**Gestion à distance à l’aide du protocole SSH  
Installation de binaires complémentaires (nslookup, ifconfig, …) et mise à jour d’un système (Debian, RHEL, SUSE, ...)  
Installation de binaires (fail2ban et git)**

**Gestion de paquets**

**Présentation**

Les gestionnaires de paquets sont utilisés pour ...

* l'installation ;
* la mise à jour et la désinstallation de logiciels en automatisant la récupération de paquets à partir de sources disponibles depuis Internet, un réseau local, un ou des cédéroms, ...) ;
* la gestion des dépendances (bibliothèques)  
  et parfois
* la compilation.

Lorsque des paquets sont installés, mis à jour ou supprimés, les programmes de gestion de paquets peuvent…

* afficher les dépendances des paquets,

et

* résoudre les dépendances automatiquement.

**Sources**

La plupart des référentiels (dépôts ou *deposit*) de chaque distribution comptent plus d'une dizaine de milliers de paquets (*packages*) disponibles.   
Pour obtenir d'autres paquets, d'autres référentiels peuvent être ajoutés aux sources d’une distribution.   
Des problèmes peuvent arriver lorsque plusieurs sources offrent des paquets de même nom.   
Les systèmes avec un tel potentiel de conflits peuvent utiliser des marqueurs afin de contrôler quelles sources doivent être préférées.

En plus des référentiels sur le réseau, des cédéroms et d'autres médiums de stockage peuvent être utilisés comme dépôts.   
Ceci permet à des systèmes sans accès au réseau d'être mis à jour.

**Remarque** ...  
Pour utiliser les commandes proposées par les différents gestionnaires de paquets, on doit disposer des droits d'administration (su ou sudo).

Gestionnaires de paquets selon la distribution ...

|  |  |
| --- | --- |
| Distribution | Gestionnaire de paquets |
| Debian | APT (*Advanced Package Tools*) |
| RHEL | dnf ou yum |
| SUSE | zypper |
| Arch | pacman |

**Mise à jour du système (Debian)**

La mise à jour d’un système Debian se fait en trois étapes ...

* **Vérification des nouveaux paquets disponibles**  
  La commande **apt update** permet de comparer les versions des paquets installés sur le système et les paquets équivalents disponibles sur les référentiels.  
  Cette procédure permet d’établir si des mises à jour sont disponibles.  
  **>> sudo apt update**
* **Mise à jour effective des paquets**  
  La commande **apt upgrade** permet d’installer les nouveaux paquets mis à jour identifiés par la commande apt update.  
  **>> sudo apt update [--yes]**
* **Supprimer les paquets et bibliothèques désuètes**Pour donner suite à la mise à jour des paquets, il peut être utile de supprimer les paquets qui ne sont plus nécessaires.  
  La commande **apt autoremove** permet de libérer de l’espace disque.  
  **>> sudo apt autoremove [--yes]**

**Remarques** ...  
Avant toute installation de paquets, il est toujours préférable de faire une mise à jour du système.  
Pour effectuer la grande majorité des commandes, un gestionnaire de paquets nécessite les privilèges de l’utilisateur root.  
Le commutateur **--yes** (forme courte **-y**) permet de répondre oui par avance aux demandes de confirmation.

Lors de l’exécution de la commande **apt update**, le système effectue la lecture des listes de paquets, la construction de l'arbre des dépendances et la lecture des informations d'état.  
Par la suite, ce dernier indique s’il existe des paquets pouvant être mis à jour.

Une image contenant texte

Description générée automatiquement

On peut exécuter la commande **apt list --upgradable** afin de voir la liste des paquets à mettre à jour.

Afin de compléter la mise-à-jour, on exécute la commande **apt update**.

Une image contenant texte

Description générée automatiquement

Avant d’effectuer la mise à jour, le système affiche la liste des paquets à mettre à jour ainsi que les statistiques concernant cette mise à jour (*n* mis à jour, *n* nouvellement installés, *n* à enlever et *n* non mis à jour).  
Le système indique également la taille du téléchargement nécessaire et l’espace disque supplémentaire nécessaire.

Si cela s’avère nécessaire, on complète l’opération de mise-à-jour à l’aide de la commande **apt autoremove**.

Une image contenant texte

Description générée automatiquement

**Installation de paquets**

Avant l’installation d’un paquet ou binaire, il important de vérifier s’il est possible de l’installer depuis un référentiel.  
La commande **apt list** permet de vérifier la présence d’un paquet et de savoir s’il est installé sur un système.

Pour vérifier la présence d’un paquet sur le référentiel de base de Debian ...  
**>> sudo apt list <Paquet>  
>> sudo apt list nginx**

Une image contenant texte

Description générée automatiquement

Si le paquet est installé sur le système, Linux va l’indiquer.  
Sinon, on peut procéder à l’installation du paquet avec la commande **apt install** …

**>> sudo apt install <Paquet(s)>** **[--yes]  
>> sudo apt install fail2ban**  
**>> sudo apt install nginx lynx**  
**>> sudo apt install vsftpd ftp git**

**Bibliothèques**

Les bibliothèques (*libraries*) partagées, également désignées objets partagés sont des morceaux de code compilé et réutilisable qui sont utilisés de manière récurrente par divers programmes.  
Ces bibliothèques de logiciels sont des collections de code destinées à être utilisées par de nombreux programmes différents.

L’installation de paquets implique souvent l’installation de bibliothèques et c’est pourquoi il est important de faire une mise à jour du système avant l’installation de nouveau paquets.

**Remarque** ...  
Le nom de fichiers des bibliothèques sont génériques et sont normalement des liens symboliques qui pointent vers le fichier réel contenant une bibliothèque, dont le nom contient le numéro de version exact.

**Suppression de paquets**

Lorsqu’un paquet n’est plus nécessaire, cela est une bonne pratique en matière de sécurité de supprimer les paquets inutiles.

Pour supprimer un ou des paquets ...  
**>> sudo apt remove <Paquet(s)> [--yes]  
>> sudo apt remove fail2ban**  
**>> sudo apt remove nginx lynx**   
**>> sudo apt remove vsftpd ftp git --yes**

La sous-commande permet de désinstaller les paquets indiqués.   
Toutefois, certains paquets ou fichiers peuvent demeurer en place à la suite de la suppression d’un binaire.

Afin de remédier à cette situation, on peut ...

* utiliser le commutateur **--purge** (forme courte **-p**) de la commande **apt remove**...  
  **>> sudo apt remove <Paquet(s)> [--purge] [--yes]   
  >> sudo apt remove nginx --purge –yes**ou  
  utiliser la commande **apt purge** …  
  **>> sudo apt purge <Paquet(s)> [--yes]   
  >> sudo apt purge nginx --yes**
* utiliser la commande **apt autoremove** afin de supprimer toutes les librairies associées ...  
  **>> sudo apt autoremove --yes**

La sous-commande **autoremove** permet de désinstaller les paquets indiqués avec leurs dépendances logicielles.

Pour supprimer un paquet et ses dépendances ...  
**>> sudo apt autoremove <Paquet(s)> [--yes]   
>> sudo apt autoremove nginx --yes**

**Remarque** ...  
Le commutateur **--purge**, passée à la commande **apt remove**, supprime les paquets indiqués, les dépendances logicielles et leurs fichiers de configuration.  
Toutefois, les fichiers de préférences et les fichiers journaux (*logs*) ne sont pas supprimés.

**Suppression de paquets d'installation**

Le gestionnaire de paquets APT conserve une copie de chaque paquet installé et désinstallé sur le périphérique de stockage du système (répertoire **/var/cache/apt/archives**).

L’utilisation des commandes suivantes permet de récupérer de l'espace disque ...

* La commande **apt clean** supprime la totalité des paquets présents dans le répertoire **/var/cache/apt/archives**.  
  Cela n'a toutefois aucun impact sur les paquets installés.  
  **>> sudo apt clean**
* La commande **apt autoclean** permet de supprimer les paquets présents dans le répertoire **/var/cache/apt/archives**, tout en conservant ceux qui ont un équivalent dans les référentiels.  
  Si le répertoire **/var/cache/apt/archives** contient plusieurs versions du même logiciel, seule la version à jour (ayant un équivalent dans le dépôt) sera conservée.   
  Ceci permet de récupérer de l'espace de stockage, mais moins toutefois qu'avec la commande **apt** **clean**.

**Recherche de paquets**

La commande **apt list** permet de rechercher des paquets et d’indiquer si le paquet est installé ou non.

La commande **apt-cache showpkg** permet d’afficher une liste exhaustive des paquet disponibles pour un motif ...  
**>> sudo apt-cache <Paquet(s)>  
>> sudo apt-cache nginx**

Le résultat de la commande permet de savoir quelles sont les dépendances obligatoires (paquets qui doivent être installés, habituellement des librairies) pour un paquet spécifique.  
Le résultat indique quels sont les paquets qui peuvent être installés comme complémentaires.

**Exploration des paquets Debian**

La commande **apt-file** permet d’explorer des paquets Debian (**.deb**) et de retrouver un paquet en fonction d’un fichier.

**Remarque** ...  
Il est possible que l’on doive installer le paquet **apt-file** afin de l’utiliser.  
Il faudra ensuite préparer **apt-file** à être utilisé en remplissant son cache à l’aide de la commande **apt-file update**.

**Liste des fichiers d’un paquet**

La commande **apt-file** permet de lister les fichiers qui sont contenus dans un paquet Debian (.deb).   
Un paquet Debian est un ensemble de fichiers (conf, script, stockage d’information) mis dans un paquet et prêt à être déployés.

Pour afficher les fichiers qui sont contenus dans un paquet Debian...  
**>> sudo apt-file show <Paquet(s)>  
>> sudo apt-file show fail2ban**

L’utilitaire **apt-file** affiche la liste les fichiers contenus dans le paquet fail2ban.  
On peut noter que différents fichiers sont dans le répertoire /etc/default et d’autre dans le répertoire /etc/fail2ban/, ...

**Identification du paquet d’un fichier**

On peut retrouver, à partir d’un fichier nommé, le paquet Debian associé à celui-ci ...  
**>> sudo apt-file search <Fichier)>  
>> sudo apt-file search /usr/bin/htpasswd**

Cette commande permet de retrouver le paquet associé à un fichier ou une commande.

**Utilitaires intéressants**

Beaucoup d’administrateurs système utilisent des commandes comme **ifconfig** ou **nslookup**.  
Toutefois, ces utilitaires sont marqués comme obsolètes, c’est-à-dire qu’il n’y a plus de développement pour ces commandes … **ifconfig** a été remplacée par la commande **ip** et **nslookup** par la commande **dig**.

Une image contenant texte

Description générée automatiquement

La plupart des distributions ne les propose plus ces utilitaires. Ils peuvent néanmoins demeurer utiles.  
Pour les installer …  
**>> sudo apt-install net-tools dnsutils**

Deux autres utilitaires peuvent également peuvent être fort utiles …

**fail2ban** est un système de prévention d’intrusions (*Intrusion Prevention System* ou IPS) qui analyse les journaux (*logs*) de différents services (SSH, Apache/nginx, FTP…) en cherchant des correspondances entre des motifs définis dans ses filtres et les entrées des journaux.   
Si une correspondance est trouvée, une ou plusieurs actions sont exécutées  
Typiquement, fail2ban cherche des tentatives répétées de connexions infructueuses dans les fichiers journaux et procède à un blocage en ajoutant une règle au pare-feu (iptables ou nftables) afin de bannir l'adresse IP de la source.  
**>> sudo apt list fail2ban  
>> sudo apt install fail2ban**

**git** est un autre utilitaire intéressant.  
Ce dernier permet de télécharger (*download*) ou téléverser (*upload*) sur un serveur central des fichiers tels que du code ou des fichiers HTML.

Ces serveurs centraux dont github et gitlab permet à des développeurs de partager leur environnement de travail.  
**>> sudo apt list git  
>> sudo apt install git**

**SSH**

Secure Shell (SSH) est à la fois un utilitaire et un protocole de communication sécurisé.

Le protocole de connexion impose un échange de clés de chiffrement en début de connexion.   
Par la suite, tous les segments TCP sont authentifiés et chiffrés.  
Il devient donc impossible d'utiliser un renifleur (comme Wireshark) afin de voir les transactions entre deux entités.

Le protocole SSH a été conçu avec l'objectif de remplacer les différents programmes   
rlogin, telnet, rcp, ftp et rsh.

**Vocabulaire**

|  |  |
| --- | --- |
| Terme | Définition |
| Cryptologie | Science du secret dans la transmission de l'information |
| Cryptographie | Discipline de la cryptologie assurant notamment la confidentialité, l'authenticité et l'intégrité du message dans une transmission |
| Chiffrer | Altération d’un message avec comme objectif qu’il ne soit plus lisible |
| Déchiffrer | Modification d’un message chiffré afin de récupérer les données d'origine, lisibles et compréhensibles |

Le principe de la cryptographie est de …

* déterminer un algorithme permettant de chiffrer un message ;
* communiquer cet algorithme à un correspondant afin qu'il puisse déchiffrer le message.

**Installation et configuration de base SSH**

Le paquet le plus répandu utilisé sous Linux pour fournir des connexions sécurisées est openSSH.

Le paquet openSSH utilise le protocole Secure Shell (SSH) afin d’établir une connexion chiffrée entre deux périphériques réseau.  
il utilise le chiffrement asymétrique comme mécanisme d'authentification.

openSSH peut gérer différents types de clés telles que **RSA**, **DSA** et **DSA basé sur les courbes elliptiques**.   
OpenSSH reconnaît également les certificats **X509** et les fichiers au format **PKCS#12**.

L'installation du service openSSH crée un répertoire **/etc/ssh** et génère un couple de clés RSA …

* /**etc/ssh/ssh\_host\_rsa\_key** contenant la clé privée du serveur ;
* **/etc/ssh/ssh\_host\_rsa\_key.pub** contenant la clé publique du serveur.

L'installation comporte donc une étape de génération des clefs de chiffrement.  
Par la suite, le service SSH est lancé.  
**>> sudo apt install openssh-server --yes**

**Remarque** …  
openSSH est habituellement installé sur les distributions populaires.

## Établissement d'une connexion SSH

Les étapes de l'établissement d'une connexion SSH sont …

* Le serveur envoie sa clef publique au client.   
  Le client vérifie qu'il s'agit bien de la clef du serveur, s'il l'a déjà reçue lors d'une connexion précédente ;
* Le client génère une clef secrète (chiffrement symétrique) et l'envoie au serveur, en chiffrant cet échange avec la clef publique du serveur (chiffrement asymétrique).   
  Le serveur déchiffre cette clef secrète en utilisant sa clé privée  
  Cette dernière étape confirme qu'il est bien le vrai serveur ;
* Comme preuve de confirmation au client SSH, le serveur chiffre un message standard avec la clef secrète et l'envoie au client.  
  Si le client est en mesure de déchiffrer ce message standard en utilisant la clef secrète, il a la preuve que le serveur est bien le vrai serveur ;
* Une fois la clef secrète échangée, le client et le serveur peuvent alors établir un canal sécurisé (un tunnel) grâce à la clef secrète commune (chiffrement symétrique) ;
* Une fois que le canal sécurisé est en place, le client peut envoyer au serveur une demande d’ouverture de session (nom et mot de passe de l’utilisateur).   
  Le canal sécurisé demeure en place jusqu'à ce que l'utilisateur se déconnecte.

**Première connexion avec SSH**

**Authentification par mot de passe**

L’authentification par mot de passe est la méthode la plus simple.

Depuis l’hôte client …  
**>> ssh <Identifiant>@<Nom DNS ou Adresse IP du serveur>  
>> ssh tux@192.168.1.100**ou **>> ssh tux@srv01.tux.local**

Si le même identifiant est utilisé sur le client et sur le serveur …  
**>> ssh <Nom DNS ou Adresse IP du serveur>  
>> ssh 192.168.1.100**ou **>> ssh srv01.tux.local**

Si on doit utiliser un port différent …  
**>> ssh <Nom DNS ou Adresse IP du serveur> -p <Port>  
>>** **ssh tux@srv01.tux.local -p 12345**Le commutateur **-p <Port>**, qui précise le port utilisé par le serveur, est facultative.   
Si rien n'est précisé, c'est le port TCP 22 qui sera utilisé par défaut.

Pour se connecter avec SSH en IPV6 depuis un terminal ...  
**>> ssh -6 <Adresse IPv6 du serveur>  
>>** **ssh -6 tux@2a01:e35:2431::2e57**

**Remarque** ...  
Afin de pouvoir se connecter en IPV6, il faut que le serveur SSH soit en mesure d’écouter les adresses IPV6.

Pour ce faire, il faut ajouter le code suivant dans le fichier **/etc/ssh/sshd\_config** sur le serveur ...  
**ListenAddress ::**

**Configuration du serveur SSH**

Le fichier de configuration du serveur SSH est **/etc/ssh/sshd\_config**.

Le répertoire **/etc/ssh** est l'emplacement de stockage des fichiers de configuration Secure Shell.   
Le fichier de configuration **/etc/ssh/sshd\_conf** définit les options du serveur au format …

**Directive Valeur**

**Attention** ...  
Si on a modifié le fichier de configuration du serveur, il faut indiquer au service de relire son fichier de configuration …  
**>> sudo systemctl reload|restart|reload-or-restart sshd.service**

Voici les directives les plus importantes de ce fichier de configuration …

**Port**Cette directive indique que le serveur SSH écoute sur le port 22, qui est le port par défaut de SSH.   
**Port 2096**

**ListenAddress**Dans certains cas, on peut avoir plusieurs interfaces réseau (ou interfaces virtuelles) et on aimerait limiter le serveur SSH afin qu'il n'écoute que certaines des interfaces réseau.   
**ListenAddress 192.168.1.100:192.168.1.101**

Il est possible d’envisager de ne pas autoriser l'utilisateur root à se connecter directement à l’aide du protocole SSH, en particulier si le système est connecté à Internet.   
Autoriser la connexion root à l’aide du protocole SSH offre aux pirates potentiels la possibilité de se connecter avec des privilèges élevés.  
Ceci peut s’avérer dangereux.

**Directive PermitRootLogin**Cette directive permet d’éviter une attaque par force brute en essayant plusieurs mots de passe pour obtenir un accès.

Cette directive, qui est définie par défaut avec la valeur prohibit-password, interdit toutes les méthodes d'authentification interactives, n'autorisant que celle utilisant une clé publique.   
Les valeurs peuvent être **yes**, **prohibit-password**, **forced-commands-only**, ou **no**.

* Si cette option est définie sur **prohibit-password**,   
  le mot de passe etl'authentification interactive au clavier sont désactivés pour l’utilisateur root ;
* Si cette option est définie sur **forced-commands-only,**   
  une connexion root avec authentification par clépublique est autorisée.  
  Toutes les autresméthodes d'authentification sont désactivées pour l’utilisateur root ;
* Si cette option est définie sur **no**, root n'est pas autorisé à se connecter.

**Directive AllowUsers et DenyUsers**

Il est également possible de limiter seulement certains utilisateurs pouvant se connecter à l’aide de SSH  
Dans ce cas, il est proposé deux choix … **AllowUsers** ou **DenyUsers**.

Ces deux mots clés permettent de spécifier une liste de noms d'utilisateurs …  
**AllowUsers tux kermit squiddly**

* Si on utilise le **directive AllowUsers**,   
  seuls ces utilisateurs seront autorisés à se connecter au serveur SSH ;
* Si on utilise le **directive DenyUsers**,   
  tous les utilisateurs peuvent se connecter au serveur SSH, sauf les utilisateurs identifiés.

Il est possible d’utiliser des caractères génériques afin de faire correspondre de motifs …  
# Refuser à tout utilisateur ayant un nom débutant par le motif invit … **DenyUsers invit\***# Refuser à tout utilisateur ayant un nom qui contient 4 caractères et qui se termine par tux …  
**DenyUsers ????tux**

**Remarque** …  
Il existe également une directive **AllowGroups** et une directive **DenyGroups**.

Le tableau suivant répertorie les directives les plus courantes que l’on devra peut-être modifier sur le serveur openSSH.

|  |  |
| --- | --- |
| Directive | Description |
| Banner | Spécifie le fichier à afficher avant la connexion d’un usager |
| AllowUsers | Autorise la liste d'utilisateurs spécifiée à se connecter à l'aide de SSH |
| DenyUsers | Empêche la liste d'utilisateurs spécifiés de se connecter à l'aide de SSH |
| PasswordAuthentication | Autorise l'authentification à l'aide de mots de passe textuels |
| MaxAuthTries | Spécifie le nombre de tentatives qu’un utilisateur peut entrer un mot de passe (défaut 6) |
| PermitRootLogin | Autorise le compte utilisateur root à se connecter à l'aide de SSH |
| PermitEmptyPassword | Spécifie si un usager peut se connecter sans mot de passe (défaut No) |
| PrintMotd | Spécifie si le contenu du fichier /etc/motd est affiché lorsqu’un usager se branche avec SSH |
| Protocol | Spécifie le niveau de protocole de chiffrement – Le niveau 2 est préféré et plus sécurisé |
| PubkeyAuthentication | Autorise l'authentification à l'aide de certificats |
| X11Forwarding | Autorise les serveurs X distants à exécuter des applications client X à l'aide d'un tunneling |
| AllowTcpForwarding | Autorise le serveur à accepter les protocoles en tunnel |
| listenAddress | Spécifie les interfaces qu’écoute SSH |
| Port | Modifie le serveur SSH pour écouter un autre port |

Après avoir apporté des modifications à la configuration, il faudra redémarrer le service openSSH pour que celles-ci prennent effet.   
Il est également possible de simplement recharger le fichier de configuration, cela permet de ne pas déconnecter les utilisateurs utilisant le service.

Pour ce faire, on utilise …  
**>> sudo systemctl reload|restart|reload-or-restart sshd.service**

**Utilisation d’openSSH**

Il est intéressant de connaître plusieurs commandes de client SSH …

* **ssh**  
  Cette commande permet de se connecter à un serveur SSH et de travailler sur le serveur à l’aide d’une ligne de commande ;
* **scp**  
  Cette commande permet de transférer des fichiers vers et/ou depuis un serveur SSH à l’aide de la ligne de commande ;
* **sftp**  
  Cette commande permet de se connecter à un serveur SSH et d'exécuter des commandes de type **ftp**.

Pour une meilleure utilisation, on se doit connaître la fonction de chacune de ces commandes ainsi que certaines directives clés du fichier **/etc/ssh/ssh\_config**.

**Fichier ssh\_config**

La directive Host permet de spécifier différents paramètres pour différents serveurs SSH auxquels on peut se connecter.   
Par exemple, les éléments suivants appliqueraient la valeur 0 de **ConnectTimeout** lors de la connexion à srv01.tux.info et la valeur 600 lors de la connexion à profsavad.info …

**Host srv01.tux.info  
 ConnectTimeout 0  
Host test.tux.org  
 ConnectTimeout 600**

**Remarque** …  
La plupart des directives du fichier ssh\_config sont liées aux directives du fichier de configuration du serveur SSH.

### Commande ssh

La première fois que l’on essaie de se connecter à un hôte à l’aide de la commande ssh, on est invité à vérifier l'empreinte de la clé RSA …  
**>> ssh** [**tux@srv01.tux.local**](mailto:tux@srv01.tux.local)

### Connexion à un serveur SSH

Depuis l’hôte client …  
**>> ssh <Identifiant>@<Nom DNS ou Adresse IP du serveur>  
>> ssh tux@srv01.tux.local  
>> ssh tux@192.168.1.100**

Si le **même identifiant est utilisé** sur le client et sur le serveur …  
**>> ssh <Nom DNS ou Adresse IP du serveur>  
>> ssh srv01.tux.local**   
**>> ssh 192.168.1.100**

Si on doit se **connecter sous un autre compte utilisateur**, …  
**>> ssh -l <Nom d’utilisateur>@<Nom DNS ou Adresse IP du serveur>  
>> ssh -l kermit@srv01.tux.local**   
**>> ssh -l kermit@192.168.1.100**

Pour exécuter une commande sur le système distant, mais revenir immédiatement à son système client, on utilise la syntaxe …  
**>> ssh <Nom DNS ou Adresse IP du serveur> <Commande>  
>> ssh 192.168.1.100 "sudo apt update && sudo apt upgrade -y"**    
**>> ssh 192.168.1.100 "sudo systemctl status sshd.service"**

**Commandes scp et sftp**

Pour copier un fichier de l’hôte actuel vers un hôte distant, on utilise la syntaxe suivante …  
**>> scp <Nom du fichier> <Nom DNS ou Adresse IP du serveur>:<Répertoire>  
>> scp /etc/hosts 192.168.1.100:/tmp**  
**>> scp monfichier.texte tux@192.168.1.100:/home/tux/Documents/monfichier.texte**

Pour transférer un fichier depuis l’hôte distant vers le système local, on utilise …  
**>> scp tux@192.168.1.100:/home/tux/Documents/monfichier.texte monfichier.texte**

Une fonctionnalité pratique de scp est la possibilité de copier des fichiers entre deux serveurs distants …  
**>> scp** tux**@192.168.1.100:monfichier.texte tux@192.168.1.101:monfichier.texte**

La commande sftp se connecte à un serveur SSH et fournit une interface client de type ftp …  
**>> sftp** [tux@srv01.tux.local](mailto:tux@srv01.tux.local)  
**sftp> pwd  
sftp> ls  
sftp> get tux.texte .  
sftp> ls  
sftp> bye  
>>**

**Connexion sans mot de passe**

**Étape 1**Sur l’hôte client SSH, on exécute la commande ssh-keygen ...   
**>> ssh-keygen -t rsa**

Pour confirmer la présence des clés …  
**>> ls ./.ssh/**

Pour afficher le contenu de la clé publique …  
**>> cat ./.ssh/id\_rsa.pub**

**Étape 2**Il existe maintenant une clé publique dans le fichier **~/.ssh/id\_rsa.pub**.

Celui-ci doit être copié sur le serveur distant dans le fichier **~/.ssh/registered\_keys** de son compte sur l’hôte distant.  
On devra peut-être créer ce répertoire et définir les autorisations sur 770) …  
**>> ssh-copy-id -i id\_rsa.pub srv01.tux.local**

La clé publique va se retrouver dans le fichier **/home/tux/.ssh/authorized\_keys** sur le serveur.

**Étape 3**On définit les autorisations du fichier allowed\_keys sur 640 …  
**>> ssh -l tux srv01.tux.local 'chmod 640 /home/tux/.ssh/authorized\_keys'**

Il est maintenant possible de **se connecter au serveur SSH sans mot de passe** …  
**>> ssh -l tux srv01.tux.local 'date'**

**Utilisation de la clef de chiffrement sans mot de passe**

**Étape 1**On démarre un nouvel interpréteur de commandes (*shell*) avec l'utilitaire **ssh-agent** …  
**>> ssh-agent /bin/bash**

**Étape 2**On exécute la commande suivante …  
**>> ssh-add ~/.ssh/id\_rsa**

À partir de ce moment, lorsque l’on utilise l’interpréteur de commande **ssh-agent** pour se connecter à distance au serveur SSH, la connexion a lieu sans mot de passe.

**Bonnes pratiques**

Bien que le protocole SSH chiffre les échanges afin de les sécuriser, il important d’ajouter des mesures de sécurité supplémentaires aux accès SSH afin de rendre les tentatives d’attaque plus complexes à réaliser.

* 01 – Désactivation de l’ouverture de session pour l’utilisateur root ;
* 02 – Modification du port d’accès SSH ;
* 03 -- Utilisation forcée de la version 2 du protocole ;
* 04 – Utilisation de l’utilitaire fail2ban ;
* 05 – Écoute sur une seule interface réseau ;
* 06 – Authentification par clés :
* 07 – Autorisation/interdiction de l’accès à certains utilisateurs spécifiques ;
* 08 – Vérification des journaux (*logs*) ;
* 09 – Fermeture de tous les accès SSH à la fin de l’utilisation (ne pas laisser de connexions ouvertes).

**Gestionnaire de paquets RHEL  
dnf** (*Dandified Yum*) est le gestionnaire de paquet pour les distributions RHEL.  
Il remplace le gestionnaire yum.

|  |  |
| --- | --- |
| Commande | Description |
| dnf update | Vérification pour les mises à jour disponibles pour le système |
| dnf check-update <Paquet(s)> | Vérification pour les mises à jour disponibles pour un ou des paquets installés |
| dnf install <Paquet(s)> [--yes] | Installation d’un ou des paquets |
| dnf download <Paquet(s)> | Téléchargement (sans installation) d’un ou des paquets |
| dnf remove <Paquet(s)> | Suppression d’un ou des paquets |
| dnf reinstall <Paquet(s)> [--yes] | Réinstallation d’un ou des paquets |
| dnf repolist | Affichage des référentiels |
| dnf search <Paquet(s)> | Recherche d’un paquet spécifique |
| dnf provides <Fichier> | Recherche du paquet qui inclut ce fichier |
| dnf info <Paquet> | Affichage de l’information sur un paquet |
| dnf history | Affichage de l’historique des transactions |
| dnf clean all | Nettoyage du cache (afin de libérer de l’espace disque |
| dnf list installed | Affichage des paquets installés |

**Gestionnaire de paquets SUSE  
zypper** est le gestionnaire de paquet pour les distributions SUSE.

|  |  |
| --- | --- |
| Commande | Description |
| zypper update | Vérification pour les mises à jour disponibles pour le système |
| zypper list-updates | Vérification pour les mises à jour disponibles |
| zypper in <Paquet(s)> [--yes] | Installation d’un ou des paquets |
| zypper rm <Paquet(s)> | Suppression d’un ou des paquets |
| zypper se <Paquet(s)> | Recherche d’un paquet spécifique |
| zypper se -- provides <Fichier> | Recherche du paquet qui inclut ce fichier |
| zypper info <Paquet> | Affichage de l’information sur un paquet |

**Remarque** …  
La distribution SUSE propose l’outil **YaST** (*Yet another Setup Tool*) qui permet d’administrer, installer et configurer SUSE Linux.