Fiches Linux – Exercices

Sacré Christopher

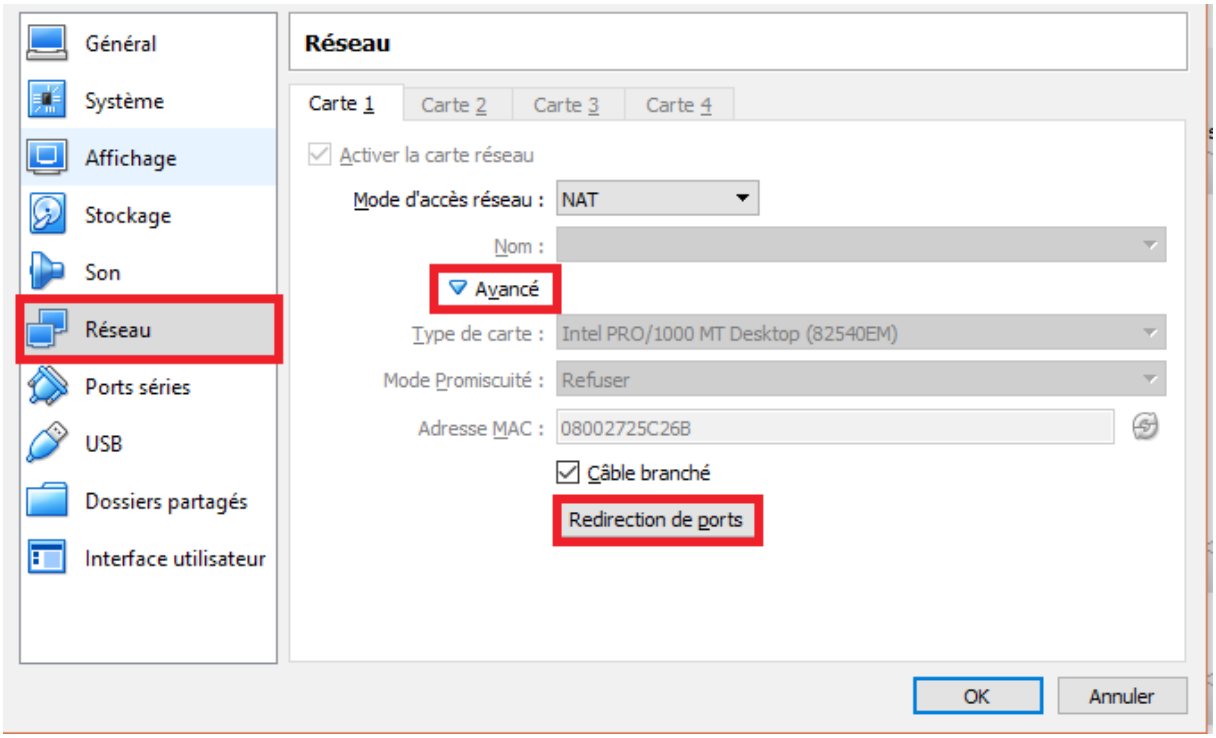
Virtual Box

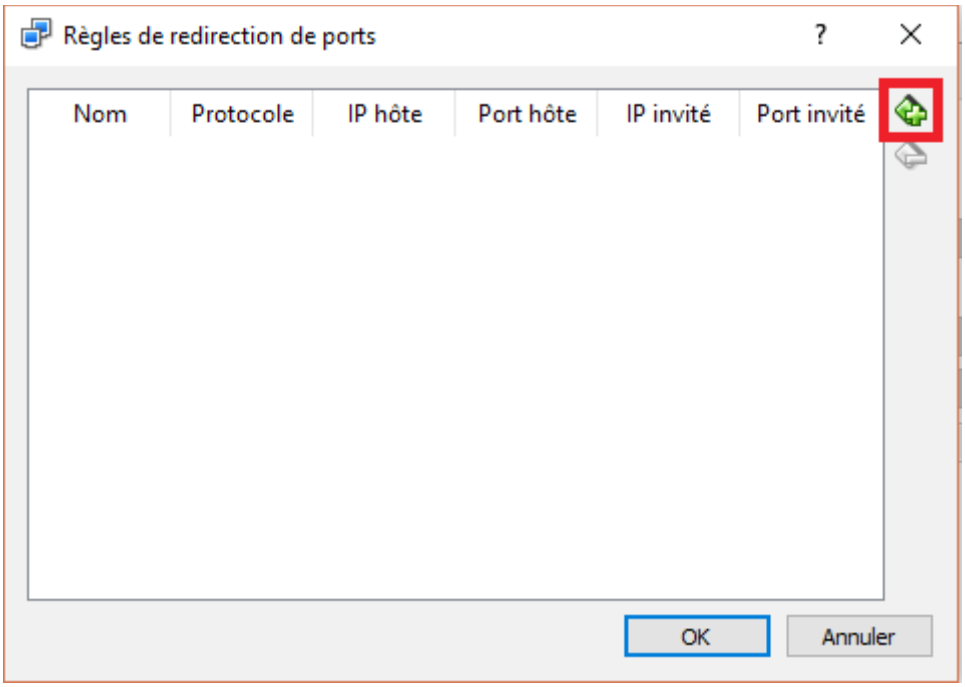
Introduction

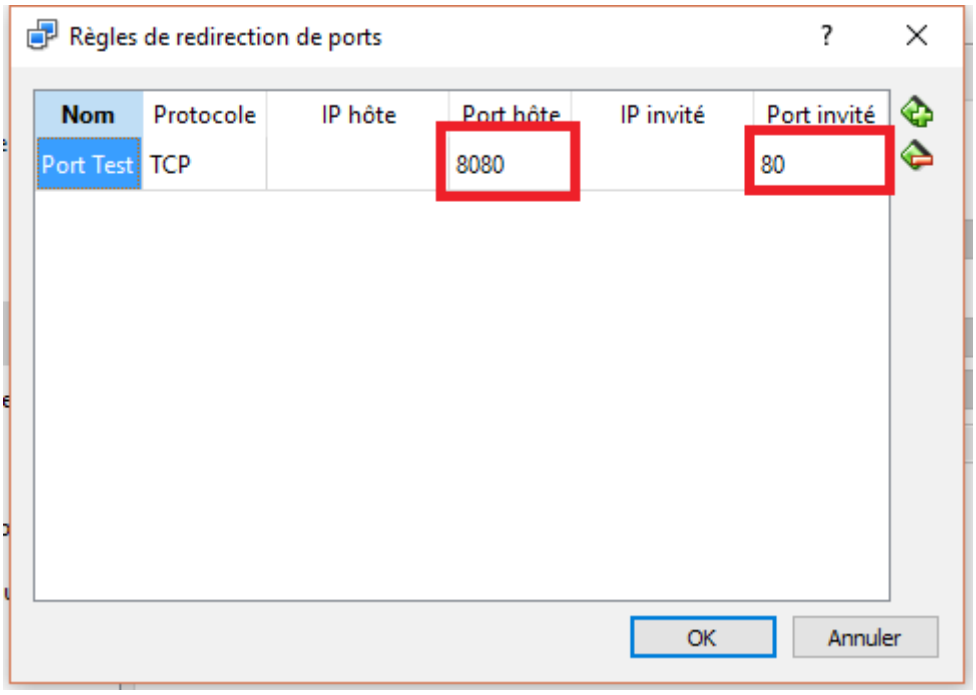
Les avantages de la virtualisation sont divers mais on notera notamment la création rapide d'un environnement de test similaire à la production et la réalisation de « snapshots » permettant de sauvegarder l'état actuel de notre machine avant toute modification.

Accéder à une machine virtuelle depuis notre machine

Pour ce qui est du réseau, une machine virtuelle est configurée en NAT. À cause de cette configuration, on ne peut par défaut pas accéder à la machine hôte depuis la machine virtuelle. Si l'on désire accéder à la machine virtuelle depuis le réseau pour notamment tester différents services (Serveur Web, SSH, ...), il faudra simplement configurer la redirection du port du NAT. On fera dès lors correspondre un numéro de port de la machine hôte à un numéro de port de la machine invitée. De cette manière, toutes les requêtes adressées vers le port hôte seront donc redirigées vers le port de la machine virtuelle.







Environnement de test

Il est souvent utile d'avoir recours à plusieurs machines virtuelles pouvant communiquer entre elles. Pour cela nous allons utiliser le « host only networking » (en bon français : le réseau privé d'hôte). Pour ce faire nous allons ajouter une carte réseau configurée sur ce mode réseau. Ainsi, les machines virtuelles pourront communiquer entre elles sans perturbations pour le « véritable » réseau physique connecté à la machine hôte. Malgré cela il faudra tout de même définir un réseau ainsi que des adresses IPs pour les machines virtuelles hôtes.

Outils pour l'administrateur système

Un administrateur système sera tôt ou tard confronté à des problèmes variés. Celui-ci devra donc disposer d'outils et de méthodes pour résoudre ces problèmes. Pour cela il pourra :

Consulter la documentation

De nombreuses informations peuvent être trouvées sur internet. Il faudra néanmoins faire attention et enter de récupérer une documentation se rapprochant le plus possible de notre distribution et de notre version de Linux.

Les pages du man sont également un bon moyen de récupérer des informations. Il existe notamment des pages « info » accesible via le logiciel/paquet « pinfo ».

De plus chaque paquet dispose d'une documentation spécifique qui est installée dans le /usr/share/doc/ « nomDeMonPaquet ». Si la documentation est malheureusement trop importante, le paquet peut se voir accompagné d'un « paquet-doc ».

Le plus souvent les logiciels sont installés dans /usr/share/ « nomLogiciel » tandis que les fichiers de configuration se trouve dans /etc/ « nomLogiciel ». (Attention ceci n'est surtout vrai que pour les machines Debian.

Consulter les logs

Le plupart des logiciels vont générer des logs qui seront stockés (la plupart du temps) au sein de « /var/log/nomLogicielOuNomPaquet ».

Actuellement les distributions Linux utilisent le système d'initialisation « systemd ». Celui ci permettra le démarrage de nos services/démons. Afin d'afficher journal de systemd on utilisera la commande:

journalctl -xf

Réseau

Dû au fait que de nombreux services / démons sont accesibles à distance (et tournent donc sur un port réseau), il est donc utile d'avoir un inventaire des services « réseaux » qui tournent sur notre machine. On utilisera pour cela la commande « netstat » :

netstat -taupe

En plus de cette commande, le fichier /etc/services nous permet également de voir les ports préréservés par le système.

APT – Gestion des packages

Toutes les distributions Linux dispose d'un système de gestion des packages permettant l'installation facile de logiciels et services. Ce système de gestion de packages résout en outre les problèmes de dépendances.

Fonctionnement

Tout d'abord il faut savoir que de nombreux dépôts contenants des paquets Debian (.deb) sont disponibles sur internet (Il s'agit de logiciel prêt à être installé). L'outil APT dispose d'un fichier de configuration (/etc/apt/sources.list) permettant de renseigner les dépôts à utiliser. Il suffit ensuite de mettre à jour depuis les dépôts (mise à jour du cache local) et de demander l'installation du logiciel à APT. L'outil installera automatiquement les dépendances nécessaires pour le logiciel demandé.

Backports

Dû au fait que la distribution Debian mise sur la stabilité, il n'est donc pas rare de ne pas avoir les dernières versions de certains logiciels. C'est là que le dépôt Backports va nous être utile, il contient tous les paquets actuellement en cours de test (« testing »), ces paquets seront donc plus récent mais également plus instables.

Afin d'utiliser les backports il faudra ajouter cette ligne au fichier /etc/apt/sources.list :

deb http://ftp.debian.org/debian jessie-backports main

Une autre manière de faire sera de procéder à une installation manuelle. Mais dans un tel cas, les mises à jour ne se feront pas à l'aide d'un simple apt-get upgrade.

Utilisation

Mettre à jour depuis les dépôts

apt-get update

Installer un logiciel

apt-get install <paquet1> <paquet2> ...

Supprimer un logiciel

apt-get remove <paquet1> <paquet2> ...

Rechercher un logiciel

apt-cache search <word>

Mettre à jour le système

apt-get upgrade

Sécurité

Les mises à jour des distributions Linux sont le plus souvent assurées par l'outil APT. La première règle en terme de sécurité informatique étant de garder un système le plus à jour possible, APT est un outil important pour se prémunir contre des attaques éventuelles.

Dû à cela, un administrateur système doit donc réaliser des mises à jour de son système régulièrement et en particulier des mises à jour de sécurité (corrections de bogues, corection de failles, ...). Debian offre donc la possibilité de distinguer ces mises à jours des autres (cela se fait au sein du fichier /etc/apt/sources.list) et il sera donc facile d'appliquer uniquement ce type de mises à jour.

Il est également possible d'automatiser cette application de mises à jours à l'aide notamment d'outils tels que Cron.

De plus il existe même un paquet pour installer automatiquement et quotidiennement les mises à jour de sécurité.

apt-get install unattended-upgrades apt-listchanges

SSH

Les systèmes Linux actuels sont le plus souvent gérés en lignes de commande (pas d'interface graphique) et à distance. Pour ce faire, on pourrait utiliser telnet mais ce protocole a le gros inconvénient de ne rien crypter. Une simple écoute réseau permettrai alors de récupérer le mot de passe root. C'est donc pour cela que SSH est venu remplacer telnet.

Fonctionnement

Le protocole SSH effectue un échange de clés de chiffrement avant d'utiliser ces dernières pour crypter toutes les communications entre le client et le serveur.

Le port 22 est le port par défaut utilisé par SSH.

SSH est un service qui est initialisé/démarré par systemd.

Installation

apt-get install ssh

Configuration

Le fichier de configuration client est : /etc/ssh/ssh_config.

Le fichier de configuration serveur est : /etc/ssh/sshd_config.

Par défaut, SSH est installé pour permettre une authentification par login et mot de passe pour tout les utilisateurs présents sur le serveur (y compris root).

Après avoir effectué une modification dans un fichier de configuration, il faut redémarrer le service pour que les modifications soient prises en compte. On utilisera pour cela :

/etc/init.d/ssh restart

OU service ssh restart

OU systemctl restart ssh

Utilisation

Le client SSH a besoin des informations suivantes : un nom de machine ou une adresse IP, un login et un mot de passe. On peut remplacer cette authentification par login/mdp par une clé. Exemple de commande :

ssh nomutilisateur@nommachineOUadresseIP

Sécurité

Il est possible de configurer le serveur SSH pour interdire l'usage du compte root pour les connexions SSH. L'option « PermitRootLogin » doit être positionnée à « No » dans le fichier de configuration du serveur SSH.

De plus, Il également possible de restreindre l'utilisation que depuis certaines machines et qu'avec certains utilisateurs.

AllowUsers utilisateurAutorisé@sousRéseauAutorisé

Copie de fichiers

Il est à noter que dès que vous avez un accès SSH, vous pouvez copier des fichiers entre votre machine hôte et invitée via SCP/SFTP. Ceci peut se faire avec le logiciel WinSCP (Windows) ou Cyberduck(Mac).

Gestion des utilisateurs

Commandes basiques

adduser

Permet d'ajouter un utilisateur. De plus cette commande crée un profil pour l'utilisateur basé sur un squelette situé dans /etc/skel. Par défaut la home directory créée par adduser sera disponible par tout le monde (Cette configuration sera présente au sein de : /etc/adduser.conf), ceci pourrait ne pas correspondre à notre politique de confidentialité.

deluser

Permet de supprimer un utilisateur.

addgroup

Permet d’ajouter un groupe.

delgroup

Permet de supprimer un groupe ?

SU

Cette commande permet de changer d’utilisateur.

su admin

Sans argument, cela permet de devenir root.

su

Sudo

Cette a pour objectif de permettre à des utilisateurs d’exécuter des commandes en tant que superutilisateur. Attention, pour qu’un utilisateur puisse exécuter une commande avec « sudo », il doit faire partie du groupe sudo.

Installation

apt-get install sudo

Configuration

Par défaut, un utilisateur ajouté au groupe sudo possède les mêmes privilèges que root. On peut cependant changer ce comportement dans le fichier de configuration /etc/sudoers. On peut par exemple faire en sorte qu’un utilisateur sudo ne puisse exécuter que certaines commandes. Le fichier /etc/sudoers s’édite via la commande particulière visudo.

sudo visudo

Utilisation

Pour ajouter un utilisateur au groupe sudo : adduser toto sudo

Pour vérifier l’appartenance d’un utilisateur au groupe sudo : groups

Pour permettre à un utilisateur d’exécuter une commande privilégiée (« root »). Ajouter une ligne dans le fichier /etc/sudoers.

user_name ALL=NOPASSWD: /usr/bin/apt-get install

Passwd

Il est possible de verrouiller ou de désactiver le compte root. Verrouiller le compte root empêche simplement de pouvoir se connecter directement avec le compte root tandis que la désactivation le rend totalement inutilisable.

Pour verrouiller le compte root :

sudo passwd -l root

Pour désactiver le compte root :

sudo usermod --expiredate 1 root

On peut cependant encore utiliser le compte root via :

sudo -s

Sécurité

Sudo

Les avantages du SUDO sont les suivants :

1. Permettre à des utilisateurs d'exécuter une commande en tant que superutilisateur sans devoir le mot de passe de root.

2. Travailler en mode non privilégié et n'utiliser le mode privilégié que quand cela est nécessaire. Ceci réduit le risque de commettre des dommages pour le systèmes.

3. Contrôler et enregistrer qui fait quoi (SUDO enregistre toutes le commandes sudo effectuées dans /var/log/auth.log).

4. Renforcer la sécurité. En désactivant le compte root et en le remplaçant par un compte « sudo », un attaquant ne connaîtra pas le mot de passe mais également le nom du compte !

Politique de sécurité des mots de passe

Il est important d'avoir des mots de passe assez solides et de s'assurer qu'ils ne pourront pas être facilement «crackés ». Les systèmes Linux ont une sécurité de mot de passe par défaut pour les utilisateurs normaux. Les mot de passes doivent avoir une longueur de 6 caractères minimum. Ceci peut s'avérer assez faible comme sécurité. Cette politique de sécurité peut être améliorée notamment ceci :

1. Imposer un minimum de 8 caractères pour les mot de passes

2. N'autoriser que x essais pour le mot de passe

3. Imposer un nombre minimum de caractères différents lors du changement de mot de passe.

4. Fixer une durée de vie minimale et maximale du mot de passe (adduser)

La politique de sécurité se gère au moyen du module PAM(Pluggable Authentication Module) sous Linux.

Pour améliorer la politique de sécurité, on peut installer le paquet suivant :

apt-get install libpam-cracklib

Ensuite dans le fichier de configuration de pam → /etc/pam.d/common-password.
password requisite pam_cracklib.so retry=3 minlen=8 difok=3

1. retry →nombre de tentatives autorisées

2. minlen → nombre minimum de caractères pour le mot de passe

3. difok →nombre de caractères différents entre ancien et nouveau mot de passe

Apache

Fonctionnement

Le principe de fonctionnement d'Apache2 repose sur l'utilisation de modules. En effet, il suffit d'installer et/ou d'activer des modules suivant nos besoins. Il existe donc un module pour PHP, pour activer SSL, pour une authentification LDAP, ...

Apache2 est un service qui est initialisé/démarré par systemd. Pour redémarrer le service :

/etc/init.d/apache2 restart

OU service apache2 restart

OU systemctl restart apache2

En production, un serveur apache s'occupe de servir plusieurs sites Web et/ou sert de serveur HTTP frontal. Ces 2 points seront abordés ci-dessous.

Installation

apt-get install apache2 apache2-doc

L'installation créée un compte et un groupe www-data. Apache 2 fonctionne par défaut sur ce compte et groupe pour des raisons de sécurité et tourne sur le port 80.

Un site de base (page HTML) est placée dans /var/www ce qui permet de tester directement Apache2 après son installation : http://adresseip.

Il est à noter que si vous voulez tester apache sur un serveur ne disposant pas d'interface graphique (et donc pas de navigateur classique), vous pouvez installer lynx qui est un navigateur en mode texte (c'est moche mais cela permet de tester !).

Configuration

Modules

Apache dispose de nombreux modules. Nous n'en ferons pas l'inventaire ici. Pour activer un module il suffit d'utiliser la commande « a2enmod » (apache2 enable module).

Il existe évidemment la commande réciproque « a2dismod ».

N'oubliez pas de redémarrer le service apache2 après activation du module.

a2enmod <<module>>

Configuration PHP

Apache peut être configuré pour servir des pages PHP. Il suffit d'installer PHP ainsi que le module PHP pour apache et de redémarrer le service apache2.

apt-get install php5 php5-mysql libapache2-mod-php5

VirtualHosts

Les virtualhosts permettent de déployer plusieurs sites Web sur un même serveur (même adresse IP). La distinction se fait en général sur le nom du site, apache doit en effet savoir suivant l'url quel site il doit présenter.

L'ajout d'un vhost se fait en créant un fichier dans /etc/apache2/sites-available/ « monsite.conf » et puis d'y insérer :

<VirtualHost *:80>

ServerName monsite.be

nécéssaire pour qu'apache fasse une distinction sur le nom du site. Toute URL comportant « monsite.be » utilisera ce vhost.

ServerAdmin webmaster@localhost

Permet de préciser le responsable du site.

DocumentRoot /var/www/htdocs/monsite

Permet de préciser l'endroit où se trouve l'arborescence du site.

ErrorLog ${APACHE_LOG_DIR}/monsite_error.log

```
        CustomLog ${APACHE_LOG_DIR}/monsite_access.log combined
```

Ces deux lignes vont permettre de déterminer où seront stockés les logs.

```
        <Directory /var/www/htdocs/monsite>
                Require all granted
                AllowOverride All
        </Directory>
</VirtualHost>
```

L'ajout de règles / restrictions sur le site se fera via la directive « Directory ».

```
# n'autoriser l'accès au site que depuis localhost
require ip localhost
# accès uniquement au site pour l'utilisateur admin
require user admin
# pas de redéfinition possible (pas de .htaccess)
AllowOverride None
```

Pour activer le vhost, il suffit d'utiliser la commande « a2ensite » (apache2 enable site).

```
a2ensite monsite
```

Il existe la commande réciproque « a2dissite ».

Ne pas oublier de redémarrer le service apache2 après activation / désactivation.

Reverse Proxy

Un proxy inverse est un serveur frontal c'est-à-dire un serveur exposé sur Internet et par lequel toutes les requêtes passeront. Ce serveur ne traitera pas les requêtes mais se contentera des les rediriger vers d'autres serveurs internes à l'entreprise.

Les intérêts de ce mécanisme sont multiples. Vu qu'il n'y a qu'un seul point d'accès, la sécurité est plus facile à gérer. Cela permet également de mettre en œuvre du « load balancing » entre des serveurs internes. C'est également un moyen simple de rendre disponible un serveur interne sur le Web (pas besoin de configuration réseau).

Pour mettre en place un reverse proxy, il faut activer le module apache « proxy_http » et « proxy ».

a2enmod proxy proxy_http

Ensuite dans le fichier VirtualHost :

<VirtualHost *:80

ServerName siteReverseProxy

ServerAdmin webmaster@localhost

ProxyPass / http://www.example.com/

ProxyPassReverse / http://www.example.com/

ErrorLog ${APACHE_LOG_DIR}/siteReverse_error.log

CustomLog ${APACHE_LOG_DIR}/siteReverse_access.log combined

</VirtualHost>

Sécurité

Un serveur Web doit être sécurisé en particulier les échanges entre le client et le serveur doivent être cryptés. Ceci se fait aisément grâce au paquet openssl. Le port par défaut pour les communications https est le 443.

Installation

apt-get install openssl

a2enmod ssl

systemctl restart apache2

Création d'un certificat auto-signé

La commande openssl permet de créer un certificat ainsi qu'un clé associée à ce certificat.

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/apache2/ssl/apache.key -out /etc/apache2/ssl/apache.crt

Ici la clé et le certificat seront déposés dans le répertoire /etc/apache2/ssl créé au préalable.

Le VirtualHost sera modifié de la sorte :

<VirtualHost *:443>

ServerName monsite.be

ServerAdmin webmaster@localhost

```
        DocumentRoot /var/www/htdocs/monsite
        ErrorLog ${APACHE_LOG_DIR}/monsite_error.log
        CustomLog ${APACHE_LOG_DIR}/monsite_access.log combined
        SSLEngine on
        SSLCertificateFile /etc/apache2/server.crt
        SSLCertificateKeyFile /etc/apache2/server.key
        <Directory /var/www/htdocs/monsite>
                Require all granted
                AllowOverride All
        </Directory>
</VirtualHost>
```

Let's Encrypt

Let's encrypt est une autorité de certification libre, gratuite et automatisée. Ceci permet d'obtenir un certificat valide pour son site Web sans trop d'effort. Cependant, la machine servant le site Web doit être « publiquement » accessible ainsi que le nom du domaine. Cela veut dire qu'en test ce procédé n'est pas applicable.

https://letsencrypt.org/

MySQL

MySQL

Il faudra donner un mot de passe sûr au compte root de MySQL.

apt-get install mysql-server mysql-client

Injecter un fichier SQL :

mysql -u utilisateur -p base_exportee < base_exportee.sql

PhpMyAdmin

Installation

apt-get install phpmyadmin

Pour tester l'installation : http://adresseip/phpmyadmin

Planification de tâches

Un administrateur système a régulièrement besoin d'outils pour exécuter des tâches récurrentes (mises à jour quotidiennes, backups quotidiens, ...). Pour faciliter ceci différents outils exite sous linux :

Cron

Cron est un démon capable d'exécuter des tâches planifiées et récurrentes. Chaque utilisateur possède une «crontab » c'est-à-dire une table reprenant les commandes que l'utilisateur souhaite exécuter et à quel moment.

Utilisation

Pour éditer sa propre crontab :

crontab -e

Le format de la crontab est le suivant :

#Format

#min heure jourDuMois mois jourSemaine commande

* peut être utilisé à la place d'une valeur pour indiquer qu'il s'agit de toutes les valeurs.

Des raccourcis fréquemment utilisés existent pour les 5 premières colonnes :

1. @yearly

2. @monthly

3. @weekly

4. @daily

5. @hourly

6.@reboot

La sortie standard et d'erreur de la commande utilisée dans la crontab peut être redirigée et envoyée vers un fichier de log.

@reboot apt-get update >> /var/log/update.log

Il est à noter que si la machine n'est pas allumée au moment où la tâche a été planifiée, celle-ci ne sera jamais exécutée.

AT

La commande at permet l'exécution d'une commande à un moment ultérieur. Celle-ci sera exécutée dès que l'horloge atteindra l'heure donnée. Si la machine

n’est pas en ligne à ce moment , la commande sera réalisée dès que celle-ci sera en ligne.

at 10:00 2017-12-31 apt-get update

NFS

NFS est un système de fichiers en réseau permettant de partager des données. Il permet aux clients d’avoir accès à un réseau.

Installation

Du côté serveur

Avant de pouvoir commencer à partager des fichiers il faut installer le module nous permettant de partager les dossiers aux clients.

apt-get install nfs-kernel-server nfs-common

nfs-kernel-server permet la gestion du serveur tandis que nfs common prend en charge les données partagées entre la machine serveur et la machine client.

Il faudra également penser à recharger le service.

Après avoir défini vos partages dans le fichier /etc/exports il suffit de relancer le service nfs:

sudo service nfs-kernel-server restart

Cette commande ne coupe pas les transferts en cours si la nouvelle configuration permet toujours leur accès au serveur. Vous pouvez donc la lancer plus ou moins à n'importe quel moment.

Pour vérifier que l'export a bien eu lieu, taper sur le serveur NFS la commande :

showmount -e

Du côté client

Du coté client il faut installer le module nfs-common et portmap.

apt-get install nfs-common portmap.

Configuration

Du côté serveur

Les 3 fichiers de configuration à modifier du côté serveur permettant le partage de fichiers sont /etc/exports, /etc/hosts.allow, /etc/hosts.deny. On pourrait ne modifier que /etc/exports pour partager des fichiers mais cela rendrait le serveur très fragile aux attaques.

/etc/hosts.allow et /etc/hosts.deny spécifient qui a accès au serveur et qui ne l'a pas.

/etc/exports est composé de ligne d'entrée indiquant quel dossier est partagé et à quel machine.

/Dossier/APartager/ ip_master(rw,sync) ip_trusty(rw,sync)

Master et trusty correspondant aux machines à laquelle le dossier est partagé, pour plus de précision il est conseillé d'utiliser les IP des machines. Rw et no_root_squash sont des options qui décrivent l'accès qu'a à l'utilisateur au fichier, les plus importantes sont :

-ro : read only.

-rw : read write, l'utilisateur peut modifier le fichier.

-sync : l'accès au partage nfs se fasse de manière synchrone.

Pour rendre les lignes de configurations effectives il faut redémarrer le nfs kernel server avec la commande :

nfs-kernel-server restart

Du côté client

Il y a 2 moyens de monter les fichiers chez le client on peut le faire manuellement avec mount, ou installer un module qui le fait automatiquement (autofs).

Version Manuelle

Disons que nous avons un dossier home disponible sur le serveur nfs et qu'on voudrait le monter chez un client. On rentrera dans le terminal une ligne de code de ce genre :

mount -t nfs -o ro,soft,intr adresse_ip_serveur:/home client/ usrXX/nfs_public

Le dossier nfs_public doit être créé préalablement

-t signifie qu'on va spécifier le type de filesystem

-intr : Le processus de montage

-soft : va tenter de se connecter pendant un temps défini et en cas d'échec renvoie une erreur

-hard: va tenter de se connecter et ne s'arrêtera pas tant qu'il n'aura pas réussi.

On peut démonter le dossier partagé chez le client en utilisant la commande umount.

Version automatique (autofs)

AutoFs est un programme qui monte automatiquement les dossiers,il resout les problemes du montage manuel ou mount_fstab tel que le fait que les dossiers restent montés et prennent de la ressource même lorsqu'ils ne sont pas utilisé ou encore que sans wifi l'auto-montage ne s'exécute pas.

apt-get install autofs

Fichiers à configurer :

1. /etc/auto.master auto.master est le fichier dans lequel il faut déclarer le répertoire parent de montage et le type de système de fichier Une ligne de configuration doit être écrite sous ce format :

/<point_de_montage_parent> /etc/auto.<type> --ghost,--timeout=30

point_de_montage_parent est le dossier dans lequel on va monter les fichiers reçus du serveur nfs.

type : le type de système de fichier qu'il s'agit.

--ghost : créer des dossiers vides pour chaque point de montage

--timeout: indique le temps d'attente avant de démonter les fichiers

exemple: /net /etc/auto.nfs –ghost,--timeout=30

Solution au problème posé

Le problème posé par le professeur était de partager à plusieurs utilisateurs un dossier public en read-only et que chaque utilisateur connecté au serveur ait son dossier privé seulement accessible à l'utilisateur lui- même.

Du côté client

Nous avons créer des utilisateurs sur la machine client avec adduser et chaque utilisateur créé à un UID unique qui va permettre de le reconnaitre sur la machine serveur.

On crée des dossiers pour chaque clients sur la machine et on mount pour chaque client son dossier privé.

Du

```
 GNU nano 2.2.6          Fichier : script_montage

#!/bin/bash

if [ -z $1 ]
then
        echo "Usage : $0 [nbr_clients]"
        exit 1
fi

if [ ! -e /nfs ]
then
        mkdir /nfs
fi

for i in `seq 1 $1`
do
        if [ ! -e /nfs/priv_client$i ]
        then
                mkdir /nfs/priv_client$i
        fi
```

```
        mount -t nfs -o rw,soft,intr "192.168.0.1:/home/client$i" "/nfs/priv_cl$

done

if [ ! -e /nfs/public_nfs ]
then
        mkdir /nfs/public_nfs
fi
mount -t nfs -o ro,soft,intr 192.168.0.1:/home/public_nfs /nfs/public_nfs
```

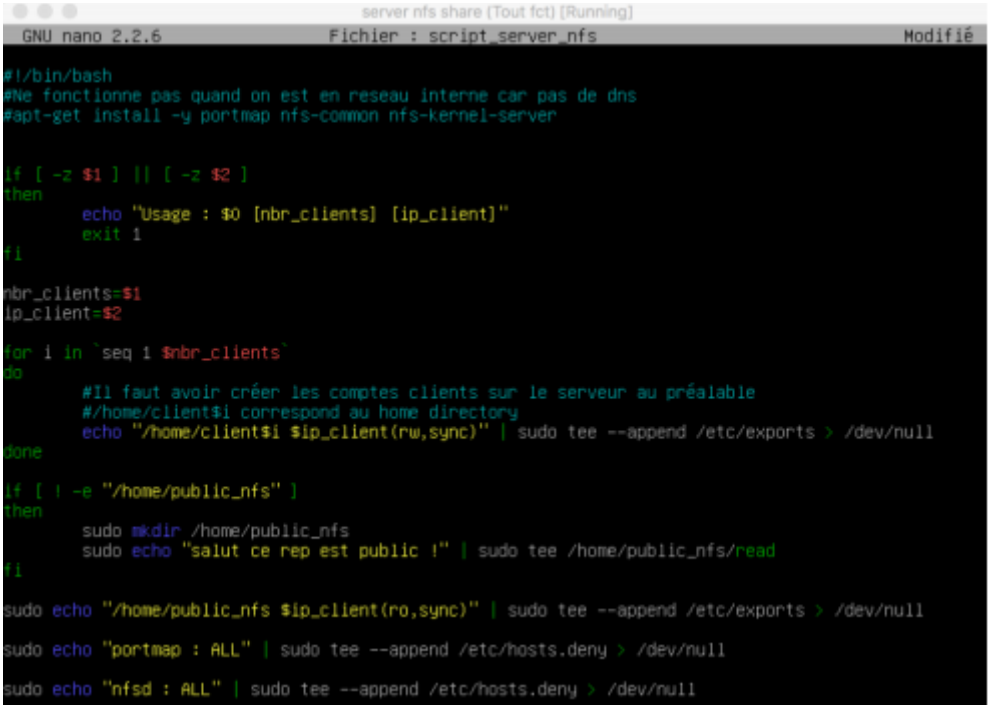
Côté serveur

-Nous avons aussi créer des utilisateurs qui ont un UID correspondant à ceux créés du côté client.

Nous avons également un script qui crée un dossier public et qui mets un fichier dedans, on ajoute ensuite une ligne de configuration dans /etc/exports qui partage le dossier en read only à l'IP de la machine client.

- Ensuite le script ajoutera une ligne de configuration, pour chaque utilisateur de la machine client, dans /etc/exports qui partagera le dossier de l'utilisateur créé sur le serveur qui a un UID correspondant à celui sur le client en rw.

-Et pour finir nous rajoutons dans /etc/hosts.deny et /etc hosts.allow des lignes de configurations qui vont donner l'accès à seulement l'ip de la machine cliente




```
sudo echo "mountd : ALL" | sudo tee --append /etc/hosts.deny > /dev/null

sudo echo "portmap : $ip_client" | sudo tee --append /etc/hosts.allow > /dev/null
sudo echo "nfsd : $ip_client" | sudo tee --append /etc/hosts.allow > /dev/null
sudo echo "mountd : $ip_client" | sudo tee --append /etc/hosts.allow > /dev/null

sudo /etc/init.d/nfs-kernel-server restart
```

NFS vs SAMBA

Comme dit plus haut, nfs est surtout utilisé dans le monde Unix et permet le partage de dossier. Samba lui est utilisé afin d'interconnecter le monde Windows au monde Unix.

La différence est que nfs ne nécessite pas l'utilisation d'un nom d'utilisateur et d'un mot de passe mais bien l'utilisation d'une identification d'hôte.

Nfs devient dès lors très complexe si on doit bien séparer les dossiers par rapport au home/utilisateurs et l'utilisation de Samba devient plus efficace.

Par contre s'il s'agit de logiciels, CD-ROM ou encore des mises à jour, il est plus simple d'utiliser NFS qui reste très simple d'utilisation dans ces cas ci.

SAMBA

Prérequis

- Un serveur Linux Debian (nous avons utilisé la version 8)

- Un serveur Windows Server, sur lequel sera installé un Windows Active Directory (nous avons utilisé la version de Windows Serveur 2016)

- Un client Windows (Nous avons utilisé une version de Windows 10)

Installation

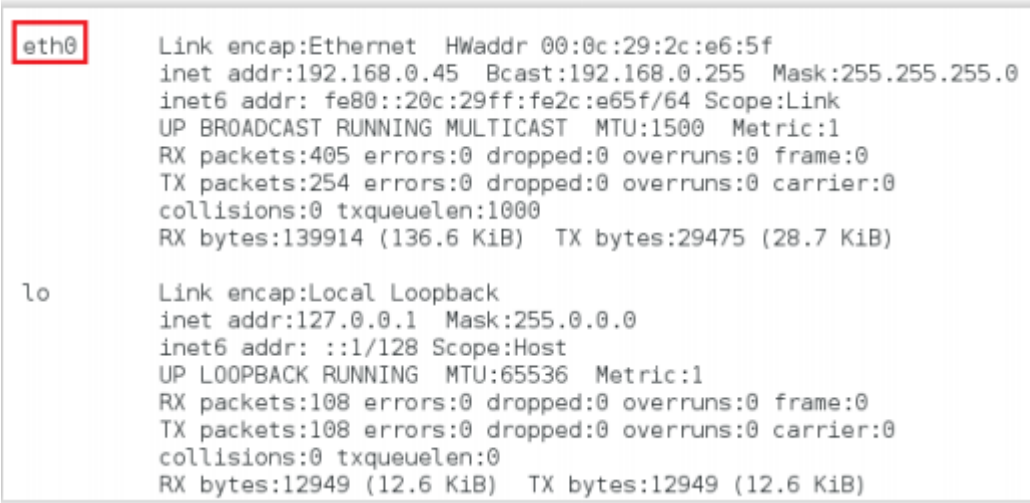
1. Installer le serveur Windows Server 2016 ainsi que le serveur Active Directory et DNS.

2. Installer le Windows 10 et l'intégrer au domaine que l'on vient de créer.

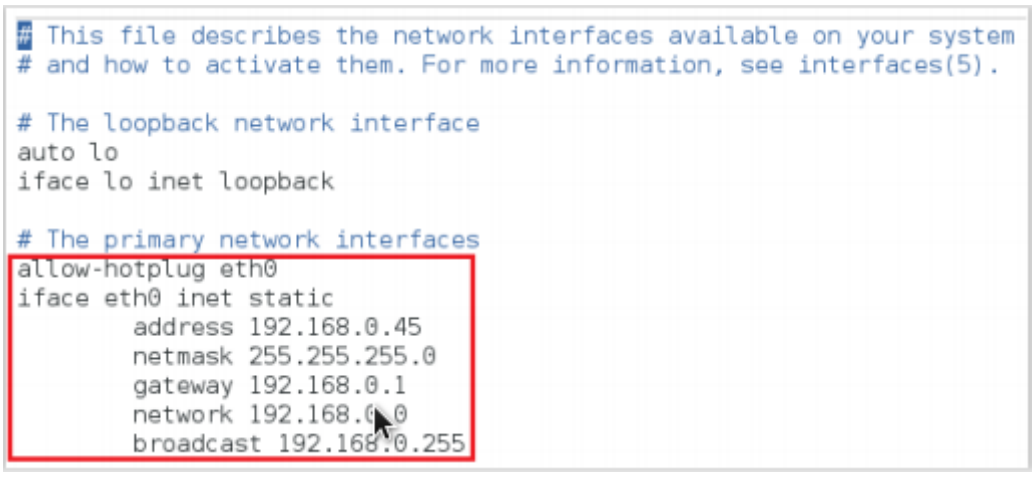
3. Installer le serveur Debian.

4. Configurer le serveur en IP statique

a. On vérifie l'état de notre interface réseau : ifconfig



b. On va modifier la configuration de la carte eht0 qui nous intéresse ici dans le fichier /etc/network/interfaces afin de définir notre adresse IP statique. (vi /etc/network/interfaces)



c. On recharge la configuration de notre interface eth0.

Ifdown eth0

Iquery (Vérification de nos paramètres modifiés)

Ifup eth0

d. Vérifiez vos paramètres définis plus tôt

Ifconfig

5. Il faut ensuite ajouter dans le fichier de résolution de nom notre serveur DNS (le serveur Windows Server 2016 sur lequel on a installé l'AD)

a. On modifie pour cela le fichier /etc/hosts. (vi /etc/hosts)

```
127.0.0.1       localhost
127.0.1.1       debian.samba.royaume    debian
192.168.0.46    samba.royaume
192.168.0.46    VWS001.samba.royaume    VWS001
```

1. LE FQDN (Le Fully Qualified Domain Name) de notre server Debian
2. Le nom de notre domaine, ici samba.royaume, associé à l'adresse IP de notre serveur DNS
3. Le nom de domaine de notre serveur d'AD et DNS.

b. Pour tester la configuration il suffit de Ping vers le nom de domaine.

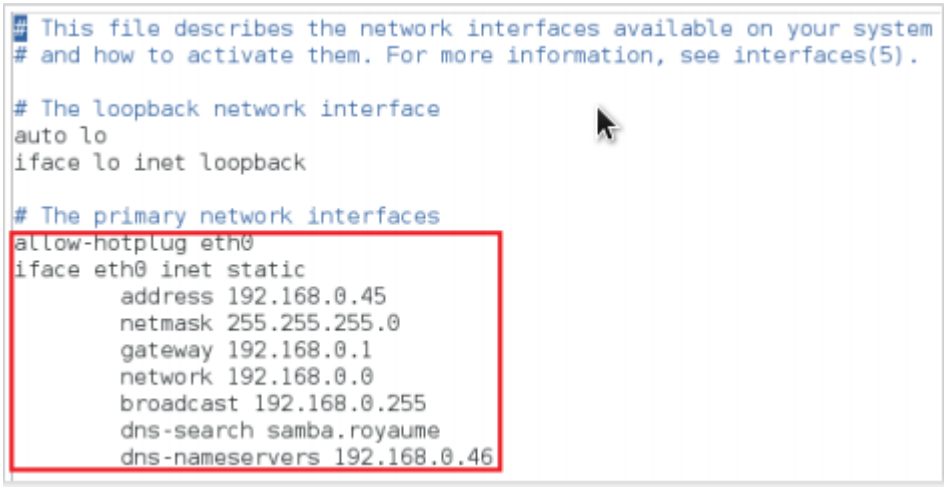
6. Entrer toutes les machines du domaine dans le fichier /etc/hosts est impossible, c'est pourquoi nous allons indiquer le serveur DNS de notre domaine.

a. Il faut installer le package resolvconf.

apt-get install resolvconf

b. Une fois installé, on retourne dans notre fichier de configuration réseau /etc/network/interfaces.

vi /etc/network/interfaces



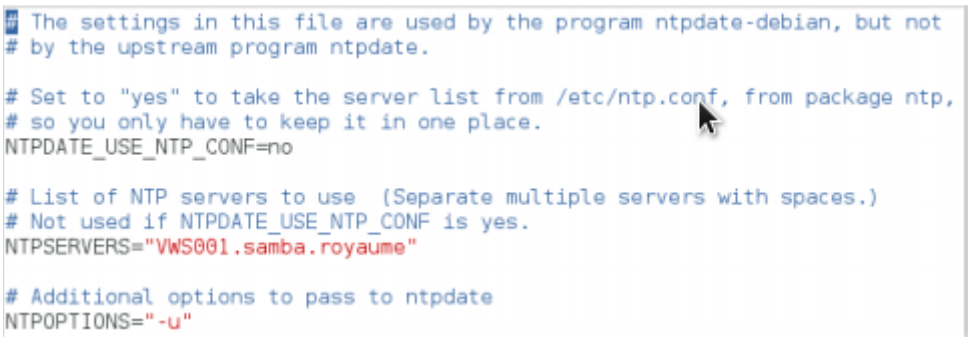
7. Il est très important que notre serveur Debian soit à la même heure que notre serveur Wiindows Server 2016, en effet l'AD est très pointilleux sur l'heure système.

a. Il faut installer le package NTP (Network Time Protocol)

apt-get install ntp

b. Il faut à présent configurer ntpdate en éditant le fichier /etc/default/ntpdate.

vi /etc/default/ntpdate



c. On teste la synchronisation

/usr/sbin/ntpdate-debian

d. Pour éviter les dérives du temps, et donc une désynchronisation on va lancer des tâches planifiées qui vont lancer ntpdate à une certaine heure et aussi au démarrage de la machine.

Crontab -e

```
# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h  dom mon dow   command
# SYNCHRO NTP
18 1 * * * /usr/sbin/ntpdate-debian #synchronise l'horloge à 18h01 tous les jours
@reboot /usr/sbin/ntpdate-debian #lance la syncrhonisation au redémarrage
```

8. On va

1. *18 1 * * * /usr/sbin/ntpdate-debian #synchronise l'horloge à 18h01 tous les jours*
2. *@reboot /usr/sbin/ntpdate-debian #lance la synchronisation au redémarrage*

maintenant intégrer notre serveur Debian au domaine.

a. Pour cela nous devons installer un package, krb5-user, qui va nous permettre d'utiliser le protocole Kerberos. Ce protocole est un protocole d'authentification réseau qui va permettre de s'authentifier sur le Windows Server 2016.

apt-get install krb5-user

b. Passons à la configuration de Kerberos pour pouvoir se connecter.

vi /etc/krb5.conf

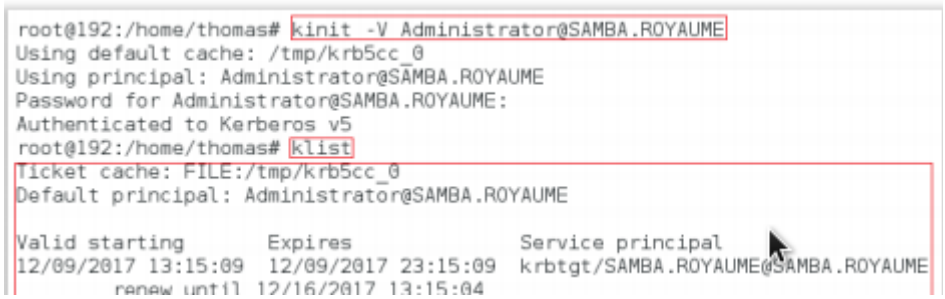
```
[libdefaults]
        ticket_lifetime = 24h
        renew_lifetime = 7d
        default_realm = SAMBA.ROYAUME
        forwardable = true
        proxiable = true
        dns_fallback = no
        dns_lookup_realm = false
        dns_lookup_kdc = false

[realms]
        SAMBA.ROYAUME = {
                kdc = VWS001.samba.royaume
                admin_server = VWS001.samba.royaume
        }

[domain_realm]
        .samba.royaume = SAMBA.ROYAUME
        samba.royaume = SAMBA.ROYAUME
```

c. Testons notre configuration

kinit -V



d. Pour continuer il vaut mieux détruire le ticket pour éviter les résidus.

Kdestroy

9. Passons maintenant à la jonction au domaine avec SAMBA et WINBIND.

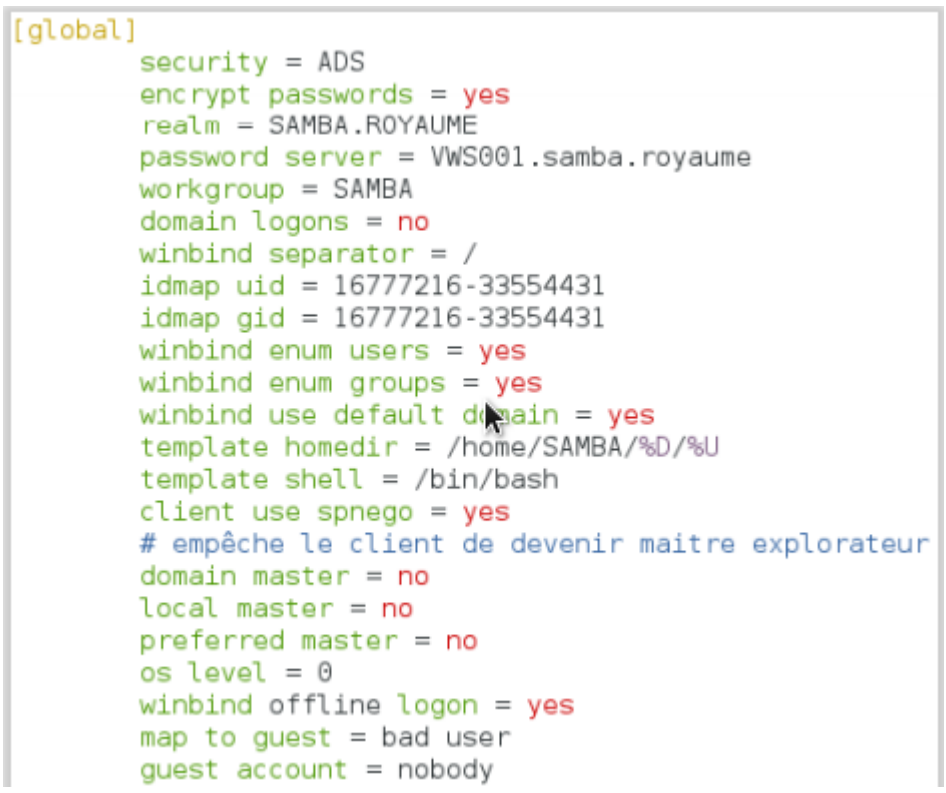
WINBIND : Permet de se loguer sur la machine Linux avec des identifiants Windows.

a. Il faut installer ces deux paquets :

apt-get install samba winbind

b. Configurons Samba grâce au fichier de configuration /etc/samba/smb.conf :

vi /etc/samba/smb.conf



c. On teste nos paramètres :

testparm

```
root@192:/home/thomas# testparm
Load smb config files from /etc/samba/smb.conf
WARNING: The "idmap uid" option is deprecated
WARNING: The "idmap gid" option is deprecated
Processing section "[partage]"
Processing section "[public]"
Loaded services file OK.
WARNING: The setting 'security=ads' should NOT be combined with the 'password server' parameter.
(by default Samba will discover the correct DC to contact automatically).

Server role: ROLE_DOMAIN_MEMBER
```

d. On redémarre Samba pour recharger la configuration :

service smb restart

e. On se connecte à l'Active Directory depuis notre serveur Debian

net join ads -U <utilisateur autorisé> -S <FQDN du contrôleur de domaine>

```
root@debian:/home/thomas# net join ads -U Administrator -S VWS001.samba.royaume
Enter Administrator's password:
Using short domain name -- SAMBA
Joined 'DEBIAN' to dns domain 'samba.royaume'
```

f. On redémarre winbind pour mettre à jour les sources d'authentifications

service winbind restart

g. On teste si on arrive à récupérer les Users/Groups de l'AD

wbinfo -u

```
root@debian:/home/thomas# wbinfo -u
administrator
guest
defaultaccount
admin
krbtgt
student1
student2
student3
student4
student5
```

wbinfo -g

```
root@debian:/home/thomas# wbinfo -g
domain computers
domain controllers
schema admins
enterprise admins
cert publishers
domain admins
domain users
domain guests
group policy creator owners
ras and ias servers
allowed rodc password replication group
denied rodc password replication group
read-only domain controllers
enterprise read-only domain controllers
cloneable domain controllers
protected users
key admins
enterprise key admins
dnsadmins
dnsupdateproxy
```

10. On configure l'authentification de compte Windows sur Linux.

    a. On modifie le fichier /etc/nsswitch.conf.

        vi /etc /nsswitch.conf

```
# /etc/nsswitch.conf
passwd:         compat winbind
group:          compat winbind
shadow:         compat
gshadow:        files

hosts:          files myhostname mdns4_minimal [NOTFOUND=return] dns
networks:       files

protocols:      db files
services:       db files
ethers:         db files
rpc:            db files

netgroup:       nis
```

    b. On teste la prise en compte des modifications

        getent passwd

```
root@debian:/home/thomas# getent passwd
root:x:0:0:root:/root:/bin/bash
admin:*:16777219:16777216:Admin:/home/SAMBA/SAMBA/admin:/bin/bash
krbtgt:*:16777220:16777216:krbtgt:/home/SAMBA/SAMBA/krbtgt:/bin/bash
student1:*:16777221:16777216:student1:/home/SAMBA/SAMBA/student1:/bin/bash
student2:*:16777222:16777216:student2:/home/SAMBA/SAMBA/student2:/bin/bash
student3:*:16777223:16777216:student3:/home/SAMBA/SAMBA/student3:/bin/bash
student4:*:16777224:16777216:student4:/home/SAMBA/SAMBA/student4:/bin/bash
student5:*:16777225:16777216:student5:/home/SAMBA/SAMBA/student5:/bin/bash
```

getent group

```
root@debian:/home/thomas# getent group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
domain computers:x:16777218:
domain controllers:x:16777219:
schema admins:x:16777220:
enterprise admins:x:16777221:
cert publishers:x:16777222:
domain admins:x:16777223:
domain users:x:16777216:
domain guests:x:16777217:
group policy creator owners:x:16777224:
ras and ias servers:x:16777225:
allowed rodc password replication group:x:16777226:
denied rodc password replication group:x:16777227:
read-only domain controllers:x:16777228:
enterprise read-only domain controllers:x:16777229:
cloneable domain controllers:x:16777230:
protected users:x:16777231:
key admins:x:16777232:
enterprise key admins:x:16777233:
dnsadmins:x:16777234:
dnsupdateproxy:x:16777235:
```

c. Les utilisateurs et groupes sont bien ajoutés au système

d. Il faut ensuite activer le module PAM winbind dans la configuration de PAM, afin de permettre l'ouverture de session avec un utilisateur du domaine. On modifie le fichier /etc/pam.d/common-account

vi /etc/pam.d/common-account

```
# here are the per-package modules (the "Primary" block)
account [success=2 new_authtok_reqd=done default=ignore]        pam_unix.so
account [success=1 new_authtok_reqd=done default=ignore]        pam_winbind.so
# here's the fallback if no module succeeds
account requisite       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
account required        pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

vi /etc/pam.d/common-auth

```
# here are the per-package modules (the "Primary" block)
auth    [success=2 default=ignore]      pam_unix.so nullok_secure
auth    [success=1 default=ignore]      pam_winbind.so krb5_auth krb5_ccache_type=FILE cached_login try_first_pass
# here's the fallback if no module succeeds
auth    requisite       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required        pam_permit.so
# and here are more per-package modules (the "Additional" block)
# end of pam-auth-update config
```

vi /etc/pam.d/common-session

```
# here are the per-package modules (the "Primary" block)
session [default=1]                     pam_permit.so
# here's the fallback if no module succeeds
session requisite       pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
session required        pam_permit.so
# and here are more per-package modules (the "Additional" block)
session required        pam_unix.so
session optional                        pam_winbind.so
session optional        pam_systemd.so

session required        pam_mkhomedir.so umask=0022 skel=/etc/skel
```

e. On crée le répertoire pour les données locales des utilisateurs du domaine.

mkdir /home/<nom du domaine>

chmod 751 /home/<nom du domaine>

11. Il ne nous reste plus qu'à peaufiner la configuration de Samba pour que les utilisateurs retrouvent leur home directory sur /home/SAMBA/<nom utilisaeur>

a. On modifie le fichier de configuration de samba

vi /etc/samba/smb.conf

```
[home]
        comment = public folder of users
        path = /home/SAMBA/%D/%U
        valid users = @"domain users"
        read only = no
        writable = yes
        browseable = yes
        inherit acls = yes
        force create mode = 0660
        create mask = 700
        directory mask = 700
        access based share enum = yes
```

b. On peut aussi créer un dossier ou tous les utilisateurs peuvent lire/écrire

vi /etc/samba/smb.conf

```
[public]
        comment = Public
        path = /home/public
        public = yes
        guest ok = yes
        read only = no
        browseable = yes
        writable = yes
        printable = no
        create mode = 0777
        directory mode = 0777
```

c. La configuration à faire pour chaque utilisateur dans l’AD