ssh

ssh est un programme pour se connecter à une machine distante et pour effectuer des commandes sur cette machine. La connexion et les échanges sont sécurisés. L'identité utilisé sur le poste distant peut être différente de l'identité du poste local utilisé.

La connexion ssh nécessite d'un service ssh sur la machine distante.

1. Connexions sécurisées et simplifiées grâce à l'authentification par clé publique et privée : principe

Pour simplifier et sécuriser la connexion à une machine distante, il est possible d'utiliser une méthode basée sur l'authentification par clé publique et privée. Cette approche élimine la nécessité de saisir un login et un mot de passe à chaque connexion. Au lieu de cela, la connexion SSH vérifiera votre clé privée avec la clé publique enregistrée sur le serveur distant.

Cette méthode présente plusieurs avantages. Elle élimine la complexité liée à la gestion des mots de passe et permet un gain de temps considérable au quotidien. De plus, elle offre un niveau de sécurité supérieur, car les clés utilisées sont beaucoup plus robustes que les mots de passe traditionnels.

Si plusieurs utilisateurs doivent accéder au serveur distant, SSH permet de gérer plusieurs paires de clés, permettant ainsi à chaque utilisateur de se connecter avec sa propre clé. Si vous avez la responsabilité de plusieurs serveurs, vous pouvez utiliser la même clé publique sur tous les serveurs.

Voici un guide détaillé pour vous aider à créer un jeu de clés sur votre poste de travail. Nous allons générer deux clés : une clé privée et une clé publique. Seule la clé publique devra être déployée sur les différents serveurs, tandis que la clé privée doit être conservée précieusement sur votre ordinateur.

2. Création d'un jeu de clés ecdsa pour une connexion SSH sécurisée

L'algorithme de signature numérique **ecdsa** est un nouveau standard utilisant les courbes elliptiques, réputé pour sa sécurité et sa performance. La taille maximale des clés supportées est de 521 bits, et la plupart des clients **SSH** le prennent en charge.

Si vous préférez utiliser l'algorithme RSA, vous pouvez simplement remplacer “ecdsa” par “rsa” dans les étapes suivantes.

Étape 1: Génération de la clé SSH

Pour créer une clé SSH de type “ecdsa”, vous pouvez utiliser la commande suivante avec ssh-keygen. L'option -t spécifie le type de clé, et l'option -b définit la longueur de la clé.

```
ssh-keygen -t ecdsa -b 521
```

– Spécification de l'emplacement du stockage de la clé (optionnel)

Si vous souhaitez spécifier un emplacement particulier pour stocker la clé, vous pouvez utiliser l'option -f suivi du chemin d'accès souhaité. Par exemple :

```
ssh-keygen -t ecdsa -b 521 -f ~/.ssh/aws-cdc001-cedric-ecdsa
```

– Ajout d'un commentaire à la clé (optionnel)

Si vous souhaitez ajouter un commentaire à la clé pour une meilleure identification, vous pouvez utiliser l'option -C suivi du commentaire souhaité. Par exemple :

```
ssh-keygen -t ecdsa -b 521 -f ~/.ssh/aws-cdc001-cedric-ecdsa -C
"cedric@dskcdc001"
```

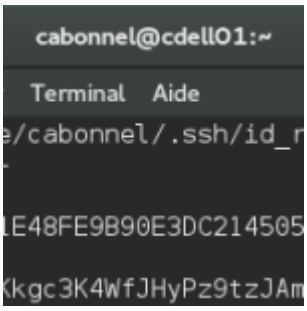
Étape 2: Sécurisation de la clé privée

Il est crucial de sécuriser la clé privée et de limiter l'accès aux personnes autorisées à l'utiliser. Lors de la création de la clé, le programme **ssh-keygen** vous demandera de définir une passphrase (mot de passe) pour la clé privée. Assurez-vous d'utiliser une passphrase sécurisée et de la mémoriser. Les caractères que vous entrez n'apparaîtront pas à l'écran pour des raisons de sécurité.

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

En suivant ces étapes, vous aurez créé un jeu de clés **ecdsa** pour une connexion SSH sécurisée. Veillez à bien protéger la clé privée et à utiliser une **passphrase** forte pour garantir la sécurité de votre connexion.

3. Contrôle et gestion des clés dans SSH



Pour contrôler vos clés SSH et effectuer des opérations de gestion, vous pouvez suivre les étapes suivantes :

Étape 1: Lister les clés présentes dans votre compte

Utilisez la commande suivante dans votre terminal pour lister les clés présentes dans votre répertoire ~/.ssh/ :

```
ls ~/.ssh/
```

Cette commande affichera la liste des clés présentes, le cas échéant.

Étape 2: Afficher le contenu d'une clé (à utiliser avec précaution)

Si vous souhaitez afficher le contenu d'une clé spécifique, vous pouvez utiliser la commande suivante :

```
cat ~/.ssh/maCle
```

Remplacez “maCle” par le nom de votre clé. Cependant, il est important de noter que l'affichage du contenu d'une clé publique ou privée à l'aide de la commande “cat” est une pratique peu recommandée en raison de la sensibilité des informations contenues dans la clé. Veillez à utiliser cette commande avec précaution et évitez de divulguer le contenu de vos clés.

Il est essentiel de prendre des mesures pour sécuriser vos clés SSH, telles que la protection de la clé privée avec une passphrase et le contrôle strict des autorisations d'accès aux fichiers de clés.

En suivant ces étapes, vous pouvez contrôler et gérer vos clés SSH de manière sécurisée. Veillez à respecter les bonnes pratiques en matière de gestion des clés et à prendre les mesures appropriées pour protéger vos informations sensibles.

4. Copier et utiliser une clé publique avec SSH

Guide étape par étape pour copier et utiliser une clé publique avec SSH sous Linx.

–

Pour utiliser la clé, vous devez procéder à la copie de votre clé publique vers le poste distant. La clé publique est généralement stockée dans un fichier nommé “id_rsa.pub” situé dans le répertoire “.ssh” de votre dossier utilisateur. L'étape suivante consiste à ajouter cette clé publique au fichier “authorized_keys” du dossier “.ssh” sur l'ordinateur distant.

Voici un exemple plus détaillé du processus :

1. Tout d'abord, identifiez l'emplacement de votre clé publique. Par défaut, elle se trouve dans le fichier “~/.ssh/id_rsa.pub”.

2. Ensuite, ouvrez une session sur le serveur distant en utilisant la commande SSH :

```
ssh utilisateur@srvprod.aceinternet.fr
```

Remplacez “utilisateur” par votre nom d'utilisateur et “srvprod.aceinternet.fr” par l'adresse du serveur distant.

3. Une fois connecté au serveur distant, créez le dossier “.ssh” dans votre répertoire utilisateur s'il

n'existe pas déjà :

```
mkdir -p ~/.ssh
```

4. Utilisez la commande “ssh-copy-id” pour copier votre clé publique sur le serveur distant et l'ajouter au fichier “authorized_keys” :

```
ssh-copy-id -i ~/.ssh/id_rsa.pub utilisateur@srvprod.aceinternet.fr
```

Cette commande copie le contenu de votre clé publique dans le fichier “authorized_keys” sur le serveur distant, ce qui vous permettra de vous connecter sans avoir à saisir de mot de passe. Cette commande est a utiliser sur votre poste local.

5. Après avoir exécuté la commande, vous serez invité à saisir votre mot de passe pour le serveur distant une dernière fois. Entrez-le et la copie de votre clé publique sera effectuée.

Une fois que vous avez suivi ces étapes, vous devriez être en mesure de vous connecter au serveur distant en utilisant votre clé privée, sans avoir à saisir votre mot de passe à chaque fois.

Veuillez noter que les noms de fichiers et les chemins d'accès peuvent varier en fonction de votre configuration spécifique, mais les étapes générales restent les mêmes.

5. Gestion des clés SSH avec un fichier de configuration

Pour faciliter la gestion des connexions SSH, vous pouvez créer un fichier de configuration qui regroupe toutes les informations nécessaires. Voici un exemple de configuration :

```
host srvprod.aceinternet.fr
  HostName srvprod.aceinternet.fr
  Port 2134
  User adminsrv
  IdentityFile ~/.ssh/id_rsa_adminsrvAceinternetFr
```

Ce fichier de configuration permet de spécifier les paramètres de connexion pour l'hôte distant “srvprod.aceinternet.fr”. Les lignes suivantes indiquent respectivement le nom d'hôte, le port, le nom d'utilisateur et le chemin vers la clé privée à utiliser pour cette connexion.

Il est important de protéger le fichier de configuration pour garantir la sécurité de vos informations sensibles. Vous pouvez définir les permissions appropriées en utilisant les commandes suivantes :

```
chmod 600 ~/.ssh/config
chown $USER ~/.ssh/config
```

La première commande `chmod 600` définit les permissions du fichier de configuration de manière à ce qu'il soit accessible en lecture et écriture uniquement par le propriétaire (vous), et aucun accès en lecture pour les autres utilisateurs. La deuxième commande `chown $USER` garantit que le fichier appartient à l'utilisateur courant.

En veillant à protéger votre fichier de configuration, vous pouvez centraliser et gérer plus facilement

vos connexions SSH en utilisant les paramètres spécifiés dans ce fichier. Cela simplifie également la maintenance et la modification des connexions SSH.

6. Conseils en cas de panne

6.1 Que faire en cas de changement de la clé publique de l'hôte distant

Une **clé publique de l'hôte distant** est un élément essentiel dans le système d'authentification et de sécurité utilisé par le protocole SSH (Secure Shell) lors des connexions à distance. Lorsque vous vous connectez à un hôte distant via SSH, l'hôte présente sa clé publique au client pour vérifier son identité.

La **clé publique de l'hôte distant** est générée lors de la première connexion SSH à cet hôte et est ensuite stockée dans le fichier **known_hosts** du client. Elle est associée à une signature numérique unique qui permet d'authentifier l'hôte distant de manière sécurisée. Cette clé publique est utilisée pour chiffrer les données envoyées au serveur, assurant ainsi la confidentialité des communications.

Lorsque vous vous reconnectez à l'hôte distant ultérieurement, le client SSH vérifie si la clé publique présentée par l'hôte correspond à celle enregistrée dans le fichier **known_hosts**. Si les clés correspondent, la connexion est établie en toute sécurité. Cependant, si la clé publique a changé depuis la dernière connexion, le client SSH émet un avertissement indiquant qu'une attaque potentielle de type “**man-in-the-middle**” est possible, et la connexion est bloquée par mesure de sécurité.

La **clé publique de l'hôte distant** joue donc un rôle crucial dans l'établissement de connexions sécurisées via SSH. Elle permet d'authentifier l'hôte distant et de détecter tout changement potentiel dans l'identité de l'hôte. La gestion appropriée des clés publiques et la vérification de leur validité contribuent à assurer la sécurité des connexions SSH.

_

Si vous rencontrez une erreur indiquant que la clé publique de l'hôte distant a changé lors d'une tentative de connexion SSH, voici les étapes à suivre pour résoudre ce problème :

1. Tout d'abord, lorsque vous essayez de vous connecter à l'hôte distant avec la commande `ssh 192.168.100.5`, vous obtenez un message d'erreur indiquant que la clé a changé et qu'une attaque de type “man-in-the-middle” est possible.

```
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
fa:f8:ec:af:20:0e:58:ca:d1:4d:47:b0:7e:fa:2b:e2.
Please contact your system administrator.
Add correct host key in /home/cedric/.ssh/known_hosts to get rid of this
message.
Offending ECDSA key in /home/cedric/.ssh/known_hosts:5
ECDSA host key for 192.168.100.5 has changed and you have requested strict
checking.
```

```
Host key verification failed.
```

2. Cela signifie que la clé ECDSA (ECDSA key) utilisée pour sécuriser la connexion entre votre client et l'hôte distant a été modifiée depuis votre dernière connexion. Cette clé est stockée localement sur votre client, dans le fichier known_hosts, qui se trouve généralement dans le répertoire caché .ssh de votre utilisateur (par exemple, /home/cedric/.ssh/known_hosts).

3. Pour résoudre ce problème, vous devez réinitialiser l'entrée de l'hôte distant dans le fichier known_hosts. Vous pouvez le faire en utilisant la commande suivante :

```
ssh-keygen -R 192.168.100.5
```

Cette commande supprimera l'enregistrement de l'hôte **192.168.100.5** du fichier **known_hosts**.

4. Une fois que vous avez réinitialisé l'entrée, vous pouvez vous connecter à nouveau à l'hôte distant en utilisant la commande `ssh 192.168.100.5`. Cette fois-ci, la nouvelle clé publique sera enregistrée dans le fichier **known_hosts**, et vous devriez pouvoir vous connecter sans erreur.

En suivant ces étapes, vous pourrez résoudre le problème lié au changement de la clé publique de l'hôte distant et vous reconnecter en toute sécurité.

6.2 Possible usurpation DNS

Lorsque vous essayez de vous connecter à un hôte distant via SSH, vous pouvez rencontrer un avertissement indiquant une possible usurpation DNS. Voici le message d'erreur associé :

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@       WARNING: POSSIBLE DNS SPOOFING DETECTED!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The ECDSA host key for raspberrypi has changed,
and the key for the corresponding IP address 192.168.100.84
is unchanged. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
Offending key for IP in /home/cedric/.ssh/known_hosts:66
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:kAmTpCxHgmxEjMRO02lHxatQPXzJ3dHcaJCXb+eqswA.
Please contact your system administrator.
Add correct host key in /home/cedric/.ssh/known_hosts to get rid of this
message.
Offending ECDSA key in /home/cedric/.ssh/known_hosts:62
ECDSA host key for raspberrypi has changed and you have requested strict
checking.
```

```
Host key verification failed.
```

Ce message indique que la clé de l'hôte distant “raspberrypi” a changé, mais l'adresse IP correspondante (192.168.100.84) est restée inchangée. Cela peut signifier deux choses : soit une usurpation DNS est en cours, soit l'adresse IP de l'hôte et sa clé de connexion ont changé simultanément.

Pour résoudre ce problème, vous devez supprimer l'enregistrement associé à l'hôte en question. Vous pouvez le faire en utilisant la commande suivante :

```
ssh-keygen -R raspberrypi
```

Cette commande supprimera l'enregistrement de l'hôte “raspberrypi” du fichier known_hosts.

Une fois que vous avez supprimé l'enregistrement, vous pouvez essayer de vous reconnecter à l'hôte. Cette fois-ci, l'association entre le nom de l'hôte et sa clé sera enregistrée à nouveau dans le fichier known_hosts.

Assurez-vous de suivre ces étapes pour garantir la sécurité de votre connexion SSH et éviter les risques potentiels liés à une usurpation DNS.

6.3 Choix entre RSA et ECDSA

Le choix entre l'utilisation d'ECDSA (Elliptic Curve Digital Signature Algorithm) ou de RSA (Rivest-Shamir-Adleman) dépend de plusieurs facteurs, notamment les considérations de sécurité et les préférences personnelles.

ECDSA utilise des courbes elliptiques pour la génération de clés et les opérations de signature numérique. Il est généralement considéré comme plus efficace en termes de performances et d'utilisation de la bande passante. Les clés ECDSA sont également plus courtes que les clés RSA équivalentes, ce qui peut être avantageux dans certains cas.

D'autre part, RSA est un algorithme de cryptographie asymétrique plus ancien et largement utilisé. Il est éprouvé et bien pris en charge par de nombreuses infrastructures et logiciels. RSA est généralement considéré comme étant plus sûr pour des longueurs de clé équivalentes, mais nécessite des clés plus longues pour offrir un niveau de sécurité comparable à ECDSA.

En fin de compte, le choix entre ECDSA et RSA dépend de la compatibilité avec les systèmes existants, des performances souhaitées et des recommandations de sécurité spécifiques. Il est recommandé de se référer aux recommandations de sécurité en vigueur et de prendre en compte les spécifications et les exigences propres à votre environnement avant de faire un choix.

7. Script Bash pour générer une clé privée SSH et configurer la connexion

Voici un script Bash qui demande à l'utilisateur de saisir le nom de l'hôte distant, le numéro de port et son nom d'utilisateur pour se connecter via SSH. Il génère ensuite une clé privée et la pousse sur l'hôte distant. Enfin, il écrit un fichier de configuration dans le répertoire “.ssh/config”.

ssh-keygen-config.sh

```
#!/bin/bash

# Demande à l'utilisateur les informations nécessaires
read -p "Nom de l'hôte distant : " hostname
read -p "Numéro de port SSH : " port
read -p "Nom d'utilisateur : " username

# Vérifie si le fichier de clé privée existe déjà
private_key=~/.ssh/id_rsa_${hostname}
if [[ -f $private_key ]]; then
  echo "La clé privée pour cet hôte existe déjà : $private_key"
  echo "Veuillez supprimer la clé existante ou utiliser un autre nom
d'hôte."
  exit 1
fi

# Génère une clé privée avec un nom dynamique basé sur l'hôte distant
ssh-keygen -t rsa -f $private_key

# Pousse la clé publique sur l'hôte distant
ssh-copy-id -i $private_key.pub -p $port $username@$hostname

# Crée le fichier de configuration
config_file=~/.ssh/config
echo "Host $hostname" >> $config_file
echo "  HostName $hostname" >> $config_file
echo "  Port $port" >> $config_file
echo "  User $username" >> $config_file
echo "  IdentityFile $private_key" >> $config_file
echo "" >> $config_file

# Définit les permissions du fichier de configuration
chmod 600 $config_file

echo "Configuration terminée. Vous pouvez maintenant vous connecter en
utilisant 'ssh $hostname' avec la nouvelle clé privée."
```

Le nom de la clé privée générée sera basé sur le nom de l'hôte distant fourni par l'utilisateur, ce qui permet de générer des clés privées uniques pour chaque hôte distant. Par exemple, si l'utilisateur saisit “**srvprod.aceinternet.fr**” comme nom d'hôte distant, la clé privée sera enregistrée sous `~/.ssh/id_rsa_srvprod.aceinternet.fr`.

Assurez-vous d'exécuter le script en tant qu'utilisateur disposant des droits nécessaires pour effectuer les opérations (par exemple, l'utilisateur courant doit pouvoir générer une clé privée, écrire dans le répertoire `.ssh` et copier la clé publique sur l'hôte distant).