Installation

Choix OS

Avant de procéder à l'installation de notre machine, il faudra choisir la distribution Linux à installer et plu globalement le système d'exploitation que l'on désire utiliser.

Pour les distributions Linux il en existe de divers types :

- Red Hat : qui est plus approprié pour le monde scientifique. Et qui s'est séparé en deux projets distincts :
    - Fedora (a gardé l'idée « Open Source » et est gratuit)
    - RHEL – Red Hat Enterprise Linux (qui a opté pour l'esprit d'entreprise).
- Debian
- Ubuntu
- …

Unix étant un monde vivant (qui va nous obliger à faire des mises à jour, ce qui peut poser des soucis de compatibilité). Il faudra faire en sorte que les OS et les softwares restent fonctionnels et compatibles entre eux. Il faut notamment se rendre compte que mettre à jour les serveurs est une opération risquée et compliquée.

Il faudra dès lors faire bien attention et veiller à la certification du matériel, à être en phase avec le métier, le logiciel imposé (parfois) par l'entreprise. Il faudra également regarder la durée de support (s'il y en a un) ainsi que le cycle entre des nouvelles versions (pas trop peu espacées : cela voudrait dire des mises à jour fréquentes mais pas trop espacées non plus : logiciel en bout de vie).

Exemple :

J'hésite entre les distributions Fedora et RHEL de Red Hat. Il faudra donc bien se rendre compte que si je prends Fedora, celle-ci sera bien entendu une version gratuite de RHEL mais je n'aurai aucun support et qu'il ne s'agit que d'une compilation des codes open source de RHEL. Cela peut donc être un risque de prendre une telle version malgré son côté gratuit.

Nom du serveur

Lors de la mise en place d'un serveur il faudra lui donner un nom. Cela peut demander peut demander réflexion, en effet dans une entreprise possédant une grande architecture, il faudra pouvoir retrouver le serveur en question ± facilement.

Pour cela on va tenter d'utiliser une règle de toponimie : nomEntreprise_numServ. En fin de compte peu importe les règles que l'on désire mettre ne place, il faudra surtout penser à les documenter. Cela en est même devenus une loi. Il faudra tout de même faire attention lors de l'attribution de noms aux serveurs (genre mail, …) car cela peut hacher le travail aux hackers le permettant de facilement comprendre notre architecture et donc les machines à cibler.

Virtualisation

Actuellement la virtualisation est en plaine expansion. Cela permet de ne plus avoir à acheter différents types de serveurs (mail, web, ..). On achète directement une machine globale qui va se charger des différents rôles. Cela nous permettra de mieux dispatcher les différentes ressources disponibles.

On notera également deux modes pour les machines virtuelles :

- Shared : la machine virtuelle utilise la même adresse IP que la machine physique.
- Bridged : la machine virtuelle utilise une adresse IP différente de la machine physique. Ce qui est obligatoire si l'on désire utiliser une adresse IP différente de la machine physique.

L'utilisation de machines virtuelles va également nous permettre de « snapshoter » notre machine virtuelle. Il s'agit de faire une copie de tous les disques durs de la machine virtuelle et permet notamment de revenir en arrière rapidement en restaurant l'état de notre machine à ce moment donné.

Installation d'une machine Debian

Lors de ce cours nous avons installé une machine Debian sans interface graphique (ne contenant donc qu'un shell nous permettant d'y entrer des commandes.

Le problème avec le GUI c'est que cela cache certains processus et que cela est plus distant de la machine. Afin d'avoir un contrôle total sur ce que l'on fait et afin d'être le plus proche possible de la machine on préfère ne pas utiliser d'interface graphique.

La bonne pratique pour le dossier d'installation d'un serveur est « /usr/local » ou « /opt » (Ce dernier étant moins conseillé).

Langage

Afin d'être sûr de ne pas avoir d'ambiguïté et afin de pouvoir obtenir une aide plus complète notamment sur internet on utilise l'anglais (une plus grande aide présente dans ce langage-là). De plus il s'agit du langage « natif ».

Mot de passe

Afin d'éviter tout soucis de compatibilité il faudra choisir un mot de passe compatible qwerty / azerty. En effet notre machine pourrait bien être en qwerty lors de son démarrage et vice versa.

On peut d'ailleurs utiliser la commande localectl qui va nous permettre de visualiser / configurer les informations concernant notre clavier.

Exemple : voici la manière de configurer son clavier en azerty :

localectl set-keymap be-latin1

En soit une bonne pratique également est d'utiliser plutôt que des mots de passe des phrases de passe (passphrase) Exemple : IPL4EVER !

Il faut également penser à privilégier la longueur. Au plus notre programme est long, au plus il sera dur à casser.

NTP

Notre serveur doit absolument être à l'heure (à la seconde près). Il arrive parfois que l'on est des dérives de temps (une différence qui s'agrandit entre le serveur et le temps réel). Afin d'éviter cette dérive on va utiliser le protocole NTP

Ces derniers sont considérés comme les gardiens de l'heure. NTP possède notamment des serveurs que l'on peut interroger afin d'obtenir l'heure actuelle. Afin d'éviter une surcharge de ces serveurs (ceux-ci reçoivent de nombreuses requêtes).

On a ajouté une couche supplémentaire (couche stratum) qui demande lui-même l'heure aux différents serveurs NTP et en fait un calcul moyen. On va utiliser plusieurs couches stratum afin de diminuer la charge des serveurs NTP (chaque couche effectuera une moyenne des serveurs de la couche précédente).

Logs

On tentera de placer tous les logs de notre système au sein de « /var/log ». Il faut absolument garder des fichiers logs, ceux-ci sont un peu nos journaux de bords.

En fait la loi nous oblige à garder nos fichiers de log pour une durée minimale de 1 an. (Ceux-ci pouvant être utilisé dans le cadre d'une enquête policière, …).

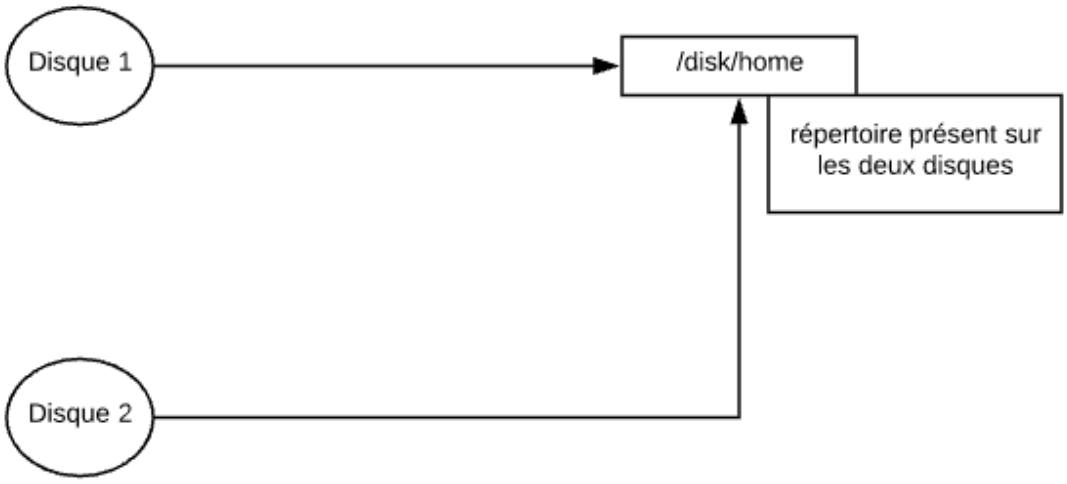
Bonnes pratiques au sujet des dossiers

Le dossier /var

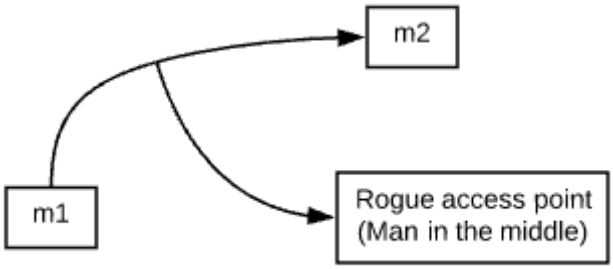
Le mieux au sujet de ce dossier est de le mettre sur une partition différente du système. Si tel n'est pas le cas quelqu'un pourrait nous envoyer de nombreux messages / erreurs afin de surcharger les disques systèmes et empêcher une connexion sur notre serveur.

Le dossier /home

Une autre bonne pratique est de déplacer /home au sein de /disk/home et ainsi faire en sorte que home soit un lien physique.



SSH



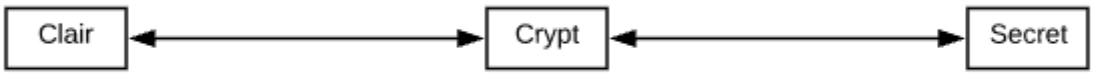
Si maintenant les communications entre moi (m1) et mon serveur (m2) ne sont pas cryptées quelqu'un pourrait lire mes messages le temps de leur transfert (man in the middle).

Quelqu'un pourrait également hacké le serveur DNS et rediriger nos utilisateurs vers un autre site.

Afin de prévenir cela on va utiliser SSH ainsi que de la cryptographie.

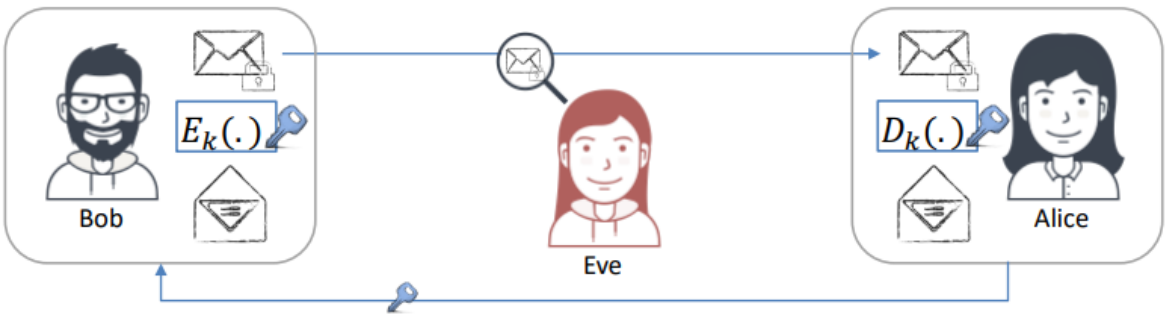
Cryptographie

On va utiliser une manière de « crypter » nos données afin que celles-ci ne transitent plus en clair sur le réseau.



La fonction de crypt utilisée peut être soit confidentielle (Il sera dès lors plus dur de trouver des bugs), soit open source (Dès qu'un bug est découvert on pourra le corriger, tout le monde peut y contribuer).

Encryption symétrique

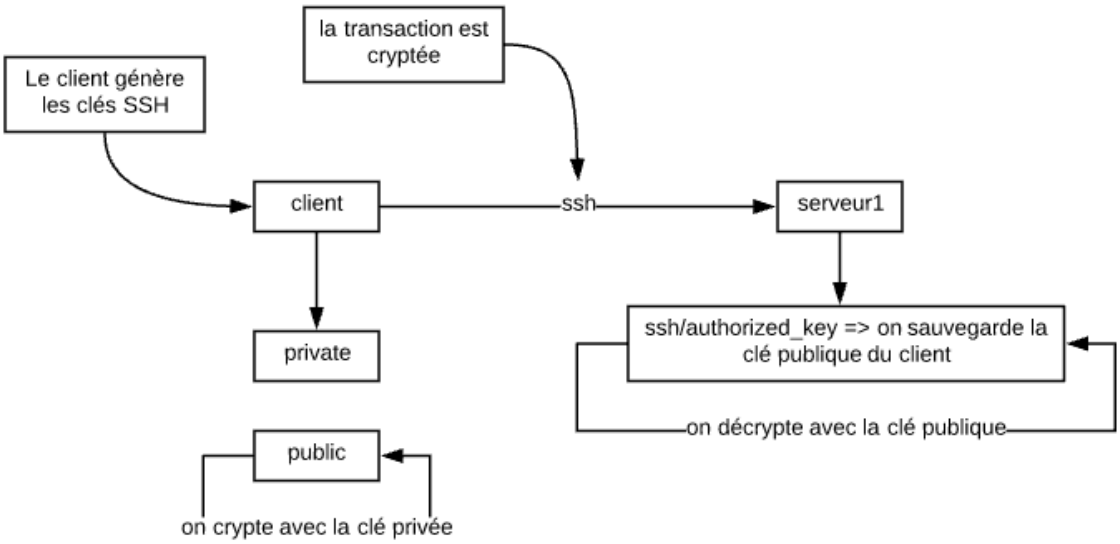


Le problème avec cette technique c'est que l'on va devoir échanger la clé à un moment donné et que celle-ci pourrait être interceptée.

Private Key Infrastructure (PKI)

On va générer deux clés de cryptage : une privée (id_rsa) et une publique(id_rsa.pub), celle-ci devant être unique vis-à-vis des autres clés existantes. Pour générer ces deux clés on utilise la commande : ssh-keygen -t rsa.

Afin de prévenir l'usurpation des messages, on va signer numériquement nos messages à l'aide de notre clé. Ainsi si l'on désire crypter le contenu de nos messages :



De cette manière on envoie de manière cryptée la clé symétrique (générée à l'aide de la clé privée) de façon à avoir un décryptage asymétrique.

Il faudra tout de même penser à protéger le fichier ssh en mode 600 (droit en lecture / écriture pour le propriétaire), mais également penser protéger le répertoire parent en mode 700 (droit en lecture/ écriture / exécution pour le propriétaire).

Backups

Les backups doivent absolument être automatisés et non manuel. Il est très vite arrivé d'oublier de faire un backup manuel, cela demande également du temps à être effectué. On veillera également à mettre le backup sur une autre machine.

On peut notamment utiliser crom afin de planifier nos backups.

Paquets

Il existe divers logiciels (paquets) pouvant être utilisé / installé sur notre machine. Afin de nous permettre leurs installations divers logiciels sont disponibles.

RPM (Red Hat Package Manager)

Il s'agit de la commande de base permettant de gérer les paquets au sein de de Red Hat

- rpm -ivh packageName.rpm
    - -i spécifie que l'on désire installer le package packageName.

    - -v spécifie que l'on ne désire afficher que les informations de niveau « verbose ».
    - -h permet un meilleur affichage.
    - Installer un paquet
- rpm -Uvh packageName.rpm
    - -U permet de que l'on désire mettre à jour le package packageName.
    - -v spécifie que l'on ne désire afficher que les informations de niveau « verbose ».
    - -h permet un meilleur affichage.
    - Installer ou mettre à jour un paquet
- rpm -q -a
    - Liste de tous les packets installés
- rpm -e packageName
    - Désinstaller un paquet
- rpm -q -l packageName
    - Renvoie tous les fichiers installés par un paquet
- rpm -q -f filename
    - Permet de connaitre la paquet d'un fichier en particulier.
- rpm -V packageName
    - Permet de vérifier l'intégrité des packages.
    - Permet de voir si quelque chose a changé au sein de nos packages (hacking, …)

Il faudra cependant faire attention lors d'une utilisation de rpm car celui-ci ne gère pas automatiquement les dépendances. Il nous est donc demandé de gérer à la main les dépendances d'un paquet, ce qui peut vite devenir fastidieux.

YellowDog Update Manager (YUM)

Il s'agit d'un gestionnaire de paquets pour les distributions Linux. Il s'agit en soit d'une surcouche de RPM.

- yum install packageName
    - Permet d'installer un paquet
- yum update [packageName]
    - Permet de mettre à jour un paquet
- yum upgrade
    - Permet de mettre à jour le système.
- yum whatprovides something
    - Permet de savori de quel package provient telle fonctionnalité ou tel fichier.

Base Repository

Il faut savoir que la configuration de yum (ainsi que ces utilitaires y étant reliés) se situera à la base au sein du fichier /etc/yum.conf. Ce fichier permet de spécifier des options de yum qui vont avoir un effet global, il va également permettre d'avoir des sections spécifiques qui vont permettre de mettre des options spécifiques à certains répertoires. Les valeurs que l'on définit au sein des sections individuelles du fichier /etc/yum.conf sont plus fortes que celle définies dans la section principale.

Extra Repository

Il est par contre recommandé de définir des répertoires individuels au sein de nouveau fichier .repo (ou ceux existants) présents au sein du répertoire /etc/yum.repos.d/.

Commande permettant de lister les ID's de chacun des répertoires :

```
yum repolist all
```

Afin d'activer un répertoire (ou des répertoires), on peut utiliser la commande :

```
yum-config-manager –enable repository
```

Si l'on desire désactiver un répertoire (ou des répertoires), on peut utiliser la commande :

```
yum-config-manager –disable repository
```

Au sein de ces méthodes repository est l'ID unique du répertoire. On peut utiliser des expressions afin de sélectionner certains répertoires.

DPKG / APT

Il s'agit de commandes permettant de gérer les paquets. Mais ceux-ci permettent de gérer automatiquement les dépendances.

dpkg est un peu plus complexe que apt.