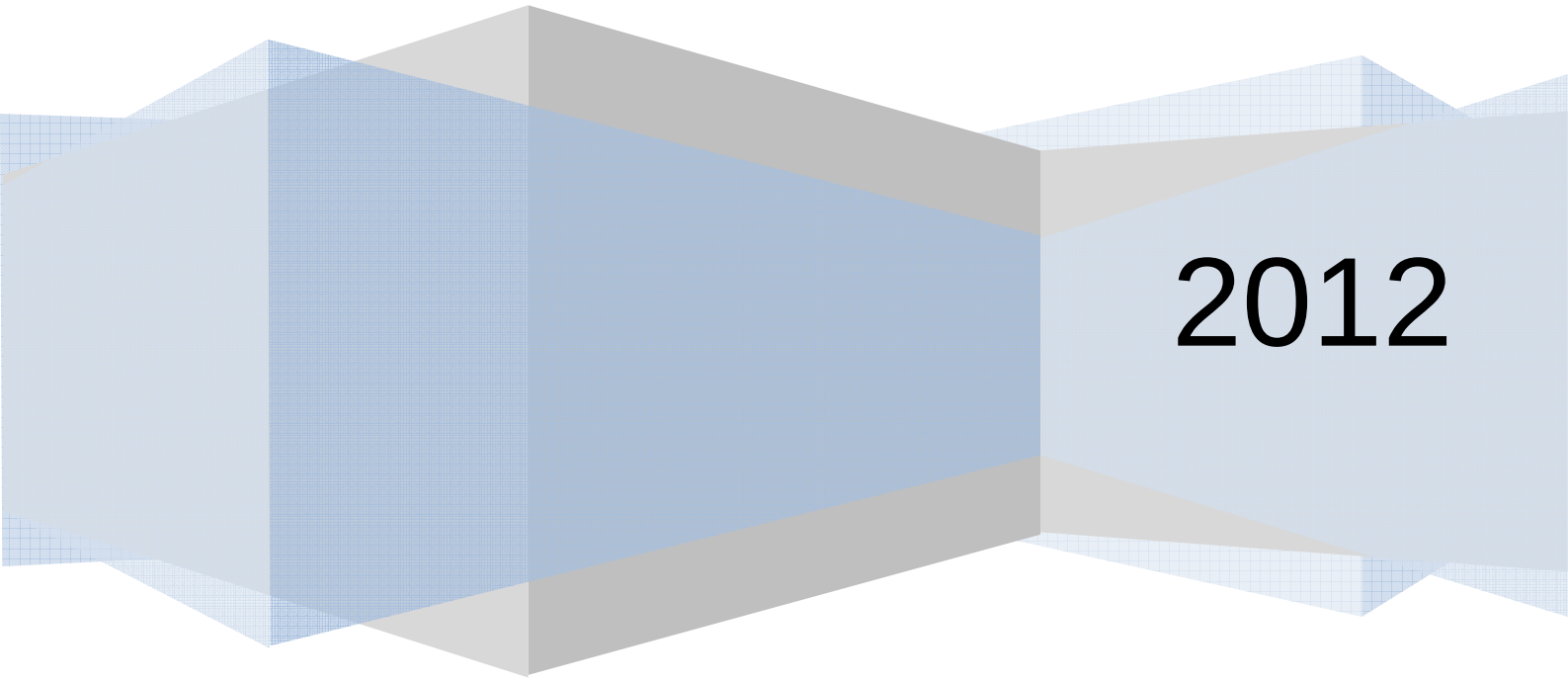
Secure SHell - utilisation

Réalisée sur CentOS 5.7

Ecrit par Charles-Alban BENEZECH



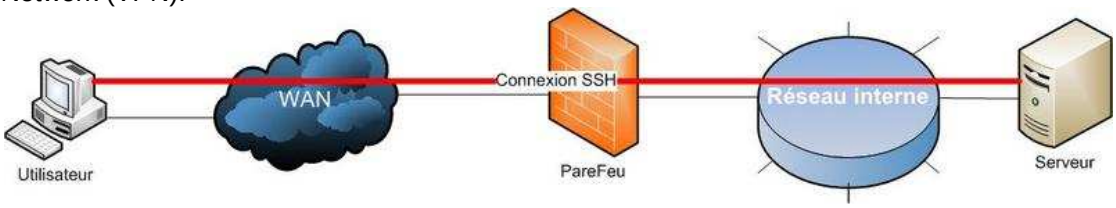
2012

Sommaire

1. Secure SHell

1.1 Introduction

Le SSH est un protocole permettant de créer un tunnel entre deux machines ou plus. La création d'un tunnel consiste en l'encapsulation par un protocole réseau d'un autre protocole réseau de même couche (modèle OSI) ou supérieur, ainsi les données échangées dans le datagramme seront illisibles car protégées par le protocole ssh qui chiffrera le contenu. Ce protocole peut-être utilisé par exemple lié avec des technologies comme le Virtual Private Network (VPN).



1.2 Fonctionnement

La technique de tunnel pour chiffrer les communications sera utilisée pour assurer la confidentialité, l'intégrité et l'authenticité de la dite communication lorsque les communications transiteront sur un réseau non fiable (internet par exemple). En effet, le tunnel prendra tout son sens lors de connexions entre sites distants échangeant des informations. Plusieurs protocoles sont utilisables afin d'établir des connexions sécurisés; voici quelques exemples: L2TP, TLS, IPSEC... SSH dans sa deuxième version intègre un protocole de systèmes de fichiers, Ssh File Tranfert Protocole (SFTP, à ne pas confondre avec FTPS (FTP over SSL)); il est aussi utilisé pour la commande scp.
SSH est un protocole mais aussi un programme basé sur ce protocole qui permet de connecter les machines entre elles en créant un tunnel. Afin d'établir la connexion entre les deux machines, ssh a besoin de vérifier l'identité de la personne qui essaye d'établir la connexion. Pour cela SSH peut utiliser deux méthodes: établir la connexion et ouvrir un prompt d'authentification où l'utilisateur devra entrer son login et son mot de passe; ou il peut utiliser un crypto-système asymétrique, pour cette dernière les clés publiques devront être distribuées sur toutes les machines sur lesquelles l'utilisateur désirera se connecter.

2. Configuration

2.1 Configuration réseau

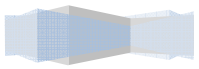
Afin d'établir une connexion entre deux machines via SSH il existe plusieurs méthodes diverses et variées. Tous d'abord, on peut commencer par dissocier les systèmes Windows, Linux et Mac. Je vais présenter les différentes manières de se connecter à une machine distante (serveur linux) via SSH sur ces différentes plate-formes. Dans un souci de permettre une meilleure compréhension voici un petit bilan d'information présentant les futures configurations à venir:
Configuration du Serveur
Nom: LCOSTST01
OS: centos
Configuration Windows
Nom: pc01
OS: Windows seven
Logiciel: putty
Configuration Linux
Nom: LCOSTST02
OS: CentOS
Logiciel: ssh

2.2 fichier de configuration

Afin de nous permettre d'affiner la configuration de notre serveur et optimiser son système de sécurité nous pouvons paramétrer le fichier de configuration. Ce fichier est placer dans le répertoire "/etc/ssh/" (appellé ssh_config sous red Hat), il contient un certains nombre de paramètre (les paramètres commenté (ligne commencant par un "#" sont des paramètre par défaut) que nous pouvons juger pertinent de modifier:
Port [22]: Désigne le port sur lequel le démon ssh écoute.
HostKey [/etc/ssh/ssh_host_key]: définit l'emplacement des clés.

2.2.1 Quelques petit conseil

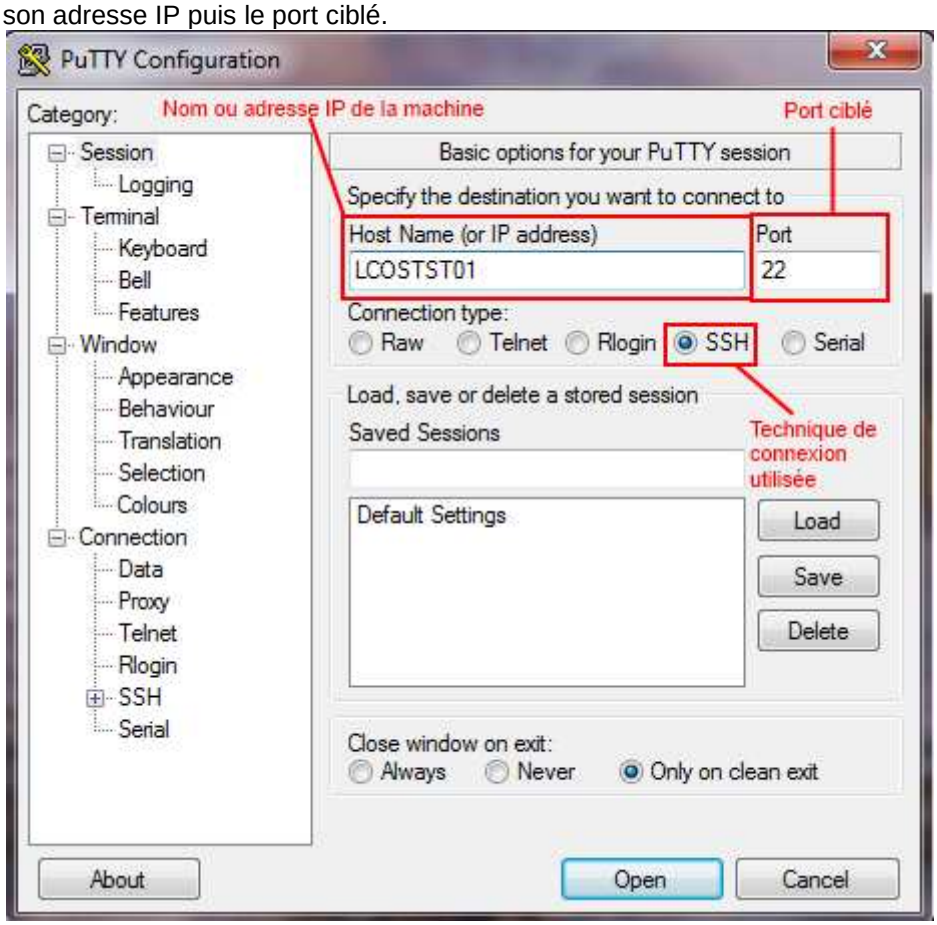
Voici quelques conseils de paramètrage de sshd pour la sécurité:
PasswordAuthentication [yes/no]: Une fois les clés ajouter avec succès nous pouvons désactiver la connexion par mot de passe car les clé suffiront à assuré la connexion et empêchera une personne à tenter d'attaquer les mot de passe ("PasswordAuthentication no").
PermitRootLogin [yes/no]: Il peut être utile d'empêcher les utilisateurs de se connecter directement avec le compte root ce qu'ils pourront ensuite via la commande su. De cette manière un utilisateur ne peut pas se loger directement sur le serveur avec le compte root (PermitRootLogin no").
AllowUsers [<utilisateur>]: Il est possible aussi de rajouter une liste d'utilisateur qui seront les seuls à pouvoir se loger sur le serveur.
ListenAddress [0.0.0.0]: permet de définir sur quel carte réseau le démon doit écouter.



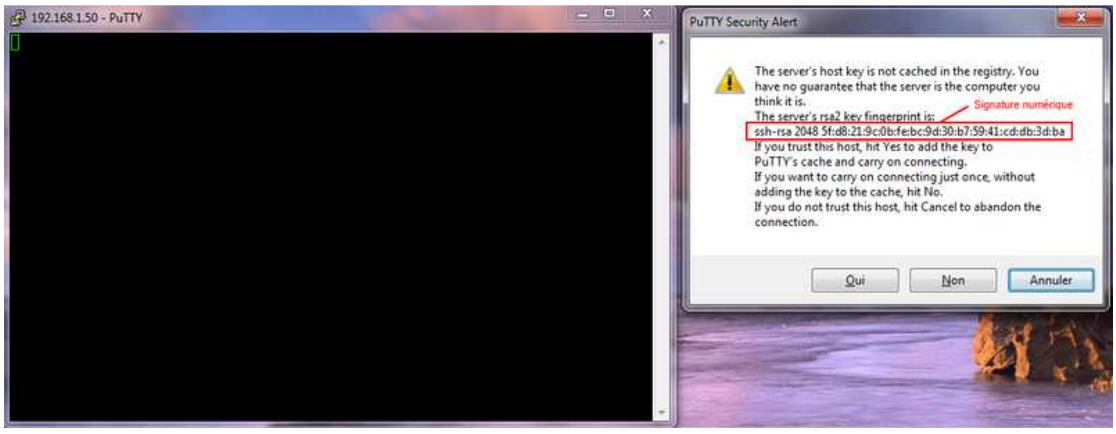
3. Utilisation

3.1 Windows

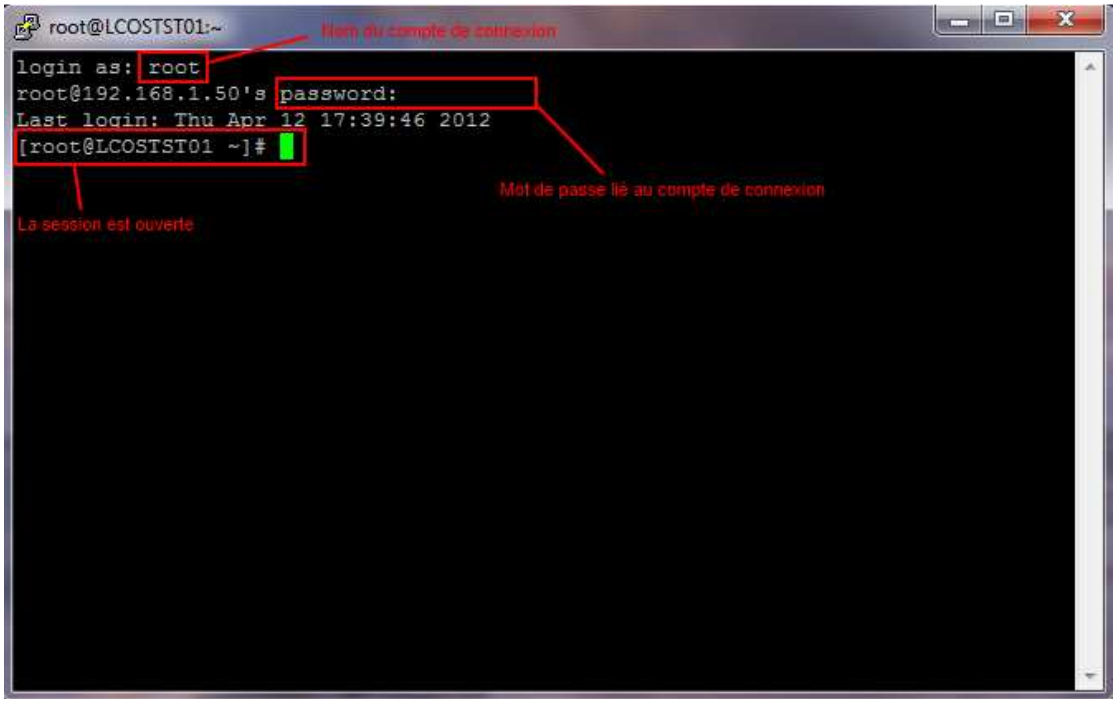
Afin d'établir une connexion SSH entre l'utilisateur et LCOSTST01 nous allons utiliser l'utilitaire Putty. Une fois l'utilitaire lancé, nous allons devoir entrer le nom de la machine ou son adresse IP puis le port ciblé.



Etant la première fois que nous établissons une connexion vers ce serveur,avant d'établir la connexion, Putty va nous demnander s'il peut rajouter le serveur (sa signature numérique)dans la liste des serveurs autorisés.



Une fois la connexion établie il ne nous reste plus qu'à nous loger sur le serveur grâce à notre identifiant.



Vous êtes maintenant connecté à votre machine au travers d'un tunnel ssh.

3.2 Linux

Pour permettre les connexions sous linux deux solutions s'offrent à nous.

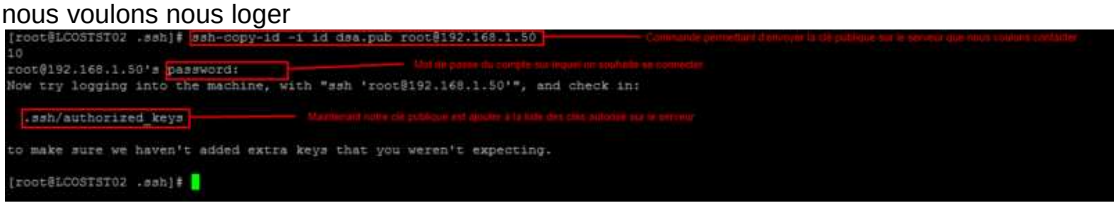
3.2.1 Méthode de connexion par mot de passe

La première consiste à se connecter directement avec le programme ssh sur la machine visée, il ouvrira de cette manière un prompt et nous devrons nous loger avec nos identifiants. Pour ce faire nous nous logerons sous notre utilisateur sur notre machine et executerons la commande ssh.

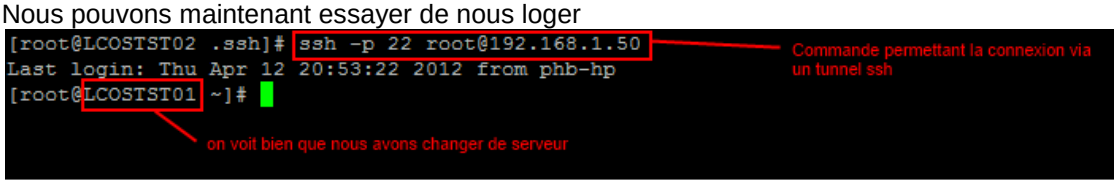
3.2.2 Méthode de connexion par crypto-système asymétrique

La deuxième consiste à utiliser un crypto-système asymétrique, ainsi nous créerons une clé publique et une clé privée. La clé privée restera sur le serveur dans son dossier (/HOMEDIR/.ssh) ainsi qu'une copie de la clé publique et une autre copie de la clé publique sera envoyée sur le serveur sur lequel on veut se connecter automatiquement, il nous suffira d'ajouter la clé publique à la liste des clés autorisées.
Dans un premier temps nous allons devoir créer la clé privée

Maintenant que notre clé est créée nous allons devoir l'envoyer à la machine sur laquelle nous voulons nous loger



Nous pouvons maintenant essayer de nous loger



et voilà! Nous avons une connexion ssh automatisée sans avoir besoin d'échanger des mots de passe ou passphrase et tout aussi sécurisée, nous pouvons aussi utiliser l'agent ssh pour nous connecter.

3.2.3 L'agent ssh

L'agent ssh (ssh-agent, lancer avec le service ssh) est un valet (service ayant pour fonction de mémoriser les mot de passe ou passphrase afin d'automatisé le log d'un utilisateur) lié au service ssh. L'agent ssh va ouvrir un shell dans lequel il chargera (grâce à la commande ssh-add) les clés et les passphrases et ainsi nous n'auront plus à les tapés à chaque connexion.