

GPG

Projet gsb.org



DUMAS Lucie

Table des matières

Qu'est-ce que GPG.....	3
Configuration de GPG	4
La commande sudo	13
Différence entre les commandes su, su- et sudo	13
Configuration de la commande sudo	13



Qu'est-ce que GPG

GPG (GNU Privacy Guard) est un logiciel de cryptographie open-source largement utilisé dans le domaine de la sécurité informatique. Il fournit des outils pour chiffrer et déchiffrer des données, ainsi que pour créer et vérifier des signatures numériques.

GPG est principalement utilisé pour sécuriser la communication et les données en garantissant la confidentialité, l'authenticité et l'intégrité des informations échangées. Il utilise le chiffrement à clé publique pour permettre des communications sécurisées entre les utilisateurs, en utilisant un système de paires de clés publiques et privées. GPG est largement utilisé pour la sécurisation des e-mails, le stockage sécurisé des fichiers et d'autres applications où la confidentialité et la sécurité des données sont essentielles.



Configuration de GPG

Dans un premier temps, nous allons télécharger sur toutes nos machines le paquet GPG :

```
apt update && apt upgrade
apt install gpg
```

Sur notre routeur, nous allons générer une paire de clés publique et privée en utilisant l'outil GPG à l'aide de la commande suivante :

```
gpg --full-generate-key --expert
```

Nous créons une clé de certification. Pour cela, nous choisissons le type de clé RSA (8) et nous enlevons les options de signature et de chiffrement en sélectionnant successivement (S) puis (C) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1 an. Nous pouvons maintenant générer la clé.

```
root@asie-rtr:~# gpg --full-generate-key --expert
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

gpg: répertoire « /root/.gnupg » créé
gpg: le trousseau local « /root/.gnupg/pubring.kbx » a été créé
Sélectionnez le type de clé désiré :
  (1) RSA et RSA (par défaut)
  (2) DSA et Elgamal
  (3) DSA (signature seule)
  (4) RSA (signature seule)
  (7) DSA (indiquez vous-même les capacités)
  (8) RSA (indiquez vous-même les capacités)
  (9) ECC et ECC
  (10) ECC (signature seule)
  (11) ECC (indiquez vous-même les capacités)
  (13) Clé existante
  (14) Existing key from card
Quel est votre choix ? 8

Actions possibles pour une clé RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Signer Certifier Chiffrer

  (S) Inverser la capacité de signature
  (C) Inverser la capacité de chiffrement
  (A) Inverser la capacité d'authentification
  (Q) Terminé
Quel est votre choix ? S

Actions possibles pour une clé RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Certifier Chiffrer

  (S) Inverser la capacité de signature
  (C) Inverser la capacité de chiffrement
  (A) Inverser la capacité d'authentification
  (Q) Terminé
Quel est votre choix ? C

Actions possibles pour une clé RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Certifier

  (S) Inverser la capacité de signature
  (C) Inverser la capacité de chiffrement
  (A) Inverser la capacité d'authentification
```



```
Actions possibles pour une clef RSA : Signer Certifier Chiffrer Authentifier
Actions actuellement permises : Certifier
```

- (S) Inverser la capacité de signature
- (C) Inverser la capacité de chiffrement
- (A) Inverser la capacité d'authentification
- (Q) Terminé

```
Quel est votre choix ? q
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
  0 = la clef n'expire pas
  <n> = la clef expire dans n jours
  <n>w = la clef expire dans n semaines
  <n>m = la clef expire dans n mois
  <n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0) 1y
La clef expire le mer. 11 sept. 2024 14:00:43 CEST
Est-ce correct ? (o/N) o
GnuPG doit construire une identité pour identifier la clef.
```

```
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.
gpg: /root/.gnupg/trustdb.gpg : base de confiance créée
gpg: répertoire « /root/.gnupg/openpgp-revocs.d » créé
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/621B1B89062734E76ECEFA7E70721FF9210606D.rev'
les clefs publique et secrète ont été créées et signées.

pub  rsa4096 2023-09-12 [C] [expire : 2024-09-11]
    621B1B89062734E76ECEFA7E70721FF9210606D
```

Nous vérifions que la clé publique soit bien générée :

gpg -k

```
root@asie-rtr:~# gpg -k
gpg: vérification de la base de confiance
gpg: marginals needed: 3  completes needed: 1  trust model: pgp
gpg: profondeur : 0  valables : 1  signées : 0
    confiance : 0 i., 0 n.d., 0 j., 0 m., 0 t., 1 u.
gpg: la prochaine vérification de la base de confiance aura lieu le 2024-09-11
/root/.gnupg/pubring.kbx

pub  rsa4096 2023-09-12 [C] [expire : 2024-09-11]
    621B1B89062734E76ECEFA7E70721FF9210606D
```

Nous vérifions également que la clé privée soit bien générée :

gpg -K



```

root@asie-rtr:~# !140
gpg -K
/root/.gnupg/pubring.kbx

sec   rsa4096 2023-09-12 [C] [expire : 2024-09-11]
      621B1B89062734E76ECEF8A7E70721FF9210606D
uid    [   ultime ] DSI Asie <macron@explosion.gsb.org>
ssb    rsa4096 2023-09-12 [S] [expire : 2024-09-11]
ssb    rsa4096 2023-09-12 [E] [expire : 2024-09-11]
ssb    rsa4096 2023-09-12 [A] [expire : 2024-09-11]

```

Nous allons maintenant créer une sous-clé de signature :

```

# Pour cette commande il faut préciser --edit-key pour éditer la clé puis indiquer le nom de la clé
que nous souhaitons modifier.
gpg --expert --edit-key DSI Asie
addkey

```

Nous sélectionnons le type de clé RSA (8), puis nous enlevons l'option de chiffrement (C) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1 an. Nous pouvons générer la sous-clé.

```

root@asie-rtr:~# gpg --expert --edit-key DSI Asie
gpg (GnuPG) 2.2.40; Copyright (C) 2022 g10 Code GmbH
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

La clef secrète est disponible.

sec   rsa4096/E70721FF9210606D
      créé : 2023-09-12  expire : 2024-09-11  utilisation : C
      confiance : ultime      validité : ultime

```



```

gpg> addkey
Sélectionnez le type de clef désiré :
  (3) DSA (signature seule)
  (4) RSA (signature seule)
  (5) Elgamal (chiffrement seul)
  (6) RSA (chiffrement seul)
  (7) DSA (indiquez vous-même les capacités)
  (8) RSA (indiquez vous-même les capacités)
  (10) ECC (signature seule)
  (11) ECC (indiquez vous-même les capacités)
  (12) ECC (chiffrement seul)
  (13) Clef existante
  (14) Existing key from card
Quel est votre choix ? 8

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises : Signer Chiffrer

  (S) Inverser la capacité de signature
  (C) Inverser la capacité de chiffrement
  (A) Inverser la capacité d'authentification
  (Q) Terminé

Quel est votre choix ? c

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises : Signer

  (S) Inverser la capacité de signature
  (C) Inverser la capacité de chiffrement
  (A) Inverser la capacité d'authentification
  (Q) Terminé

Quel est votre choix ? q
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
  0 = la clef n'expire pas
  <n> = la clef expire dans n jours
  <n>w = la clef expire dans n semaines
  <n>m = la clef expire dans n mois
  <n>y = la clef expire dans n ans

```

De nombreux octets aléatoires doivent être générés. Vous devriez faire autre chose (taper au clavier, déplacer la souris, utiliser les disques) pendant la génération de nombres premiers ; cela donne au générateur de nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.

```

sec  rsa4096/E70721FF9210606D
     créé : 2023-09-12  expire : 2024-09-11  utilisation : C
     confiance : ultime  validité : ultime
ssb  rsa4096/E6D076F52F282700
     créé : 2023-09-12  expire : 2024-09-11  utilisation : S

```

Nous créons maintenant une sous-clé de chiffrement :

```
addkey
```

Nous sélectionnons le type de clé RSA (8), puis nous enlevons l'option de signature (S) et enfin (Q) pour quitter. Nous lui choisissons une taille de 4096 bits puis une durée de validité de 1 an. Nous pouvons générer la sous-clé.




```

gpg> addkey
Sélectionnez le type de clef désiré :
(3) DSA (signature seule)
(4) RSA (signature seule)
(5) Elgamal (chiffrement seul)
(6) RSA (chiffrement seul)
(7) DSA (indiquez vous-même les capacités)
(8) RSA (indiquez vous-même les capacités)
(10) ECC (signature seule)
(11) ECC (indiquez vous-même les capacités)
(12) ECC (chiffrement seul)
(13) Clef existante
(14) Existing key from card
Quel est votre choix ? 8

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises : Signer Chiffrer

(S) Inverser la capacité de signature
(C) Inverser la capacité de chiffrement
(A) Inverser la capacité d'authentification
(Q) Terminé

Quel est votre choix ? s

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises : Chiffrer

(S) Inverser la capacité de signature
(C) Inverser la capacité de chiffrement
(A) Inverser la capacité d'authentification
(Q) Terminé

Quel est votre choix ? q
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.
0 = la clef n'expire pas
<n> = la clef expire dans n jours
<n>w = la clef expire dans n semaines
<n>m = la clef expire dans n mois
<n>y = la clef expire dans n ans
Pendant combien de temps la clef est-elle valable ? (0) 1y
La clef expire le mer. 11 sept. 2024 15:06:54 CEST
Est-ce correct ? (o/N) o
Faut-il vraiment la créer ? (o/N) o
De nombreux octets aléatoires doivent être générés. Vous devriez faire
autre chose (taper au clavier, déplacer la souris, utiliser les disques)
pendant la génération de nombres premiers ; cela donne au générateur de
nombres aléatoires une meilleure chance d'obtenir suffisamment d'entropie.

sec  rsa4096/E70721FF9210606D
    créé : 2023-09-12  expire : 2024-09-11  utilisation : C
    confiance : ultime    validité : ultime
ssb  rsa4096/E6D076F52F2B2700
    créé : 2023-09-12  expire : 2024-09-11  utilisation : S
ssb  rsa4096/3F33961EAD7D6B62

```

Nous créons une sous-clé d'authentification :

addkey

Nous sélectionnons le type de clé RSA (8), puis nous enlevons l'option de signature (S), de chiffrement (C), nous ajoutons l'option d'authentification (A) et enfin, nous quittons le menu de configuration (Q). Nous lui choisissons une taille de 4096 bits, puis une durée de validité de 1 an. Nous pouvons générer la sous-clé.




```

gpg> addkey
Sélectionnez le type de clef désiré :
(3) DSA (signature seule)
(4) RSA (signature seule)
(5) Elgamal (chiffrement seul)
(6) RSA (chiffrement seul)
(7) DSA (indiquez vous-même les capacités)
(8) RSA (indiquez vous-même les capacités)
(10) ECC (signature seule)
(11) ECC (indiquez vous-même les capacités)
(12) ECC (chiffrement seul)
(13) Clef existante
(14) Existing key from card
Quel est votre choix ? 8

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises : Signer Chiffrer

(S) Inverser la capacité de signature
(C) Inverser la capacité de chiffrement
(A) Inverser la capacité d'authentification
(Q) Terminé

Quel est votre choix ? s

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises : Chiffrer

(S) Inverser la capacité de signature
(C) Inverser la capacité de chiffrement
(A) Inverser la capacité d'authentification
(Q) Terminé

Quel est votre choix ? c

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises :

(S) Inverser la capacité de signature
(C) Inverser la capacité de chiffrement
(A) Inverser la capacité d'authentification
(Q) Terminé

Quel est votre choix ? a

Actions possibles pour une clef RSA : Signer Chiffrer Authentifier
Actions actuellement permises : Authentifier

(S) Inverser la capacité de signature
(C) Inverser la capacité de chiffrement
(A) Inverser la capacité d'authentification
(Q) Terminé

Quel est votre choix ? q
les clefs RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clef désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clef devrait être valable.

```

Nous sauvegardons les changements effectués :

```

save
# L'option --list-keys permet de lister les clé présente
# L'option --with-keygrip permet d'afficher le grip de la clé d'authentification.
gpg --list-keys --with-keygrip

```



```

gpg> save
root@asie-rtr:~# gpg --list-keys --with-keygrip
/root/.gnupg/pubring.kbx

pub   rsa4096 2023-09-12 [C] [expire : 2024-09-11]
      621B1B89062734E76ECEF8A7E70721FF9210606D
      Keygrip = D2C8A1D5CD749D8AEE8F2E2D2A375FDB7F0D91EE
uid   [   ultime   ] DSI Asie <macron@explosion.gsb.org>
sub   rsa4096 2023-09-12 [S] [expire : 2024-09-11]
      Keygrip = DB0C7A2334862BB8D2074173BA22ED6047D60E77
sub   rsa4096 2023-09-12 [E] [expire : 2024-09-11]
      Keygrip = 43CA02BA81EFBAC7564326049E8C1AC154F3412A
sub   rsa4096 2023-09-12 [A] [expire : 2024-09-11]
      Keygrip = 44750BFD68325FBA4FCF6D14D0FED6C4DD8814FB
root@asie-rtr:~#

```

Une fois nos clés créées, nous allons copier les clés publiques suivantes dans le fichier `~/.ssh/authorized_keys` afin que les machines possédant ces clés publiques puissent se connecter au serveur :

- la clé publique du routeur Asie
- la clé publique du routeur Monde

Sur le serveur, nous allons importer la clé publique du routeur ainsi que sa clé privée pour autoriser la connexion du serveur vers le routeur et du routeur vers le serveur. Nous importons la clé publique dans le fichier `~/.ssh/authorized_keys`.

Pour qu'une machine autorise une connexion SSH par clé GPG, nous devons importer la clé publique du routeur dans le fichier `~/.ssh/authorized_keys`. Pour qu'une machine puisse se connecter à une autre machine autorisant la clé publique du routeur, il faut que cette dernière soit en possession de la clé privée du routeur.

Etant donné que, par défaut, l'agent SSH ne reconnaît pas les clés GPG, nous devons activer pour chaque machine (conteneurs et machines virtuelles inclus) l'agent GPG dans le fichier `~/.gnupg/gpg-agent.conf` afin que ce dernier puisse prendre en charge les clés.

```
enable-ssh-support >> $HOME/.gnupg/gpg-agent.conf
```

Nous modifions également notre fichier `~/.bash_profile` pour y ajouter un script permettant d'échanger les sockets SSH et GPG afin que l'on puisse utiliser les clés GPG pour une connexion SSH.

```
nano ~/.bash_profile
```



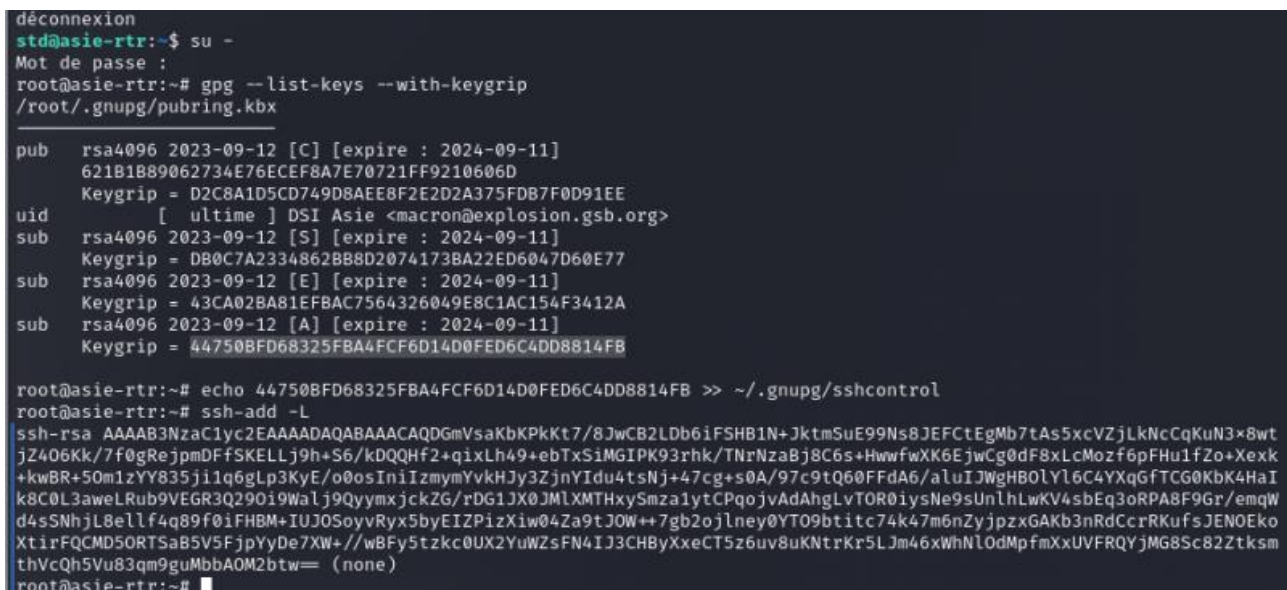
```
# Script d'activation de l'agent GPG
unset SSH_AGENT_PID
if [ "${gnupg_SSH_AUTH_SOCK_by:-0}" -ne $$ ]; then
    export SSH_AUTH_SOCK="$(gpgconf --list-dirs agent-ssh-socket)"
fi
export GPG_TTY=$(tty)
gpg-connect-agent updatestartuptty /bye >/dev/null
```



```
GNU nano 7.2 .bash_profile
unset SSH_AGENT_PID
if [ "${gnupg_SSH_AUTH_SOCK_by:-0}" -ne $$ ]; then
    export SSH_AUTH_SOCK="$(gpgconf --list-dirs agent-ssh-socket)"
fi
export GPG_TTY=$(tty)
gpg-connect-agent updatestartuptty /bye >/dev/null
```

Pour que les machines puissent prendre en compte tous les changements, nous nous déconnectons du compte utilisateur et nous nous reconnectons. Nous pouvons afficher la liste des clés afin de vérifier que les changements aient bien été pris en compte.

```
# Afficher la liste des clés avec le grip de la clé d'authentification
gpg --list-keys --with-grip
echo 44750BFD68325FBA4FCF6D14D0FED6C4DD8814FB >> ~/.gnupg/sshcontrol
ssh-add -L
```



```
déconnexion
std@asie-rtr:~$ su -
Mot de passe :
root@asie-rtr:~# gpg --list-keys --with-keygrip
/root/.gnupg/pubring.kbx

pub   rsa4096 2023-09-12 [C] [expire : 2024-09-11]
      621B1B89062734E76ECEF8A7E70721FF9210606D
      Keygrip = D2C8A1D5CD749D8AEE8F2E2D2A375FDB7F0D91EE
uid   [  ultime ] DSI Asie <macron@explosion.gsb.org>
sub   rsa4096 2023-09-12 [S] [expire : 2024-09-11]
      Keygrip = DB0C7A2334862BB8D2074173BA22ED6047D60E77
sub   rsa4096 2023-09-12 [E] [expire : 2024-09-11]
      Keygrip = 43CA02BA81EFBAC7564326049E8C1AC154F3412A
sub   rsa4096 2023-09-12 [A] [expire : 2024-09-11]
      Keygrip = 44750BFD68325FBA4FCF6D14D0FED6C4DD8814FB

root@asie-rtr:~# echo 44750BFD68325FBA4FCF6D14D0FED6C4DD8814FB >> ~/.gnupg/sshcontrol
root@asie-rtr:~# ssh-add -L
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDGmVsaKbKPkKt7/8JwCB2LDdb6iFSHB1N+JktmSue99Ns8JEFCTeGmb7tAs5xcVZjLkNcCqKuN3x8wt
jZ4O6Kk/7f0gRejpmDFfSKELLj9h+S6/kDQQHf2+qixLh49+ebTxSiMGIPK93rhk/TNrNzaBj8C6s+HwwfwXK6EjwCg0dF8xLcMozf6pFHu1fZo+Xexk
+kwBR+50m1zYY835j1lq6gLP3KyE/o0osIniIzmymYvKHJy3ZjnYIdu4tsNj+47cg+s0A/97c9tQ60FFdA6/aluIJWgHB0LYL6C4YXqGFTCG0KbK4HaI
k8C0L3aweLRub9VEGR3Q290i9Walj9QyymxjckZG/rDG1JX0JMLXMTXhxySmza1ytCPqojvAdAhgLvTOR0iysNe9sUnlhLwKV4sbEq3oRPA8F9Gr/emqW
d4sSNhjlBellf4q89f0iFHBm+IUJOSoyvRyx5byEIZPizXiw04Za9tJOW++7gb2ojlney0YT09btitc74k47m6nZyjpzGAKb3nRdCcrRKufsJENOEko
XtirFQCMD5ORTSaB5V5FjpYyDe7XW+//wBFy5tzkc0UX2YuWZsFN4IJ3CHByXxeCT5z6uv8uKntRkr5LJm46xWhNlOdMpfmXxUVFRQYjMG8Sc82Ztksm
thVcQh5Vu83qm9guMbbAOM2btw= (none)
root@asie-rtr:~#
```



Nous pouvons ensuite modifier le Time To Live de notre passphrase afin de ne pas avoir à la rentrer chaque jours. Pour cela, nous modifions le fichier `~/.gnupg/gpg-agent.conf` :

```
GNU nano 7.2 .gnupg/gpg-agent.conf
enable-ssh-support
default-cache-ttl 34560000
max-cache-ttl 345560000
```



La commande sudo

Différence entre les commandes su, su- et sudo

La commande su permet de se connecter en tant que root en gardant les variables d'environnement de l'utilisateur précédent.

La commande su - permet de se connecter en tant que root en utilisant les variables d'environnement de l'utilisateur root, ce qui permet d'utiliser des commandes apparaissant comme introuvables pour les autres utilisateurs.

La commande sudo permet d'entrer une commande avec les privilèges administrateurs si l'utilisateur fait parti des groupes pouvant utiliser la commande sudo.

Configuration de la commande sudo

Pour configurer la commande sudo, nous devons ajouter l'utilisateur std dans le groupe sudo :

```
usermod -a -G sudo std
```

Pour que les changements soient effectifs, nous devons nous déconnecter puis nous reconnecter à l'utilisateur std. Nous pouvons répéter cette manipulation sur chaque machine de notre réseau pour nous assurer que l'utilisateur std puisse utiliser la commande sudo.

Pour vérifier que l'utilisateur std soit dans le groupe sudo, nous utilisons la commande suivante :

```
cat /etc/group
```

Nous pouvons voir la liste des groupes et des utilisateurs qui leurs sont associés :

```
sudo:x:27:std
```

