SCR SAÉ S2 03 ⊥ :

Installation de services réseau

sae.s2.03.Part3

Installation et configuration d'accès distants via `openssh`

Les clés SSH

Ce travail consiste à installer sur sa vm Linux, un serveur SSH auquel on pourra se connecter pour utiliser, par exemple, une application particulière à distance.

Note. Les configurations réalisées lors de toutes les précédentes parties doivent demeurer opérationnelles.

Résultat attendu : Un serveur `openssh` devra être installé et configuré sur la machine Linux. Il devra se lancer au démarrage de la machine. Les clients s'y connectent, depuis Linux ou Windows par leur login et mot de passe, puis on modifiera la configuration du serveur pour que l'authentification des clients se fasse par la méthode de paire de clés SSH.

Délai de finalisation du travail : mercredi 17 mai 2023, 23h59.

Pour pouvoir effectuer les tests, on travaillera de concert avec un autre binôme. La machine d'un binôme jouera le rôle de la machine qui tourne le serveur, pendant que l'autre jouera le rôle du client Linux, puis du client Windows. Ensuite, on inversera les rôles. Ainsi, chaque vm sera configurée à la fois comme serveur `openssh` sous Linux, et comme client Linux et Windows du serveur `openssh` de l'autre binôme.

Livrables.

1. La vm opérationnelle.
2. Un fichier PDF par binôme contenant les informations suivantes :
   (a) les mots de passe root Linux et admin Windows qui seront valables au moment de la correction des vms. Vous n'êtes pas censés avoir changé les mots de passe des autres utilisateurs.
   (b) le nom des vms dans le groupe de vms qui a été constitué pour faire les tests.

   Le fichier PDF doit être nommé `login1-login2-info.pdf`

I. Installation et configuration du serveur et client Linux.

1. En utilisant l'authentification par mot de passe.
   (a) Installer le package `openssh`. Avant d'installer un nouveau package ...
   (b) Quel est le fichier de configuration de `sshd` (le dæmon ssh) ? Dans ce fichier, vérifier que l'authentification du client se fait par mot de passe uniquement. Noter qu'il y a une page `man` dédiée à ce fichier de configuration.
   (c) Dans le wiki de *Arch Linux*, on indique le répertoire qui contient les fichiers correspondant aux paires de clés (privée,publique) du serveur. Noter le contenu de ce répertoire.
   (d) Lancer le service `sshd`. Vérifier que le port correspondant est bien ouvert. Noter de nouveau le contenu du répertoire précédent.
   (e) On se met par binôme de vms Linux, et on teste une connexion `ssh` avec *alice*, *bob*. Sur la machine client, lors de la première connexion à un serveur `ssh`, un message avertit que l'authenticité du serveur ne peut pas être établie. Pourquoi ? Hint : les serveurs que le client considère comme légitimes sont, par défaut, dans le fichier `~/.ssh/known_hosts`. Ce fichier existe-t-il ? Le serveur auquel on tente de se connecter est-il enregistré dans ce fichier ?

(f) Pour permettre au client d'établir l'authenticité du serveur, celui-ci envoie au client l'empreinte (un condensé) du contenu du fichier correspondant à la clé publique du serveur. Noter, côté client, l'empreinte envoyée par le serveur. Sur le serveur, on va utiliser la commande `ssh-keygen` pour générer cette empreinte. Consulter la page manuel de cette commande pour trouver l'option adéquate. Comparer les deux empreintes. Si elles ne sont pas identiques, il faudra répondre "no" à la finalisation de la connexion.

(g) Si les empreintes sont identiques, on répond "yes" à la la finalisation de la connexion du client. Une fois connecté, consulter de nouveau, le fichier `~/.ssh/known_hosts` (du client ou du serveur ?)

(h) Inverser les rôles dans le binôme, et refaire le travail précédent.

2. Le client s'authentifie par la méthode de paire de clés SSH.
Référence : SSH KEYS dans le wiki de *Arch Linux*.

(a) À l'aide de `ssh-keygen`, chacun des utilisateurs *alice* et *bob*, génère sa paire de clé en utilisant les paramètres par défaut. Quel est le type de clés par défaut ? Dans quel répertoire de l'utilisateur la paire de clé est-elle enregistrée par défaut ?
Pour les besoins de cette saé, on ne positionne pas de passphrase.

(b) On se remet par binôme de vms Linux : client et serveur. L'utilisateur disons *alice*, comme client du serveur doit avoir, au préalable, le contenu sa clé publique concaténée au contenu de `~alice/.ssh/authorized_keys` côté serveur.
Vérifier que, dans `~alice/.ssh/authorized_keys`, chaque clé commence sur une nouvelle ligne. Faire le nécessaire pour que *alice* et *bob* aient chacun sa clé publique ajoutée au contenu de son fichier `authorized_keys`. Vérifier que les permissions d'accès aux fichiers et répertoires sont comme il faut.

(c) Une fois le point précédent réalisé, on modifiera le fichier de configuration du serveur `ssh` afin que l'authentification ne soit plus par mot de passe mais par la méthode de paire de clés SSH. Tester.

(d) *alice* ajoute une paire de clés de type ecdsa. Réaliser le travail. Tester une session ssh en présentant à la ligne de commande la clé à utiliser lors de l'authentification.

(e) Inverser les rôles dans le binôme, et refaire le travail précédent.

II. Installation et configuration du client Windows.

On se remet par binôme de vms : client sur Windows et serveur sur Linux. L'utilisateur *Albatros* va se connecter à son compte *alice* sur la machine Linux via SSH.

(a) On boote sur Windows et on installe `puTTY` : *https://www.putty.org/*. Dans tout ce qui suit, on ne doit pas utiliser l'agent `Pageant` de `puTTY`.

(b) *Albatros* lance le programme `PuTTYgen` pour générer sa paire de clés de type rsa et la sauvegarder dans `C:\Users\Albatros\`. **Ne pas fermer encore la boîte de dialogue** `PuTTYgen`.

(c) Comme dans la partie I. 2, il faudra ajouter la clé publique à la suite du contenu du fichier `~alice/.ssh/authorized_keys`. Pour ce faire, on copie la zone indiquée par `PuTTYgen`, on la place dans un fichier texte brut d'extension `.dat` qu'on envoie dans `~alice` par le programme `PSFTP` de `PuTTY`.

(d) Dans `PuTTY`, définir une session SSH associée à la clé privée sauvegardée. Tester la connexion SSH.

(e) *Albatros* ajoute une paire de clés de type ecdsa, et définit, dans `PuTTY`, une session SSH associée à cette clé. Tester la connexion SSH.

(f) Inverser les rôles dans le binôme, et refaire le travail précédent.