HEH.be Sciences et technologies

Avenue V. Maistriau 8a
B-7000 Mons
+32 (0)65 33 81 54
scitech-mons@heh.be

WWW.HEH.BE

Linux

Rapport : Réalisation d'un serveur Linux

Année Académique 2020-2021
Groupe N°1 Sécurité

Quentin Mettens
Mathieu Bawin
Clément Raulier

WALLONIE-BRUXELLES ENSEIGNEMENT
pôle hainuyer

Table des matières

1. Introduction

Dans le cadre du cours de projet Linux, il nous est demandé de réaliser un serveur linux, devant être constituer de :

- Un serveur SSH.
- Un serveur WEB.
- Un serveur DNS.
- Un serveur FTP.
- Un serveur MariaDB.
- Un serveur SAMBA | NFS.
- Un serveur NTP.

Nous serons aussi amenés à gérer différentes options sur ce serveur :

- Le partitionnement, le raid, ainsi que le LVM.
- La configuration du pare-feu, d'un antivirus.
- La mise en place d'un système de sauvegarde.
- Automatisation de la création de client (scripts).

2. Choix de la distribution

Nous avons décidé de travailler sur la distribution CentOS (version 7) pour réaliser le projet en Linux. Cette distribution présente notamment plusieurs avantages :

- D'une part, la sécurité est plus que jamais importante, CentOS est donc un excellent choix, puisqu'elle se base sur le code source de RHEL, qui possède déjà un très haut niveau de sécurité. Elle prend en charge notamment la fonctionnalité SELinux, qui permet de mettre en place des contrôles pour l'utilisation des ressources et protège donc contre les accès qui ne sont pas autorisés. CentOS est donc généralement préféré par les entreprises et par les grandes organisations.
- De plus, elle bénéficie d'une imposante et active communauté de développeurs, ce qui permettra de trouver et de récupérer facilement de la documentation pour la réalisation du projet. La documentation en ligne de Red Hat est notamment compatible avec la distribution de CentOS et est disponible en anglais mais aussi en français.
- D'autre part, la distribution CentOS est beaucoup plus compliquée à configurer que Debian mais cela se justifie par le fait qu'il y a beaucoup plus de fonctionnalité à configurer (notamment orienté sécurité). Elle possède aussi beaucoup moins de paquets préinstallés, donc cela prendra un peu plus de temps pour la configurer mais au final on a aussi plus de contrôle.

3. Partitionnement du serveur

3.1. Pourquoi créer des partitions ?

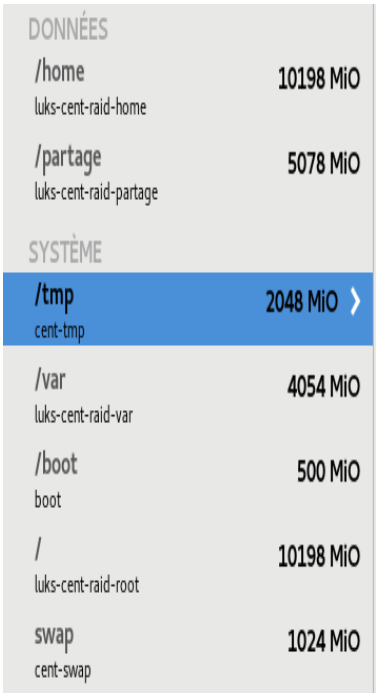
L'intérêt de créer plusieurs partitions sur un même disque dur présente de nombreux avantages :

- Elle permet de séparer clairement la répartition des données sur le disque dur, permettant ainsi de rendre la recherche de fichiers et de documents plus rapide car elles seront confinées à une partie de l'ensemble du disque.
- Elle permet aussi d'améliorer la sécurité, puisque si une des partitions est défaillante, cette défaillance sera contenue dans cette partition et ne se propagera pas vers les autres.

3.2. Schéma de partitionnement

Au niveau du partitionnement, nous avons opté pour :

- Une partition /, /boot et swap
- Une partition /tmp qui permet de limiter la quantité de données temporaires stockées pour éviter qu'elle ne remplisse le système et provoquent des pertes de données ou une interruption des services. Un autre avantage, est de pouvoir monter la partition en tant que nosuid et noexec.
- Une partition /var permet d'éviter que les logs ne remplissent tout le disque dur.
- Une partition /home permet à l'utilisateur de remplir son dossier /home (nous réaliserons des quotas pour chaque utilisateur), sans remplir le disque dur.
- Une partition /partage permettant aux utilisateurs qui y auront accès de pouvoir remplir la partition, sans remplir le disque dur
- Nous avons laissé +/- 8Go d'espace libre, au cas où nous en aurions besoin pour agrandir une des partitions du disque dur

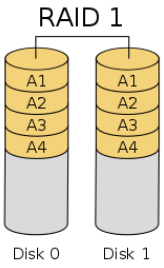


3.3. Configuration du LVM

Nous avons configuré presque l'entièreté des partitions en LVM puisque celle-ci permet d'agrandir ou de réduire l'espace disques des partitions à chaud.

La seule partition que nous n'avons pas configurée en LVM, est la partition /boot, puisque cela peut poser des problèmes pour le démarrage du système. Normalement, ça doit fonctionner correctement depuis GRUB2, mais cela reste encore extrêmement sensible.

3.4. Configuration du RAID



Dans la suite de ce projet, nous utiliserons le RAID 1 qui permet d'avoir de la redondance. Les disques durs */dev/sda* (40Go) et */dev/sdb* (40Go) seront utilisés pour configurer du RAID.

Attention : la technologie RAID n'est pas considérée comme une solution de sauvegarde. Il servira notamment en cas de perte ou de défaillance d'un des deux disques durs. On ne recherchera pas ici, la performance (RAID 0), mais plutôt la sécurité.
Au niveau du RAID, nous utiliserons cette technologie sur les différentes partitions du disque dur SAUF sur la partition swap et /tmp, puisque ce n'est en principe pas utile dans les cas courants.

Attention : Nous n'oublions pas de configurer le RAID sur la partition /boot, puisque si nous ne le faisons pas, et que notre disque principal tombe en panne, la machine ne pourra pas redémarrer.

3.5. Chiffrement

Pour améliorer la sécurité, nous allons chiffrer la partition /home, /partage ainsi que /var et / à l'aide de LUKS (Linux Unified Key Setup), qui est le standard GNU/Linux pour le chiffrement de disques. Nous avons choisi ces partitions, pour la simple et bonne raison que se sont celles qui contiennent les données les plus importantes du système.
Une partition chiffrée est chiffrée via une clé, cette clé est générée lors de la création de la partition, qui est protégée par mot de passe.

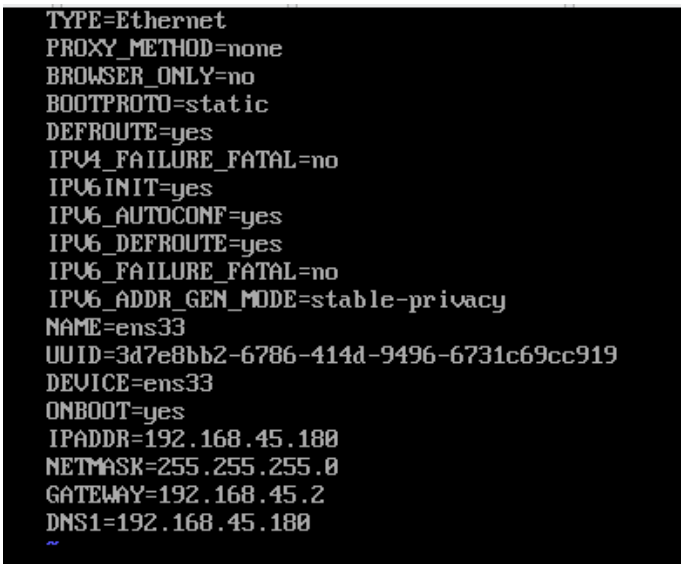
3.6. Option de partionnement

Pour améliorer la sécurité :

- /home sera monté avec les options nodev/nosuid
- /partage sera monté avec les options nodev/noexec
- /var sera monté avec les options nodev/noexec
- /tmp sera monté avec les options nodev/nosuid/noexec
    - **noexec** : n'autorise pas l'exécution de binaires sur le système de fichier.
    - **nosuid** : bloque les opérations sur les bits suid et sgid.
    - **nodev** : indique que le système de fichiers ne peut pas contenir de périphériques spéciaux

4. Configuration du réseau

Nous avons décidé de connecter notre machine virtuelle en NAT au niveau du réseau. Cela permet à la VM de se masquer sur l'hôte physique, tout en profitant d'un accès au réseau local ou est connecté l'hôte physique.
Voici la configuration de la machine virtuelle linux qui se trouve dans le fichier */etc/sysconfig/network-scripts/ifcfg-ens33* :



5. Configuration du SSH

Nous allons ensuite configurer le serveur SSH sur notre machine virtuelle Linux. Pour cela, nous avons accédé au fichier */etc/ssh/sshd_config.*

Nous avons juste changé quelques paramètres qui étaient configurés par défaut :

- Nous avons interdit les utilisateurs de se connecter en root (PermitRootLogin *no*).
- Nous avons diminué le nombre de tentatives d'authentification (MaxAuthTries 3) et le nombre maximal de sessions SSH (MaxSessions 5), afin de limiter l'impact d'une attaque par brute-force.
- Nous n'autorisons que les administrateurs à se connecter en ssh au serveur (*AllowUsers qmettens mathieu clement*).

  Pour permettre à des utilisateurs d'acquérir des droits administrateurs, nous devons les rajouter au groupe « wheel » à l'aide de la commande :

  - *usermod -aG wheel UserName*

- Nous avons configuré une bannière (*/etc/banner*) pour avertir les personnes qui tenteraient de se connecter en SSH
- Nous refusons la connexion par mot de passe (PasswordAuthentication no), nous n'acceptons que les connexions via clé SSH

5.1. Configuration des clés SSH ?

Avant de configurer le serveur SSH, nous allons générer les clés SSH du client via Windows (putty ou powershell) à l'aide de la commande *ssh-keygen* ou via *puttygen.*
Nous avons ensuite enregistré la clé publique générer par le client dans le fichier */home/$USER/.ssh/authorized_keys.*
Le dossier *.ssh* et le fichier *authorized_keys* n'ont pas été créer automatiquement lors de la création de l'utilisateur. C'est pour cela, que nous avons dû les créer et leur attribuer des droits de permissions (*.ssh* - 700 et *authorized_keys* - 600).

5.2. Configuration du service Fail2Ban

Fail2ban est une application qui lit et analyse les logs du service SSH et cherche les tentatives de connexion infructueuses. Les adresses IP trouvées seront ensuite bannies.

Avant de commencer, nous devons installer le paquet fail2ban (*yum install fail2ban-server*) qui se trouve dans le dépôt de paquets EPEL (*yum install epel-release*).
Nous devons activer le service fail2ban (*systemctl enable | start fail2ban.service*).

Ensuite, la bonne pratique consiste à ne pas toucher aux fichiers de configuration principal */etc/fail2ban/jail.conf*. C'est pour cela, que nous allons créer un fichier *sshd.local* dans le répertoire */etc/fail2ban/jail.d/.*

Nous devons activer le jail sur le service SSH pour le protéger (*enabled = true*). À l'aide de la commande *fail2ban-client status sshd*, nous pourrons voir les statistiques concernant le filtre et les actions pour sshd.

```
# /etc/fail2ban/jail.d/sshd.local

[DEFAULT]
#Defini la duree de bannissement
bantime = 86400
#Fail2Ban bannira les clients qui essaiera de se connecter sans succes trois fois, en 10 minutes
findtime = 600
maxretry = 3
#Adresse IP pour lequel Fail2Ban fait exception
ignoreip = 192.168.45.180

[sshd]
enabled = true
port=ssh
~
```

6. Configuration de HTTPD

Lors de ce projet, nous avons décidé d'utiliser le serveur web http Apache car celui-ci est beaucoup plus adapté aux sites web de petite et de moyenne taille où le contenu est généralement dynamique. Tandis que Nginx est notamment plus intéressant pour les projets de grandes tailles ou lorsque le volume de trafic est très important.

Voici les différentes étapes pour configurer le service HTTPD :

1. Installation du service : *yum install httpd*
2. Lancement du service : *systemctl enable | start httpd*
3. Autoriser l'accès au service apache à notre dossier /home/$USER ➔ *chmod 750 /home/$USER* et *chgrp apache /home/$USER*
4. Création du dossier www ainsi que de son fichier index.html :
   - 4.1. *Mkdir /home/$USER/www*
   - *4.2. Touch /home/$USER/www/index.html*
   - 4.3. *Chmod 750 /home/$USER/www | chown $USER:apache /home/$USER/www*
   - 4.4. *Chmod 640 /home/$USER/www/index.html | chown $USER:apache /home/$USER/www/index.html*
5. Création du fichier qui contiendra les mots de passe des utilisateurs :
   - *5.1. Touch /etc/httpd/.htpasswd*
   - *5.2. Chmod 600 /etc/httpd/.htpasswd*
   - *5.3. Chown apache: apache /etc/httpd/.htpasswd*
   - *5.4. htpasswd /etc/httpd/.htpasswd -c $USER*
6. Création du VirtualHost de chaque utilisateur :
   - 6.1. Création du dossier */etc/httpd/conf.d/VirtualHost*. Ce dossier contiendra les fichiers de configuration des utilisateurs sous la forme $USER.conf.
   - 6.2. Chaque fichier de configuration contiendra :

```
<VirtualHost *:80>
# Permet de definir l'URL permettant d'acceder au site web
ServerName www.$1.lan
# Permet de rediriger les users sur le port 443
Redirect permanent / https://www.$1.lan/
</VirtualHost>

<VirtualHost *:443>
DocumentRoot /home/$1/www
ServerName www.$1.lan
ServerAlias $1.lan
# Pemet de definir des droits sur le dossier www
<Directory /home/$1/www/>
# Type d'authentification utilise
AuthType Basic
AllowOverride AuthConfig
AuthName 'Restricted Files'
AuthBasicProvider file
# Fichier contenant les MDP de l'user
AuthUserFile /etc/httpd/.htpasswd
# Permet de definir quel users peut se connecter
Require user $1
Options Indexes FollowSymLinks
</Directory>
SSLEngine on
# Permet de definir quel protocles en seront pas utilisee
SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
# Permet de definir le niveau de chiffrement
SSLCipherSuite HIGH:3DES:!aNULL:!MD5:!SEED:!IDEA
# Pemet de definir le chemni des cles publics et privees
SSLCertificateKeyFile /etc/pki/tls/private/http.key
SSLCertificateFile /etc/pki/tls/certs/http.crt
```

Attention : Pour pouvoir autoriser la connexion sur le port 443, nous devons installer le mod SSL à l'aide de la commande *yum install mod_ssl* ainsi que *yum install openssl* qui permet de générer des certificats auto-signés.

Nous pouvons générer ces clés via la commande *: openssl req -x509 -nodes -days 365 -newkey rsa:1024 -out /etc/pki/tls/certs/http.crt -keyout /etc/pki/tls/private/http.key*.

La clé publique sera stockée dans le fichier */etc/pki/tls/certs/http.crt*, tandis que la clé privée sera stockée dans le fichier */etc/pki/tls/private/http.key.* Nous leur attribuons la permission suivante 600.

6.3. Nous avons ensuite rajouté quelque option dans le fichier */etc/httpd/conf/httpd.conf* :

- Nous devons inclure au fichier de configuration les différents fichiers des utilisateurs en ajoutant la ligne : *IncludeOptional conf.d/VirtualHost/*.conf*
- Pour améliorer la sécurité, nous rajoutons les lignes suivantes :

```
#Permet de changer l'en-tete en production,"apache"
ServerTokens Prod
# Supprime les informations de version de la page generee par apache
ServerSignature Off
# Permet d'empecher aux attaquants d'obtenir le numero d'inode via l'en-tete Etag
FileEtag none
IncludeOptional conf.d/VirtualHost/*.conf

#------------------------
```

7. Configuration du DNS

Un serveur de cache DNS permet de garder en cache les résolutions DNS des requêtes effectuées par des clients afin de répondre plus vite à des requêtes identiques.
Nous allons tout d'abord installer le service DNS ainsi que diverse commande comme *dig*, host ou encore *nslookup* :

- *yum install bind bind-utils*

Ensuite, nous devons attribuer les bonnes permissions au fichier de configuration et au fichier de zone :

- *chown root :named /etc/named.conf /var/named/zone.lan*
- *chmod 0640 /etc/named.conf /var/named/zone.lan*

Nous pouvons ensuite démarrer et lancer le service :

- *systemctl enable named*
- *systemctl start named*

Nous allons ensuite configurer le fichier */etc/named.conf* :

```
options {
        listen-on port 53 { 127.0.0.1; 192.168.45.180; };
        directory       "/var/named";

        #Permet de definir les hotes qui peuvent interroger le serveur
        allow-query     { localhost; 192.168.45.0/24; };

        # Permet d'eviter les attaques DDoS, limite l'acces aux services DNS recursifs
        allow-query-cache       {127.0.0.1; 192.168.45.0/24;};

        # Permet d'empecher le transfert de zone
        allow-transfer  { none; };

        recursion yes;

        dnssec-enable yes;
        dnssec-validation yes;

        # Pour faire office de cache DNS et pouvoir acceder a internet
        forwarders { 208.67.222.222;208.67.220.220;};

        # Permet de changer la version du serveur Apache en "SECRET"
        version "SECRET";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
        type hint;
        file "named.ca";
};
```

```
zone "lan" IN {
        type master;
        file "zone.lan";
        # Indique qui a le droit d'update leur information de maniere dynamique
        allow-update    {none; };
};
```

Ensuite, nous devons configurer le fichier */var/named/zone.lan* :

```
$TTL 1D
@       IN SOA  lan.    root.lan. (
                                        0       ; serial
                                        1D      ; refresh
                                        1H      ; retry
                                        1W      ; expire
                                        3H )    ; minimum

# NS correspond a nos serveur dns
# CNAME correspond a CANONIAL NAME qui permet d'appeler nos site web de plusieurs maniere
@       IN      NS      dns.lan.
dns     IN      A       192.168.45.180
@       IN      A       192.168.45.185
client  IN      A       192.168.45.185
www.qmettens    IN      CNAME   dns
www.clement     IN      CNAME   dns
www.mathieu     IN      CNAME   dns
~
~
```

Chaque fois qu'un utilisateur voudra rendre son site web disponible, il faudra qu'il rajoute l'URL dans le fichier de zone.

Si nous voulons vérifier la configuration du DNS, nous pouvons utiliser la commande *nslookup*

```
[root@dns vsftpd]# nslookup www.qmettens.lan
Server:         192.168.45.180
Address:        192.168.45.180#53

www.qmettens.lan        canonical name = dns.lan.
Name:   dns.lan
Address: 192.168.45.180

[root@dns vsftpd]# nslookup www.google.com
Server:         192.168.45.180
Address:        192.168.45.180#53

Non-authoritative answer:
Name:   www.google.com
Address: 172.217.17.132
Name:   www.google.com
Address: 2a00:1450:400e:803::2004
```

8. Configuration de Mariadb

MariaDB est un système de gestion de base de données relationnelle qui constitue une solution de remplacement des bases de données MySQL.
Nous allons d'abord installer le service : *yum install mariadb-server mariadb* pour pouvoir ensuite activer et démarrer le *service systemctl enable | start mariadb.*
Lors d'une nouvelle installation, nous pouvons utiliser la commande mysql_secure_installation, qui permet d'assurer un minimum de sécurité à notre utilitaire :

- Définit un mot de passe root MySQL
- Supprime les comptes root MySQL accessible de l'extérieur
- Supprime les connexions anonymes
- Supprime la BD de test

Après s'être connecté à notre compte root, nous allons pour chaque utilisateur :

- *Créer sa database : CREATE DATABASE $USERdb;*
- *Accorder tous les privilèges à $User : GRANT ALL PRIVILEGES ON $USERdb. *TO $USER@localhost IDENTIFIED BY $password ;*
- *FLUSH PRIVILEGES;*

9. Configuration du VSFTPD

Pour configurer FTP, nous avons décidé d'utiliser VSFTPD au lieu de proFTPD. Le problème avec proFTPD, c'est que sa configuration est beaucoup plus lourde et très mal documentée, tandis que VSFTPD est très sécurisé mais aussi très léger.
Pour commencer, nous allons installer le service vsftpd :

- *yum install vsftpd*

Pour ensuite le démarrer :

- *systemclt enable | start vsftpd*

Nous allons configurer le fichier principal à savoir *vsftpd.conf*, nous allons complètement le réécrire car la plupart des lignes de configuration présentent dans le fichier ne sont pas forcément utile pour nous.

Voici le fichier de configuration principal :

```
#Interdire les connexions anonymes au serveur
anonymous_enable=NO

#Autoriser les users locaux
local_enable=YES
write_enable=YES
guest_enable=YES
guest_username=vsftpd
allow_writeable_chroot=YES

#Logs
xferlog_enable=YES
xferlog_file=/var/log/vsftpd.xfer.log
xferlog_std_format=YES
log_ftp_protocol=YES
vsftpd_log_file=/var/log/vsftpd.log

#Authentification
pam_service_name=vsftpd
userlist_enable=YES
userlist_deny=NO

#Permet de definir les users autoriser
userlist_file=/etc/vsftpd/vsftpd_user_list

# Permet de definir les droits specifiques pour chaque users
user_config_dir=/etc/vsftpd/vsftpd_user_conf

listen=YES
listen_port=3060
pasv_enable=YES
pasv_min_port=3650
pasv_max_port=3660
listen_ipv6=NO
tcp_wrappers=NO
connect_from_port_20=NO
```

```
ftpd_banner=Bienvenue sur le serveur FTP
max_per_ip=5
max_clients=20

#Configuration SSL

ssl_enable=YES
allow_anon_ssl=NO

force_local_data_ssl=YES
force_local_logins_ssl=YES

ssl_sslv2=NO
ssl_sslv3=NO
ssl_tlsv1=YES

ssl_ciphers=HIGH

rsa_cert_file=/etc/pki/tls/certs/vsftp.crt
rsa_private_key_file=/etc/pki/tls/private/vsftp.key
```

Ensuite, nous allons créer le fichier */etc/vsftpd/vsftpd_user_list* qui contiendra les utilisateurs qui seront autorisés à se connecter avec vsftpd.

Nous allons créer la base de données de vsftpd :

```
MariaDB [(none)]> CREATE DATABASE vsftpd
    -> Nous allons attribuer tous les droits de la base de donne a vsftpd
    -> GRANT SELECT ON vsftpd.* TO 'vsftpd'@'localhost' IDENTIFIED BY 'test123'
    -> FLUSH PRIVILEGES_
```

Ensuite, nous allons créer la table '*accounts*' qui contiendra l'*id*, l'*username* ainsi que son MDP :

```
MariaDB [vsftpd]> CREATE TABLE 'accounts' (
    -> 'id' INT NOT NULL AUTO_INCREMENT PRIMARY KEY
    -> 'username' VARCHR(30) NOT NULL
    -> 'pass' VARCHAR(50) NOT NULL
    -> UNIQUE ('usernme')
    -> ) ENGINE = MYISAM
```

Nous devons ensuite installer *pam_mysql*. Malheureusement, celui-ci n'est pas présent nativement dans les dépôts de CentOS. C'est pour cela, que nous allons le télécharger à l'aide de la commande *:*

- *wget http://www.nosuchhost.net/~cheese/fedora/packages/epel-8/x86_64/pam_mysql-0.8.1-0.6.el8.x86_64.rpm.*

Ensuite, nous avons dû l'installer à l'aide de la commande :

- *rpm -ivh pam_mysql-0.8.1-0.6.el8.x86_64.rpm*

Nous avons ensuite créé le fichier PAM qui se trouve dans */etc/pam.d/vsftpd* et qui permet à vsftpd d'utiliser mariadb pour authentifier ses utilisateurs virtuels.

```
auth required pam_mysql.so user=vsftpd passwd=test123 host=localhost db=vsftpd table=accounts userco
lumn=username   passwdcolumn=pass crypt=3
account required pam_mysql.so user=vsftpd passwd=test123  host=localhost db=vsftpd table=accounts us
ercolumn=username  passwdcolumn=pass crypt=3
~
~
```

Nous avons ensuite rajouté des utilisateurs virtuels FTP dans la base de données à l'aide de la commande :

- INSERT INTO accounts (username, pass) VALUES ('ftp-$1' md5('test')) ;

Chaque utilisateur possèdera son propre fichier de configuration dans le dossier */etc/vsftpd/vsftpd_user_conf/ftp-$USER*. Voici la configuration du fichier personnel de chacun des utilisateurs :

```
local_root=/home/$1/www
write_enable=YES
guest_username=$1
anon_world_readable_only=NO
anon_umask=027
anon_mkdir_write_enable=YES
anon_upload_enable=YES
anon_other_write_enable=YES
```

Pour améliorer la sécurité du service VSFTPD, nous allons chiffrer les communications à l'aide de SSL, notamment en commençant par créer le certificat :

- *openssl req -x509 -nodes -days 365 -newkey rsa:1024 -out /etc/pki/tls/certs/vsftp.crt -keyout /etc/pki/tls/private/vsftp.key.*

Nous allons leur attribuer à tous les deux les permissions 600 et ensuite rajouter leur chemin dans le fichier de configuration principal

10. Configuration de Chronyd

Le service NTP permet de synchroniser l'horloge locale des ordinateurs du réseau avec l'horloge du serveur.

10.1. Configuration du serveur

De base, le service chrony est déjà préinstallé, mais pour s'en assurer, nous pouvons faire un simple :

- yum install chrony

Nous pouvons ensuite le démarrer :

- systemctl enable | start chronyd

Pour commencer, nous allons d'abord éditer le fichier */etc/chrony.conf*. Nous allons rajouter les noms de serveurs suivants :

```
server 0.be.pool.ntp.org
server 1.be.pool.ntp.org
server 2.be.pool.ntp.org
server 3.be.pool.ntp.org
```

Ces noms de serveur pointent en faites vers un ensemble aléatoire de serveurs qui change toutes les heures.

Pour améliorer la sécurité du service chrony, nous allons restreindre l'accès au serveur seulement aux utilisateurs du réseau local :

- allow 192.168.45.0/24

```
# Use public servers from the pool.ntp.org project.
# Please consider joining the pool (http://www.pool.ntp.org/join.html).
server 0.be.pool.ntp.org iburst
server 1.be.pool.ntp.org iburst
server 2.be.pool.ntp.org iburst
server 3.be.pool.ntp.org iburst

# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift

# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3

# Enable kernel synchronization of the real-time clock (RTC).
rtcsync

# Enable hardware timestamping on all interfaces that support it.
#hwtimestamp *

# Increase the minimum number of selectable sources required to adjust
# the system clock.
#minsources 2

# Allow NTP client access from local network.
allow 192.168.45.0/24

# Serve time even if not synchronized to a time source.
#local stratum 10

# Specify file containing keys for NTP authentication.
#keyfile /etc/chrony.keys

# Specify directory for log files.
logdir /var/log/chrony
```

Pour nous assurer que notre serveur s'est bien connecté à notre serveur de temps distant, nous allons utiliser la commande :

- chronyc sources

```
[root@dns ~]# chronyc sources
210 Number of sources = 4
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^- host-95-182-219-178.dyna>     2  10   377   480  -1263us[-1263us] +/-   48ms
^* 45.87.77.15                   2  10   377   594   -975us[-1065us] +/-   11ms
^- dns-rec-2-brudie.belnet.>     2  10   377   285  -1996us[-1996us] +/-   43ms
^+ ntp.devrandom.be              2  10   377   469  -1054us[-1054us] +/-   10ms
```

10.2. Configuration du client

Nous devons installer le service chrony, si celui-ci n'est pas déjà disponible. Ensuite, nous devons rajouter dans le fichier de configuration */etc/chrony.conf*, le serveur ntp que nous avons configuré dans le point précédent (*server 192.168.45.180*).

Pour nous assurer que notre client s'est bien connecté à notre serveur, nous pouvons utiliser la commande :

- chronyc sources

```
[qmettens@client /]$ chronyc sources
210 Number of sources = 1
MS Name/IP address         Stratum Poll Reach LastRx Last sample
===============================================================================
^* 192.168.45.180                3   6   177    51    +11us[  +69us] +/-   11ms
[qmettens@client /]$
```

11. Configuration de NFS

11.1. Configuration du serveur

Nous allons d'abord installer le service : *yum install nfs-utils* pour pouvoir ensuite activer et démarrer le *service systemctl enable | start rpcbind | nfs-server.*
Nous allons tout d'abord créer le sous-répertoire *nfs* dans le répertoire principal */partage* et nous lui attribuons ensuite les permissions 1777.
Ensuite, nous déclarons le répertoire à partager via le fichier */etc/exports :*

- */partage/nfs 192.168.45.0/24(rw,sync,root_squash)*

Le partage du dossier nfs ne se fera que pour l'ensemble des machines présentes dans le réseau 192.168.228.0/24. Pour ce partage :

- L'option *rw* permet la lecture ainsi que l'écriture
- L'option *sync* garantit la solidité des données en cas de coupure de courant ou lors d'une interruption anormale du serveur
- L'option *root_squash* permet de spécifier que le root de la machine cliente n'aura pas les droits de root sur le dossier partagé.

Nous allons mettre à disposition le partage NFS à l'aide de la commande : *exportfs -av*

11.2. Configuration du client

Pour monter les différents partages sur le poste client, nous allons utiliser la méthode via autofs qui est moins gourmande en ressources que le montage manuel via fstab.
D'abord, nous installerons le package via la commande : *yum install autofs*
Nous allons d'abord créer le dossier /autofs.
Nous éditons ensuite le fichier */etc/auto.master* afin d'activer autofs, en y ajoutant cette ligne :

```
/autofs /etc/auto.nfs   --timeout=60, --ghost
"
```

- Le premier champ représente le point de montage parent : */autofs.*
- Le deuxième champ correspond aux différents répertoires que contiendra le point de montage parent : */etc/auto.nfs*
- L'option --*timeout* correspond au temps pendant lequel, le montage reste activé après le dernier accès au dossier partagé : --*timeout=60*
- L'option --*ghost* permet de créer automatiquement le répertoire partageNFS, si cette option est omise, aucun répertoire n'est créé et l'utilisateur devra connaître le point de montage pour y accéder

Nous modifions le fichier */etc/auto.nfs* en y ajoutant cette ligne :

```
PartageNFS      -fstype=nfs,rw  192.168.45.180:/partage/nfs
~
```

- Le premier champ correspond au nom du partage
- Le deuxième champ correspond aux autorisations puis le chemin vers le dossier partager distant
- Le troisième champ correspond au chemin vers le dossier partagé distant

12. Configuration de Samba

Samba permet l'échange de fichiers dans un réseau, avec des postes clients tournant sous Windows, Mac OS X ou Linux.
Tout d'abord, nous devons installer le service : *yum install samba samba-client*
Nous devons ensuite démarrer et lancer le service nmb ainsi que smb :

- *systemctl enable |start nmb.service|smb.service*

Le démon serveur *smbd* fournit des services de partage de fichiers et d'impression aux clients Windows. Tandis que le démon serveur *nmbd* comprend et répond à toutes les requêtes de service de nom NetBIOS.

Nous allons tout d'abord commencer par créer le répertoire qui contiendra les fichiers partagés uniquement accessible par mot de passe (/partage/samba), pour l'autre répertoire, les utilisateurs n'auront pas besoin de mot de passe (/partage/nfs).

- *mkdir /partage/samba*
- *chmod 1777 /partage/samba*

Nous allons ensuite renommer le fichier de configuration en tant que */etc/samba/smb.conf.org*, pour pouvoir ensuite créer notre propre fichier de configuration */etc/samba/smb.conf :*

```
[global]
        workgroup = WORKGROUP
        netbios name = LINUX
        netbios aliases = LINUX2
        server string = Serveur de fichiers LINUX
        #Ppermet de creer un fichie de log pour chaque user
        log file = /var/log/samba/log.%m
        # Seuls les messages d'erreur sont journalisés
        log level = 0
        # Limite la taille maximale du fichier
        max log size = 1000
        # Permet de definir les reseaux autorises
        hosts allow = 192.168.45.0/24  192.168.0.0/24
        security = user
        # Imprimante desactivees
        load printers = no
        disable spoolss = yes
        # Permet de definir la gestion de MDP
        passdb backend = tdbsam
        # Desactive la synchronisation des MDP samba avec ceux de LINUX
        unix password sync = no
        # User non autorises
        invalid users = root
        #Chiffrement des MDP
        encrypt passwords = yes
        # User invites seront mapper sur cet utilisateur
        guest account = smbguest
        # Tous les user qui n'ont aucun login/MDP seront logués sur ct utilisateur
        map to guest = bad user
        # Forcer les users a se connecter sur ce groupe
        force group = users

[nfs]
        path = /partage/nfs
        comment = Partage Public
        public = yes
        guest ok = yes
        only guest = yes
        read only = no
        writeable = yes
        create mode = 0660
        force group = users
        directory mode = 0770

[samba]
        path = /partage/samba
        comment = Partage Privee
        public = no
        guest ok = no
        read only = no
        writeable = yes
        invalid users = root smbguest
        create mode = 0640
        directory mode = 0750
```

Pour les utilisateurs qui ne possèdent pas de login/MDP, nous allons leur créer un utilisateur : *smbguest*

```
useradd -g users -d /dev/null -s /sbin/nologin smbguest
```

Cette commande permet d'ajouter un utilisateur :

- Qui appartiendra aux groupes users : *-g users*
- Qui ne possède pas de /home : *-d /dev/null*
- Qui ne possède pas de shell de connexion : *-s /sbin/nologin*

```
passwd -l smbguest
```

Cette commande permet de bloquer la connexion au shell via un mot de passe

```
smbpasswd -a smbguest -d
```

Cette commande rajoute l'utilisateur à la base de données samba, mais il sera désactivé.

```
usermod -G users $1
```

Les utilisateurs qui pourront avoir accès aux différents partages de samba, doivent appartenir au groupe users.

```
smbpasswd -a $1
```

Ensuite, nous devons les rajouter à la base de données, mais cette fois-ci, celui-ci sera activé

13. Sauvegarde

13.1. Ajouter un disque dur

Avant de commencer, nous allons tout d'abord vérifier que le disque a bien été détecté par le système :

- *Dmesg | grep sd[a-z]*

Ici, nous allons rajouter le disque sdc. Nous allons donc créer une nouvelle partition primaire à l'aide de la commande :

- *Fdisk /dev/sdc*
- *n* (création d'une nouvelle partition)
- *p* (création d'une partition primaire)
- *1* (création de la partition numéro 1 è /dev/sdc1)
- *w* (nous allons ensuite créer la nouvelle table de partition)

Nous allons ensuite formater la nouvelle partition à l'aide de la commande :

- *mkfs -t ext4 /dev/sdc1*

Nous devons ensuite créer le point de montage à savoir :

- *mkdir /sauvegarde*

Ensuite, nous devons rajouter cette ligne dans le fichier /etc/fstab :

- */dev/sdc1 /sauvegarde ext4 defaults 1 2*

13.2. Configuration de rsnapshot

Rsnapshot est une solution de sauvegarde efficace écrite en PERL et basée sur Rsync.
Une fois que la première synchronisation des données est faite, les sauvegardes suivantes se feront à l'aide du planificateur de tâche : *crontab*.

Rsnapshot est une solution permettant de faire des sauvegardes quotidiennes complètes.

Dans un premier temps, on va installer rsnapshot fourni par le dépôt de paquets EPEL :

- *yum install rsnapshot*

Ensuite, on va partir d'un fichier de config vide en gardant le fichier d'origine de côté :

- *mv rsnapshot.conf rsnapshot.conf.orig*

On va donc pouvoir éditer rsnapshot.conf comme ceci :

```
# /etc/rsnapshot.conf

#Config file version
config_version  1.2

snapshot_root   /sauvegarde

#External program dependencies
cmd_cp  /usr/bin/cp
cmd_rm  /usr/bin/rm
cmd_rsync       /usr/bin/rsync
cmd_ssh /usr/bin/ssh
cmd_logger      /usr/bin/logger
cmd_du  /usr/bin/du
cmd_rsnapshot_diff      /usr/bin/rsnapshot-diff

#Backup intervals
retain  daily   7
retain  weekly  4
retain  monthly 12
retain  yearly  4

verbose 2

#Fichier de Log
loglevel        3
logfile /var/log/rsnapshot
lockfile        /var/run/rsnapshot.pid

backup  /home/  localhost/
backup  /etc/   localhost/
backup  /var/   localhost/
backup  /usr/   localhost/
backup  /partage/       localhost/
~
```

Une fois le fichier édité, on peut vérifier qu'il n'y a pas d'erreur de syntaxe grâce à la commande suivante :

- *rsnapshot configtest*

Si tout est bon, on obtient ceci :

```
[root@localhost etc]# rsnapshot configtest
Syntax OK
```

Maintenant que le fichier de configuration est bien édité, on peut lancer une première simulation :

- *rsnapshot daily*

Cette première sauvegarde va prendre un peu de temps, les prochaines seront plus rapides étant donné qu'elles se feront de manière incrémentale.

Il ne reste plus qu'à effectuer cette sauvegarde automatiquement afin qu'elle s'effectue tous les jours à 0:20 :

- *crontab -l*

```
20 00 * * *     /usr/bin/rsnapshot daily
30 00 * * 1     /usr/bin/rsnapshot weekly
40 00 1 * *     /usr/bin/rsnapshot monthly
50 00 1 1 *     /usr/bin/rsnapshot yearly
```

On peut vérifier que notre première sauvegarde a bien été effectuée :

- *ls -l /sauvegarde*

```
[root@dns sauvegarde]# ls -l /sauvegarde
total 48
drwxr-xr-x. 3 root root  4096 29 mai 19:30 daily.0
```

Chaque nouvelle sauvegarde portera le nom de daily.x (x étant le numéro de la sauvegarde, la 4e portera le nom de : daily.3).

13.3. Sauvegarde de la base de données

La base de données ne peut pas être sauvegardée par rsnapshot, c'est pour cela, que nous avons créé un script (sqldump.sh) permettant de faire une sauvegarde de notre base de données tous les jours à 12 :40 à l'aide de l'utilitaire *mysqldump* :

```
#!/bin/bash

YEAR=$(date +%Y)
MONTH=$(date +%m)
DAY=$(date +%d)

mkdir -p /sauvegarde/mysql/$DAY-$MONTH-$YEAR
mysqldump -u root -ptest123 --all-databases | gzip -c > /sauvegarde/mysql/$DAY-$MONTH-$YEAR/sqlsave.
sql.gz
~
```

Ce script permettra de sauvegarder l'ensemble des bases de données dans le fichier */sauvegarde/mysql/$DAY-$MONTH-$YEAR/sqlsave.sql.gz.*

- *$DAY* : correspond au jour de la sauvegarde
- *$MONTH* : correspond au jour de la sauvegarde
- *$YEAR* : correspond à l'année de la sauvegarde

Nous allons la compresser puisque l'ensemble des bases de données peuvent être lourde à stocker.

14. Antivirus

Bien que Linux soit connu pour être l'un des systèmes d'exploitation les plus sécurisés, cela ne signifie pas qu'il est impénétrable aux attaques de virus. Une étude a montré que **36%** du total des principaux cas de logiciels malveillants ont été signalés sur des systèmes Linux.

Heureusement, nous avons ClamAV, l'une des meilleures solutions de sécurité Linux pour se débarrasser de nombreuses menaces de logiciels malveillants. C'est fiable, facile à installer et gratuit

Pour commencer, nous allons tout d'abord installer ClamAV :

- *yum install clamav-server clamav-data clamav-update clamav-filesystem clamav clamav-scanner-systemd clamav-devel clamav-lib clamav-server-systemd*

Nous devons décommenter la ligne socket pour activer l'antivirus :

- *LocalSocket /var/run/clamd.scan/clamd.sock*

Par la suite, nous pouvons activer et démarrer le service :

- *systemctl enable | start clamd@scan*

Une fois que cela est fait, la première chose à faire est de mettre à jour la base de données de l'antivirus à l'aide de la commande : *freshclam.*
Pour automatiser ces mises à jour, nous avons décidé de lancer cette commande tous les premiers du mois à midi.

```
00 12 1 * *     /usr/bin/freshclam &
```

Dès lors, nous pouvons dire à l'antivirus d'effectuer un scan régulièrement dans le */home* de chaque utilisateur, mais aussi dans le dossier */partage,* pour éviter tout problème.

Si l'antivirus détecte un virus ou tout autre objet malveillant, celui-ci sera déplacé dans le dossier */quarantaine.*

```
30 12 * * *     clamscan -r --move=/quarantaine /home /partage &
```

15. Firewall

Pour le firewall, nous avons décidé de nous passer de firewalld, qui est préinstallé par défaut sur CentOS 7 et de télécharger à la place iptables.

- systemclt stop firewalld
- systemctl disable firewalld
- yum remove firewalld
- yum install iptables-services
- systemctl enable iptables
- systemctl start iptables

Pour pouvoir configurer efficacement le firewall, nous avons décidé de réaliser un script qui se lancera à l'aide du service (startFirewalld.service) à chaque redémarrage de la machine.

```
[Unit]
Description=Configuration des regles de pare-feu lors du redemarrage

[Service]
Type=oneshot
RemainAfterExit=yes
ExecStart=/usr/local/bin/scriptFirewall.sh

[Install]
WantedBy=multi-user.target
~
```

Le script sera créé dans le dossier /usr/local/bin/scriptFirewall.sh :

```
#! /bin/bash

# Vider les tables
iptables -F

# On bloque l'ensemble du traffic
iptables -P INPUT DROP
iptables -P OUTPUT DROP
iptables -P FORWARD DROP

# Accepter le traffic local
iptables -A INPUT -i lo -j ACCEPT
iptables -A OUTPUT -o lo -j ACCEPT

# Autoriser les connexions déjà établies
iptables -A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A OUTPUT -m state --state RELATED,ESTABLISHED -j ACCEPT

# Autoriser les pings

iptables -A OUTPUT -p icmp -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT

# Autoriser le SSH (I/O)
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
iptables -A INPUT -p tcp --dport 22 -j ACCEPT


# Autoriser le DNS (I/O)

iptables -A OUTPUT -p udp --sport 53 --dport 1024:65535 -j ACCEPT
iptables -A OUTPUT -p udp --sport 1024:65535 --dport 53 -j ACCEPT
iptables -A INPUT -p udp --sport 53 --dport 1024:65535 -j ACCEPT
iptables -A INPUT -p udp --sport 1024:65535 --dport 53 -j ACCEPT

#iptables -A OUTPUT -p udp --sport 53 -j ACCEPT
```

```
#iptables -A OUTPUT -p udp --sport 53 -j ACCEPT
#iptables -A INPUT -p udp --dport 53 -j ACCEPT

# Autoriser HTTP

iptables -A INPUT -p tcp --sport 1024:65535 --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 80 --dport 1024:65535 -j ACCEPT

# Autoriser le HTTPS

iptables -A OUTPUT -p tcp --sport 443 --dport 1024:65535 -j ACCEPT
iptables -A INPUT -p tcp --sport 1024:65535 --dport 443 -j ACCEPT

# Autoriser le NTP
# Serveur
iptables -A INPUT -p udp --dport 123 -j ACCEPT
iptables -A OUTPUT -p udp --sport 123 -j ACCEPT

# Client
iptables -A INPUT -p udp --sport 123 -j ACCEPT
iptables -A OUTPUT -p udp --dport 123 -j ACCEPT

# Autoriser le FTP

iptables -A INPUT -p tcp --dport 3060 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 3060 -j ACCEPT
iptables -A INPUT -p tcp --dport 3650:3660 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 3650:3660 -j ACCEPT
iptables -A INPUT -m state --state ESTABLISHED, RELATED -j ACCEPT

# Autoriser Samba


iptables -A INPUT -p tcp --dport 445 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 445 -j ACCEPT
iptables -A INPUT -p tcp -m multiport --dports 137,138,139 -j ACCEPT
iptables -A OUTPUT -p tcp -m multiport --sports 137,138,139 -j ACCEPT

# Autoriser le NFS


iptables -A INPUT -p udp --dport 111 -j ACCEPT
iptables -A INPUT -p tcp --dport 111 -j ACCEPT
iptables -A OUTPUT -p udp --sport 111 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 111 -j ACCEPT


iptables -A INPUT -p tcp --dport 2049 -j ACCEPT
iptables -A INPUT -p udp --dport 2049 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 2049 -j ACCEPT
iptables -A OUTPUT -p udp --sport 2049 -j ACCEPT

# Autoriser l'AV

# Autoriser MariaDB
iptables -A INPUT -p tcp --dport 3306 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 3306 -j ACCEPT
```

16. Quota

Pour les quotas, nous avons décidé d'en mettre uniquement sur les partitions à risques, c'est-à-dire le */home* et le */partage*.

Nous commençons par modifier le fichier */etc/fstab,* pour rajouter les options *usrquota et grpquota* sur les partitions */home* et */partage.*

Ensuite, nous pouvons soit démonter/remonter les dossiers, ou nous pouvons tout simplement reboot le système.

- *shutdown -r now*

Avant de commencer à créer les quotas, nous devons utiliser les commandes :

- *quotacheck -avug* (Permet de créer les fichiers aquota.group et aquota.user)
- *quotaon -avug* (Activer les quotas)

Pour plus de sécurité, nous allons changer les permissions sur les fichiers *aquota.group* et *aquota.user* :

- *chmod 600 aquota.group aquota.user*

Ensuite, pour que chaque utilisateur ne surcharge son dossier /home, nous allons lui attribuer des quotas de 200 Mo (soft) et de 250 Mo (hard)

- *setquota -u $USER 200000 250000 0 0 /home*

```
Utilisateur     utilisé souple stricte sursis utilisé souple stricte sursis
-------------------------------------------------------------------------
root       --      20       0       0              2     0     0
qmettens   --      32  200000  250000              9     0     0
clement    --      28  200000  250000              8     0     0
mathieu    --      28  200000  250000              8     0     0
```

Ensuite, nous allons limiter le groupe utilisateur (users) sur le dossier /partage de 1Go (soft) et de 1.5Go (hard) :

- *setquota -g users 1000000 1500000 0 0 /partage*

```
Groupe          utilisé souple stricte sursis utilisé souple stricte sursis
-------------------------------------------------------------------------
root       --      28       0       0              4     0     0
users      --      12 1000000 1500000              2     0     0
#1000      --       0       0       0              2     0     0
```

Enfin, nous allons limiter l'utilisateur smbguest sur le dossier /partage de 500Mo (soft) et de 550Mo (hard) :

- *setquota -u smbguest 500000 550000 0 0 /partage*

```
Utilisateur     utilisé souple stricte sursis utilisé souple stricte sursis
-------------------------------------------------------------------------
root       --      28       0       0              4     0     0
smbguest   --       4  500000  550000              3     0     0
qmettens   --       8       0       0              1     0     0
```

Pour vérifier nos configurations, nous allons utiliser la commande :

- *repquota -avug*

17. SELinux

SELinux est une architecture de sécurité développé par la NSA pour Linux et utilisé principalement par les systèmes Red Hat, CentOS et Fedora.

Il améliore de manière significative la sécurité des serveurs sur lesquels il est déployé, en apportant une couche supplémentaire aux traditionnels droits d'accès des fichiers.

17.1. WEB

- setsebool -P httpd_enable_homedirs 1

17.2. FTP

- setsebool -P ftpd_full_access on

17.3. SAMBA

- semanage fcontext -a -t samba_share_t "/partage/samba(/.*)?"
- setsebool -P smbd_anon_write=1
- semanage fcontext -a -t public_content_rw_t "/partage/nfs(/.*)?"
- setsebool -P samba_export_all_rw = 1
- setsebool -P samba_create_home_dirs = 1

17.4. ANTIVIRUS

- setsebool -P antivirus_can_scan_system 1

18. Script

18.1. scriptCreation.sh

```
#! /bin/bash
# Permet de verifier que seul ces utilsateurs peuvent actionner le script
if [ $(whoami) == root ] || [ $(whoami) == qmettens ] || [ $(whoami) == clement ] || [ $(whoami) ==
mathieu ]
then
        id -u $1
        # Si l'utilisateur n'existe pas,
        if [ $? == '1' ]
        then
                useradd $1
                passwd $1
                usermod -a -G users $1
                setquota -u $1 200000 250000 0 0 /home
        fi
        id -u $1
        # Si l'utilisateur existe,
        if [ $? == '0' ]
        then
                boolean=true
                while("$boolean" = true)
                do
                        echo "1. Réalisation du site web de l'utilisateur"
                        echo "2. Réalisation du compte FTP de l'utilisateur"
                        echo "3. Réalisation de la BD de l'utilisateur"
                        echo "4. Accès à Samba"
                        echo "5. All"
                        echo "6. Quitter le programme"
                        read number

                        case $number in

                        1)      # Creation des differents fichiers et dossiers de l'user
                                mkdir /home/$1/www
                                touch /home/$1/www/index.html
                                chmod 750 /home/$1
                                chgrp apache /home/$1
```

```
                        chgrp apache /home/$1
                        chmod 0750 /home/$1/www
                        chown $1:apache /home/$1/www
                        chmod 640 /home/$1/www/index.html
                        chown $1:apache /home/$1/www/index.html
                        echo "Création de votre passwords pour votre site Web"
                        htpasswd /etc/httpd/.htpasswd $1
                        echo "
                                <VirtualHost *:80>
                                # Permet de definir l'URL permettant d'acceder au site web
                                ServerName www.$1.lan
                                # Permet de rediriger les users sur le port 443
                                Redirect permanent / https://www.$1.lan/
                                </VirtualHost>

                                <VirtualHost *:443>
                                DocumentRoot /home/$1/www
                                ServerName www.$1.lan
                                ServerAlias $1.lan
                                # Pemet de definir des droits sur le dossier www
                                <Directory /home/$1/www/>
                                # Type d'authentification utilise
                                AuthType Basic
                                AllowOverride AuthConfig
                                AuthName 'Restricted Files'
                                AuthBasicProvider file
                                # Fichier contenant les MDP de l'user
                                AuthUserFile /etc/httpd/.htpasswd
                                # Permet de definir quel users peut se connecter
                                Require user $1
                                Options Indexes FollowSymLinks
                                </Directory>
                                SSLEngine on
                                # Permet de definir quel protocles en seront pas utilisee
                                SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
                                # Permet de definir le niveau de chiffrement
```

```
                    SSLCipherSuite HIGH:3DES:!aNULL:!MD5:!SEED:!IDEA
                    # Permet de definir le chemni des cles publics et privees
                    SSLCertificateKeyFile /etc/pki/tls/private/http.key
                    SSLCertificateFile /etc/pki/tls/certs/http.crt
                    </VirtualHost> " > /etc/httpd/conf.d/VirtualHost/$1.conf

                    echo " <!DOCTYPE html>
                    <html>
                    <head>
                    <title>Site Web</title>
                    </head>
                    <body>
                    <h1>Bienvenue sur ton site $1</h1>
                    </body>
                    </html> " > /home/$1/www/index.html

                    echo "www.$1    IN      CNAME   dns" >> /var/named/zone.lan
                    systemctl restart named
                    systemctl restart httpd.service
    ;;
    2)
                    echo "Création de votre utilisateur ftp-$1"
                    touch /etc/vsftpd/vsftpd_user_conf/ftp-$1

                    echo "
                            # Dossier racine de l'user
                            local_root=/home/$1/www
                            write_enable=YES
                            # Sur quel utilisateur, il est mapper
                            guest_username=$1
                            anon_world_readable_only=NO
                            anon_umask=027
                            #Autoriser les users a creer des repertoires
                            anon_mkdir_write_enable=YES
                            anon_upload_enable=YES
                            # Autoriser les users a supprimer ou a changer nom
```

```
                                    # Autoriser les users a supprimer ou a changer nom
                                    anon_other_write_enable=YES " > /etc/vsftpd/vsftpd_u
ser_conf/ftp-$1
                                # Autoriser l'user en rajoutant son nom dans le fichier
                                echo "ftp-$1" >> /etc/vsftpd/vsftpd_user_list

                                echo "Veuillez entrer votre password"
                                read password
                                # Rajout de l'username et de son mot de passe dans la BD
                                mysql -u root -ptest123 -e "USE vsftpd;INSERT INTO accounts(
username,pass) VALUES ('ftp-$1',md5('$password'));"
                                echo "Création terminé"
                                systemctl restart vsftpd.service

                        ;;

                        3)              echo "Création de la BD de $1, pour cela, veuillez entrez so
n password"
                                read password
                                echo "Connectez-vous à votre compte root"
                                # Creation de la BD de l'user et des droits de celle-ci
                                mysql -u root -ptest123 -e "CREATE DATABASE $1db; GRANT ALL
PRIVILEGES ON $1db .* TO '$1'@'localhost' IDENTIFIED BY '$password'; FLUSH PRIVILEGES;"
                                echo "Création terminé"
                                systemctl restart mariadb.service
                        ;;
                        4)
                                echo "Veuillez entrer votre mot de passe d'accès au fichier
de partage samba"
                                smbpasswd -a $1
                                systemctl restart smb.service
                                systemctl restart nmb.service
                        ;;

                        5)              mkdir /home/$1/www
                                touch /home/$1/www/index.html
-
```

```
                chmod 750 /home/$1
                chgrp apache /home/$1
                chmod 0750 /home/$1/www
                chown $1:apache /home/$1/www
                chmod 640 /home/$1/www/index.html
                chown $1:apache /home/$1/www/index.html
                echo "Création de votre passwords pour votre site Web"
                htpasswd /etc/httpd/.htpasswd $1
        echo "
                <VirtualHost *:80>
                ServerName www.$1.lan
                Redirect permanent / https://www.$1.lan/
                </VirtualHost>

                <VirtualHost *:443>
                DocumentRoot /home/$1/www
                ServerName www.$1.lan
                ServerAlias $1.lan
                <Directory /home/$1/www/>
                AuthType Basic
                AllowOverride AuthConfig
                AuthName 'Restricted Files'
                AuthBasicProvider file
                AuthUserFile /etc/httpd/.htpasswd
                Require user $1
                Options Indexes FollowSymLinks
                </Directory>
                SSLEngine on
                SSLProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
                SSLCipherSuite HIGH:3DES:!aNULL:!MD5:!SEED:!IDEA
                SSLCertificateKeyFile /etc/pki/tls/private/http.key
                SSLCertificateFile /etc/pki/tls/certs/http.crt
                </VirtualHost> " > /etc/httpd/conf.d/VirtualHost/$1.conf

                echo " <!DOCTYPE html>
                <html>
```

```
                                        <title>Site Web</title>
                                        </head>
                                        <body>
                                        <h1>Bienvenue sur ton site $1</h1>
                                        </body>
                                        </html> " > /home/$1/www/index.html

                                        echo "www.$1    IN      CNAME   dns" >> /var/named/zone.lan
                                        systemctl restart named
                                        systemctl restart httpd.service
                                        echo "Création de votre utilisateur ftp-$1"
                                        touch /etc/vsftpd/vsftpd_user_conf/ftp-$1

                                        echo "
                                                local_root=/home/$1/www
                                                write_enable=YES
                                                guest_username=$1
                                                anon_world_readable_only=NO
                                                anon_umask=027
                                                anon_mkdir_write_enable=YES
                                                anon_upload_enable=YES
                                                anon_other_write_enable=YES " > /etc/vsftpd/vsftpd_u
   ser_conf/ftp-$1

                                        echo "ftp-$1" >> /etc/vsftpd/vsftpd_user_list

                                        echo "Veuillez entrer votre password"
                                        read password
                                        mysql -u root -ptest123 -e "USE vsftpd;INSERT INTO accounts(
   username,pass) VALUES ('ftp-$1',md5('$password'));"
                                        echo "Création terminé"
                                        systemctl restart vsftpd.service

                                        echo "Création de la BD de $1, pour cela, veuillez entrez so
   n password"
                                        read password
```

```
                                    echo "Connectez-vous à votre compte root"
                                    mysql -u root -ptest123 -e "CREATE DATABASE $1db; GRANT ALL
PRIVILEGES ON $1db .* TO $1@localhost IDENTIFIED BY $password; FLUSH PRIVILEGES;"
                                    echo "Création terminé"
                                    systemctl restart mariadb.service

                                    echo "Veuillez entrer votre mot de passe d'accès au fichier
de partage samba"
                                    smbpasswd -a $1
                                    systemctl restart smb.service
                                    systemctl restart nmb.service
                        ;;

                        6)          boolean=false
                                    break
                        ;;
                        esac
                done
        fi
else
        echo "Vous n'êtes pas autorisé à exécuter ce script"
fi
```

18.2. scriptRemove.sh

```
#! /bin/bash

if [ $(whoami) == root ] || [ $(whoami) == qmettens ] || [ $(whoami) == clement ] || [ $(whoami) ==
mathieu ]
then
        id -u $1
        if [ $? == '0' ]
        then
                boolean=true
                while("$boolean" = true)
                do
                        echo "1. Suppression du site Internet"
                        echo "2. Suppresion du compte FTP de l'utilisateur"
                        echo "3. Suppresion de la BD de l'utilisateur"
                        echo "4. Suppresion de l'Accès à Samba"
                        echo "5. All"
                        echo "6. Quitter le programme"
                        read number

                        case $number in

                        1)      # Retablissement des parametres par defaut du dossier /home
                                rm -r /home/$1/www
                                chmod 700 /home/$1
                                chown $1:$1 /home/$1
                                echo "Suppresion du password pour l'authentification d'apache"
                                htpasswd -D /etc/httpd/.htpasswd $1
                                echo "Suppresion du VirtualHost de l'utilisateur"
                                rm /etc/httpd/conf.d/VirtualHost/$1.conf
                                # Suppresion de l'URL de l'user dans /var/named/zone.lan
                                # Ainsi que des espaces vides
                                sed -i "/^www.$1/d" /var/named/zone.lan
                                systemctl restart named
                                systemctl restart httpd.service
                        ;;
```

```
                        2)
                                echo "Suppresion de votre utilisateur ftp-$1"
                                rm /etc/vsftpd/vsftpd_user_conf/ftp-$1
                                # Suppresion de l'user ftp-$1 dans la liste des users autoriser
                                # Ainsi que des espaces vides
                                sed -i "/^ftp-$1/d" /etc/vsftpd/vsftpd_user_list
                                # Suppresion de l'user et de son password dans la table accounts
                                mysql -u root -ptest123 -e "USE vsftpd; DELETE FROM accounts WHERE u
 sername= 'ftp-$1';"
                                systemctl restart vsftpd.service
                        ;;

                        3)      echo "Suppression de la database de l'utilisateur : $1"
                                # Suppresion de la BD de l'user
                                mysql -u root -ptest123 -e "DROP DATABASE $1db;"
                                systemctl restart mariadb.service
                        ;;
                        4)      echo "Suppresion de votre mot de passe d'accès au fichier de partage
  samba
"
                                smbpasswd -x $1
                                systemctl restart smb.service
                                systemctl restart nmb.service
                        ;;

                        5)      rm -r /home/$1/www
                                chmod 700 /home/$1
                                chown $1:$1 /home/$1
                                echo "Suppresion du password pour l'authentification d'apache"
                                htpasswd -D /etc/httpd/.htpasswd $1
                                echo "Suppresion du VirtualHost de l'utilisateur"
                                rm /etc/httpd/conf.d/VirtualHost/$1.conf
                                sed -i "/^www.$1/d" /var/named/zone.lan
                                systemctl restart named
                                systemctl restart httpd.service
```

```
                               echo "Suppresion de votre utilisateur ftp-$1"
                               rm /etc/vsftpd/vsftpd_user_conf/ftp-$1
                               sed -i "/^ftp-$1/d" /etc/vsftpd/vsftpd_user_list
                               mysql -u root -ptest123 -e "USE vsftpd; DELETE FROM accounts WHERE u
sername= 'ftp-$1';"
                               systemctl restart vsftpd.service


                               echo "Suppression de la database de l'utilisateur : $1"
                               mysql -u root -ptest123 -e "DROP DATABASE $1db;"
                               systemctl restart mariadb.service

                               echo "Suppresion de votre mot de passe d'accès au fichier de partage
 samba"
                               smbpasswd -x $1
                               systemctl restart smb.service
                               systemctl restart nmb.service
                        ;;
                        6)      boolean=false
                                break
                        ;;
                        esac
                done
        fi
else
        echo "Vous n'êtes pas autorisé à exécuter ce script"
fi
```

19. Conclusion

Lors de ce deuxième quadrimestre, nous avons été amenés à réaliser le déploiement d'un serveur dans le cadre du cours de Linux.

Celui-ci, nous aura permis de découvrir, de configurer et de sécuriser les services demandés, qui étaient disponibles pour notre distribution.

De plus, ce projet, nous aura permis de collaborer et d'échanger nos idées en équipe, ce qui a considérablement amélioré l'expérience de travail.

20. Bibliographie

20.1. Syllabus

- Malaise A., *Projet Linux*, Bachelier en informatique et système finalité réseaux et télécommunications, Année académique 2020-2021

20.2. Source électronique

20.2.1. SSH

- Linuxtricks.fr, 2018, SSH : Installer et configurer un serveur SSH, [en ligne] https://www.linuxtricks.fr/wiki/ssh-installer-et-configurer-un-serveur-ssh

20.2.2. HTTPD

- Kikinovak,2020, *Hébergement sécurisée avec Apache et SSL sous CentOS 7*, [en ligne] https://blog.microlinux.fr/apache-ssl-centos-7/
- Kikinovak,2020, *Serveur web apache sous CentOS 7*, [en ligne] https://blog.microlinux.fr/apache-centos-7/
- Denis, 2015, *[Tuto] Mettre en place un serveur web sous C*entOS, [en ligne] http://denisrosenkranz.com/tuto-mettre-en-place-un-serveur-web-sous-centos-apache-mysql-php-vsftpd/

20.2.3. DNS

- Sery N., 2018, *Mise en place d'un serveur DNS local sur CentOS7*, [en ligne] https://linux-note.com/centos-7-serveur-dns-local/
- Kikinovak, 2020, *Configurer un serveur DNS avec BIND sous CentOS 7*, [en ligne] https://blog.microlinux.fr/bind-centos-7/

20.2.4. MariaDB

- Kikinovak,2020, *Serveur de base de données MySQL/MariaDB sous CentOS 7*, [en ligne] https://blog.microlinux.fr/mysql-centos-7/#securiser

20.2.5. VSFTPD

- Kikinovak, 2020, *Serveur FTP avec vsftpd et let's Encrypt sous CentOS 7*, [en ligne] https://blog.microlinux.fr/vsftpd-letsencrypt-centos-7/
- HowTos, 2019, *Using virtual accounts with vsftpd and mysql on CentOS 5* , [en ligne] https://wiki.centos.org/HowTos/VirtualVsFtpd

20.2.6. CHRONY

- Mairien A., 2019, *Créer un serveur NTP sur CentOS 7 (Chron*y), [en ligne] https://notamax.be/creer-un-serveur-ntp-sur-centos7-chrony/

20.2.7. NFS

- Outscale, 2019, *Mettre en place un serveur NFS*, [en ligne] https://wiki.outscale.net/display/FR/Mettre+en+place+un+serveur+NFS
- Wiki Ubuntu-fr, 2021, *AutoFS – montage automatique de systèmes de fichier*, [en ligne] https://doc.ubuntu-fr.org/autofs

20.2.8. SAMBA

- Kikinovak, 2020, *Installer un serveur Samba sous CentOS 7*, [en ligne] https://blog.microlinux.fr/serveur-samba-centos-7/
- Linux-training.be, 2020, *Chapter 21. Samba securing shares*, [en ligne] http://linux-training.be/networking/ch21.html#idp70224256

20.2.9. SAUVEGARDE

- Kikinovak, 2020*, Serveur de sauvegarde avec Rsnapshot sous CentOS 7*, [en ligne] https://blog.microlinux.fr/rsnapshot-centos-7/
- Kikinovak, 2020, *Serveur de base de données MySQl/MariaBD sous CentOS 7*, [en ligne] https://blog.microlinux.fr/mysql-centos-7/#automatique

20.2.10. ANTIVIRUS

- Mercise SA, 2021, *Installer Clamav sur CentOS 7.6*, [en ligne] https://www.mercise.ch/installer-clamav-sur-centos-7-3/
- Edgaras G., 2019*, How to install Clamav on CentOS 7 : A step-by-step guide*, [en ligne] https://www.hostinger.com/tutorials/how-to-install-clamav-centos7