

Installation Guide

Customer Reference • Document Classification: Public

Overview

Please check the signing document below before installing the components.

Please run following commands to install all the necessary components

```
1: sudo chmod +x install.sh
2: sudo ./install.sh
```

Note: the kernel module is compiled under kernel version 6.8.0-40-generic and ubuntu version 22.04(<https://releases.ubuntu.com/>).

If there is any change on kernel version, please contact SE support, otherwise the kernel module will not work.

Secure Boot

Enrollment & Kernel Module Signing Guide

Customer Reference • Document Classification: Public

Overview

This document provides step-by-step instructions for enrolling a Machine Owner Key (MOK) and signing a custom kernel module to enable compatibility with UEFI Secure Boot on Ubuntu-based systems.

These steps are required when deploying a custom kernel or loading a custom kernel module (usbcmd.ko) that is not signed by a trusted certificate authority recognized by the system firmware.

Prerequisites

Before proceeding, ensure the following are in place:

- Ubuntu system with UEFI Secure Boot enabled
- Administrative (sudo) access to the target system
- OpenSSL installed (openssl command available)
- The mokconfig.cnf template file — edit all fields enclosed in < > before use
- The usbcmd.ko kernel module from the components/driver folder

Part 1: Enrolling the MOK Key

A Machine Owner Key (MOK) is a cryptographic key pair that allows you to sign custom kernel modules. The public certificate must be enrolled in the system firmware before signed modules will load under Secure Boot.

Step 1 — Generate the MOK Key Pair

From the directory containing your mokconfig.cnf file, run the following command to generate a new RSA 2048-bit key pair. This produces two files: MOK.priv (private key) and MOK.der (public certificate).

```
openssl req -config ./mokconfig.cnf \  
    -new -x509 -newkey rsa:2048 \  
    -nodes -days 36500 -outform DER \  
    -keyout "MOK.priv" \  
    -out "MOK.der"
```

Note: Keep MOK.priv secure. This private key must not be distributed. Only MOK.der (the public certificate) needs to be enrolled on the system.

Step 2 — Import the Certificate for Enrollment

Import the MOK certificate into the MOK enrollment queue. You will be prompted to create a one-time password. This password is used in the next step to confirm the enrollment.

```
ubuntu@localhost:~$ sudo mokutil --import MOK.der  
input password:  
input password again:
```

⚠ Important: The certificate is not enrolled yet at this point. It is queued for enrollment by the shim bootloader on the next reboot. Proceed to the next step.

Step 3 — Verify the Import Queue

To confirm that the certificate has been successfully queued for enrollment, run:

```
ubuntu@localhost:~$ sudo mokutil --list-new
```

Review the output to confirm your certificate details appear. Once verified, reboot the system to proceed with the firmware-level enrollment.

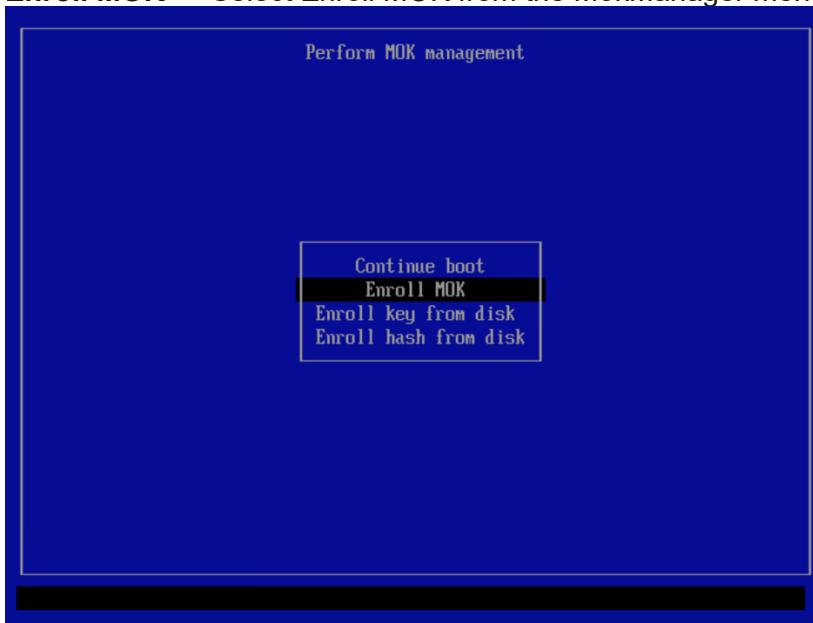
Step 4 — Complete Enrollment via MokManager

Upon reboot, the shim bootloader will detect the pending MOK enrollment and launch MokManager before loading GRUB. Follow these steps at the MokManager screen:

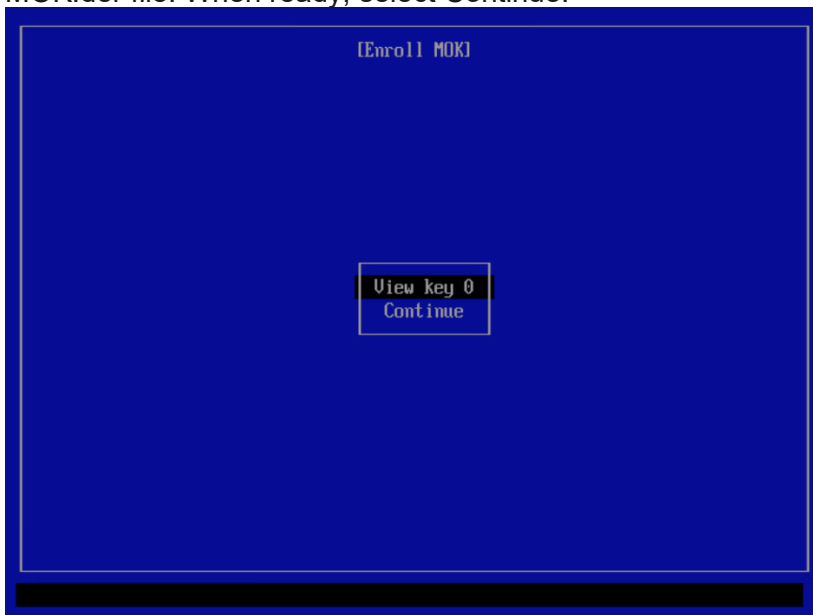
1. **Blue Screen Prompt** — Once the Shim UEFI key management screen is present, a key must be pressed to enter Machine Owner Key management.



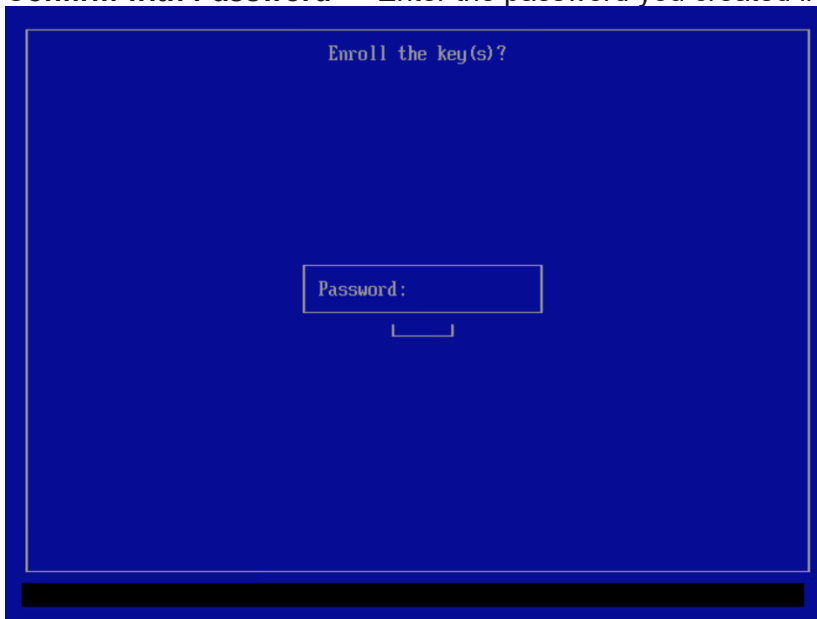
2. **Enroll MOK** — Select Enroll MOK from the MokManager menu.



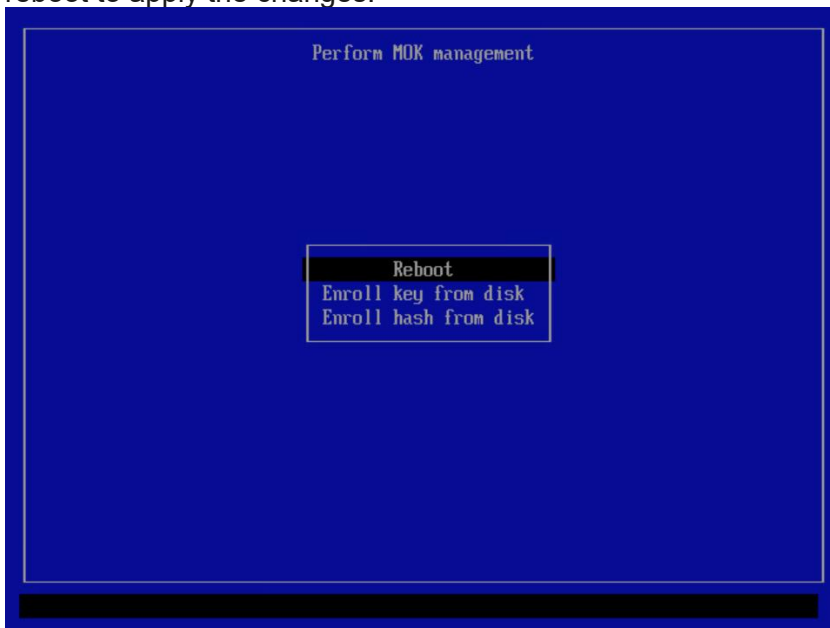
3. **View Key (Optional)** — Select View key to verify that the certificate details match your MOK.der file. When ready, select Continue.



4. **Confirm with Password** — Enter the password you created in Step 2 when prompted.



5. **Reboot** — MokManager will save the enrolled key. The system will prompt for another reboot to apply the changes.



Note: After the second reboot, the key is permanently enrolled in the firmware. You can verify enrolled keys at any time using: `sudo mokutil --list-enrolled`

Part 2: Signing the Kernel Module

With the MOK key enrolled, you can now sign the `usbcmd.ko` kernel module. The signature allows Secure Boot to verify the module is trusted before loading it into the kernel.

Step 1 — Locate the Kernel Module

The kernel module is located in the `components/driver` folder of your installation package. Ensure you have the unsigned `usbcmd.ko` file available before continuing.

Step 2 — Sign the Module with `kmodsign`

Use `kmodsign` to apply a SHA-512 signature to the module using your MOK key pair. This produces a new signed module file named `usbcmd-signed.ko`.

```
ubuntu@localhost:~$ kmodsign sha512 MOK.priv MOK.der usbcmd.ko usbcmd-signed.ko
```

Step 3 — Replace the Original Module

Rename the signed output file to replace the original module, then return it to the `components/driver` directory:

```
mv usbcmd-signed.ko usbcmd.ko
```

Place the renamed `usbcmd.ko` back into the `components/driver` folder. The module is now ready for installation.

Next Steps

With the MOK key enrolled and the kernel module signed, you may proceed to install all components. The signed `usbcmd.ko` module will load successfully under UEFI Secure Boot.

For further assistance or troubleshooting, please contact your support representative.