Accès aux installations de linux après réactivation du secure boot

Le fait d'avoir rétabli le **Secure Boot** bloque probablement le démarrage d'Ubuntu 26.04 car son chargeur d'amorçage (shim) ou son noyau n'est plus reconnu comme signé et digne de confiance par le firmware UEFI.
Voici les solutions :

1. Désactiver temporairement le Secure Boot (solution rapide)

Le moyen le plus simple de retrouver l'accès à Ubuntu est de désactiver le Secure Boot dans le **BIOS/UEFI** de votre machine.

1. Redémarrez et appuyez sur la touche d'accès au BIOS (souvent F2, F10, Del).
2. Allez dans l'onglet **Security** ou **Boot**.

- Désactivez **Secure Boot**.
- Sauvegardez et quittez. Ubuntu devrait alors démarrer normalement.

2. Réactiver le Secure Boot avec une clé MOK (solution permanente)

Si vous souhaitez garder le Secure Boot activé, votre installation d'Ubuntu a probablement besoin d'enregistrer sa clé propriétaire (MOK).

1. **Désactivez temporairement le Secure Boot** (comme ci-dessus) pour démarrer sur Ubuntu.

- Ouvrez un terminal et vérifiez l'état :

```
sudo mokutil --sb-state
```

- Si des clés doivent être enregistrées, redémarrez. Pendant le démarrage, une interface **MOK Management** (blue screen) devrait apparaître.
- Suivez les instructions pour **enregistrer une nouvelle clé (Enroll MOK)**.
- Une fois l'inscription réussie, **réactivez le Secure Boot** dans le BIOS/UEFI.

Cela permettra à Ubuntu de démarrer tout en conservant la sécurité du Secure Boot.

Comment obtenir une clé MOK ?

Pour obtenir une clé **MOK (Machine Owner Key)** afin de réactiver **Secure Boot** de façon permanente, vous devez générer une paire de clés (privée/publique) que le gestionnaire **shim** pourra reconnaître. Voici la procédure :

Générer une clé MOK

1. **Créez le répertoire, s'il n'existe pas, pour stocker la clé :

```
sudo mkdir -p /var/lib/shim-signed/mok
cd /var/lib/shim-signed/mok
```

3. **Générez la clé privée et le certificat DER** (format requis par UEFI) :

```
sudo openssl req -new -x509 -newkey rsa:2048 -keyout MOK.priv -outform DER -out MOK.der -days 36500 -subj "/CN=MaCléMOK/" -nodes
```

5. **Convertissez le certificat en format PEM** (nécessaire pour certaines opérations) :

```
sudo openssl x509 -inform DER -in MOK.der -out MOK.pem
```

7. **Restreignez les permissions** :

```
sudo chmod 600 MOK.priv
sudo chmod 644 MOK.der MOK.pem
```

Enregistrer la clé dans le système (Bios)

1. **Importez la clé pour enregistrement** :

```
sudo mokutil --import MOK.der
```

- Vous devrez définir un **mot de passe temporaire** (à utiliser au prochain démarrage).

2. **Redémarrez** :

```
sudo reboot
```

3. **Au démarrage**, l'écran **MokManager** (bleu) apparaît :

- Sélectionnez **« Enroll MOK » → « Continue » → « Yes »**
- Saisissez le mot de passe défini précédemment
- Confirmez et redémarrez

4. **Vérifiez l'enregistrement** :

```
sudo mokutil --list-enrolled | grep "MaCléMOK"
```

Après avoir "enroll" le nouvel OS

1- Aller dans le bios et activer si nécessaire le secureboot
2- Aller dans le menu boot :
  Sélectionner l'entrée que vous venez de créer
3- Redémarrer après avoir sauvegardé
Le démarrage devrait alors fonctionner

> **Nota** : il ne faut pas choisir le nouvel OS que vous venez d'enregistrer sans changer le démarrage dans le bios : Si par exemple vous aviez juqu'à présent un menu Ubuntu séléectionné, le nouvel OS qui n'est pas Ubuntu ne fonctionnera pas en mode secureboot activé. Il faut donc passerr par le bios et sélectionner le nouvel OS qui va maintenant apparaître et ensuite sauvegarder en sortant du bios. Le nouvel OS devrait fonctionner avec secure boot activé.

Si vous avez besoin de votre clé MOK (ex : pour des modules DKMS)

Alors, **ne signez pas **shim, mais **signez uniquement les modules** (comme vboxdrv, nvidia.ko, etc.) avec votre clé MOK, **sans toucher au chargeur**.

- Utilisez sign-file pour signer les modules.
- Assurez-vous que le shim et grub restent ceux signés par Debian.

□□ Vérification finale

Après correction :

```
mokutil --sb-state
```

Doit afficher : `SecureBoot enabled` Et :

```
dmesg | grep -i secureboot
```

Doit confirmer que le noyau a démarré en mode Secure Boot.

Comment gérer plusieurs clés MOK ?

Il est possible de gérer **plusieurs clés MOK (Machine Owner Key)** sur un même système. Chaque clé est stockée dans la **NVRAM UEFI** et peut être utilisée pour signer des modules noyau ou des binaires EFI. Voici comment les gérer efficacement.

1. Lister les clés MOK existantes

Pour voir les clés déjà inscrites :

```
sudo mokutil --list-enrolled
```

Pour voir les demandes d’importation ou de suppression en attente :

```
sudo mokutil --list-new
sudo mokutil --list-delete
```

2. Importer plusieurs clés

Vous pouvez importer plusieurs clés .der indépendamment :

```
sudo mokutil --import /chemin/vers/MOK1.der
sudo mokutil --import /chemin/vers/MOK2.der
```

Un mot de passe temporaire est demandé pour chaque clé. L'inscription se fait au **redémarrage** via l'interface **MokManager**.

3. Supprimer une clé MOK inutile

Si une clé n'est plus nécessaire :

```
sudo mokutil --delete /chemin/vers/MOK.der
```

Le fichier .der original est requis. Au redémarrage, **MokManager** demandera confirmation de la suppression.

4.Exporter les clés pour sauvegarde

Exporter les clés inscrites :

```
sudo mokutil --export
```

Cela sauvegarde les certificats .cer dans le répertoire courant.

5. Bonnes pratiques

- **Nommez vos clés différemment** (ex. MOK-nvidia.der, MOK-vbox.der) pour éviter les conflits.
- Stockez les clés dans /var/lib/shim-signed/mok/ ou /etc/keys/.
- Une **limite matérielle** existe (dépend de la NVRAM), mais une dizaine de clés est généralement supportée.

.Pour définir l’ordre de démarrage :

Pour définir l'ordre de démarrage avec efibootmgr, utilisez l'option -o ou --bootorder suivie de la liste des identifiants de démarrage, séparés par des virgules.

Commande de base

```
sudo efibootmgr -o XXXX,YYYY,ZZZZ
```

Remplacez `XXXX`, `YYYY`, `ZZZZ` par les numéros des entrées de démarrage (visibles via `efibootmgr` sans arguments), dans l'ordre souhaité.

Exemple

Si vous souhaitez que **Windows Boot Manager** (Boot0005) démarre en premier, suivi d' **Ubuntu** (Boot0006), puis d'autres entrées :

```
sudo efibootmgr -o 0005,0006,0001,0002
```

□□ Utilisez efibootmgr ou efibootmgr -v pour afficher la liste actuelle des entrées et leur ordre.

Cas à gérer :

J'ai les entrées suivantes :
Boot0001* Ubuntu HD(..à quoi correspond ce contenu ?.)/\EFI\ubuntu\shimx64.efi pour l'entrée unbuntu 24.4.
Boot0004* debian, HD(...)/\EFI\debian\shimx64.efi
Boot0002* Linux Boot Manager HD()\EFI\systemd\systemd-bootx64.efi
Boot0003* NixOS-boot HD()/\EFI\NixOS-boot\grubx64.efi
L'entrée 4 debian ne fonctionne pas avec secureboot activé : pourquoi ?
les entrées 2 et 3 ne fonctionnent pas

Pouvez vous m'expliquer ?

Explication des entrées UEFI et problèmes liés à Secure Boot

Les entrées UEFI que vous observez correspondent à différents chargeurs d'amorçage installés sur votre système. Voici ce qu'elles signifient :

- *Boot0001 Ubuntu**: Pointe vers \EFI\ubuntu\shimx64.efi. C’est le chargeur utilisé pour démarrer Ubuntu avec **Secure Boot activé**, car shimx64.efi est signé par Microsoft.
- *Boot0004 Debian**: Pointe vers \EFI\debian\shimx64.efi. Normalement, cette entrée devrait fonctionner avec Secure Boot, **mais seulement si** :
- Le fichier shimx64.efi est correctement signé.
- L’entrée UEFI est bien configurée et reconnue par le firmware.
- Aucune corruption ou mauvaise installation du paquet shim-signed n’a eu lieu.
- *Boot0002 Linux Boot Manager**: Correspond à systemd-bootx64.efi. Ce chargeur **ne supporte pas nativement Secure Boot** sauf si le binaire est signé ou si le firmware l’a explicitement approuvé.
- *Boot0003 NixOS-boot**: Utilise grubx64.efi via GRUB. **GRUB non signé ne fonctionne pas avec Secure Boot**, sauf si shim est utilisé comme intermédiaire.

Pourquoi Debian (entrée 4) ne démarre pas avec Secure Boot ?

L’entrée Debian ne fonctionne pas car, bien qu’elle pointe vers shimx64.efi, il est possible que :

- Le paquet shim-signed ou grub-efi-amd64-signed ne soit **pas correctement installé**.
- Le firmware **n’a pas chargé la clé Debian** via shim, ou une mise à jour a corrompu la chaîne de confiance.
- L’entrée NVRAM est mal configurée ou obsolète.
  Solution : Réinstallez les paquets nécessaires avec :

```
sudo apt install --reinstall shim-signed grub-efi-amd64-signed
```

Puis réinstallez GRUB :

```
sudo grub-install --target=x86_64-efi --efi-directory=/boot/efi --uefi-secure-boot
```

Pourquoi les entrées 2 (**systemd-boot**) et 3 (**GRUB NixOS**) ne fonctionnent pas ?

- systemd-boot** (entrée 2)** : Ce chargeur **n’est pas signé** pour Secure Boot. Il ne démarrera pas si Secure Boot est activé, sauf si vous avez désactivé la vérification ou ajouté vos propres clés.
- GRUB NixOS** (entrée 3)** : Utilise grubx64.efi directement. **Sans **shim, ce binaire n’est pas approuvé par Secure Boot, donc le démarrage échoue.
  Rappel : Seuls les chargeurs signés (comme shimx64.efi) peuvent démarrer sous Secure Boot. grubx64.efi ou systemd-bootx64.efi doivent passer par shim ou être signés manuellement.

Résumé

Entrée	Chargeur	Fonctionne avec Secure Boot ?	Cause probable d’échec
Ubuntu	shimx64.efi	Oui	Correctement configuré
Debian	shimx64.efi	Non	Paquets mal installés ou clé non chargée
systemd-boot	systemd-bootx64.efi	Non	Non signé, incompatible sans configuration
NixOS (GRUB)	grubx64.efi	Non	Lancé directement, sans shim

Quelques explications sur le MOK

La désactivation de Secure Boot est un paramètre de micrologiciel ("BIOS setting", bien que tous les systèmes avec Secure Boot aient un UEFI au lieu du BIOS hérité).
Donc, à moins que vous ne réinitialisiez vos paramètres de micrologiciel (nom de l'héritage: "réinitialisation CMOS"), ou que vous effectuiez une mise à jour du micrologiciel système ("mise à jour BIOS") et que la mise à jour réinitialise vos paramètres de micrologiciel pour vous, Secure Boot devrait rester désactivé à moins que vous ne l'activiez à nouveau.
Réinstaller le système d'exploitation ou le pilote Nvidia ne devrait pas du tout avoir d'effet sur l'état de démarrage sécurisé.

Le "mot de passe MOK" semble souvent mal compris.
Ubuntu inclut l'automatisation pour créer et enregistrer une clé de propriétaire de machine (MOK) pour Secure Boot, si le système a Secure Boot activé. Le processus d'enregistrement peut être démarré, mais ne peut pas être terminé pendant l'exécution d'un système d'exploitation, car le processus d'enregistrement doit être certain que la commande d'enregistrement du MOK provient réellement de l'utilisateur, et non de tout programme qui prétend être l'utilisateur.
Ainsi, lorsque Ubuntu crée un MOK, il vous oblige à définir un mot de passe unique pour compléter le processus d'enregistrement MOK. La prochaine fois que vous démarrez, Secure Boot validera d'abord la signature de Microsoft sur le shimx64.efiChargeur de démarrage sécurisé, puis shimx64.efidétectera qu'un processus d'enregistrement MOK a été lancé. Il interrompra le processus de démarrage et vous présentera l'écran bleu MOK Manager. Si vous choisissez de terminer le processus d'inscription MOK, il vous demandera de saisir le mot de passe que vous avez défini avant de redémarrer pour confirmer que 1.) vous êtes l'utilisateur qui a commencé le processus et 2.) vous voulez vraiment terminer le processus d'enregistrement MOK.
Après cela, le "mot de passe MOK" n'est plus jamais nécessaire: son travail est fait. Si le MOK est perdu parce que les paramètres du micrologiciel système sont réinitialisés pour une raison quelconque, vous devrez refaire le processus d'enregistrement MOK dès le début.

Le MOK réel sera situé dans /var/lib/shim-signed/mok/répertoire une fois qu'il a été créé. Il s'agit d'une clé cryptographique en deux parties: la partie publique, également appelée *certificat*, sera dans le MOK.derfichier. C'est la partie qui est enregistrée dans le firmware du système. Vous pouvez le visualiser sous une forme lisible par l'homme avec n'importe quelle commande qui peut gérer les certificats X.509 au format DER, par exemple:
sudo openssl x509 -inform DER -in /var/lib/shim-signed/mok/MOK.der -noout -text
La partie privée *sera seulement* dans le MOK.privfichier, lisible uniquement par root. Ce fichier sera automatiquement utilisé par dkmset d'autres outils de gestion de noyau et de modules si nécessaire pour signer par ex. Les modules de pilote Nvidia dans le cadre de leur processus d'installation si Secure Boot est activé.

Récupéré sur le web

J'ai un ordinateur portable double boot Windows 10 / Linux mint 20. Démarrage sécurisé activé et aussi cryptage de disque dur, mais ce dernier n'est peut-être pas important pour la question.

En passant, ma question est très similaire à celle-ci: https://forums.linuxmint.com/viewtopic.php?t=274365
J'ai installé Windows et après ça - Linux Mint. Après l'installation de la Monnaie, l'ordinateur a redémarré et on m'a demandé de "Continuer le démarrage" ou "Enrôler MOK". Je ne savais pas quoi faire et je suis allé chercher en ligne sur l'autre ordinateur portable. Pendant la recherche, le dialogue semble chronométré et l'ordinateur a juste continué à démarrer. L'ordinateur portable est Dell Vostro 5581 mis à jour vers le dernier BIOS.
Plus tard le même jour, j'ai installé Virtualbox à partir du site Web d'Oracle (pas à partir du dépôt). Lors de l'installation en mode texte sur la console il m'a demandé de confirmer que je voulais

inscrire MOK sur le prochain redémarrage et entrer un mot de passe temporaire. J'ai fait et redémarré et inscrit le MOK. (Ne sachant pas ce que je fais, au fait)
Alors, voici mes questions. Tout ce truc de démarrage sécurisé est très nouveau pour moi.

1. Quelle est la boîte de dialogue initiale "Continuer le démarrage" ou "Enrôler MOK" qui s'affiche lorsque vous installez Mint et redémarrez pour la première fois? Cela apparaît avant même que le système d'exploitation ne démarre. Je pense que c'est un truc de BIOS. Et cela ne semble pas être important si vous continuez simplement à démarrer ou à inscrire la clé. J'ai fait 2 installations Linux et dans la 1-st, j'ai choisi d'inscrire la clé, mais sur le 2-et l'a ignoré. Il ne semblait pas y avoir de différence.
2. Si j'avais inscrit la clé MOK, Virtualbox aurait-il installé sans me demander de faire quoi que ce soit? Qu'est-ce que VirtualBox exactement quand il inscrit sa propre clé?
3. Comment puis-je faire le "Enroll MOK" maintenant après avoir installé et configuré mon système et vraiment ne pas vouloir réinstaller à nouveau? L'autre question sur le forum (voir lien ci-dessus) a une réponse disant:
   J'ai eu le même problème. Pour définir un nouveau mot de passe MOK, j'ai utilisé la commande
   sudo update-secureboot-policy --enroll-key
   Cependant, sur mon installation, il n'y a pas de telle commande update-secureboot-policy.
4. Maintenant, j'ai peur d'installer les pilotes propriétaires NVidia, parce que je n'ai pas inscrit MOK et j'ai peur que cela ne fonctionne pas.
5. Et, généralement, que fait cette chose "Enrôlement MOK" après le 1-st reboot? Je ne comprends vraiment pas. Cela signifie-t-il qu'il met certaines clés Ubuntu dans le BIOS? Est-ce que cela signifie que si je le fais, alors tous les futurs modules de noyau propriétaires que j'installe se produiront sans problème sans inscrire leurs propres MOK?

Réponses

1. Quelle est la boîte de dialogue initiale "Continuer le démarrage" ou "Enrôler MOK" qui s'affiche lorsque vous installez Mint et redémarrez pour la première fois?

Cela est produit par shimx64.efilorsqu'il détecte qu'il existe un nouveau MOK dans une variable UEFI NVRAM accessible au système d'exploitation, en attente d'installation.

2. Si j'avais inscrit la clé MOK, Virtualbox aurait-il installé sans me demander de faire quoi que ce soit?

Très probablement, oui.

2.5. Qu'est-ce que VirtualBox exactement quand il inscrit sa propre clé?

Ça ne fait probablement que déclencher update-secureboot-policy --enroll-keysi elle est disponible.

3. Comment puis-je faire le "Enroll MOK" maintenant après avoir installé et configuré mon système et vraiment ne pas vouloir réinstaller à nouveau?

sudo apt install shim-signed
sudo update-secureboot-policy --enroll-key

4. Maintenant, j'ai peur d'installer les pilotes propriétaires NVidia, parce que je n'ai pas inscrit MOK et j'ai peur que cela ne fonctionne pas.

Techniquement pas une question, mais ne vous inquiétez pas. Si vous installez le pilote NVidia via l'outil de gestion de pilote tiers d'Ubuntu/Mint, il fera probablement les étapes énumérées dans 3.) ci-dessus pour vous si vous n'avez pas déjà fait cela.
Si vous utilisez le package d'installation téléchargé directement depuis NVidia, installez d'abord un dkmsoutil de gestion pour les modules tiers, puis exécuter le programme d'installation du pilote NVidia:

```
sudo apt install dkms


sudo ./NVIDIA-Linux-x86_64-<version number>.run --dkms \
    --module-signing-secret-key=/var/lib/shim-signed/mok/MOK.priv \
    --module-signing-public-key=/var/lib/shim-signed/mok/MOK.der
```

dkmsautomatise la reconstruction des modules de noyau tiers (comme le pilote NVidia) de sorte que vous n'aurez pas à le faire manuellement chaque fois que vous recevez une mise à jour de sécurité du noyau.

5. Et, généralement, que fait cette chose "Enrôlement MOK" après le 1-st reboot?

Si vous ne faites pas le "Enroll MOK" sur le prochain redémarrage juste après la course update-secureboot-policy --enroll-key, la procédure d'inscription sera en attente, en attendant que vous *le complétiez* en sélectionnant "Enrôler MOK" sur une chaussure ultérieure, ou *de l'annuler* avec sudo mokutil --revoke-importdans Linux.
Une fois que vous avez terminé la procédure d'inscription MOK, vous ne devriez pas voir cette

invite à nouveau à moins de perdre l'ancien MOK et d'en inscrire un nouveau.

5.1. Cela signifie-t-il qu'il met certaines clés Ubuntu dans le BIOS?

Non, la procédure d'inscription fait une clé unique à **votre** système et le place dans /var/lib/shim-signed/mok/accessible à root uniquement, de sorte que les processus d'installation du module du noyau peuvent l'utiliser, et inscrit une copie de la partie publique de la clé à une variable UEFI NVRAM, de sorte qu'il peut être utilisé par shimx64.efilors du démarrage.

5.2. Est-ce que cela signifie que si je le fais, alors tous les futurs modules de noyau propriétaires que j'installe se produiront sans problème sans inscrire leurs propres MOK?

C'est ça l'idée, oui. Malheureusement, tous les paquets sources de modules de noyau tiers n'ont pas encore été mis à jour pour détecter de manière transparente la présence de MOK et l'utiliser automatiquement si nécessaire.
sur mon installation, il n'y a pas de telle mise à jour de commande-secureboot-policy
Sur mon système Ubuntu, cette commande est dans le shim-signedpaquet.

Revision #2
Created 13 May 2026 16:56:40 by Thorgan
Updated 13 May 2026 17:26:08 by Thorgan