Auteur: P. Petit	*Titre: TD Unix SSH*	*Version: 1.1*
Date: 25/02/2008	Licence: Gnu Free Documentation Licence	Durée: 1h00

U06: SSH

Objectifs

- compréhension du fonctionnement de ssh
- utilisation des fonctionnalités avancées (clefs, redirections de ports, ...)

Configuration initiale

Ce TD est à réaliser avec deux stations de linux ou solaris nommées station1 (une seule carte réseau en mode NAT) et station2 (une seule carte réseau en mode NAT). Ces deux machines doivent pouvoir communiquer entre elles. On aura crée deux comptes utilisateurs test1 et test2 sur chacun des deux ordinateurs (mots de passe respectifs: passtest1 et passtest2)

Prérequis

- configuration réseau
- installation de paquets, démarrage des services
- notion générales sur les algorithes de chiffrement à clefs publiques
- notion sur ssh

Exercice 1: ssh: utilisation de base

1. ouvrez une session en tant que test2 sur station1.
2. ouvrez un fenêtre de commande sur station1 et connectez vous à distance sur station2 via ssh en tant que test1. La syntaxe de base de ssh est la suivante : « ssh login@machinedistante ». Lors de cette première connexion, vous avez un message d'avertissement. A quoi correspond-il ? A quoi servent ces clefs ? acceptez la clef proposée. lancez une ou deux commandes distantes et quittez la session ssh.
3. Reconnectez vous à distance sur station2. Le message d'avertissement s'affiche-t-il ?
4. Citez deux endroits de stockage des clefs d'hôtes.
5. Après avoir vérifié sur station2 que l'option « X11forwarding » était activée (cf /etc/ssh/sshdconfig). Connectez vous à station2 en tant que test1 en utilisant l'option « -X ». Lancez l'application xeyes sur station2 via votre connexion ssh. Que se passe-t-il ? Que vaut la variable DISPLAY sur station2 ? décrivez le trajet du flux de données permettant l'affichage de xeyes?

Exercice 2: ssh: génération de clefs privées/publiques personnelles

1. La commande ssh-keygen permet de générer un couple de clef privée/publique. Un paramètre obligatoire (option -t) est le type chiffrement utilisé pour chiffré la clef : rsa1 (rsa pour ssh version 1: déconseillé), rsa (rsa pour ssh2: conseillé), dsa: (dsa pour ssh version 2). La clef privée peut optionnellement être protégée par un mot de passe. Ce couple de clef privée/publique peut servir à la connexion à des hôtes distants. Pour cela, il suffit d'ajouter la clef privée au fichier ~/.ssh/.authorized_keys de la machine distante. Votre travail consiste, en tant que test2 sur station 1 à:

- créer un couple de clef privée/publique sur station1 (acceptez les choix par défaut, fournissez une passphrase: linuxrulez);
- ajouter la clef publique au fichier ~/.ssh/authorized_keys de station2
- vérifier que vous arrivez à vous connecter sans mot de passe de station1 et à station2
- modifier le fichier authorized_keys de façon à ce que seule la commande 'who » puisse être exécutée par ce moyen (voir le format du fichier authorized_keys dans la page de manuel de sshd)

Exercice 3: ssh: gestion des clefs utilisateurs

Fournir la « passphrase » à chaque connexion distante peut être fastidieux. Expliquez comment l'outil ssh-agent peut nous aider à résoudre ce problème. Sur votre poste de travail linux (le système hôte): ssh-agent s'exécute-t-il ? Quel processus l'a lancé ?

La commande ssh-add permet d'ajouter une passphrase à celles que mémorise ssh-agent. Lancez ssh-agent sur station1 s'il n'est pas déjà lancé et utilisez ssh-add pour ajouter votre passphrase. Êtes-vous encore obligé de la fournir en vous connectant à station2 depuis station1 ?

Fermez puis ouvrez votre sessions sur station1. Connectez-vous à station2. Êtes vous obligés de fournir une passphrase ? Pourquoi ?