Comprendre l'Attaque de l'Homme du Milieu (MITM) dans SSH

Riyad Derguini

January 22, 2025

Contents

1 Introduction

Lorsque vous vous connectez à un système distant via SSH, il est essentiel que la connexion soit **chiffrée** pour éviter qu'un attaquant ne puisse observer ou modifier les commandes que vous envoyez. Cependant, même avec un chiffrement robuste, une attaque appelée **homme du milieu (MITM)** peut compromettre la sécurité de la connexion. Ce document explique en détail cette attaque, ses implications, et comment SSH la prévient.

—

2 Problématique : L'Attaque de l'Homme du Milieu (MITM)

2.1 Contexte

Lorsque vous établissez une connexion SSH, le client SSH (votre système local) et le serveur SSH (le système distant) utilisent le protocole de **Diffie-Hellman** pour générer une clé de chiffrement commune. Cette clé est utilisée pour chiffrer toutes les communications ultérieures.

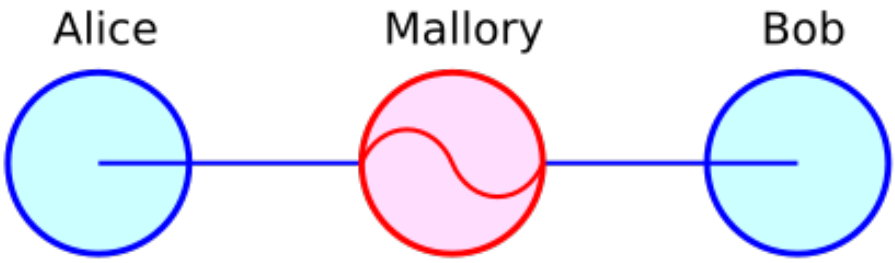
Cependant, un attaquant situé entre votre système local et le système distant peut intercepter et manipuler les échanges. C'est ce qu'on appelle une **attaque de l'homme du milieu (MITM)**.

2.2 Comment fonctionne l'attaque MITM ?

- L'attaquant se fait passer pour le **serveur SSH** auprès de votre client SSH.
- Il se fait passer pour le **client SSH** auprès du serveur SSH.
- Il établit deux connexions chiffrées :
  - Une entre votre client SSH et lui-même.

  - Une autre entre lui-même et le serveur SSH.

- L'attaquant peut alors **déchiffrer, observer et modifier** les messages échangés entre les deux parties sans qu'elles ne s'en aperçoivent.



—

3 Authentification du serveur SSH

Pour prévenir l'attaque MITM, SSH utilise un mécanisme d'**authentification du serveur**. Voici comment cela fonctionne :

3.1 Paire de clés du serveur

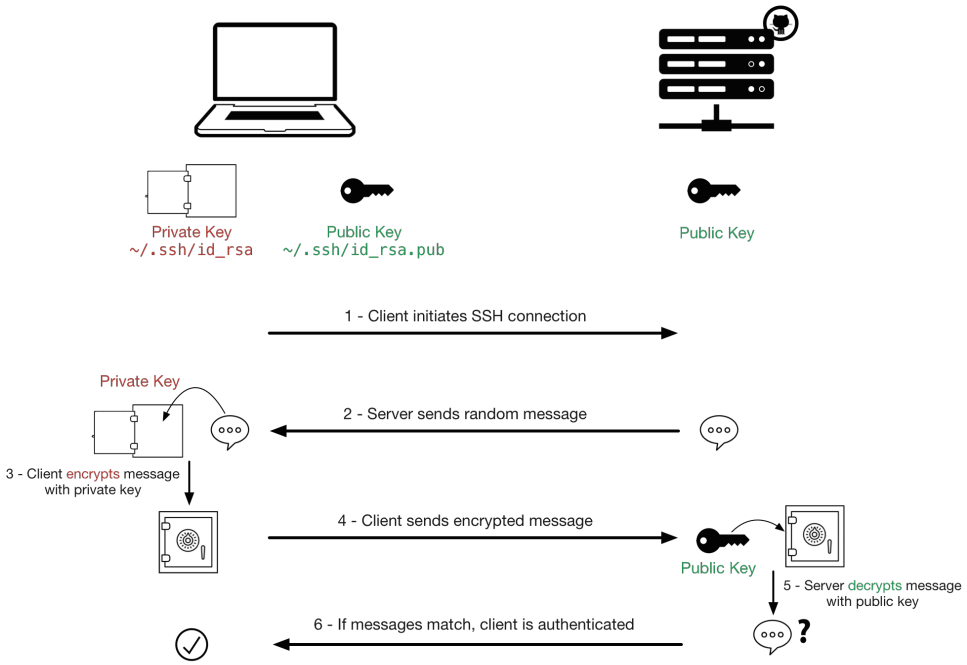
Le serveur SSH possède une **paire de clés asymétriques** : - Une **clé privée** : Gardée secrète par le serveur. - Une **clé publique** : Partagée avec les clients.

3.2 Échange de clés publiques

Lors de la première connexion, le serveur envoie sa clé publique au client. Le client doit vérifier que cette clé publique correspond bien au serveur auquel il souhaite se connecter.

3.3 Challenge et réponse

Le client envoie un **challenge** au serveur, qui ne peut être résolu que si le serveur possède la clé privée correspondante. Le client vérifie ensuite la réponse pour s'assurer qu'il communique bien avec le bon serveur.



—

4 Le Rôle des Fingerprints

4.1 Qu'est-ce qu'un fingerprint ?

Un **fingerprint** est un hash (empreinte) de la clé publique du serveur. Il est plus court et plus facile à vérifier par un être humain que la clé publique complète.

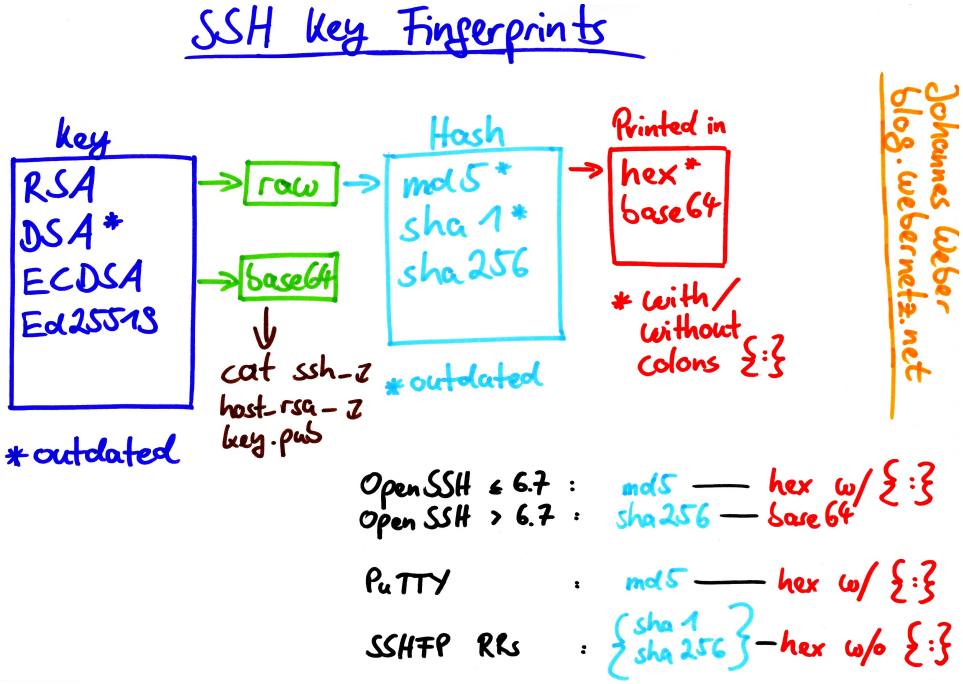
4.2 Pourquoi utiliser un fingerprint ?

Lors de la première connexion, le client SSH affiche le fingerprint de la clé publique du serveur. L'utilisateur doit vérifier ce fingerprint par un **moyen sûr** (par exemple, en le comparant avec une version transmise en main propre ou via un canal sécurisé).

4.3 Exemple de fingerprint

Un fingerprint ressemble à ceci :

```
SHA256:AbCdEfGhIjKlMnOpQrStUvWxYz0123456789+/-=
```



—

5 Conclusion

L'attaque de l'homme du milieu (MITM) est une menace sérieuse pour les connexions SSH. Cependant, grâce à l'**authentification du serveur** et à l'utilisation de **fingerprints**, SSH garantit que vous vous connectez bien au bon système et non à un attaquant. Il est essentiel de toujours vérifier le fingerprint du serveur lors de la première connexion pour éviter toute compromission.