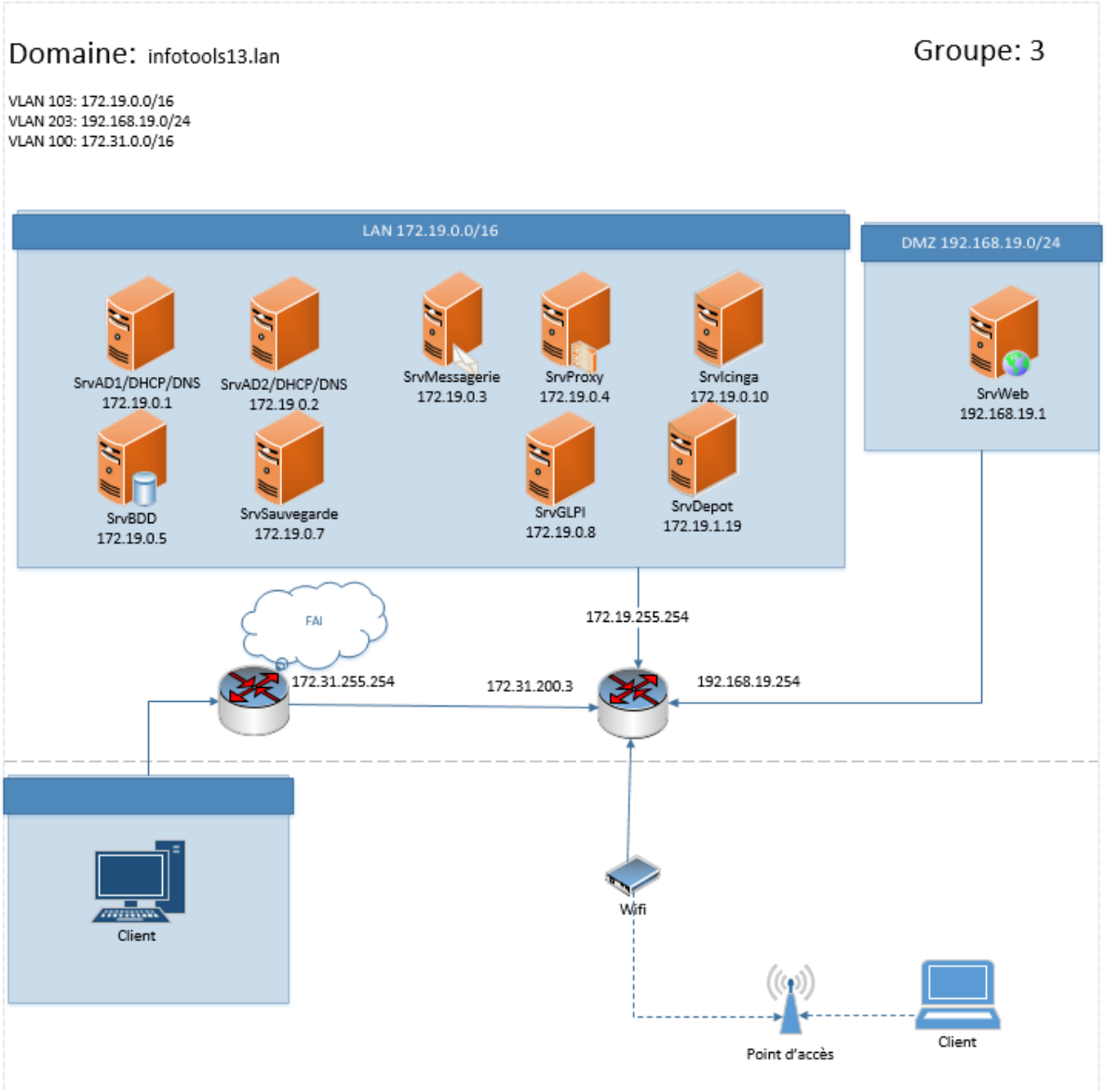
Rapport Infotools

Sommaire

1-Schéma réseau :



2-Mise en place d'un serveur web

2.1 Installation d'apache et module php

Pour installer un serveur web, voici les instructions qu'il faut suivre pour que tout se passe bien :

apt install apache2
apt install php libapache2-mod-php php-mysql

On active apache et on le redémarre :

systemctl enable apache2

systemctl restart apache2

2.2 Création d'utilisateur pour l'accès en ssh

Maintenant que le serveur web est opérationnel, il faut créer des utilisateurs qui pourront se connecter sur notre serveur via ssh. Pour cela, on a la commande useradd pour ajouter un utilisateur comme ceci :

Ici, j'ai créé deux utilisateurs pour accéder dans le répertoire /var/www/html/ et /html2/ parce que je collabore avec deux groupes de SLAM (Solutions Logicielles et Applications Métiers).

Là, je leur donne de droit sur les deux répertoires.

2.3-Installation de PhpMyadmin

Pour raison de sécurité, on va installer PhpMyadmin sur le serveur web qui est dans le réseau DMZ mais pas sur la base de données qui dans le réseau LAN.

En effet, cela nous permet d'éviter de créer une règle de redirection vers le LAN depuis l'extérieur. Au lieu de rédiger vers de LAN, on va rédiger vers la DMZ.

Maintenant, on va procéder à l'installation de notre PhpMyadmin sur notre serveur web.

Il faut se rendre sur le site de PhpMyadmin à l'adresse suivante pour télécharger la dernière version du paquet : https://www.phpmyadmin.net/downloads/

```
#cd /tmp/
# wget https://files.phpmyadmin.net/phpMyAdmin/4.9.2/phpMyAdmin-4.9.2-all-
languages.tar.gz
# tar xvf phpMyAdmin-4.9.2-all-languages.tar.gz
```

Déplacer le dossier dans le répertoire /user/share :

```
mv phpMyAdmin-4.9.2-all-languages /usr/share/phpmyadmin
```

Donner les droits du dossier à l'utilisateur www-data du service apache :

```
chown -R www-data:www-data /usr/share/phpmyadmin
```

Il faut créer un dossier temporaire pour PhpMyadmin :

```
mkdir -p /var/lib/phpmyadmin/tmp
sudo chown www-data:www-data /var/lib/phpmyadmin/tmp
```

Lancer l'installation des paquets additionnels d'apache pour PhpMyadmin :

```
apt-get install -y php7.3-common php7.3-mysql php7.3-gd php7.3-imap php7.3-json
php7.3-curl php7.3-zip php7.3-xml php7.3-mbstring php7.3-bz2 php7.3-intl php7.3-
gmp
```

On redémarre le service apache

```
systemctl restart apache2
```

Nous allons créer le fichier de configuration qui va nous permettre d'accès à la page web de PhpMyadmin via apache et il faut utiliser l'éditeur **nano** :

```
vi  /etc/apache2/conf-available/phpmyadmin.conf
```

Coller et enregistrer le fichier de configuration suivant :

```
# phpMyAdmin default Apache configuration

Alias /phpmyadmin /usr/share/phpmyadmin

<Directory /usr/share/phpmyadmin>
    Options SymLinksIfOwnerMatch
    DirectoryIndex index.php

    <IfModule mod_php5.c>
        <IfModule mod_mime.c>
            AddType application/x-httpd-php .php
        </IfModule>
        <FilesMatch ".+\.php$">
            SetHandler application/x-httpd-php
        </FilesMatch>

        php_value include_path .
        php_admin_value upload_tmp_dir /var/lib/phpmyadmin/tmp
```

Attention sur la capture ci-dessus, on n'a pas tous les éléments. Il faut donc récupérer tous les éléments sur ce lien :

https://www.gabinhocity.eu/installer-apache-php7-3-mysql-et-phpmyadmin-sur-debian-10-buster/

Activer le fichier de configuration via la commande suivante :

```
a2enconf phpmyadmin.conf
```

On redémarre le service apache :

```
systemctl restart apache2
```

Il faut créer et modifier le fichier de configuration de PhpMyadmin :

```
 # cp /usr/share/phpmyadmin/config.sample.inc.php
/usr/share/phpmyadmin/config.inc.php

vi  /usr/share/phpmyadmin/config.inc.php
```

Modifier les informations suivantes :

```
. . .
$cfg['blowfish_secret'] = ''; /* YOU MUST FILL IN THIS FOR COOKIE AUTH! */
. . .
```

Par

```
$cfg['blowfish_secret'] = '2O:.uw6-8;Oi9R=3W{tO;/QtZ]4OG:T:'; /* YOU MUST FILL IN
THIS FOR COOKIE AUTH! */
```

Et les lignes suivantes :

```
/* User used to manipulate with storage */
// $cfg['Servers'][$i]['controlhost'] = '';
// $cfg['Servers'][$i]['controlport'] = '';
// $cfg['Servers'][$i]['controluser'] = 'pma';
// $cfg['Servers'][$i]['controlpass'] = 'pmapass';
/* Storage database and tables */
// $cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
// $cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
// $cfg['Servers'][$i]['relation'] = 'pma__relation';
```

Par

```
$/* User used to manipulate with storage */
// $cfg['Servers'][$i]['controlhost'] = '';
// $cfg['Servers'][$i]['controlport'] = '';
 $cfg['Servers'][$i]['controluser'] = 'phpmyadmin';
 $cfg['Servers'][$i]['controlpass'] = 'Votre_mot_depasse_phpmyadmin';
/* Storage database and tables */
$cfg['Servers'][$i]['pmadb'] = 'phpmyadmin';
$cfg['Servers'][$i]['bookmarktable'] = 'pma__bookmark';
```

Attention sur la capture ci-dessus, on n'a pas tous les éléments. Il faut donc récupérer tous les éléments sur ce lien :

https://www.gabinhocity.eu/installer-apache-php7-3-mysql-et-phpmyadmin-sur-debian-10-buster/

Enfin, on modifie le fichier /usr/share/phpmyadmin/config.inc.php pour indiquer notre serveur de base de données :

```
/**
 * First server
 */
$i++;
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
$cfg['Servers'][$i]['verbose'] = 'Localhost';
$cfg['Servers'][$i]['host'] = 'localhost';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;

/**
 * Second server
 */
$i++;
/* Authentication type */
$cfg['Servers'][$i]['auth_type'] = 'cookie';
/* Server parameters */
$cfg['Servers'][$i]['host'] = '172.19.0.5';
$cfg['Servers'][$i]['verbose'] = 'Srv BDD';
$cfg['Servers'][$i]['compress'] = false;
$cfg['Servers'][$i]['AllowNoPassword'] = false;
```

On redémarre le service apache

```
systemctl restart apache2
```

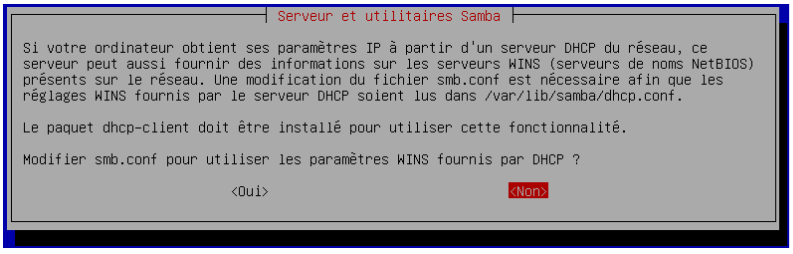
3-Mise en place d'un outil de supervision

3.1 Installation d'Icinga

Icinga est un fork de nagios qui permet de superviser des différents services et équipements informatiques.

Pour installer Icinga, il faut installer les paquets d'Icinga. Pour cela, il suffit de taper la commande suivante :

apt install icinga



On rentre le mot de passe pour se connecter à l'interface d'icinga.

Modifier le fichier **/etc/icinga/icinga.cfg** :

Check_external_commands=1

Modifier le fichier **/etc/apache2/conf-enabled/icinga.conf** et commenter la ligne ***Require all granted***.

On redémarre apache.

service restart apache2

Ensuite exécuter les commandes suivantes :

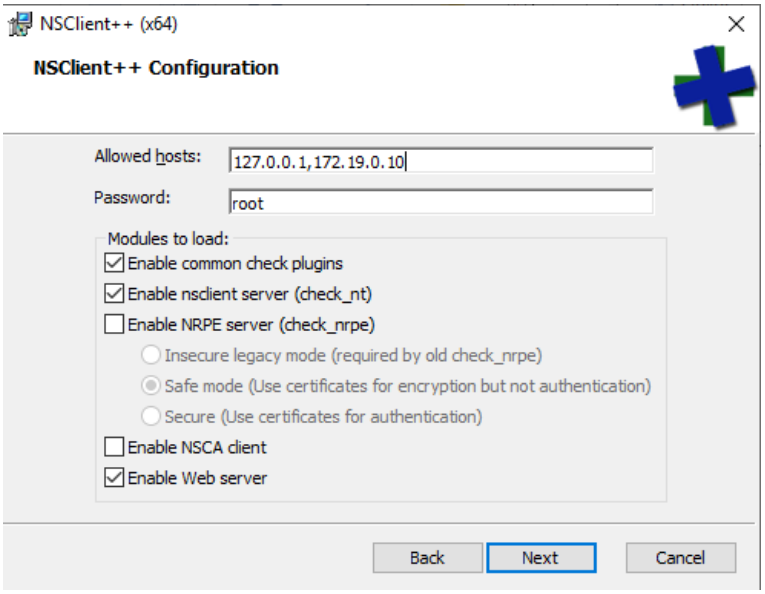
- `service icinga stop`
- `dpkg-statoverride --update --add nagios www-data 2710 /var/lib/icinga/rw/`
- `dpkg-statoverride --update --add nagios nagios 751 /var/lib/icinga/`
- `service icinga start`

3.2 Installation d'agent Nsclient

Il faut télécharger la dernière version de Nsclient.

https://nsclient.org/download/

Puis lancer l'installation sur nos clients Windows.



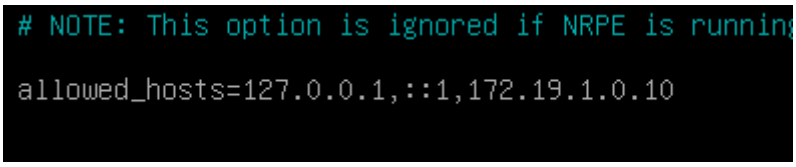
3.3 Installation de l'agent nrpe

Pour installer l'agent nrpe sur nos clients linux, il faut installer le paquet **nagios-nrpe-server**.

apt install nagios-nrpe-server

On indique à l'agent l'adresse ip de notre serveur Icinga.

nano /etc/nagios/nrpe.cfg



On redémarre le service :

Systemctl restart nagios-nrpe-server.service

3.4 Supervision d'un client Windows

On commence par la création d'hôte dans le dossier **/etc/icinga/objects**

```
  GNU nano 3.2                                  SrvAD.cfg                                  Mo

# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in icinga-common_services.cfg may also apply.
#

define host{
        use                     generic-host            ; Name of host template to use
        host_name               SrvAD
        alias                   Serveur Active Directory
        address                 172.19.0.1
        }
```

Ensuite, on crée la commande pour chaque service.

```
  GNU nano 3.2                                  command.cfg                                Modifié

# If you are confused about this command definition, cause you was
# reading other suggestions, please have a look into
# /usr/share/doc/monitoring-plugins/README.Debian


# commande check disk
define command {
        command_name    check_nscp_disk
        command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -p 12489 -s root -v USE$
}

#commande check time
define command {
        command_name    check_nscp_time
        command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -p 12489 -s root -v UPT$
}

#commande check memoire
define command {
        command_name    check_nscp_memoire
        command_line    /usr/lib/nagios/plugins/check_nt -H '$HOSTADDRESS$' -p 12489 -s root -v MEM$
}
```

Enfin, on l'ajoute les services dans le fichier de configuration de l'hôte.

```
  GNU nano 3.2                                  SrvAD.cfg                                  Modifié

# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in icinga-common_services.cfg may also apply.
#

define host{
        use                     generic-host            ; Name of host template to use
        host_name               SrvAD
        alias                   Serveur Active Directory
        address                 172.19.0.1_
        }

#check disk

define service{
        use                             generic-service         ; Name of service template to use
        host_name                       SrvAD
        service_description             Disk
        check_command                   check_nscp_disk
        }

#check time

define service{
        use                             generic-service         ; Name of service template to use
        host_name                       SrvAD
        service_description             Time
        check_command                   check_nscp_disk
        }
```

3.5 Supervision d'un client Linux

Tout d'abord, avant de créer l'hôte, il faut installer le plugin nrpe sur le serveur.

apt install nagios-nrpe-plugin

Ensuite, on crée l'hôte dans le dossier **/etc/icinga/objects**.

```
  GNU nano 3.2                                  SrvBDD.cfg

# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in icinga-common_services.cfg may also apply.
#

define host{
        use                     generic-host            ; Name of host template to use
        host_name               SrvBDD
        alias                   serveur de base de données
        address                 172.19.0.5_
        }
```

Puis certaines commandes sont déjà créées lors d'installation de l'agent nrpe sur le client. On peut les voir dans le fichier **/etc/nagios/nrpe.cfg** du client.

```
command[check_users]=/usr/lib/nagios/plugins/check_users -w 5 -c 10
command[check_load]=/usr/lib/nagios/plugins/check_load -r -w .15,.10,.05 -c .30,.25,.20
command[check_hda1]=/usr/lib/nagios/plugins/check_disk -w 20% -c 10% -p /dev/hda1
command[check_zombie_procs]=/usr/lib/nagios/plugins/check_procs -w 5 -c 10 -s Z
command[check_total_procs]=/usr/lib/nagios/plugins/check_procs -w 150 -c 200
```

Enfin, il manque plus qu'ajouter les services dans le fichier de configuration de l'hôte **/etc/icinga/objects/SrvBDD.cfg**.

```
  GNU nano 3.2                          SrvBDD.cfg                          Modifié

# A simple configuration file for monitoring the local host
# This can serve as an example for configuring other servers;
# Custom services specific to this host are added here, but services
# defined in icinga-common_services.cfg may also apply.
#

define host{
        use                     generic-host            ; Name of host template to use
        host_name               SrvBDD
        alias                   serveur de base de données
        address                 192.168.1.32
        }

#check users
define service{
        use                             generic-service         ; Name of service template to use
        host_name                       SrvBDD
        service_description             Users
        check_command                   check_nrpe!check_users
        }
#check disk_
define service{
        use                             generic-service         ; Name of service template to use
        host_name                       SrvBDD
        service_description             Disk
        check_command                   check_nrpe!check_disk
        }
```

3.6 Installation de nullmailer

On va installer nullmailer pour renvoyer des mails.

apt install nullmailer

On ajoute nos utilisateurs dans le fichier **/etc/icinga/objects/contacts_icinga.cfg**.

```
define contact{
        contact_name                    toto
        alias                           Toto
        service_notification_period     24x7
        host_notification_period        24x7
        service_notification_options    w,u,c,r
        host_notification_options       d,r
        service_notification_commands   notify-service-by-email
        host_notification_commands      notify-host-by-email
        email                           toto@infotools.lan_
        }
```

On peut aussi créer un groupe de contact dans le fichier **/etc/icinga/objects/contacts_icinga.cfg**.

```
 GNU nano 3.2                                contacts_icinga.cfg

# no need to create more than one contact group.

define contactgroup{
        contactgroup_name       admins
        alias                   Nagios Administrators
        members                 root
        }
define contactgroup{
        contactgroup_name       GrpLin
        alias                   Admin Linux
        members                 simon, toto
        }
define contactgroup{
        contactgroup_name       GrpWin
        alias                   Admin Windows
        members                 felix, toto
        }
```

Maintenant, il faut créer les deux groupes (Linux, Windows) dans **/etc/icinga/objects**.

On crée le groupe Linux :

```
 GNU nano 3.2                                          GrpLin.cfg

# Some generic hostgroup definitions

# A simple wildcard hostgroup
define hostgroup {
        hostgroup_name  GrpLin
                alias           groupe linux
                members          SrvBDD
        }
```

On crée le groupe Windows :

```
 GNU nano 3.2                                          GrpWin.cfg

# Some generic hostgroup definitions

# A simple wildcard hostgroup
define hostgroup {
        hostgroup_name  GrpWin
                alias           groupe windows
                members          SrvAD_
        }
```

4- Mise en place d'un proxy

4.1 Installation de Squid

Un serveur proxy va nous permettre de filtrer tous les trafics internet.

Pour l'installer, il suffit de faire :

apt update

apt install squid

4.2 Configuration de Squid

Maintenant, on va passer la configuration de notre serveur proxy. On va modifier le fichier de configuration de squid.

nano /etc/squid/squid.conf

On indique l'adresse de notre réseau local et l'autoriser d'aller sur internet.

acl lan src 172.19.0.0/16

http_access allow lan

4.3 Installation de SquidGuard

Pour filtrer les trafics qui vont sur internet, on va utiliser squidGuard. On exécute les commandes suivantes :

apt install squidguard

cd /var/lib/squidguard/db

wget http://cri.univ-tlse1.fr/blacklists/download/blacklists.tar.gz

tar -xzf blacklists.tar.gz

4.4 Configuration de SquidGuard

On va indiquer à Squid qu'il faut rediriger son programme vers Squidguard pour effectuer un filtrage de sites.

nano /etc/squid/squid.conf

On ajoute ceci :

```
# Ajouter à la fin du fichier :
redirect_program /usr/bin/squidGuard -c /etc/squidguard/squidGuard.conf
redirect_children 10
```

On va ajouter des règles dans le fichier **/etc/squidguard/squidGuard.conf**

```
src lan {
   ip 172.19.0.0/16
}

dest games {
domainlist games/domains
urllist games/urls
}

dest local{
}

acl {
lan {
pass !games all
redirect http://localhost/proxy.html
}
default {
pass local none
}
}
```

On reconstruit la base, c'est-à-dire la liste noire.

squidGuard -C all -d /var/lib/squidguard/db/blacklists

On crée la page proxy.html dans le répertoire suivant **/var/www/html/squidguard/proxy.html.**

```
<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <title>Proxy Squid</title>
</head>
<body>
    <h1>L'accès à ce site n'est pas autorisé !</h1>
</body>
</html>
```

On change la configuration dans Apache2 pour faire pointer vers ce fichier dans **/etc/apache2/sites-available/000-default.conf.**

DocumentRoot /var/www/html/squidguard

On recharge Apache2 :

systemctl reload apache2

On change le propriétaire du dossier blacklist pour l'attribuer à l'utilisateur et au groupe proxy sur le dossier suivant pour permettre à Squid d'exécuter le filtrage et on recharge Squid.

chown -Rf proxy.proxy /var/lib/squidguard/db/blacklists

systemctl reload squid

4.5 Mise en place de l'authentification Active Directory

On voit que notre squid && squidGuard fonctionne bien. On peut aller plus loin en ajoutant l'authentification avec les utilisateurs de l'Active Directory. Pour cela, il faut modifier le fichier **/etc/squid/squid.conf** et ajouter les éléments suivants :

auth_param basic program /usr/lib/squid3/basic_ldap_auth -R -b "DC=infotools,DC=lan" -D "CN=administrateur,CN=Users,DC=infotools,DC=lan" -w "mot de passe" -f sAMAccountName=%s -h grp3_srvad.infotools.lan

auth_param basic children 5 startup=5 idle=1

auth_param basic realm Squid proxy-caching web server

auth_param basic credentialsttl 2 hours

Pour plus information sur squid && squidGuard :

https://alexandre-dosreis.me/articles/squid-sous-debian-10.x#VI-Filtrer-le-trafic-web-avec-SquidGuard

5- Mise en place d'un serveur de messagerie

5.1 Installations de Postfix

J'ai installé Postfix. Postfix est un serveur de messagerie modulaire, cela veut dire qu'il est composé de plusieurs petits programmes qui ont chacun un rôle particulier à effectuer. Cela permet d'apporter beaucoup plus de flexibilité et de sécurité car il suffit de toucher à la partie concernée du code pour modifier Postfix. De plus, chaque sous-programme peut être sécurisé individuellement.

Tout d'abord, on va installer les paquets nécessaires à la mise en place de ce serveur de messagerie :

```
root@messagerie:~# apt install postfix
```

On configure le programme de manière suivante :

Configuration générale de type : Internet Site

Nom du courrier (nom du domaine) : infotools.lan

Ensuite, on configure d'avantage postfix avec la commande suivante : **dpkg-reconfigure postfix.**

- ✓ Ajouter l'utilisateur de base de l'ordinateur
- ✓ Ajouter notre réseau 172.19.0.0/16
- ✓ On peut également limiter les protocoles à ipv4

Editer le fichier /etc/postix/main.cf :

```
myhostname = messagerie.infotools.lan
alias_maps = hash:/etc/aliases
alias_database = hash:/etc/aliases
myorigin = /etc/mailname
mydestination = messagerie.infotools.lan, infotools.lan, localhost, localhost.localdomain, localhost
relayhost =
mynetworks = 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128  172.19.0.0/16
mailbox_size_limit = 0
recipient_delimiter = +
inet_interfaces = all
inet_protocols = ipv4
home_mailbox= Maildir/
```

Redémarrer postfix :

```
root@messagerie:~# systemctl restart postfix
```

Puis, j'ai créé trois utilisateurs (toto,simon,felix) avec la commande adduser :

```
root@messagerie:~# adduser toto
Ajout de l'utilisateur « toto » ...
Ajout du nouveau groupe « toto » (1001) ...
Ajout du nouvel utilisateur « toto » (1001) avec le groupe « toto » ...
Création du répertoire personnel « /home/toto »...
Copie des fichiers depuis « /etc/skel »...
Nouveau mot de passe :
Retapez le nouveau mot de passe :
Les mots de passe ne correspondent pas.
passwd: Erreur de manipulation du jeton d'authentification
passwd: password unchanged
Essayer à nouveau ? [o/N]o
Nouveau mot de passe :
Retapez le nouveau mot de passe :
passwd: password updated successfully
Changing the user information for toto
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Cette information est-elle correcte ? [O/n]o
root@messagerie:~# _
```

On peut regarder dans le repertoire /home nos utilisateur.

```
root@messagerie:/home# ls
felix  grp3  simon  toto
```

On test envoyer un mail à partir d'une connexion locale en utilisant Telnet sur le port 25

```
root@messagerie:/home# telnet localhost 25
Trying ::1...
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 messagerie.infotools.lan ESMTP Postfix (Debian/GNU)
ehlo messagerie
250-messagerie.infotools.lan
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250-SMTPUTF8
250 CHUNKING
data
503 5.5.1 Error: need RCPT command
.
502 5.5.2 Error: command not recognized
quit
221 2.0.0 Bye
Connection closed by foreign host.
root@messagerie:/home# _
```

On définit la variable d'environnement pour utiliser Maildir

```
root@messagerie:/home# echo 'export MAIL=$HOME/Maildir/' > /etc/profile.d/mail.sh
```

Vérifier que le mail est bien arrivé dans le répertoire Maildir de l'utilisateur destinataire.

```
root@messagerie:/home/toto# tree
.
└── Maildir
    ├── cur
    ├── new
    │   └── 1646840167.V801I40b9aM138334.messagerie
    └── tmp

4 directories, 1 file
root@messagerie:/home/toto# _
```

Maintenant, on va installer le protocole IMAP et POP :

```
root@messagerie:/home/toto# apt install dovecot-imapd dovecot-pop3d
```

Par défaut dovecot recherche les mails dans /var/mail et non dans Maildir. On va faire en sorte qu'ils lisent les mails dans /home/user/Maildir.

Modifier la ligne mail_location dans le fichier /etc/dovecot/conf.d/10-mail.conf :

```
mail_location = maildir:~/Maildir
```

Redémarrer le service dovecot.

```
systemctl restart dovecot
```

5.2 Sécurisation du serveur de messagerie

Il faut créer ne autorité de certification locale et un certificat pour notre serveur de messagerie dans le dossier /etc/postfix/certs/ca et /etc/postfix/certs/mail. On va créer l'arborescence ci-contre :

```
|-- certs
|   |-- ca
|   |   |-- ca.crt
|   |   |-- ca.key
|   |   `-- ca.srl
|   `-- mail
|       |-- mail.crt
|       |-- mail.csr
|       |-- mail.ext
|       |-- mail.key
|       `-- mail.key.insecure
```

Création de l'autorité de certification dans le dossier ca

- ✓ Création de la clé privée de l'autorité de certification

```
openssl genrsa -des3 -out ca.key 2048
```

- ✓ Création du certificat de l'autorité de certification (FQDN : infotools.lan)

```
openssl req -new -x509 -days 3650 -key ca.key -out ca.crt_
```

Création de la clé privée du fichier d'extension et de la demande de certification du serveur de messagerie dans le dossier mail

- ✓ Création de la clé privée du serveur de messagerie

```
openssl genrsa -des3 -out mail.key 2048
```

- ✓ Création de la demande de certificat du serveur de messagerie (FQDN : infotools.lan)

```
openssl req -new -key mail.key -out mail.csr
```

- ✓ Création d'un fichier d'extensions afin d'ajouter les autres noms DNS et l'IP (mail.ext)

```
 GNU nano 3.2                                    mail.ext

authorityKeyIdentifier=keyid,issuer
basicConstraints = CA:FALSE
keyUsage = digitalSignature, nonRepudiation, keyEncipherment, dataEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = srvMessagerie.infotools.lan
DNS.2 = smtp.infotools.lan
DNS.3 = pop.infotools.lan
DNS.4 = imap.infotools.lan
IP.1 = 172.19.0.3
```

- ✓ Signature du certificat du serveur de messagerie par l'autorité de certification

```
root@srvMessagerie:/etc/postfix/certs/mail# openssl x509 -req -in mail.csr -out mail.crt -CA ../ca/c
a.crt -CAkey ../ca/ca.key -CAcreateserial -days 3650 -extfile mail.ext
```

- ✓ Création d'une clé non sécurisée pour le serveur de messagerie

```
openssl rsa -in mail.key -out mail.key.insecure
```

Ensuite on va modifier les fichiers main.cf, master.cfg et 10-ssl.conf pour prendre en compte le certificat et les clés (insecure) du serveur de messagerie.

/etc/postfix/main.cf :

Adapter les valeurs de *smtpd_tls_cert_file* et *smtpd_tls_key_file* avec le chemin vers le certificat et les clés (insecure) du serveur de messagerie

```
# TLS parameters
smtpd_tls_cert_file=/etc/postfix/certs/mail/mail.crt
smtpd_tls_key_file=/etc/postfix/certs/mail/mail.key.insecure
```

Ajouter smtpd_tls_CAfile avec comme valeur le chemin vers le certificat de l'autorité de certification

```
smtpd_tls_CAfile=/etc/postfix/certs/ca/ca.crt
```

/etc/dovecot/conf.d/10-ssl.conf

Modifier les valeurs *ssl_cert, ssl_key* (prendre la clé insecure et certificat)

```
# certificate, just make sure to update the domains in dovecot-openssl.cnf
ssl_cert = </etc/postfix/certs/mail/mail.crt
ssl_key = </etc/postfix/certs/mail/mail.key.insecure
```

Décommenter *ssl_ca* et affecter la bonne valeur

```
# followed by the matching CRL(s). (e.g.
ssl_ca = </etc/postfix/certs/ca/ca.crt
```

Décommenter *ssl_verify_client_cert* et mettre la valeur à *yes*

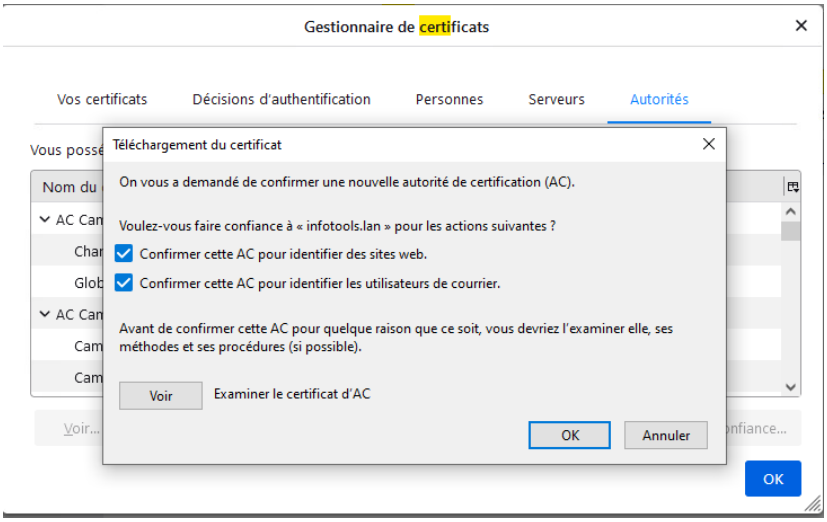
```
# PEM encoded trusted certificate
ssl_verify_client_cert=yes
```

Redémarrer les services postfix et dovecot

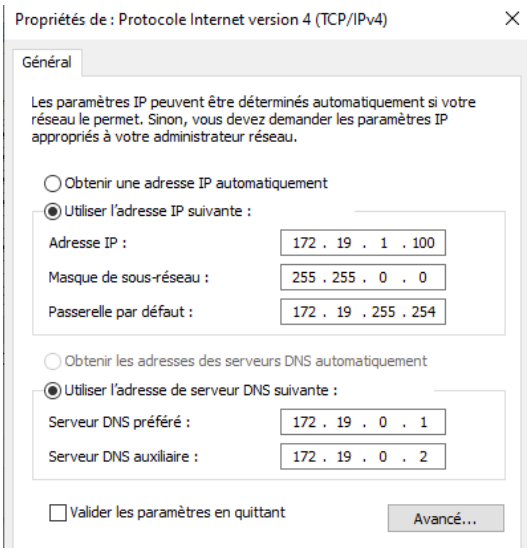
5.3 Configuration thunderbird

On transfère le certificat de l'autorité de certification **(ca.crt)** sur notre client windows via winscp**.**

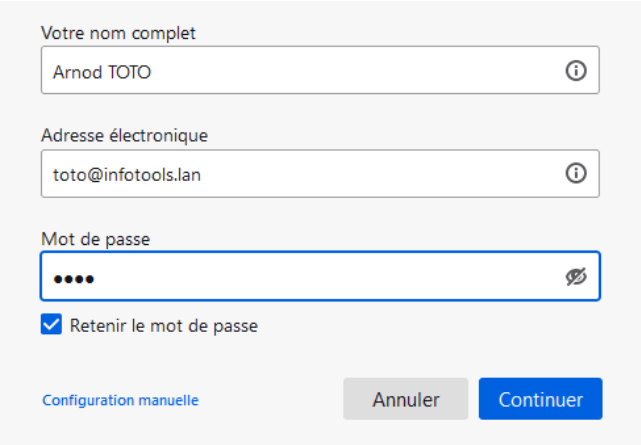
On installe thunderbird sur le client windows et ajoute le certificat de l'autorité de certification dans thunderbird (Burger>Preference>Chercher certificats).



Configuration de client :



On va ajouter nos utilisateurs :



On va créer maintenant un alias pour chaque utilisateur dans le fichier /etc/aliases:

```
# See man 5 aliases for format
postmaster:    root
simon.guigue:   simon
arnod.toto:     toto
felix.bruchot:  felix
root: toto
```

Pour prendre en compte les modifications, on exécute la commande suivante :

```
newaliases
```

Enfin, on va tester d'envoyer un mail. Et voici le résultat :

6-mise en place d'un serveur de sauvegarde

6.1 Installation de backuppc

Tout d'abord, on va changer le nom de la machine. On va nommer la machine « **SrvSauvegarde »** et mettre la bonne **IP.**

On installe les paquets nécessaires en exécutant la commande suivante :

```
apt install ssh rsync backuppc
```

On modifie le mot de passe du compte backuppc ave la commande suivante :

```
htpasswd /etc/backuppc/htpasswd backuppc
```

On modifie le fichier **/etc/backuppc/apache.conf** et commentez la ligne **Require local.**

On redémarre le service apache :

```
service apache2 restart
```

On modifie le fichier /etc/resolv.conf pour mettre le serveur dans le domaine

```
domain infotools.lan
search infotools.lan
nameserver 172.19.0.1
ameserver 172.19.0.2
```

6.2 Sauvegarde des données de SrvAD

Pour cela, il faut se connecter sur l'interface http://ip_SrvSauvegarde/backuppc avec l'utilisateur « backuppc » et le mot de passe « root »

Une fois connecté, on va modifier étape par étape ces éléments :

Etape 0 : Changement de la langue :

Menu **Server - Edit Config**
Onglet **CGI**, puis choisissez dans la zone **Language** : **fr**
Cliquez sur **Save**. Actualisez la page…

Etape 1 : Création de la machine SrvAd sur BackupPC

Cliquez sur le menu **Serveur - Modifier les machines**
Cliquez sur le bouton **Ajouter**.
Dans la zone **host**, saisissez **SrvAD**
Dans la zone **user**, saisissez **BackupPC**
Cliquez sur **Sauvegarder**

Etape 2 : Paramétrage de la sauvegarde des données de la machine SrvAd sur BackupPC

Dans la liste déroulable "**Choisissez un hôte…**", sélectionnez **SrvAD**.
Cliquez ensuite sur **Modifier la configuration**, puis sur **Xfer**.
Dans la zone "**xferMethod**", sélectionnez **smb**.
Dans la zone "**Paramètres de Smb**", Cliquez sur **Ajouter**.
Dans la zone vierge, saisissez "**Data**".
Cliquez sur **Détruire** en face du partage **C$**.
Dans la zone de saisie "**SmbShareUserName**", saisissez **BackupPC**
Dans la zone de saisie "**SmbSharePasswd**", saisissez **P@ssw0rd**
Cliquez sur **Sauvegarder**.

Etape 3 : Sauvegarde des données de la machine SrvAd sur BackupPC

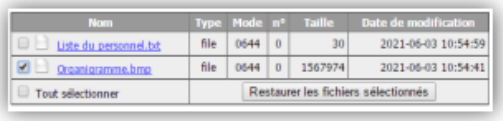
Cliquez en haut à gauche sur **srvad Accueil**
Cliquez sur le bouton **Démarrer la sauvegarde complète**.
Cliquez à nouveau sur le bouton **Démarrer la sauvegarde complète**.

Etape 4 : Restauration des données sur la machine SrvAd depuis BackupPC

Supprimez 1 fichier du dossier **C:\Data**

Sur l'interface de BackupPC, sélectionnez la sauvegarde en cliquant sur son numéro.

Le contenu s'affiche. Sélectionnez les fichiers que vous souhaitez restaurer et cliquez sur "**Restaurer les fichiers sélectionnés**"



6.3 Sauvegarde des données de SrvWeb

Il faut se connecter sur le serveur de sauvegarde avec l'utilisateur « root » et on va changer le mot de passe de l'utilisateur backuppc en « root ».

passwd backuppc

Pour autoriser l'utilisateur root à utiliser le service ssh, on va modifier le fichier **/etc/ssh/sshd_config** en remplaçant la ligne suivante **PermitRootLogin prohibit-password** par **PermitRootLogin yes.**

Ensuite, on se connecte avec l'utilisateur backuppc.

su backuppc

/bin/bash

Pour éviter une saisie de mot de passe, on va utiliser le système de clé publique. On crée une clé publique.

ssh-keygen -t rsa ; **#sans aucune saisie.**

Puis, on copie la clé publique vers notre serveur web.

ssh-copy-id -i /var/lib/backuppc/.ssh/id_rsa.pub root@SrvWeb

On essaie de se connecter sans mot de passe pour vérifier si ça fonctionne.

ssh root @SrvWeb

Il ne faut plus autoriser l'utilisateur root à utiliser le service ssh avec mot de passe. On modifie le fichier **/etc/ssh/sshd_config** en mettant **PermitRootLogin prohibit-password.**

On redémarre le service :

service ssh restart

exit

Dans l'interface de notre serveur de sauvegarde, on va ajouter notre serveur web.

Etape : Création de la machine SrvWeb sur BackupPC

Ouvrez un navigateur à l'adresse **http://10.X.0.10/backuppc**
Rappel : Login : **backuppc** et Mot de passe : **root**
Cliquez sur le menu **Serveur - Modifier les machines**
Cliquez sur le bouton **Ajouter**.
Dans la zone **host**, saisissez **SrvWeb**
Dans la zone **user**, saisissez **root**
Cliquez sur **Sauvegarder**

Etape : Paramétrage de la sauvegarde des données de la machine SrvWeb sur BackupPC

Dans la liste déroulable "**Choisissez un hôte...**", sélectionnez **SrvWeb**.
Cliquez ensuite sur **Modifier la configuration**, puis sur **Xfer**.
Dans la zone "**xferMethod**", sélectionnez **rsync**.
Dans la zone " **RsyncShareName** ", Cliquez sur **Ajouter**.
Dans la zone vierge, saisissez "**/var/www/html** ".
Cliquez sur **Détruire** en face du dossier "**/**".
Cliquez sur **Sauvegarder**.

Etape : Sauvegarde des données de la machine SrvWew sur BackupPC

Cliquez en haut à gauche sur **srvweb Accueil**
Cliquez sur le bouton **Démarrer la sauvegarde complète**.
Cliquez à nouveau sur le bouton **Démarrer la sauvegarde complète**.
Comme pour le serveur **SrvAD**, la sauvegarde s'effectue. Vous obtiendrez ceci :

Sauvegarde n°	Type	Fusionnée	Niveau	Date de démarrage	Durée (min)	Âge (jours)	Chemin d'accès de la sauvegarde sur le serveur
0	complète	oui	0	2021-06-03 13:44	0.0	0.0	/var/lib/backuppc/pc/srvweb/0

En cliquant su "0", vous pouvez consulter la sauvegarde.

Pour la restauration, supprimez le fichier index.html sur **SrvWeb** et restaurez-le.