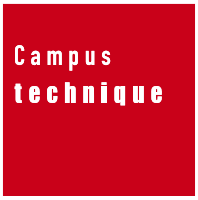
Rapport de projet

2ème Bachelier en Informatique

Linux

Auteur :
Timothée Simon
Florian Giiarrusso
Fabio Cumbo

Enseignant :
Antoine Malaise



Année académique 2017 - 2018

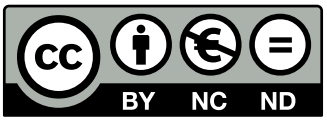


Table des matières

1 Introduction

1.1 Choix de la distribution

Pour le choix de la distribution nous avons commencé pas mettre en place certains critères de recherche :

- La gratuité
- La stabilité
- La légèreté

Nos recherches nous montrent plusieurs choix possible :

- RHEL : Payante
- Arch Linux : Pas la plus stable car elle fonctionne en rolling release
- Ubuntu Server et Debian : Ne possèdent pas de version core ce qui les rend plus lourdes
- CentOS : Répond au mieux à tout nos critères

Nous allons donc faire notre serveur sous CentOS car celui ci répond à tous nos critères dans sa version Core.

1.2 Mode d'installation

Pour l'installation nous avons opté pour l'écriture de scripts pour chacune des fonctionnalités de notre serveur. Cela nous permet de nous rappeller de nos procédure et de toujours les comprendre dans quelques années. Il nous suffit maintenant de copier nos scripts sur une machine réelle ou virtuelle pour pouvoir immédiatement commencer à configurer Linux.
Bien évidement nos scripts ne gères pas beaucoup d'erreurs et ne vérifient pas ce que l'utilisateur à entré, ils ne sont donc pas vraiment prêt pour tout les usage mais sont très utiles comme notes.

1.3 Organisation du groupe

Pour nous organiser nous avons utilisé les outils de GitHub. Nous avons donc commencé par créer un repo (privé pour le moment mais nous le passerons surement en publique après les examens) nous permettant de travailler ensemble sur notre code. Nous avons aussi utilisé la ToDo list de GitHub pour nous organiser dans notre développement.

1.4 Machine physique

Nous avons eu l'occation d'utiliser une machine physique pour mettre en pratique nos scripts. Celle-ci est bien évidement une machine de récupération, elle est composée de Intel Pentium 4, de 1Go de DDR2 et de deux disques dur en RAID 1.

2 Installation

Pour l'installation nous avons suivis les étapes suivantes :

- Mise en place de la machine avec l'iso bootable (sur machine virtuelle ou physique)
- Sélection des locale (English US)
- Modification de l'heure (Bruxelles)
- Partitionnement manuel (LVM avec encryption AKA LUKS) :
  - /dev/sda1 monté sur /boot : 1024 Mo en XFS
  - LVM (encryptée)
    - centos-root monté sur / : 5320 Mo en XFS
    - centos-home monté sur /home : 1023 Mo en XFS
    - centos-swap monté comme partition de swap : 819 Mo
- Activation de la NIC (enp0s3)
- Nom de domaine
- Lancement de l'installation
- Mise en place du mot de passe root

3 Serveur NTP

Le protocole NTP (Network Time Protocol) va permettre de synchroniser les horloges des ordinateurs connectés au même réseau local que celle du serveur de temps. Celui-ci devra synchroniser sa propre horloge en contactant un serveur de temps de référence à distance donc par internet. Cette synchronisation des heures permettra entre autres de ne pas perturber certaines applications utilisant l'horloge du système mais aussi pour donner plus de cohérence en cas de comparaison des messages de « logs » de plusieurs ordinateurs sur le réseau.

```
#!/bin/bash

source ../Common.sh

RootCheck

#Installation de NTP
Installe ntp ntpdate ntp-doc

#On stop le service ntpd
systemctl stop ntpd

#On met le serveur à la bonne heure au préalable
ntpdate be.pool.ntp.org

#Configuration du serveur ntp
cp ntp.conf /etc/ntp.conf

#Démarrage et activation du service
Service ntpd

echo "Le service NTP est maintenant installé et activé."
```

../scripts/ntp/NTP.sh

```
# For more information about this file, see the man pages
# ntp.conf(5), ntp_acc(5), ntp_auth(5), ntp_clock(5), ntp_misc(5),
    ntp_mon(5).

driftfile /var/lib/ntp/drift

#On ajoute un directory pour les logs
logfile /var/log/ntp.log

# Permit time synchronization with our time source, but do not
# permit the source to query or modify the service on this system.
#restrict default nomodify notrap nopeer noquery

# Permit all access over the loopback interface.  This could
# be tightened as well, but to do so would effect some of
# the administrative functions.
#restrict 127.0.0.1
#restrict ::1

```

```
#On établis les règles pour l'accès au service, et sécurisation
#On empêche les machines distantes de modifier la configuration du serveur, protection ddos,..
restrict default nomodify nopeer notrap noquery

#On fait confiance à la machine elle-même
restrict 127.0.0.1 mask 255.0.0.0
#IpV6
restrict ::1
# Hosts on local network are less restricted.
#restrict 192.168.1.0 mask 255.255.255.0 nomodify notrap

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

#Synchronisation des horloges avec la pool zone de Belgique
server 3.be.pool.ntp.org
server 1.europe.pool.ntp.org
server 0.europe.pool.ntp.org

#On indique au serveur qu'il doit se synchroniser sur l'horloge locale
server 127.127.1.0

#serveur “”bidon en guise 'dIP fallback, au cas où la source de temps extérieure deviendrait momentanément indisponible. En cas 'dindisponibilité du serveur distant, NTP continuera à tourner en se basant sur ce fonctionnement-là.
fudge 127.127.1.0 stratum 10


#broadcast 192.168.1.255 autokey  # broadcast server
#broadcastclient      # broadcast client
#broadcast 224.0.1.1 autokey    # multicast server
#multicastclient 224.0.1.1    # multicast client
#manycastserver 239.255.254.254   # manycast server
#manycastclient 239.255.254.254 autokey # manycast client

#Enable public key cryptography.
#crypto

includefile /etc/ntp/crypto/pw

# Key file containing the keys and key identifiers used when operating
# with symmetric key cryptography.
#
keys /etc/ntp/keys

# Specify the key identifiers which are trusted.
#trustedkey 4 8 42

# Specify the key identifier to use with the ntpdc utility.
#requestkey 8
```

```
72 # Specify the key identifier to use with the ntpq utility.
   #controlkey 8
74
   # Enable writing of statistics records.
76 #statistics clockstats cryptostats loopstats peerstats

78 # Disable the monitoring facility to prevent amplification attacks
       using ntpdc
   # monlist command when default restrict does not include the noquery
       flag. See
80 # CVE-2013-5211 for more details.
   # Note: Monitoring will not be disabled with the limited restriction
       flag.
82 disable monitor
```

../scripts/ntp/ntp.conf

On déclare d'abord le fichier de dérive « driftfile ». Il va permettre de corriger les dérives de l'horloge système en l'absence de connexion réseau au serveur de référence. On déclare ensuite le répertoire et le fichier pour stocker les « logs » du service ntpd. On permet la synchronisation avec notre source de temps mais on interdit à la source de modifier ou d'interroger le service sur ce système. On autorise les accès sur l'interface de bouclage Ensuite, on se synchronise avec les serveurs NTP belge de référence. Pour résoudre les problèmes de charge on entre plusieurs adresse. Il s'agit d'un groupement de serveur et la redistribution se fait à l'aide du Round Robin DNS (association de plusieurs adresses IP à un FQDN) Enfin, on précise au serveur de se synchroniser sur l' « Undisciplined Local Clock » et on indique un faux « pilote » destiné à la sauvegarde de l'heure dans le cas où aucune source externe d'heure synchronisée n'est disponible.

4 Connexion SSH

Pour l'administration à distance nous avons décicé d'utilier SSH (***S**ecure **SH**ell*) car se protocol est plus sécurisé que Telnet. Pour la connexion nous n'autorisons que l'authentification avec une clé RSA ou avec un mot de passe et un facteur de double autentification. Pour la double autentification nous avons décidé d'utiliser une clé Yubikey, ce facteur étant physique cela limite fortement l'accès aux personnes non-authorisée. Nous avons choisi de permettre une autre méthode d'authentification car l'utilisation d'une clé RSA seule est un risque en cas de perte du fichier et nous voulions toujours pouvoir être capable d'administer notre serveur même si ce fichier est perdu ou corrompu.

```
#!/bin/bash

source ../Common.sh

RootCheck

#Installation du paquet open-ssh (vérification) permettant de partager
    un service ssh, par défaut il est normalement déjà installé
Installe openssh-server

# Utilsation du fichier de config
cp sshd_config /etc/ssh/sshd_config
cp sshbanner /etc/ssh/banner

#démarrage service sshd
Service sshd
```

../scripts/ssh/SSH.sh

Après avoir vérifié que l'utilisateur qui lance les script est bien root et installé le serveur openSSH si celui-ci n'étais pas déjà installé, nous copions simplement le fichier de configuration et la bannière au bon endroit. Pour finir nous lançons et activons au demarrage le daemon.

5 Serveur NTP

Le protocole NTP (Network Time Protocol) va permettre de synchroniser les horloges des ordinateurs connectés au même réseau local que celle du serveur de temps. Celui-ci devra synchroniser sa propre horloge en contactant un serveur de temps de référence à distance donc par internet. Cette synchronisation des heures permettra entre autres de ne pas perturber certaines applications utilisant l'horloge du système mais aussi pour donner plus de cohérence en cas de comparaison des messages de « logs » de plusieurs ordinateurs sur le réseau.

```
#!/bin/bash

source ../Common.sh

RootCheck

#Installation de NTP
Installe ntp ntpdate ntp-doc

#On stop le service ntpd
systemctl stop ntpd

#On met le serveur à la bonne heure au préalable
ntpdate be.pool.ntp.org

#Configuration du serveur ntp
cp ntp.conf /etc/ntp.conf

#Démarrage et activation du service
Service ntpd

echo "Le service NTP est maintenant installé et activé."
```

../scripts/ntp/NTP.sh

On déclare d'abord le fichier de dérive « driftfile ». Il va permettre de corriger les dérives de l'horloge système en l'absence de connexion réseau au serveur de référence. On déclare ensuite le répertoire et le fichier pour stocker les « logs » du service ntpd. On permet la synchronisation avec notre source de temps mais on interdit à la source de modifier ou d'interroger le service sur ce système. On autorise les accès sur l'interface de bouclage Ensuite, on se synchronise avec les serveurs NTP belge de référence. Pour résoudre les problèmes de charge on entre plusieurs adresse. Il s'agit d'un groupement de serveur et la redistribution se fait à l'aide du Round Robin DNS (association de plusieurs adresses IP à un FQDN) Enfin, on précise au serveur de se synchroniser sur l' « Undisciplined Local Clock » et on indique un faux « pilote » destiné à la sauvegarde de l'heure dans le cas où aucune source externe d'heure synchronisée n'est disponible.

6 Serveurs de fichiers

Nous avons mis en place deux serveurs de fichier, un serveur NFS et un serveur Samba.

6.1 Serveur NFS

Le serveur NFS (***N**etwork **F**ile **S**ystem*) permettant à un utilisateur d'accéder à des fichier sur un serveur de la même façon qu'il accèdes à des fichiers stoqués en local.

```
#!/bin/bash
source ../Common.sh

RootCheck

s="./NFS.sh [DOSSIER] [ARGUMENTS] [IP]
DOSSIER: Dossier de partage (Defaut: /Partage)
ARGUMENTS: Arguments du partage nfs (Defaut: (rw,sync,no_root_squash,
    no_subtree_check))
IP: Adresse ip du serveur (Defaut: adresse IP local de la machine)
"

Aide $1 $s
# Défault du dossier de partage
DossierPartage=`Argument $1 "/Partage"`
ARG=`Argument $2 "(rw,sync,no_root_squash,no_subtree_check)"`
IP=`Argument $3 \`ip addr | grep 'state UP' -A2 | tail -n1 | awk '{
    print $2}' | cut -f1  -d'/'\``

# Installation du serveur nfs
Installe nfs-utils

#création du dossier partagé si celui-ci n'existe pas encore
mkdir -p $DossierPartage

#Modification des permissions d'accès
chmod 755 $DossierPartage

echo "Le dossier $DossierPartage est maintenant crée"


#Activation et démarrage des services nfs au boot
Service rpcbind
Service nfs


#Configuration du fichier /etc/exports
echo "$DossierPartage $IP$ARG" >> /etc/exports

#On exporte le partage
exportfs -a
showmount -e
```

../scripts/nfs/NFS.sh

Après avoir vérifié que l'utilisateur qui lance le script est bien root et importé les variables en argument, nous installons le serveur nfs si celui-ci n'est pas déjà installé. Ensuite nous créons le dossier de partage avec les bons droits et activons les processus requis pour le serveur. Finalemement nous ajoutons les dossier partagé ainsi que les arguments au fichier et partage et nous en informons le nfs.

6.2 Serveur SMB

Le serveur SMB (***S**erver **M**essage **B**lock*) permet à des ordinateur de partages des fichier et des imprimantes entre eux.

```
#/bin/bash

souce ../Common.sh

RootCheck
s=" ./SAMBA.sh DOSSIER UTILISATEUR NOM GROUPE
DOSSIER: Dossier de partage (Defaut: /Partage)
UTILISATEUR: Utilisateur possédant les droits (Defaut: Utilisateur
    courant)
NOM: Nom du partage (Defaut: DOSSIER)
GROUPE: Nom du groupe utilisé pour le partage (Defaut: GroupePartage)
"

Aide $1 $s

#On crée le dossier de partage
DossierPartage=`Argument $1 "/Partage"`
UserP=`Argument $2 $USER`
NameP=`Argument $3 $DossierPartage`
GroupePartage=`Argument $4 "GroupePartage"`

#Installation samba
Installe samba

#démarrage et activation du démon au démarrage
Service smb

#création du dossier partagé si celui-ci n'existe pas encore
mkdir -p $DossierPartage

#Ajout du groupe sharedFolder contenant user1 et user2
groupadd $GroupePartage
useradd -g sharedFolder $UserP

#On gère les permissions du dossier partagé
chmod 770 $DossierPartage
chown -R $UserP:$GroupePartage $DossierPartage
```

```
38 echo "Veuilez choisir un mot de passe pour l'utilisateur $UserP"
   smbpasswd -a $UserP
40
   # Copie des réglages de samba
42 echo"
   [${NameP}]
44 path=${DossierPartage}
   comment=Partage crée par un script
46 public=yes
   force directory mode=777
48 force create mode=777
   writeable=yes
50 browseable=yes" >> smb.conf
   cp smb.conf /etc/samba/smb.conf
52
   # Désavtivation de SELinux (car autrement impossible de se connecter
       depuis le client, il faudra configurer tout ça)
54 setenforce 0

56 #Redémarrage du service smb
   sudo systemctl restart smb.service
```

../scripts/samba/SAMBA.sh

Tout d'abord nous intégrons un menu d'aide et nous importons les variables passées en arguments. Ensuite nous installons le serveur samba si celui-ci n'est pas encore installé et nous démarrons ses services.Nous créons le groupe relatif au partage et nous y ajoutons l'utilisateur. Maintenant il nous faut créer demander le mot de passe du partage à l'utilisateur et créer la configuration. Finalement nous redémarons le daemon.

7 Serveur FTP

Le protocole FTP (File Transfer Protocol) permet de transferer des fichier d'un serveur à un client.

```
#!/bin/bash
source ../Common.sh
RootCheck

s='./FTP.sh [DOSSIER]
DOSSIER: Dossier de partage (Defaut: /Partage)
'

Aide $1 $s

Installe vsftp

DossierPartage=`Argument $1 "/Partage"`

systemctl start vsftp.service
systemctl enable vsftp.service

mkdir -p $DossierPartage

echo "# Allow all connections \nvsftpd: ALL\n# IP adress range\nvsftpd:
    10.0.0.0/255.255.255.0" > /etc/hosts.allow
```

../scripts/ftp/FTP.sh

8 Anti-virus

```
#!/bin/bash
source ../Common.sh

RootCheck
#Check installation du support EPEL
Installe epel-release

#Installation de tous les composants de ClamAV

Installe clamav-server clamav-data clamav-update clamav-filesystem clamav clamav-scanner-systemd clamav-devel clamav-lib clamav-server-systemd

#Configuration du daemon Clam
#Copie du template dans le cas où l'on a pas de fichier de configuration
cp /usr/share/clamav/template/clamd.conf /etc/clamd.d/clamd.conf

#Activation de Freshclam pour garder la DB à jour

#Création du service freshclam et configuration
echo "# Run the freshclam as daemon
[Unit]
Description = freshclam scanner
After = network.target

[Service]
Type = forking
ExecStart = /usr/bin/freshclam -d -c 4
Restart = on-failure
PrivateTmp = true

[Install]
WantedBy=multi-user.target" > /usr/lib/systemd/system/clam-freshclam.service

#Démarrage et activation du service au démarrage
Service clam-freshclam.service

#Changement des fichiers service autrement  clamd@.service ne démarre pas
#On renomme le fichier si on l'a pas déjà fais
if [ ! -e "/usr/lib/systemd/system/clamd.service" ];then
    mv /usr/lib/systemd/system/clamd@.service /usr/lib/systemd/system/clamd.service
fi

#On modifie le fichier clamd@scan.service et on change la référence vers
# /lib/systemd/system/clamd.service
echo ".include /lib/systemd/system/clamd.service

[Unit]
```

```
Description = Generic clamav scanner daemon

[Install]
WantedBy = multi-user.target" > /usr/lib/systemd/system/clamd@scan.service

#On modifie le fichier /usr/lib/systemd/system/clamd.service et on configure
echo "[Unit]
Description = clamd scanner daemon
After = syslog.target nss-lookup.target network.target

[Service]
Type = simple
ExecStart = /usr/sbin/clamd -c /etc/clamd.d/clamd.conf --foreground=yes
Restart = on-failure
PrivateTmp = true

[Install]
WantedBy=multi-user.target" > /usr/lib/systemd/system/clamd.service

#Démarrage et automatistion des services
Service clamd.service
Service clamd@scan.service
```

../scripts/antivirus/antivirus.sh

Ce script va permettre d'installer l'antivirus sur le serveur pour amélioré sa sécurité On vérifie d'abord que l'utilisateur exécute bien le script en tant que root, ce qui est nécessaire pour configurer les fichiers de ClamAV. On installe ensuite EPEL (Extra Package for Entreprise Linux) qui est un repo fournissant des package additionnels pour les distributions de type RedHat/CentOs.

Ensuite, au cas où on ne possède pas de fichier de configuration, on copie le template clamd.conf. On peut créer le service freshclam et le configurer de sortie qu'il vérifie 4 fois par jour la précense de mise à jour. On le démarre et on l'active au démarrage ensuite.

On renomme également le fichier /usr/lib/systemd/system/clamd@.service en /usr/-lib/systemd/system/clamd.service autrement, par défaut le service ne fonctionne pas. Puis, il faut indiquer le bon chemin du clamd.service dans le fichier /usr/lib/-systemd/system/clamd@scan.service et on peut configurer le fichier clamd.service

```
#!/bin/bash

date=$(date +%d_%m_%Y_%H_%M)

echo Arret du service freshclam et mise à jour

service clam-freshclam stop

sudo freshclam

service clam-freshclam start
```

```
13 echo Service  redémarré et mise à jour effectuée
   echo Entrez le nom du repertoire ou fichier à analyser :
15 read nom

17 if [ -e "$nom" ];then
     echo Analyse en cours ....
19
     echo "
       ----------------------------------------------------------------------
       ---\n" >> $HOME/analyseVirus.log
21   echo " " >> $HOME/analyseVirus.log
     echo "Analyse du $date\n" >> $HOME/analyseVirus.log
23   echo " "  >> $HOME/analyseVirus.log

25   clamscan -l $HOME/analyseVirus.log -r $nom
   else
27   echo Vous avez entré un nom de repertoire ou fichier inexistant
   fi
```

../scripts/antivirus/analyseVirus.sh

Les dernières lignes vont pemettre d'activer et de démarrer automatiquement les services clamd et clamd@scan au démarrage

Ce script va permettre d'exécuter une analyse virale d'un dossier ou fichier choisis par l'utilisateur qui lance le script. On met d'abord à jour la base de données de l'antivirus et puis on demande à l'utilisateur de spécifier la cible. Si elle existe, on procède à l'analyse.

9 Remerciement

Je remercie Terencio Agozzino pour avoir réalisé la mise en page de ce document en LaTeX.

Configuration du ssh https ://www.linux.com/learn/advanced-ssh-security-tips-and-[redacted]
tricks