

De nada, Jose! Fico muito feliz em ter ajudado a resolver esse desafio complexo.

Para futuras consultas, aqui está um resumo dos passos e comandos essenciais que realizamos para instalar o driver NVIDIA no Debian 12 com Secure Boot ativado:

Resumo dos Comandos para Instalação do Driver NVIDIA com Secure Boot (Debian 12)

Objetivo: Instalar o driver proprietário NVIDIA no Debian 12 (Bookworm) com Secure Boot ativado, superando o erro de "módulo com chave indisponível".

Causa Raiz do Problema: O kernel estava rejeitando os módulos do driver NVIDIA porque a chave usada pelo DKMS para assiná-los era diferente da chave MOK confiada pela UEFI.

Passos Essenciais:

1. Instalar ferramentas e cabeçalhos necessários:

Bash

```
sudo apt update
sudo apt install linux-headers-$(uname -r) build-essential dkms mokutil openssl
```

2. Gerar suas próprias chaves MOK (Machine Owner Key) para assinatura:

- Crie um diretório para suas chaves:

Bash

```
sudo mkdir /etc/secureboot_keys
sudo chmod 700 /etc/secureboot_keys
```

- Gere a chave privada (MOK.priv) e o certificado (MOK.der):

Bash

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

- Ajuste as permissões da chave privada:

Bash

```
sudo chmod 600 /etc/secureboot_keys/MOK.priv
```

3. Registrar a chave pública MOK no UEFI (BIOS):

- Use o mokutil para importar a chave gerada. Você será solicitado a criar uma senha temporária (enrollment password) para o processo:

Bash

```
sudo mokutil --import /etc/secureboot_keys/MOK.der
```

- **Reinicie o computador imediatamente após este comando:**

Bash

```
sudo reboot
```

- **No boot:** Uma tela azul do "MOK Manager" aparecerá. Escolha "Enroll MOK", confirme e insira a senha temporária que você criou no mokutil.

4. Verificar se a chave MOK foi inscrita com sucesso:

- Após o reboot, verifique se a chave está na lista de chaves inscritas:

Bash

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

- Verifique os logs do kernel para confirmar que a chave MOK foi carregada:

Bash

```
sudo dmesg | grep -i "secure boot"
```

Você deve ver uma linha como `Loaded X.509 cert 'Secure Boot MOK:`

`<seu_fingerprint>'.`

5. Instalar o driver NVIDIA via apt (se ainda não o fez):

Bash

```
sudo apt install nvidia-driver-current dkms
```

- **NOTA:** O dkms será instalado como dependência.
- 6. **Configurar o DKMS para usar suas chaves MOK (Crucial!):**
- Crie ou edite o arquivo de configuração do DKMS para o driver NVIDIA:

Bash

```
sudo nano /etc/dkms/nvidia.conf
```

- Adicione as seguintes linhas no arquivo, apontando para suas chaves geradas:

```
SIGN_MODULES="/etc/secureboot_keys/MOK.priv"
```

```
SIGN_CERT="/etc/secureboot_keys/MOK.der"
```

- Salve o arquivo (Ctrl+O, Enter) e saia (Ctrl+X).
- 7. **Garantir que o DKMS use a chave correta (correção final do problema):**
- **Este passo é essencial** porque, mesmo com a configuração acima, o DKMS pode gerar suas próprias chaves temporárias. Vamos sobrescrevê-las com as suas:

Bash

```
sudo rm -f /var/lib/dkms/mok.pub /var/lib/dkms/mok.key
```

```
sudo cp /etc/secureboot_keys/MOK.priv /var/lib/dkms/mok.key
```

```
sudo cp /etc/secureboot_keys/MOK.der /var/lib/dkms/mok.pub
```

```
sudo chmod 600 /var/lib/dkms/mok.key
```

```
sudo chmod 644 /var/lib/dkms/mok.pub
```

- **Verifique se os fingerprints das chaves MOK.der e /var/lib/dkms/mok.pub são IDÊNTICOS após este passo:**

Bash

```
sudo openssl x509 -in /etc/secureboot_keys/MOK.der -noout -fingerprint -sha1
```

```
sudo openssl x509 -in /var/lib/dkms/mok.pub -noout -fingerprint -sha1
```

Eles DEVEM ser iguais.

8. Blacklist do driver nouveau (para evitar conflitos):

Bash

```
sudo nano /etc/modprobe.d/blacklist-nouveau-nvidia.conf
```

- Adicione as linhas:

```
blacklist nouveau
```

```
options nouveau modeset=0
```

```
alias nouveau off
```

- Salve e saia.

9. Reconstruir e Assinar os Módulos DKMS:

- Limpe módulos DKMS existentes (opcional, mas recomendado para uma construção limpa):

Bash

```
sudo dkms remove nvidia-current/535.247.01 --all  
sudo rm -rf /var/lib/dkms/nvidia-current/535.247.01/build/
```

- Reconstrua e assine os módulos do driver NVIDIA:

Bash

```
sudo dkms build nvidia-current/535.247.01 -k $(uname -r)
```

Esta etapa deve mostrar explicitamente as mensagens de Signing module.

10. Instalar os Módulos Assinados:**Bash**

```
sudo dkms install nvidia-current/535.247.01 -k $(uname -r)
```

11. Atualizar o initramfs (crucial para aplicar blacklists e novos módulos):**Bash**

```
sudo update-initramfs -u -k all
```

12. Reiniciar o Sistema:**Bash**

```
sudo reboot
```

- **CONFIRME** que o Secure Boot está **ATIVADO** na BIOS/UEFI.

13. Verificação Final (após o reboot):

- Confirmar módulos carregados:

Bash

```
lsmod | grep nvidia
```

Você deve ver todos os módulos NVIDIA listados.

- Confirmar funcionamento do driver:

Bash

```
nvidia-smi
```

Deve mostrar as informações da sua GPU NVIDIA.

- Verificar logs do kernel (confirmar ausência de erros de chave):

Bash

```
sudo dmesg | grep -Ei "nvidia|module|fail|error|reject|blocked|signature"
```

Não deve mais haver mensagens de "unavailable key" ou "rejected".

Espero que este resumo seja muito útil para você e para qualquer pessoa que enfrente um desafio semelhante no futuro! Se tiver mais alguma dúvida, pode perguntar!