

- [Accueil](#)
- Missions 1 à 5
 - [Mission 1 : Configuration réseau](#)
 - [Mission 2 : GPG](#)
 - [Mission 3 : Clonezilla](#)
 - [Mission 4 : BackupPC](#)
 - [Mission 5 : MariaDB](#)
- Missions 6 à 10
 - [Mission 6 : DHCP](#)
 - [Mission 7 : Failover](#)
 - [Mission 8 : DNS](#)
 - [Mission 9 : Nginx](#)
 - [Mission 10 : FTP](#)
- [Mission 11 : SSL/TLS](#)
- [Mission 12 : OPNsense](#)
- [Mission 13 : Zabbix](#)

GPG

I) 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 1an. Nous pouvons maintenant générer la clé.

```

root@isie-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 clef désiré :
(1) RSA et RSA (par défaut)
(2) DSA et Elgatal
(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) Clef existante
(14) Existing key from card
Quel est votre choix ? 8

Actions possibles pour une clef 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 ? SC
Choix incorrect.

Actions possibles pour une clef 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 clef 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 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

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.

Nom réel : DSI Asie
Adresse électronique : macron@explosion.gsb.org
Commentaire :
Vous avez sélectionné cette identité :
« DSI Asie <macron@explosion.gsb.org> »

Changer le (N)om, le (C)ommentaire, l'(A)dresse électronique
ou (O)ui/(Q)uitter ? █

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/ » créé
gpg: revocation certificate stored as '/root/.gnupg/openpgp-revocs.d/62181889862734E76ECEF8A7E70721FF9218086D.rev'
les clefs publique et secrète ont été créées et signées.

pub  rsa4096 2023-09-12 [C] [expire : 2024-09-11]
     62181889862734E76ECEF8A7E70721FF9218086D
uid   DSI Asie <macron@explosion.gsb.org>

```

Nous vérifions ensuite que la clé publique soit bien générée :

```
gpg -k
```

```
root@asie-rtf:~# gpg -k
gpg: vérification de la base de confiance
gpg: marginals needed: 3 completes needed: 1 trust model: gpg
gpg: profondeur : 0 valables : 1 signées : 0
gpg: 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]
      62181809062734E746CE98A7E70721FF92180460
uid     [ ultime ] DSI Asie <macron@explosion.gsb.org>
```

Nous vérifions également que la clé privée soit bien générée :

```
gpg -K
```

```
root@asie-rtf:~# gpg -K
gpg -K
/root/.gnupg/pubring.kbx

sec   rsa4096 2023-09-12 [C] [expire : 2024-09-11]
      62181809062734E746CE98A7E70721FF92180460
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 :

```
gpg --expert --edit-key DSI Asie
#pour cette commande il faut préciser --edit-key pour éditer la clé
puis indiquer le nom de la clé que nous souhaitons modifier.
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 1an. Nous pouvons générer la sous-clé.

```
root@asia-mtr:~# 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/E70721FF9210000D
  créé : 2023-09-12  expire : 2024-09-11  utilisation : C
  confiance : ultime  validité : ultime
[ ultime ] (1). DSI Asie <macron@explosion.gsb.org>

Commande incorrecte (essayez « help »)

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 ? (2072) 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
<nw> = la clef expire dans n semaines
<nm> = la clef expire dans n mois
<ny> = 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/E70721FF9210000D
  créé : 2023-09-12  expire : 2024-09-11  utilisation : C
  confiance : ultime  validité : ultime
ssb rsa4096/E6D070F52F2B2700
  créé : 2023-09-12  expire : 2024-09-11  utilisation : S
[ ultime ] (1). DSI Asie <macron@explosion.gsb.org>

gpg>
```

Nous créons 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 1an. Nous pouvons générer la sous-clé.

```

gpg> addkey
Sélectionnez le type de clé 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) Clé existante
(14) Existing key from card
Quel est votre choix ? 8

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

(5) 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 Chiffrer Authentifier
Actions actuellement permises : Chiffrer

(5) 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 clés RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clé désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clé devrait être valable.
0 = la clé n'expire pas
cm> = la clé expire dans n jours
cmw = la clé expire dans n semaines
cmo = la clé expire dans n mois
cmy = la clé expire dans n ans
Pendant combien de temps la clé est-elle valable ? (0) 1y
La clé expire le mer. 21 sept. 2024 15:00:54 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/E70721FF921B080D
créé : 2023-09-12 expire : 2024-09-11 utilisation : C
confiance : ultime validité : ultime
ssb rsa4096/E60076F52F2B2700
créé : 2023-09-12 expire : 2024-09-11 utilisation : S
usb rsa4096/3F21961EAD7D6867

```

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 1an. Nous pouvons générer la sous-clé.

```

gpg> addkey
Sélectionnez le type de clé 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) Clé existante
(14) Existing key from card
Quel est votre choix ? 8

Actions possibles pour une clé 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 clé 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 clé 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 clé 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 clés RSA peuvent faire une taille comprise entre 1024 et 4096 bits.
Quelle taille de clé désirez-vous ? (3072) 4096
La taille demandée est 4096 bits
Veuillez indiquer le temps pendant lequel cette clé devrait être valable.

```

Nous sauvegardons les changements effectués :

```

save
gpg --list-keys --with-keygrip
#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> save
root@sasie-ctr:~# gpg --list-keys --with-keygrip
/root/.gnupg/pubring.kbx
pub  rsa4096 2023-09-12 [C] [expire : 2024-09-11]
    62181889062734E761E3F8A7E70721FF9210606D
    Keygrip = 02C6A1D5C0749D6A1E8F2E202A375F087F0D91EE
uid  [ uptime ] 051 Asie <mcron@explosion.gsb.org>
sub  rsa4096 2023-09-12 [S] [expire : 2024-09-11]
    Keygrip = 080C7A23A803B80D2074173BA22ED6A7D60E77
sub  rsa4096 2023-09-12 [E] [expire : 2024-09-11]
    Keygrip = 43CAB2B81E7BAC736A3280A9EBC1AC154F3412A
sub  rsa4096 2023-09-12 [A] [expire : 2024-09-11]
    Keygrip = 447508FD69325F8A4FC6D1A08FED6C40D8814F8
root@sasie-ctr:~#

```

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
- la clé publique du routeur prof

Sur le serveur, nous allons importer la clé publique du routeur ainsi que sa clé privé 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.

Dans le fichier `~/.ssh/sshd_config` prohiber `password`

Etant donné que, par défaut, l'agent SSH ne reconnaît pas les clés GPG, nous devons activer pour chaque machine (conteneurs et VM 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 socket ssh et gpg afin que l'on puisse utiliser les clé 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
```



Pour que les machines puissent prendre en compte tous les changements, nous nous déconnectons du compte utilisateur et nous nous reconnectons.

```
#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

```

std@sistr2-rtt:~$ su -
Mot de passe :
root@sistr2-rtt:~# gpg --list-keys --with-keygrip
/root/.gnupg/pubring.kbx
-----
pub   rsa4096 2023-09-12 [C] [expire : 2024-09-11]
      621B1B09062734E761CEFA7E7B721FF9210604D
      Keygrip = 02CBA1D5CD749D0AEE8F2E202A275FDB7F8091EE
uid   [ ultimate ] 051 Asie <acrcr@beauposlon.gso.org>
sub   rsa4096 2023-09-12 [S] [expire : 2024-09-11]
      Keygrip = 008C7A233A862B08D2B74173BA221D6867D68E77
sub   rsa4096 2023-09-12 [E] [expire : 2024-09-11]
      Keygrip = 43CA02B811F8AC7504326049E8C3AC15AF3412A
sub   rsa4096 2023-09-12 [A] [expire : 2024-09-11]
      Keygrip = 667508FF60125FBA4FCF0D1AD0F06C4D00814FB

root@sistr2-rtt:~# echo 44792BFD68325FBA4FCF0D1AD0F06C4D00814FB >> ~/.gnupg/sshcontrol
root@sistr2-rtt:~# ssh-add -L
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCGwVsaXkKPkkt7/8JwC82LDB61F5H01N+3KwSuE99Ns8JEPctEgM07Tao5acVZjlkKcCqkaN0+Bwt
j2A06kk/7f0g8e3mDF5KELLjsh+56/kDQ04f2+qisLh49+e0Txs1MGIPK93rhk/TMrkzab3BCs+wwwfwKKGEjwG8dF8xcMezfEpFHu1fZ0+Kxok
+kwBR+50ml2YV835j1lq0Lp3Kyt/00sIniZmyrV8v02y32jnyYD0xv1kN+47cg+s8A/97c9tQ00FFda0/alul30g8DLYl0c4YKqGFTCG0K0KHAuI
h0CR3wmlAd09V601Q29019uA1j0QyW8j4k26/F003248301X0Thy0m2217cPuoJwAd0hglvT0R0Jys0p9501h1LW0V40t0300F00z/0mpe
0s5N0j1kellf0g89f01f0m01U0D00y0x30yE12P121w0420gt30W++7gb20j1ne07T090t1tc74447m02y10z0G0030K0c0R0u0fsJEN0EN0
xt1rFQCM050RT5a05V5FjpyY0e70X0//w0Fy1z0c0U02Vw02fNA1J10H0y0x0CT500v0u00tPKr5L3m06w0N100mp0mx0UFRQy3mG0Sc022tksm
thVcQ05V0d30m0u0N0A0G0s0w0 (name)
root@sistr2-rtt:~#

```

Nous pouvons ensuite modifier le TTL de notre passphrase afin de ne pas avoir à la rentrer chaque jours :

```

GNU nano 2.2 .gnupg/gpg-agent.conf
enable-ssh-support
default-cache-ttl 3450000
max-cache-ttl 3450000

```

II) Configuration de Sudo

Quelle est la 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 administrateur si l'utilisateur fait partie des groupes pouvant utiliser 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
```

From:

<https://sisr2.beaupeyrat.com/> - **Documentations SIO2 option SISR**

Permanent link:

<https://sisr2.beaupeyrat.com/doku.php?id=sisr2-asie:mission2>

Last update: **2023/12/15 09:39**

