TP 1 : SSH

1 Consignes générales :

— Un compte-rendu par binôme
— Justifiez vos réponses mais soyez concis.

2 Objectifs

— Configuration basique du serveur SSH
— Echange de fichiers

3 Pré-requis

Modèle OSI, configuration IP par CLI

4 Secure SHell

L'un des intérêts à un réseau interconnectant des machines est la possibilité d'accéder à distance aux ressources. Dans ce contexte, vous allez manipuler l'un des outils souvent rencontrés dans un environnement professionnel : Secure SHell (SSH).

Le protocole SSH permet la mise en place d'un tunnel ou canal de communication chiffré. Typiquement, le serveur disposera d'un couple de clés et sa clé publique sera journie aux clients. Partant du fait que les algorithmes de chiffrement symétrique sont moins consommateurs en ressources que les algorithmes asymétriques, le client et le serveur basculeront vers ce type de chiffrement une fois la clé de session échangée via le canal sécurisé.

L'implémentation du protocole la plus couramment rencontrée est la suite d'outils OpenSSH, développée et maintenue par OpenBSD. Cette suite logicielle est composée de nombreux outils :

- un serveur, `sshd`,
- plusieurs clients, selon les usages
  - connexion shell distant : `ssh` ;
  - transfert et téléchargement de fichier : `scp`, `sftp` ;
- un outil de génération de clés, `ssh-keygen`,
- un service de trousseau de clés, `ssh-agent` et `ssh-add`,
- un scanner de clés publiques présentes sur les serveurs SSH, `ssh-keyscan`.

5 Démarrage

Vous utiliserez pour ce TP les machines virtuelles Kali (K) (credentials : root/toor) et Ubuntu-TP-Asterisk (U) (credentials : jfc/jfc). Vous considérerez U comme le serveur. Listez les adresses IP des deux machines que vous aurez fait démarrer en mode pont. Assurez-vous que les adresses soient différentes.

En vous basant sur le fichier de configuration, répondez aux questions suivantes :

— Qui peut se connecter sur le serveur ?

— Quel est le numéro de port d'écoute du serveur ?
— Un autre protocole d'accès à distance est telnet :
  — Quels sont les paramètres utilisés pour se connecter via telnet ?
  — Comment sont traités ces paramètres par le protocole ?

En ce qui concerne le serveur, vous accèderez à sa configuration avec la commande :
`cat /etc/ssh/sshd_config`

Pour que les modifications apportées à la configuration soient prises en compte, vous redémarrerez le serveur à chaque fois, en utilisant les commandes suivantes :
`/etc/init.d/ssh start`
`/etc/init.d/ssh stop`
`/etc/init.d/ssh restart`

Pour vous connecter à un serveur SSH, vous utiliserez la commande :
`ssh nom_utilisateur@IP_serveur`
Pour vous déconnecter, vous utiliserez la commande :
`exit`

6 Accès aux fichiers

Sur le serveur, créez un fichier msg.txt dans le répertoire test. Le contenu du fichier sera "Hello World !"
Connectez-vous en ssh au serveur et vérifiez que vous avez accès au serveur.

— Quels sont les paramètres de la commande employée ?
— Pourquoi, lors de la première connexion au serveur, vous est-il demandé de confirmer que la clé publique récupérée est bien celle du serveur ? A votre avis, sous quelle forme cette clé publique peut-elle être communiquée ?
— Une fois l'empreinte de la clé validée, comment validez-vous votre identité auprès du serveur ?
— Que se passerait-il si les deux étapes étaient inversées ?

Déconnectez-vous du serveur. A présent, examinez le contenu du fichier `~/.ssh/known_hosts` sur le client. Ce fichier contient une ligne pour chaque hôte reconnu. Le format d'une ligne est le suivant :

- Marqueur : ce champ est optionnel et peut prendre l'un des deux valeurs suivantes "@cert-authority" et "@revoked" : le premier cas indique la présence de la clé d'une CA tandis que le second spécifie que la clé ne doit plus jamais être acceptée.
- Nom de l'hôte : cette information peut être une expression régulière ou la version hashée d'un nom unique. Quand le nom est hashé, le champ débute par un caractère "|"
- Type de clé
- Clé exprimée dans la base 64 (base64-encoded key)
- Commentaire : champ optionnel

— A votre avis, pourquoi hashe-t-on le nom des hôtes ?
— Retrouvez dans le fichier de configuration du client (`cat /etc/ssh/ssh_config`) l'option permettant de spécifier le format de stockage des noms d'hôtes.
— Changez la configuration du fichier. Effacez le contenu du fichier `~/.ssh/known_hosts`. Reconnectez-vous au serveur et vérifiez que l'identité du serveur n'est plus masquée.
— En vous aidant de la documentation en annexe, ajoutez une ligne au fichier de configuration du serveur pour boquer l'accès en ssh au compte jfc. Décrivez votre procédure ainsi que la

vérification.
— Rétablissez l'accès pour jfc. Vérifiez que vous arrivez à consulter le fichier msg.txt

7 Clé publique, clé privée

Vous avez appris que les couples de clés publique/privée permettaient d'ajouter l'authentification à la confidentialité. Dans cette étape, vous allez vous authentifier auprès du serveur en utilisant ces informations. Le process est le suivant :

- Générez un couple de clés avec la commande `ssh-keygen -t rsa -b 4096` . Conservez les réponses par défaut aux questions posées.
- Copiez votre clé sur le serveur avec la commande :
  `ssh jfc@IP_serveur "cat >> ~/.ssh/authorized_keys" < ~/.ssh/id_rsa.pub`

Reconnectez-vous au serveur après avoir appliqué ces modifications : qu'est-ce qui a changé par rapport à la connexion précédente ?

Générez à nouveau un couple de clé mais cette fois-ci, appliquez une passphrase à votre clé privée. Après avoir rappelé le rôle de la passphrase, comparez la nouvelle solution à la précédente en termes de sécurité et de commodité.

8 Echange de fichier

Avec la commande scp, vous pouvez combiner la copie de fichier aux protections apportées par SSH. Le modèle est :
`scp source destination`
Créez sur le serveur le répertoire `abe` et placez-y les fichiers `a.txt`, `b.txt` et `c.txt`.
Créez sur le client le repertoire `jugem` et placez-y les fichiers `d.txt`, `e.txt`, `f.txt` et `g.txt`.
Renseignez dans votre compte-rendu les commandes permettant de copier le répertoire abe vers le client et le répertoire `jugem` vers le serveur.
Recherchez la différence entre scp et SFTP.

9 X11 forwarding

Connectez-vous au serveur SSH, installez nautilus si besoin et lancez la commande suivante : `nautilus &`
Décrivez le résultat.
A présent, vérifiez au niveau du serveur que le X11Forwarding est activé ? Cette option permet le déport de l'affichage : la GUI associée à l'application exécutée sur le serveur est alors affichée sur le terminal du client.
Vérifiez qu'il fonctionne en vous connectant avec la commande `ssh -X jfc@IP_server` et en appelant `nautilus`.

10 Annexe

Composantes du fichier de configuration : *https* : *//www.freebsd.org/cgi/man.cgi?sshd_config*(5)