 2 - Déverrouiller la partition LUKS avec le mot de passe

Monter le système Linux

Pour réaliser cette étape, le déverrouillage de la partition LUKS est nécessaire. Le déblocage de la partition LUKS doit faire apparaître des mapper :

```
ls /dev/mapper
> control  sda5_crypt  sirius--vg-root  sirius--vg-swap_1
```

> Vérifier que les mappers existent bien

Tout d'abord si vous avez réaliser l'option 1 vous devez démonter la clé USB :

```
umount /mnt
# OU
umount /dev/sdb1
```

Maintenant nous allons réutiliser le dossier mnt pour monter le système dedans :

```
mount -t ext4 /dev/mapper/sirius--vg-root /mnt
```

Puis monter les dossiers nécessaires pour que le vrai système fonctionne :

```
mount -t proc /proc /mnt/proc
mount -t sysfs /sys /mnt/sys
mount --bind /dev /mnt/dev
```

et ainsi faire basculer la racine dans ce nouveau point de montage :

```
chroot /mnt
```

Nous sommes maintenant dans notre vrai systèmes. Mais ce n'est pas encore finis !

Pour retravailler le initramfs il faut monter également le `boot` qui se trouve dans une autre partition non chiffré.

La commande `lsblk -f` doit lister les périphériques ainsi que leurs partitions. Parmi la liste des partitions de notre disque principal nous devons en avoir une qui corresponds au `boot`.

```
sdb

├─sdb1                  ext2        1.0            978c28c7-0e7b-4172-ac8c-411d91a045f9
309M    27%
```

```
├─sdb2

└─sdb5                  crypto_LUKS 2              a27bc3e3-1d00-42bf-a312-
d77c19132baf
  └─sda5_crypt          LVM2_member LVM2 001       4da22ec6-4b19-4c5d-8080-
e5eeed0ec682
    ├─sirius--vg-root   ext4        1.0            935e9b50-e913-4af5-9e44-c68caf7393fc
102,8G     6% /
    └─sirius--vg-swap_1 swap        1              afa4e557-7b48-4f8a-96ec-
391a3c663d87                  [SWAP]
```

Dans notre cas c'est la partition `sdb1`.

Donc on fait la commande suivante pour monter le boot dans notre système : `mount /dev/sda1 /boot`

Puis vérifier que tout le montage c'est bien passé avec la commande `ls /boot` le dossier ne doit pas être vide.

Revision #5
Created 2025-07-03 07:28:27 UTC by gpatruno
Updated 2025-07-03 10:06:23 UTC by gpatruno