S'authentifier avec SSH

Amélie LEDEIN Louis LEMONNIER Yoan GERAN

2022-2023

Générer une clé SSH

Une clé SSH permet de s'authentifier de manière sécurisée à une machine sans donner son mot de passe. Le client dispose d'une clé publique et d'une clé privée qui sont « liées » et le serveur dispose de la clé publique ce qui lui permet de savoir que le client a les droits d'accès. [1]

Cette méthode d'authentification est (généralement) plus sécurisée qu'une authentification par mot de passe. Une personne malveillante devra alors trouver la clé privée plutôt que la clé publique et contrairement au mot de passe, la clé privée n'est jamais transmise au serveur pour être vérifiée.

La génération de la clé se fait avec la commande `ssh-keygen`. L'option `-t` permet de spécifier l'algorithme utilisé (nous allons utiliser `ed25519`), l'option `-C` de rajouter un commentaire et l'option `-f` d'indiquer le nom du fichier à générer (sachant que les clés SSH sont généralement dans le dossier `~/.ssh`).

```
ssh-keygen -t ed25519 -f ~/.ssh/test_keygen -C "Test"
```

Là, on vous demandera d'entrer une *passphrase*. C'est cette phrase qui vous permettra ensuite de vous identifier. Une fois cette étape finie, on se retrouve avec deux fichiers `test_keygen.pub` (la clé publique à partager) et `test_keygen` (la clé privée à ne surtout pas partager).

La *passphrase* sert en quelque sorte de mot de passe, donc elle doit être sécurisée. De plus, créez une clé par serveur et par machine (donc si vous voulez vous connecter à un même serveur depuis deux ordinateurs, créez deux clés). Ainsi, une personne qui vole la clé privée qui vous permet d'accéder au serveur A n'aura pas accès qu'à A.

Le commentaire se retrouve à la fin de la clé publique. Il peut permettre d'indiquer à quel service la clé permet de se connecter et il n'est pas rare d'y retrouver des courriels.

1. Les détails de fonctionnement de ce protocole pourront être vu dans un cours de cryptographie.

Configurer le client SSH

Nous pouvons configurer notre client SSH, notamment pour lui indiquer quelle clé utiliser lorsque nous connectons à un serveur particulier. Pour cela, on modifie (ou l'on crée) le fichier `~/.ssh/config`. Après avoir créé une clé `id_github_key` pour Github, nous indiquons comment l'utiliser.

```
Host github.com
    HostName github.com
    User <github_pseudo>
    IdentityFile ~/.ssh/id_github_key
```

Mainteant, une connexion à Github en SSH utilisea la clé `id_github_key`.

Connexion à Github en SSH

Notre but est maintenant de pouvoir nous connecter à Github en SSH, et notamment de pouvoir pousser des modifications en ligne depuis un clone d'un de nos dépôts. Pour commencer, il faut rajouter la clé **publique** (**pas la clé privée**) à Github ; cette page [2] indique comment faire. Nous pouvons maintenant vérifier que tout est correct.

1. Créer un projet sur Github (voir cette page [3]).
2. Cloner le projet pour du SSH, (voir cette page [4]).
3. Entrer dans le dossier du projet, faire une modification.
4. Enregistrer la modification (`commit`) et la pousser (`push`) sur Github.

2. https://docs.github.com/en/authentication/connecting-to-github-with-ssh/adding-a-new-ssh-key-to-your-github-account
3. https://docs.github.com/en/get-started/quickstart/create-a-repo
4. https://docs.github.com/en/repositories/creating-and-managing-repositories/cloning-a-repository