Utiliser des certificats OpenSSH

Présentation

La connexion à distance à un serveur en SSH peut se faire avec des certificats dont le format est spécifique à OpenSSH.

Rappel : l'authentification sur le serveur en SSH consiste à renseigner votre clé publique dans le dossier .ssh/authorized_keys du compte existant sur le serveur.

Votre clé privée reste sur votre ordinateur client et ne doit pas être communiquée. Cette clé privée peut en plus être protégée par une passphrase.

Configuration du service SSH du serveur

Avec Debian, la configuration d'un serveur SSH se fait dans le fichier **/etc/ssh/sshd_config** :

- La directive **PasswordAuthentication** doit être fixée à **no** pour interdire l'authentification par mot de passe.
- dans le dossier **/etc/ssh** se trouve plusieurs types de clés privées, avec les clés publiques associées, générées automatiquement, que peut utiliser le serveur SSH : **ssh_host_ecdsa_key** (ssh_host_ecdsa_key.pub) , **ssh_host_rsa_key** (ssh_host_rsa_key.pub) et **ssh_host_ed25519_key** (ssh_host_ed25519_key.pub).

Pour **vérifier** le paramétrage du serveur SSH, lancez la commande suivante en mode debug :

```
$ sudo /usr/sbin/sshd -d
```

La commande suivante d'afficher ces différents clés publiques d'un serveur SSH (RSA, ECDSA et ED25519):

```
$ ssk_keyscan adresseIPserveurssh
```

Génération du couple de clés privée / publique par le client

Utilisation de l'utilitaire **ssh-keygen** avec les options possibles suivantes :

- option **-t** : préciser le tp de clés : dsa, rsa, ecdsa ou ed25519 (valeur par défaut) ;
  - option **-b** : préciser la longueur de la clé (2048 par défaut).
  - option **-C** : préciser un commentaire pour identifier la clé publique : courriel, nom, etc..

```
$ ssh-keygen -C "Charles Técher"
```

```
* sécuriser la clé privée
```

```
$ chmod 600 ~/.ssh/id_ed25519
```

Les clés privée et publique sont enregistrées dans le dossier **~/.ssh** : **id_ed25519** et **id_ed25519.pub**.

La passphrase de la clé privée peut être modifiée avec la commande suivante :

```
$ ssh-keygen -p -f .ssh/id_ed25519
```

La clé publique peut être affichée à partir de la clé privée :

```
$ ssh-keygen -y -f .ssh/id_ed25519
```

Paramètrage du compte du serveur avec votre clé publique

Sur le serveur distant, vous souhaitez permettre une authentification avec votre clé publique shh.

Pour cela, il suffit de **copier** le contenu de votre clé publique dans le fichier **~/.ssh/authorized_keys** du compte du serveur.

- Exemple de copie automatique de la clé publique pour un compte appelé sio créé sur le serveur distant :

```
ssh-copy-id sio@ipserveur
```

- Exemple de copie manuelle de la clé publique pour un compte appelé sio créé sur le serveur distant :

```
cat ~/.ssh/id_ed25519.pub | ssh sio@ipserveur "cat >> ~/.ssh/authorized_keys"
```

Gestion des serveurs connus

Votre client SSH va **vérifier l'identité du serveur distant** avant de se connecter. Le fichier **/etc/ssh/ssh_config** permet de configurer le client SSH pour tous les utilisateurs de votre ordinateur, et le fichier **~/.ssh/config** uniquement pour l'utilisateur qui a ouvert une session.

Lors de la **première connexion** du client sur le serveur distant, vous recevez un **message avertissement** et vous devez confirmer, à vos risques et périls, que vous acceptez de vous connecter à ce serveur. Dans ce cas, une trace du serveur (**empreinte SSHFP** - SSH FingerPrint), est enregistrée dans le fichier **~/.ssh/known_hosts**. Lors d'une prochaine connexion, si la trace d'une connexion précédente est trouvée, vous n'aurez plus le message d'avertissement.

La commande suivante permet de **prendre connaissance de l'empreinte** de la **clé publique** du serveur distant, et de la communiquer aux utilisateurs client pour **vérification** :

```
$ ssh-keygen -l -f /etc/ssh/ssh_host_ed25519_key.pub
```

Inconvénients de l'usage des clés publiques

Les clés publiques d'un serveur peuvent être changées et ne plus être celles initialement crées. C'est le cas lors d'une attaque de type Man-In-The-Middle.

De même, il n'est pas possible d garantir que la clé publique d'un client est légitime.

L'utilisation d'un certificat ajoute des informations d'identité et ce certificat est signé par une autorité de certification reconnue.

La signature du certificat est un haché des données du certificat chiffrée par la clé privée de l'autorité de certification reconnue.

En utilisant la clé publique de l'autorité de certification reconnue, il est possible de déchiffrer la signature et de comparer ensuite les hachés.

Génération des certificats OpenSSH

OpenSSH permet de générer des certificat spécifiques à OpenSSH. Ce ne sont pas des certificats x509 utilisés avec SSL/TLS (https, sftp, etc.)

La création d'un certificat utilisateur ou serveur nécessite la signature par une clé privée qui représente alors l'autorité de certification.

> L'autorité de certification (CA) doit être hébergée sur le serveur très sécurisé, et de manière idéale, offline quand il n'est pas utilisé.

Pour cette démarche, la CA sera hébergée sur le serveur distant.

création de la clé privée sur le serveur distant avec le nom ca_key

```
$ ssh-keygen -f ca_key -C "SSH cle privee de la CA"
```

Cela génère 2 fichiers :

- c*akey : clé privée de la CA * c*akey.pub : clé publique de la CA

Création d'un certificat pour le client

Pour la création du certificat du client, il est nécessaire de préciser :

- -s : la clé privée de la CA qui signe le certificat
- -I : l'identifiant du certificat client. Pour l'exemple un utilisateur appelé charles.
- -n : le login autorisé pour ce client sur le serveur. Il s'agit dans l'exemple du compte linux sio.
- -V : la validité du certificat. En général une année.
- la clé publique de l'utilisateur qui est nécessaire pour créer son certificat

- Transférez sur le serveur de CA la clé publique de l'utilisateur :

```
$ scp ~/.ssh/id_ed25519.pub sio@adresseIPCA:/home/sio
```

- Sur le serveur de CA, utilisez la commande suivante :

```
$ ssh-keygen -s ca_key -I CLIENT-CHARLES -n sio -V +365d id_ed52519.pub
```

Le certificat a été généré et le fichier obtenu porte le nom de la clé publique en ajoutant -cert.

La commande suivant permet de visualiser le contenu du certificat :

```
$ ssh-keygen -L -f id_ed25519-cert.pub
id_ed25519-cert.pub:
        Type: ssh-ed25519-cert-v01@openssh.com user certificate
        Public key: ED25519-CERT SHA256:q3u+woluMmljCo+HCKHr55oztAiZ7QVDXWkw20W02UY
        Signing CA: ED25519 SHA256:5R5DL2MZrQelfe88lyf2zdv3UepovFcdm9NiHtplWUQ (using ssh-ed25519)
        Key ID: "CLIENT-CHARLES"
        Serial: 0
        Valid: from 2026-06-28T20:27:00 to 2027-06-28T20:28:40
        Principals:
                sio
        Critical Options: (none)
        Extensions:
                permit-X11-forwarding
                permit-agent-forwarding
                permit-port-forwarding
                permit-pty
                permit-user-rc
```

Remarques :

- il s'agit d'un certificat utilisateur : Type user certificate
- ce n'est pas un certificat autosigné car l'empreintes de la clé publique de l'utilisateur (Public Key) est différente de l'empreinte de la clé qui a signé le certificat (Signing CA).
- le principal est sio, c'est à dire le login associé à ce certificat

* Transférez sur le client le certificat client nouvellement généré et placez le dans le dossier .ssh de l'ordinateur client avec la clé privée id_ed25519 et la clé publique ed_ed25519.pub :

```
$ scp sio@adresseIPCA:/home/sio/id_ed25519-cert.pub ~/.ssh/
```

Création du certificat pour le serveur distant

Pour la création du certificat du serveur il est nécessaire de préciser :

- -s : la clé privée de la CA qui signe le certificat
- -I : l'identifiant du certificat client. Pour l'exemple un utilisateur appelé charles.
- -h : précise qu'il s'agit d'un certificat serveur.
- -n : les noms / adresse IP du serveur.
- -V : la validité du certificat. En général une année.
- la clé publique du serveur situé dans le dossier /etc/ssh qui est nécessaire pour créer son certificat

- Sur le serveur de CA, utilisez la commande suivante, en utilisant sudo afin que le certificat puisse être sauvegardé dans le dossier /etc/ssh :

```
$ sudo ssh-keygen -s ca_key -I SERVEUR-LOCAL -h -n localhost,127.0.0.1,adresseIP -V +365d
```

```
/etc/ssh/ssh_host_ed25519_key.pub
```

La commande suivante permet de visualiser le contenu du certificat serveur :

```
$ ssh-keygen -L -f /etc/ssh/ssh_host_ed25519_key-cert.pub
/etc/ssh/ssh_host_ed25519_key-cert.pub:
        Type: ssh-ed25519-cert-v01@openssh.com host certificate
        Public key: ED25519-CERT SHA256:++xMosJ1oo91GoxCV77GSb7tSBzKqKVvY2kDXvXzBr0
        Signing CA: ED25519 SHA256:5R5DL2MZrQelfe88lyf2zdv3UepovFcdm9NiHtplWUQ (using ssh-ed25519)
        Key ID: "SERVEUR-LOCAL"
        Serial: 0
        Valid: from 2026-06-28T20:58:00 to 2027-06-28T20:59:19
        Principals:
                localhost
                127.0.0.1
                adresseIP
        Critical Options: (none)
        Extensions: (none)
```

Remarques :

- il s'agit d'un certificat utilisateur : Type host certificate
- Configuration du serveur SSH pour utiliser ce certificat en modifiant le fichier **/etc/ssh/sshd_config** :

```
...
HostKey /etc/ssh/ssh_host_ed25519_key
HostCertificate /etc/ssh/ssh_host_ed25519_key-cert.pub
...
```

- lançez manuellement le serveur en mode debug, pour visualiser qu'il charge bien les 2 fichiers (la clé privée et le certificat) :

```
$  sudo /usr/sbin/sshd -d
sudo /usr/sbin/sshd -d
```

- redémarrez le service SSH

```
$ sudo systemctl restart ssh
```

Configuration du client pour pour vérifier le certificat du serveur

```
* transférez le fichier de la clé publique de la CA au client afin de disposer de son contenu :
*
```

Il est nécessaire d'ajouter la directive **@cert-authority** dans le fichier **~/.ssh/known_hosts** du client en précisant la clé publique de la CA :