PR02 : SIO SISR – Mise en place d'une passerelle et d'un Firewall sous Debian 11

Sommaires

Contexte :

- Notre centre de formation regroupant plusieurs enseignes dont MBWay et DigitalSchool, met à disposition des élèves un serveur Web hébergeant un intranet pour chacune d'elle : il s'agit d'un serveur web mutualisé.

- Dans l'architecture initiale, les sites web de chaque enseigne étaient hébergés sur un serveur dans le LAN Administratif.

- Suite à quelques tentatives d'intrusion dans les serveurs locaux du réseau administratif, il a été décidé de sécuriser celui-ci en le limitant strictement aux employés.

- Dans le cadre d'un stage, vous avez été chargé par votre centre de formation de mettre en place une maquette, au moindre coût, pour montrer la faisabilité de la solution.

Objectifs :

1) Maquetter le nouveau réseau et filtrer les flux.

2) Créer un sous-réseau nommé DMZ pour héberger les services partagés par le personnel et les stagiaires

3) A termes, ce réseau DMZ devrait être accessible depuis Internet

4) Le serveur Web héberge un site pour chaque établissement. Pour sécuriser les transactions les sites ne doivent être accessibles qu'en https soit https://www.mbway.lan ou https://www.digitalschool.lan. Les sites web sont accessibles à TOUS.

5) Mettre en place un service DNS

6) Mettre en place des règles de pare-feu afin de sécuriser l'installation ainsi qu'un protocole SSH.

7) Effectuer une démonstration montrant la sécurité et la fiabilité de notre projet, répondant au cahier des charges.

Cahier des charges :

- Permettre l'accès au serveur Web dans la DMZ pour tous, LAN Administratif et Formation. Le serveur Web hébergera aussi les service DNS.

- Permettre l'accès à internet pour tous en utilisant le Routeur Debian (R) comme passerelle. Ce routeur fera office de Firewall pour filtrer les accès à la DMZ.

- Permettre l'accès au service FTP à un seul poste, celui de l'administrateur situé dans

le LAN Administratif

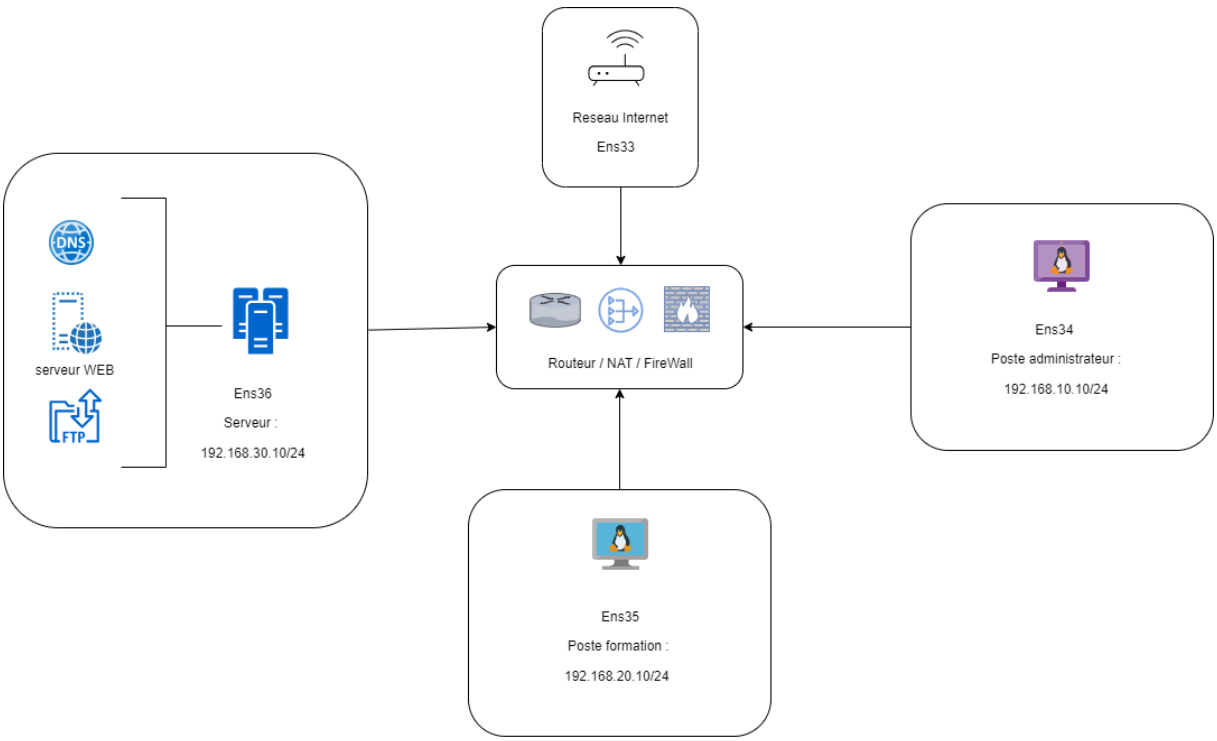
- Les postes de l'espace Formation ne pourront pas accéder au service FTP.
- Permettre un accès SSH à un seul poste, celui de l'administrateur situé dans le LAN Administratif
- Les autres périphériques du réseau Administratif et ceux du réseau Formation ne pourront pas accéder en SSH au serveur Web.
- Mettre en place les tests de validation des règles ci-dessus.
- Fournir une documentation expliquant et validant chacune des demandes du cahier des charges

Solution :

Pour répondre au problème du campus, j'ai donc mis en place une DMZ (Demilitarized Zone) hébergeant mon serveur WEB (Apache2) avec les 2 intranets, un serveur Proftpd pour le transfert de fichiers et une application serveur SSH avec OpenSSH.

J'ai également mis en place un routeur Debian hébergeant mon service DNS et mes règles de Firewall, ainsi que 2 machines clients. Une machine linux pour le réseau formation et une machine Linux pour l'administrateur.

Schéma ASI :



Prérequis :

Tout d'abord nous allons devoir installer puis paramétrer une machine virtuelle qui nous servira de Serveur WEB (Apache2), DNS (bind9), FTP (Proftpd) et une application serveur SSH avec OpenSSH.

Ensuite on va configurer deux VM sous Debian : une pour le réseau administrateur et une autre pour le réseau formation.

Les cartes réseau seront configurer en mode « Accès par pont »

Nous utiliserons et configurerons un routeur sous Debian 11 avec 4 interfaces réseau, qui nous servira de pare-feu grâce aux règles iptables que nous allons déployer.

Configuration du routeur et du Nat :

On va d'abord aller dans le fichier /etc/network/interfaces afin de paramétrer notre carte réseaux avec la commande sudo nano /etc/network/interfaces.

On redémarrera les interfaces avec la commande sudo systemctl restart networking.service pour que les changements prennent effet.

Il ne faut pas oublier d'activer l'accès par pont.

On aura pour le routeur les interfaces ci- dessous :

```
auto ens33
iface ens33 inet dhcp
post-up iptables-restore < /etc/iptables_rules.save


auto ens34
iface ens34  inet static
        address 192.168.10.254
        netmask 255.255.255.0

auto ens35
iface ens35 inet static
        address 192.168.20.254
        netmask 255.255.255.0

auto ens36
iface ens36 inet static
        address 192.168.30.254
        netmask 255.255.255.0
root@debianRouter:~#
```

Ens33 sert à avoir l'accès internet.

Ens34 sera la carte pour le lan Administrateur.

Ens35 sera la carte pour le lan Formation.

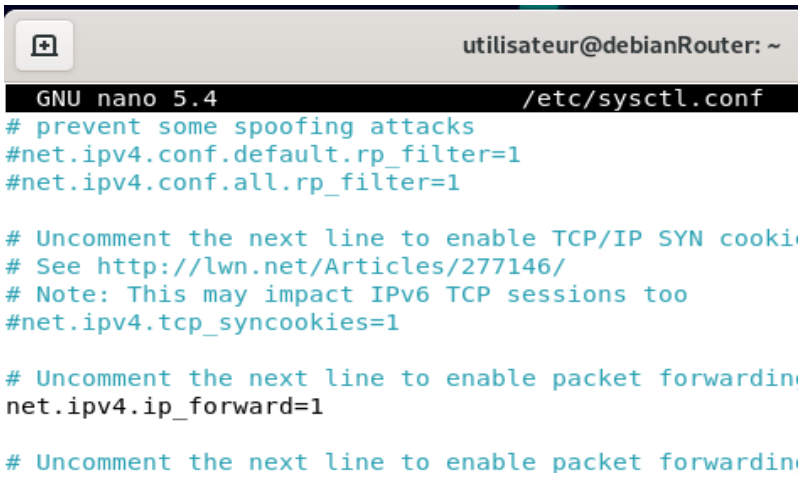
Ens36 sera la carte pour la DMZ.

La ligne post-up nous sert pour le Nat et les règles iptables, nous le reverrons plus loin dans le dossier.

Maintenant que les interfaces réseaux sont configurer on va pouvoir configurer notre Vm en mode routeur et installer le NAT.

Mise en place du mode routeur :

- On va aller dans le fichier /etc/sysctl.conf et on décommente la ligne net.ipv4.ip_forward=1



On configure ensuite le Nat avec la commande :

- iptables -t nat -A POSTROUTING -o ens33 -j MASQUERADE avec ens33 qui est la carte qui sort sur internet.

On vérifie que la commande et bonne avec iptables -L -t nat et on vérifie que l'on a notre règle dans la chaine postrouting

```
utilisateur@debianRouter:/$ sudo iptables -L -t nat
Chain PREROUTING (policy ACCEPT)
target     prot opt source               destination

Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination

Chain POSTROUTING (policy ACCEPT)
target     prot opt source               destination
MASQUERADE  all  --  anywhere             anywhere
utilisateur@debianRouter:/$
```

On automatise le montage des règles au démarrage :

- iptables-save > /etc/iptables_rules.save

Et on rajoute cette ligne dans notre fichier networking :

```
auto ens33
iface ens33 inet dhcp
post-up iptables-restore < /etc/iptables_rules.save
```

Configuration du serveur et mise en place des services :

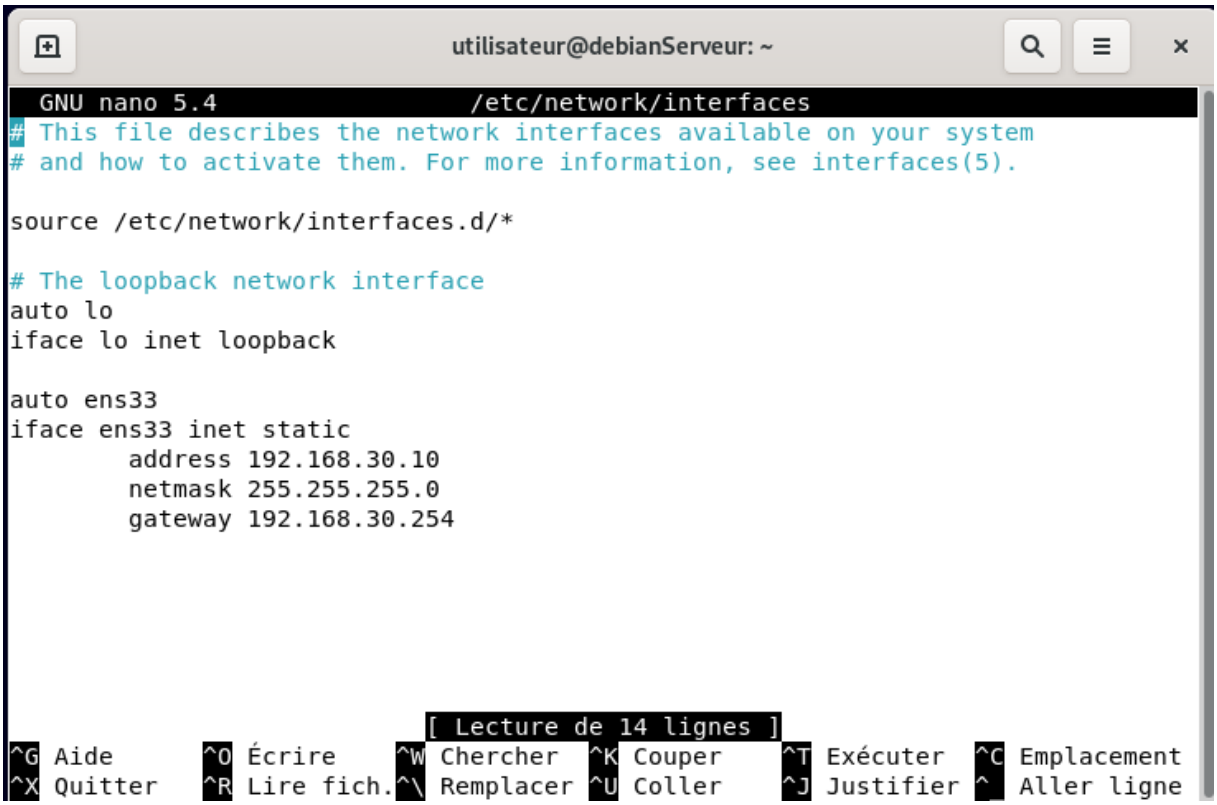
Configuration réseaux :

On va d'abord aller dans le fichier /etc/network/interfaces afin de paramétrer notre carte réseaux avec la commande sudo nano /etc/network/interfaces.

On redémarrera les interfaces avec la commande sudo systemctl restart networking.service pour que les changements prennent effet.

Il ne faut pas oublier d'activer l'accès par pont.

Pour notre serveur on lui définit comme adresse IP : 192.168.30.10



Maintenant que l'interface réseau est configurée on va pouvoir installer les services nécessaires.

Configuration du serveur FTP :

Nous devons installer au préalable ProFTPD sur notre VM SERVEUR Debian11

Installation de Proftpd :

- sudo apt-get update

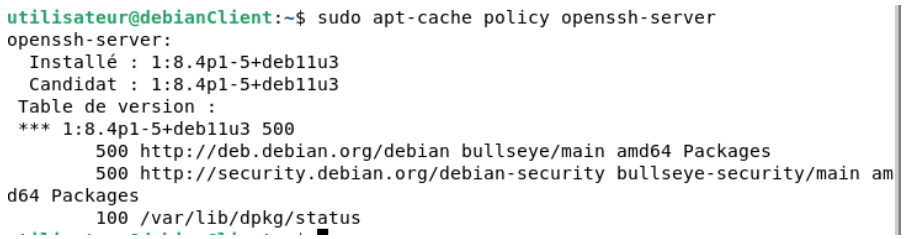
- sudo apt install proftpd

Une fois installé nous pouvons choisir de créer des utilisateurs mais nous allons nous connecter en mode utilisateur pour nos tests.

FTP n'est pas un protocole sécurisé. Pour éviter la transmission d'informations en clair, il est nécessaire de crypter les données en transit. Nous utiliserons donc principalement le protocole SFTP.

Pour cela, nous devrons installer OpenSSH sur les serveurs ainsi que sur les postes clients via la commande :

- sudo apt install openssh-server



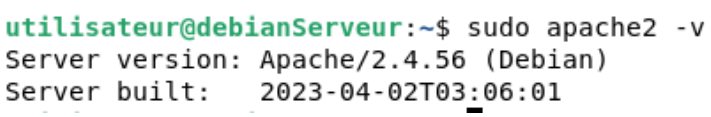
Configuration du Serveur Web :

On va devoir configurer Apache 2 avec nos 2 intranets et mettre en place le protocole HTTPS

Dans un premier temps faire sudo apt-get update afin d'avoir les dernières versions disponibles (il faut le faire avant toutes installations sous Debian)

Installation d'Apache2 :

- sudo apt install Apache2



On constate donc qu'Apache2 est correctement installé.

Configuration des deux intranets :

On va créer deux dossiers MBWay et DigitalSchool :

Aller dans le dossier cd /var/www/html est créé deux dossiers avec la commande mkdir :

```
utilisateur@debianServeur:~$ cd /var/www/html/
utilisateur@debianServeur:/var/www/html$ ls -l
total 20
drwxr-xr-x 2 root root  4096 15 avril 12:46 digitalschool
-rw-r--r-- 1 root root 10701  7 avril 21:19 index.html
drwxr-xr-x 2 root root  4096 13 avril 18:18 mbway
utilisateur@debianServeur:/var/www/html$
```

Pour chaque dossier nous allons créer un fichier index.html que nous allons ensuite configurer :

```
utilisateur@debianServeur:/var/www/html$ cd mbway
utilisateur@debianServeur:/var/www/html/mbway$ ls -l
total 4
-rw-r--r-- 1 root root 123 13 avril 18:18 index.html
utilisateur@debianServeur:/var/www/html/mbway$ cd ..
utilisateur@debianServeur:/var/www/html$ cd digitalschool
utilisateur@debianServeur:/var/www/html/digitalschool$ ls -l
total 8
-rw-r--r-- 1 root root 131 14 avril 11:13 index.html
```

Fichier index.html pour DigitalSchool :

```
utilisateur@debianServeur:/var/www/html/digitalschool$ cat index.html
<!DOCTYPE html>
<html>
<body>
        <h1>Bienvenue sur le site de Digitalschool</h1>
        <p>Ceci est la page d'acceuil.</p>
</body>
</html>
```

On va créer le même type pour MBWay.

Nous avons donc créé un dossier où nous importerons nos fichiers html, PHP, JavaScript etc… respectifs à chaque site. Ici nous n'avons déposé qu'un « index.html » pour le moment.

Configuration des Virtual hosts :

Grâce à l'étape précédente nous avons créé nos intranets.

De ce fait pour y accéder nous sommes obligés d'écrire : http://192.168.30.5/mbway/ pour accéder au site d'MBWay par exemple.

Pour corriger cela nous allons donc configurer des Virtual Host pour accéder à nos sites depuis l'adresse http://www.mbway.lan

On va aller dans le dossier /etc/apache2/sites-available/

On a un fichier 000-default.conf avec une configuration par default que l'on va copier (commande cd) et créer deux fichiers un pour MBWay et un autre pour DigitalSchool.

```
utilisateur@debianServeur:~$ cd /etc/apache2/sites-available/
utilisateur@debianServeur:/etc/apache2/sites-available$ ls -l
total 36
-rw-r--r-- 1 root root 1332  2 avril  2023 000-default.conf
-rw-r--r-- 1 root root 6387 14 avril 19:53 default-ssl.conf
-rw-r--r-- 1 root root 1351 13 avril 15:31 digitalschool.conf
-rw-r--r-- 1 root root 6390 16 avril 09:20 digitalschool-ssl.conf
-rw-r--r-- 1 root root 1636 14 avril 19:38 mbway.conf
-rw-r--r-- 1 root root 6374 16 avril 09:19 mbway-ssl.conf
utilisateur@debianServeur:/etc/apache2/sites-available$
```

On va ensuite les configurer :

```
utilisateur@debianServeur:/etc/apache2/sites-available$ cat mbway.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port that
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) this
        # value is not decisive as it is used as a last resort host regardless.
        # However, you must set it for any further virtual host explicitly.
        ServerName www.mbway.lan

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/mbway
```

```
utilisateur@debianServeur:/etc/apache2/sites-available$ cat digitalschool.conf
<VirtualHost *:80>
        # The ServerName directive sets the request scheme, hostname and port
        # the server uses to identify itself. This is used when creating
        # redirection URLs. In the context of virtual hosts, the ServerName
        # specifies what hostname must appear in the request's Host: header to
        # match this virtual host. For the default virtual host (this file) th
        # value is not decisive as it is used as a last resort host regardless
        # However, you must set it for any further virtual host explicitly.
        ServerName www.digitalschool.lan

        ServerAdmin webmaster@localhost
        DocumentRoot /var/www/html/digitalschool
```

On modifie ServerName par le nom de nos serveurs : www.mbway.lan et www.digitalschool.lan.

On lui indique la route avec DocumentRoot : /var/www/html/mbway et /var/www/html/digitalschool .

On va activer nos sites avec la commande a2ensite + le nom de notre site

Maintenant si on ajoute www.mbway.lan en alias à notre serveur dans le fichier /etc/hosts de notre client, vous pouvez voir la page d'accueil de notre site dans notre navigateur.

On vérifie que nos sites fonctionnent :

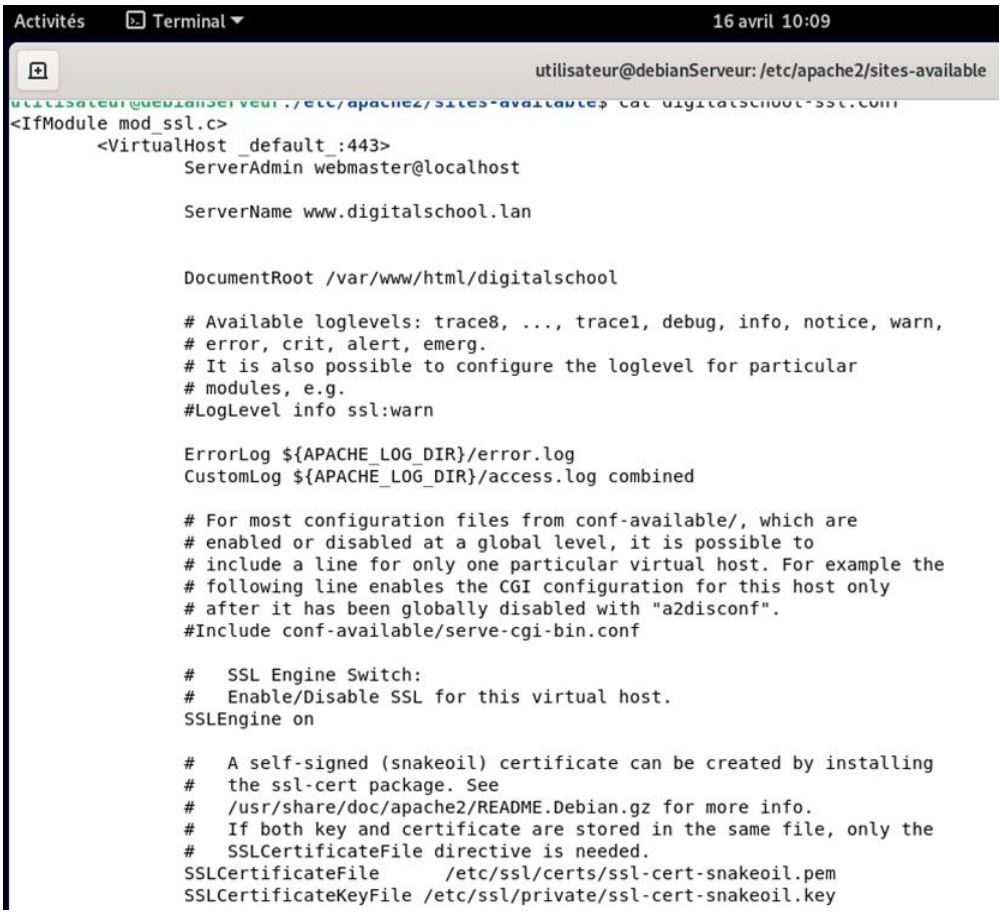
Mise en place du protocole HTTPS :

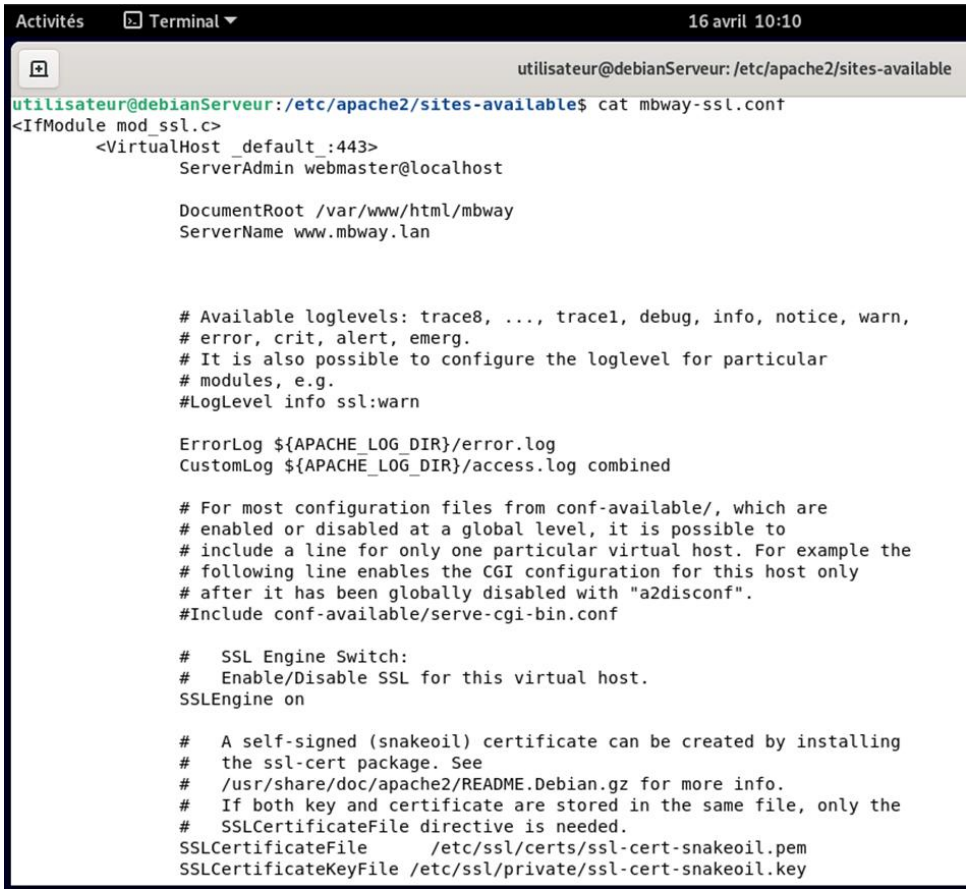
La connexion à nos sites intranets ne se fait de base qu'en HTTP ce qui n'est absolument pas sécurisé. Pour éviter de faire circuler des informations en clair, nous devons crypter les données qui transitent par le biais du protocole HTTPS.

On a un fichier avec une configuration déjà présente sous Debian :

```
utilisateur@debianServeur:/etc/apache2/sites-available$ ls -l
total 36
-rw-r--r-- 1 root root 1332  2 avril  2023 000-default.conf
-rw-r--r-- 1 root root 6387 14 avril 19:53 default-ssl.conf
-rw-r--r-- 1 root root 1351 13 avril 15:31 digitalschool.conf
-rw-r--r-- 1 root root 6390 16 avril 09:20 digitalschool-ssl.conf
-rw-r--r-- 1 root root 1636 14 avril 19:38 mbway.conf
-rw-r--r-- 1 root root 6374 16 avril 09:19 mbway-ssl.conf
utilisateur@debianServeur:/etc/apache2/sites-available$
```

On va donc copier le fichier default-ssl et créer deux fichiers pour chaque site puis les configurer :


Activités
Terminal
16 avril 10:09
utilisateur@debianServeur: /etc/apache2/sites-available
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin webmaster@localhost
ServerName www.digitalschool.lan
DocumentRoot /var/www/html/digitalschool
Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
error, crit, alert, emerg.
It is also possible to configure the loglevel for particular
modules, e.g.
#LogLevel info ssl:warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
For most configuration files from conf-available/, which are
enabled or disabled at a global level, it is possible to
include a line for only one particular virtual host. For example the
following line enables the CGI configuration for this host only
after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
SSL Engine Switch:
Enable/Disable SSL for this virtual host.
SSLEngine on
A self-signed (snakeoil) certificate can be created by installing
the ssl-cert package. See
/usr/share/doc/apache2/README.Debian.gz for more info.
If both key and certificate are stored in the same file, only the
SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key


Activités
Terminal
16 avril 10:10
utilisateur@debianServeur: /etc/apache2/sites-available
utilisateur@debianServeur:/etc/apache2/sites-available$ cat mbway-ssl.conf
<IfModule mod_ssl.c>
<VirtualHost _default_:443>
ServerAdmin webmaster@localhost
DocumentRoot /var/www/html/mbway
ServerName www.mbway.lan
Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
error, crit, alert, emerg.
It is also possible to configure the loglevel for particular
modules, e.g.
#LogLevel info ssl:warn
ErrorLog ${APACHE_LOG_DIR}/error.log
CustomLog ${APACHE_LOG_DIR}/access.log combined
For most configuration files from conf-available/, which are
enabled or disabled at a global level, it is possible to
include a line for only one particular virtual host. For example the
following line enables the CGI configuration for this host only
after it has been globally disabled with "a2disconf".
#Include conf-available/serve-cgi-bin.conf
SSL Engine Switch:
Enable/Disable SSL for this virtual host.
SSLEngine on
A self-signed (snakeoil) certificate can be created by installing
the ssl-cert package. See
/usr/share/doc/apache2/README.Debian.gz for more info.
If both key and certificate are stored in the same file, only the
SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

On donne le nom de notre serveur :

ServerName www.mbway.lan et ServerName www.digitalschool.lan ainsi que la route avec DocumentRoot /var/www/html/mbway et /var/www/html/digitalschool .

On va activer a2enmod ssl puis a2ensite default-ssl

On va activer nos sites avec la commande a2ensite + le nom du virtual Host

Puis redemarrer apache2 : service apache2 reload

La recommandation est de créer/acheter un certificat pour chaque site plutôt que d'utiliser la configuration de base d'Apache2

C'est donc ce que nous allons faire :

Dans un premier temps nous allons verifier que le module SSL est bien activée :

- sudo a2enmod ssl

Nous allons ensuite générer nos clés privées et des CSR(certificate signing request) :
Pour Mbway : openssl req -newkey rsa:2048 -nodes -keyout /etc/ssl/private/mbway.key -out /etc/ssl/certs/mbway.csr

Et pour Digitalschool : openssl req -newkey rsa:2048 -nodes -keyout /etc/ssl/private/digitalschool.key -out /etc/ssl/certs/ digitalschool.csr

Nous devrons répondre à une suite de question :

```
Mot de passe :
root@debian11:/home/utilisateur# openssl req -newkey rsa:2048 -nodes -keyout /etc/ssl/private/mbway.key -out /etc/ssl/certs/mbway.csr
Generating a RSA private key
.....+++++
.......+++++
writing new private key to '/etc/ssl/private/mbway.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:fr
State or Province Name (full name) [Some-State]:paris
Locality Name (eg, city) []:paris
Organization Name (eg, company) [Internet Widgits Pty Ltd]:thomasit
Organizational Unit Name (eg, section) []:it
Common Name (e.g. server FQDN or YOUR name) []:thomas
Email Address []:thomas.fr

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:root
An optional company name []:
root@debian11:/home/utilisateur#
```

Nous allons enfin passer a la création de nos certificats auto-signés :

Pour Mbway : sudo openssl req x509 -req -days 365 -in /etc/ssl/certs/mbway.csr -signkey

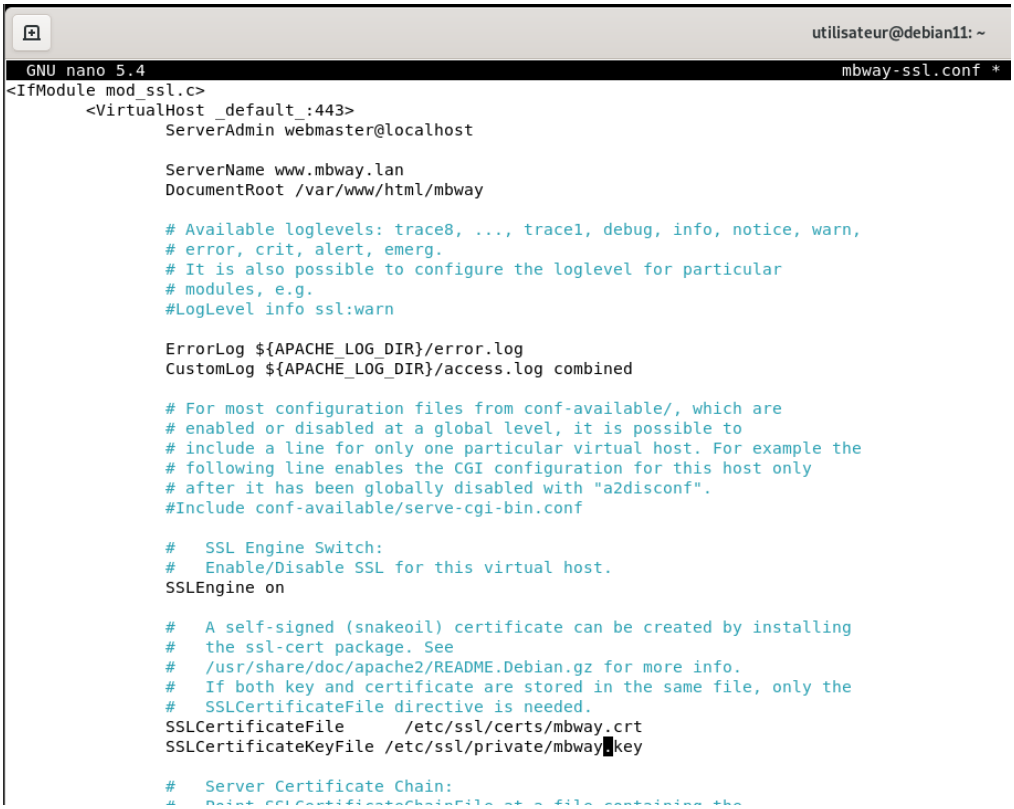
/etc/ssl/private/mbway.key -out /etc/ssl/certs/mbway.crt

Et pour digitalschool : sudo openssl req x509 -req -days 365 -in /etc/ssl/certs/digitalschool.csr -signkey /etc/ssl/private/digitalschool.key -out /etc/ssl/certs/digitalschool.crt

Nous devons ensuite modifier les lignes :

- SSLCertificateFile
- SSLCertificateKeyFile

Pour les sites Mbway et Digitaschool dans mbway-ssl.conf et digitalschool-ssl.conf



```
utilisateur@debian11: ~
 GNU nano 5.4                                    digitalschool-ssl.conf
<IfModule mod_ssl.c>
        <VirtualHost _default_:443>
                ServerAdmin webmaster@localhost

                ServerName www.digitalschool.lan
                DocumentRoot /var/www/html/digitalschool

                # Available loglevels: trace8, ..., trace1, debug, info, notice, warn,
                # error, crit, alert, emerg.
                # It is also possible to configure the loglevel for particular
                # modules, e.g.
                #LogLevel info ssl:warn

                ErrorLog ${APACHE_LOG_DIR}/error.log
                CustomLog ${APACHE_LOG_DIR}/access.log combined

                # For most configuration files from conf-available/, which are
                # enabled or disabled at a global level, it is possible to
                # include a line for only one particular virtual host. For example the
                # following line enables the CGI configuration for this host only
                # after it has been globally disabled with "a2disconf".
                #Include conf-available/serve-cgi-bin.conf

                #   SSL Engine Switch:
                #   Enable/Disable SSL for this virtual host.
                SSLEngine on

                #   A self-signed (snakeoil) certificate can be created by installing
                #   the ssl-cert package. See
                #   /usr/share/doc/apache2/README.Debian.gz for more info.
                #   If both key and certificate are stored in the same file, only the
                #   SSLCertificateFile directive is needed.
                SSLCertificateFile      /etc/ssl/certs/digitalschool.crt
                SSLCertificateKeyFile /etc/ssl/private/digitalschool.key

                #   Server Certificate Chain:
```

On va activer nos sites avec la commande a2ensite + le nom du virtual Host

Puis redemarrer apache2 : sudo systemctl restart apache2

Maintenant si on ajoute www.mbway.lan en alias à notre serveur dans le fichier /etc/hosts de notre client, vous pouvez voir la page d'accueil de notre site dans notre navigateur :

Nous arrivons donc bien a nous connecter en https !

Configuration du DNS :

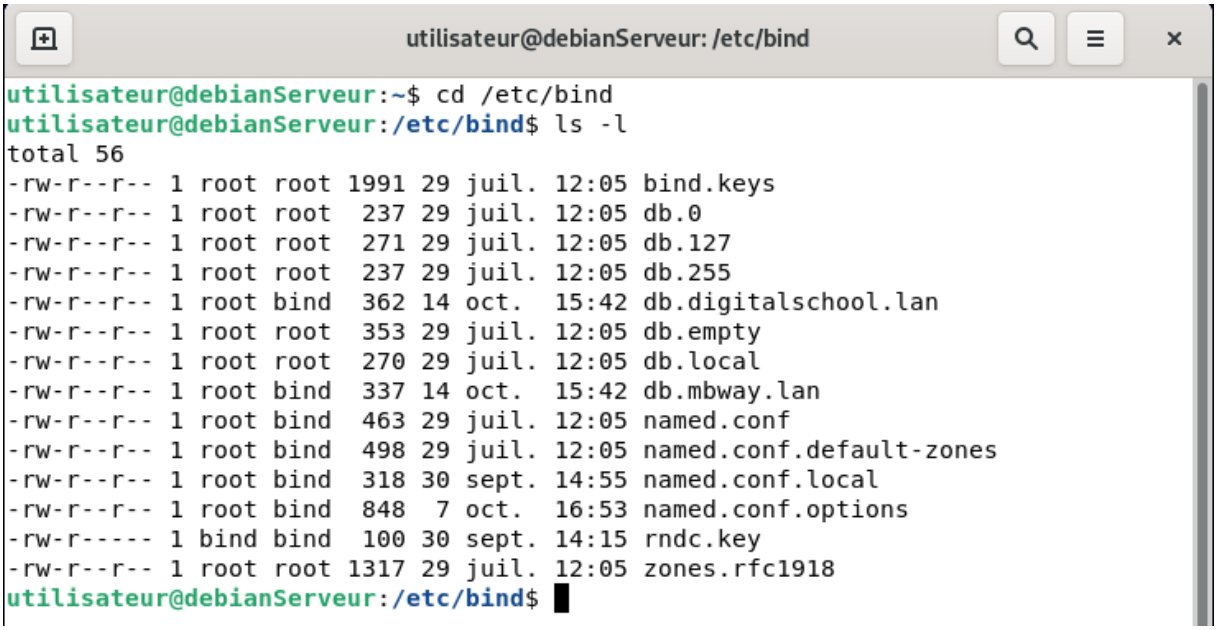
On commence par faire apt-get upgrade

Installation de BIND9 :

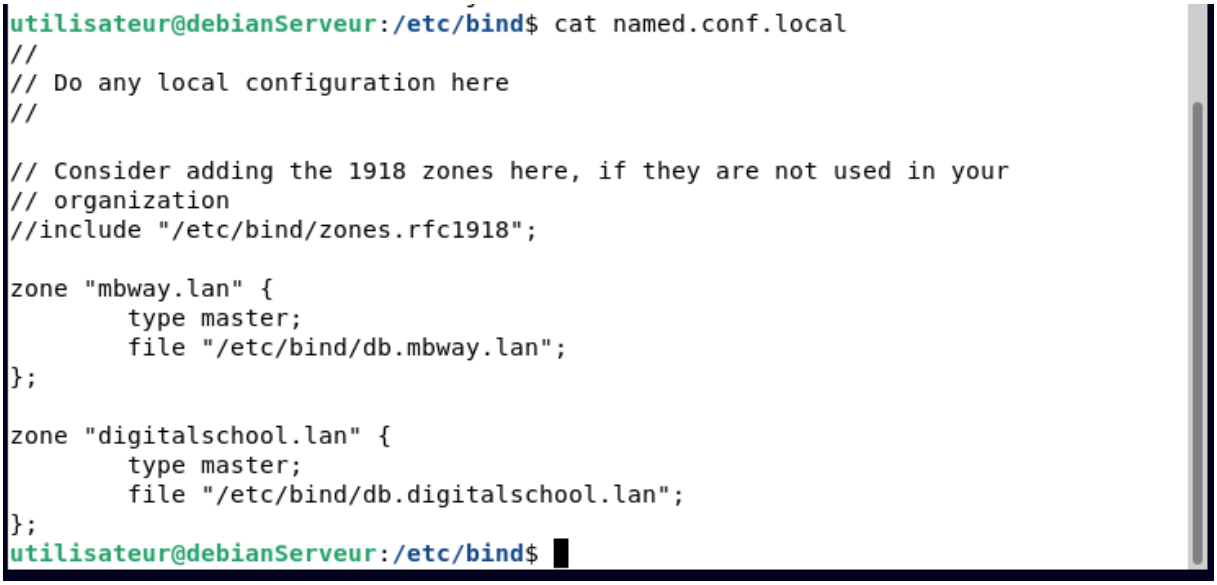
On fait ensuite sudo apt install Bind9 puis sudo apt install bind9utils.

Déclaration et création des zones :

On va commencer pas déclarer les zones dans /etc/bind



On va aller dans le fichiers named.conf.local et le configurer :



On le configure en mode master et ont créé deux zones pour chacun de nos sites et on respecte la convention db.nomDomaine pour le déclarer.

On va ensuite créer nos fichiers de zones :

```
  GNU nano 5.4                    db.mbway.lan
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     debianServer.mbway.lan. root.debianServer.mbway.lan. (
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      debianServer.mbway.lan.
debianServer    IN      A       192.168.30.10
www     IN      A       192.168.30.10
```

```
                        [ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^\ Remplacer ^U Coller    ^J Justifier ^_ Aller ligne
```

```
  GNU nano 5.4                 db.digitalschool.lan
;
; BIND data file for local loopback interface
;
$TTL    604800
@       IN      SOA     debianServer.digitalschool.lan. root.debianServer.digit>
                              2         ; Serial
                         604800         ; Refresh
                          86400         ; Retry
                        2419200         ; Expire
                         604800 )       ; Negative Cache TTL
;
@       IN      NS      debianServer.digitalschool.lan.
debianServer    IN      A       192.168.30.10
www     IN      A       192.168.30.10
```

```
                        [ Lecture de 14 lignes ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement
^X Quitter   ^R Lire fich.^\ Remplacer ^U Coller    ^J Justifier ^_ Aller ligne
```

On lui donne les bons noms et adresse IP : debianRouteur.mbway.lan. et debianRouteur.digitalschool.lan.

On ajoute WWW pour avoir le www.mbway.lan et on voit également que l'on a un enregistrement de type A.

On va également configurer le fichier named.conf.options avec le DNS de Google.

Les forwarders sont d'autres serveurs DNS vers lesquels le serveur DNS local peut envoyer les requêtes qu'il ne peut pas résoudre.

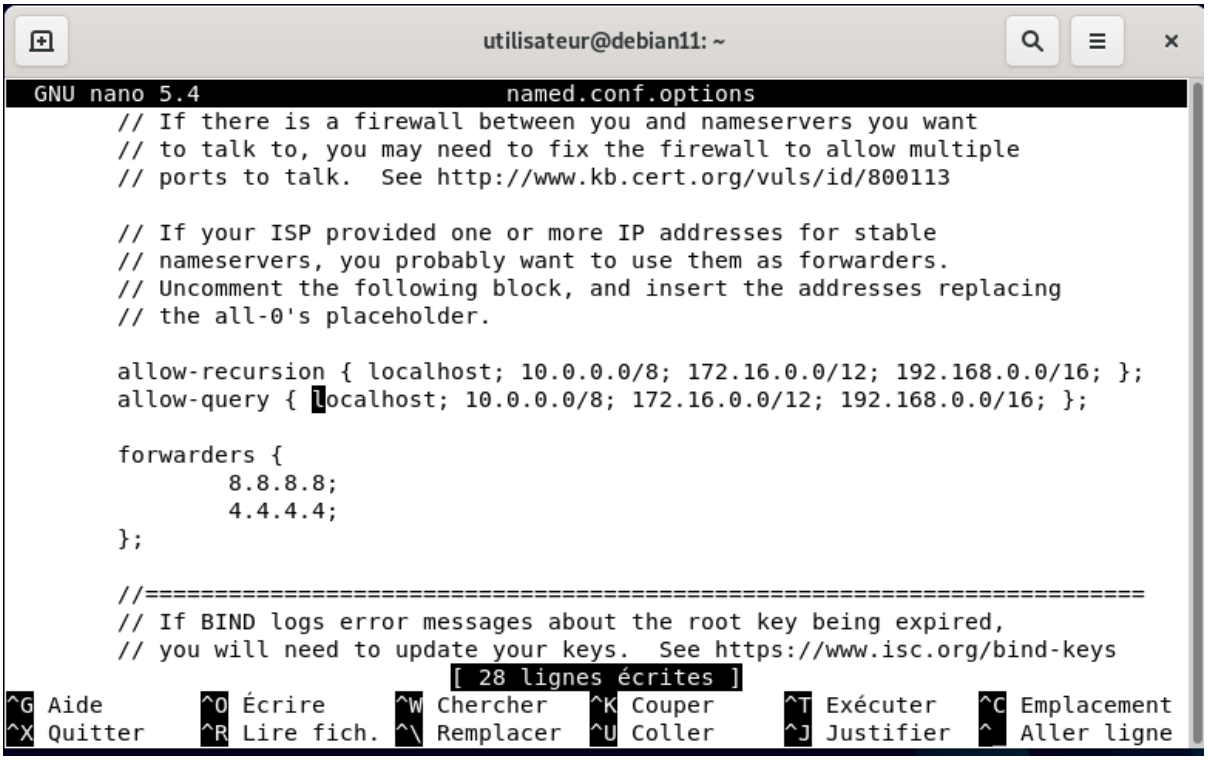
On doit aussi configurer Bind9 pour accepter toutes les requêtes provenant d'adresses privées définies par les standards RFC1918 (qui inclut les réseaux LAN).

Voici les plages d'adresses locales selon RFC1918 :

- 10.0.0.0/8 : Pour les réseaux de classe A.
- 172.16.0.0/12 : Pour les réseaux de classe B.
- 192.168.0.0/16 : Pour les réseaux de classe C.

Dans notre fichier named.conf.options, remplacez la directive allow-recursion par une liste incluant ces plages avec :

- allow-query : qui contrôle qui peut poser n'importe quelle question DNS (locale ou externe).
- allow-recursion : qui contrôle qui peut poser des questions récursives (requêtes nécessitant que le serveur DNS interroge d'autres serveurs).



On redémarre Bind9 afin que les modifications prennent effet : sudo systemctl restart bind9

Dans nos machines Debian on va devoir leur indiquer le nouveau DNS dans le fichier

Nano /etc/resolv.conf :

Nameserver : 192.168.30.10

On peut vérifier notre configuration grâce à la commande nslookup :

```
utilisateur@debianClient2:~$ nslookup www.mbway.lan
Server:         192.168.30.10
Address:        192.168.30.10#53

Name:   www.mbway.lan
Address: 192.168.30.10
```

On peut donc maintenant taper https://www.mbway.lan/

On teste la connexion :

Configuration réseaux :

On va d'abord aller dans le fichier /etc/network/interfaces afin de paramétrer nos cartes réseaux avec la commande sudo nano /etc/network/interfaces.

On redémarrera les interfaces avec la commande sudo systemctl restart networking.service pour que les changements prennent effet.

Il ne faut pas oublier d'activer l'accès par pont.

Pour le poste Administrateur :

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet static
        address 192.168.10.10
        netmask 255.255.255.0
        gateway 192.168.10.254
```

Pour le poste Formation :

```
source /etc/network/interfaces.d/*

# The loopback network interface
auto lo
iface lo inet loopback

auto ens33
iface ens33 inet static
        address 192.168.20.10
        netmask 255.255.255.0
        gateway 192.168.20.254
```

Filtrage du trafic par Iptables :

Réseau administrateur : ens34 192.168.10.0/24 IP Poste Administrateur : 192.168.10.10/24

Réseau formation : ens35 192.168.20.0/24 IP Poste Formateur192.168.20.50/24

Réseau serveur : ens36 192.168.30.0/24 IP Serveur 192.168.30.10/24

Réseau internet : ens34 DHCP

Nous allons utiliser iptables afin de créer des règles de filtrage est sécurisée notre connexion :

Nous allons regarder si on a déjà des règles en place :

```
root@debianRouter:~# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
root@debianRouter:~#
```

On va bloquer le trafic qui traverse le routeur ainsi que le trafic entrant. Mais on ne bloque pas le trafic sortant.

On va autoriser le retour pour les connexions déjà établie :

- iptables -A FORWARD -m state --state ESTABLISHED -j ACCEPT
- iptables -A INPUT -m state --state ESTABLISHED -j ACCEPT
- iptables -A OUTPUT -m state --state ESTABLISHED -j ACCEPT

Puis on va établir les règles de filtrages en suivent le cahier des charges suivant :

- ***Permettre l'accès au serveur Web dans la DMZ pour tous, LAN Administratif et Formation :***

Nous devons d'abord créer la règle de filtrage autorisant la connexion au DNS :

Autoriser le trafic DNS (UDP) depuis le Lan Formation et Lan Administration vers le serveur 192.168.30.10.

- iptables -A FORWARD -p udp --dport 53 -s 192.168.10.0/24 -d 192.168.30.10 -j ACCEPT
- iptables -A FORWARD -p udp --dport 53 -s 192.168.20.0/24 -d 192.168.30.10 -j ACCEPT

Ensuite :

- *iptables -A FORWARD -s 192.168.10.0/24 -p tcp --dport 80:443 -d 192.168.30.10 -j ACCEPT*
- *iptables -A FORWARD -s 192.168.20.0/24 -p tcp --dport 80:443 -d 192.168.30.10 -j ACCEPT*

- ***Permettre l'accès à internet pour tous en utilisant le Routeur Debian (R) comme passerelle.:***

- iptables -A FORWARD -s 192.168.10.0/24 -p tcp --dport 80 -j ACCEPT
- iptables -A FORWARD -s 192.168.20.0/24 -p tcp --dport 443 -j ACCEPT

- ***Permettre un accès SSH à un seul poste, celui de l'administrateur situé dans le LAN Administratif :***

- iptables -A FORWARD -p tcp --dport 22 -s 192.168.10.10 -d 192.168.30.10 -j ACCEPT

On bloque toutes les autres connexions :

- iptables -A FORWARD -p tcp --dport 22 -d 192.168.30.10 -j DROP

- ***Permettre l'accès au service FTP à un seul poste, celui de l'administrateur situé dans le LAN Administratif :***

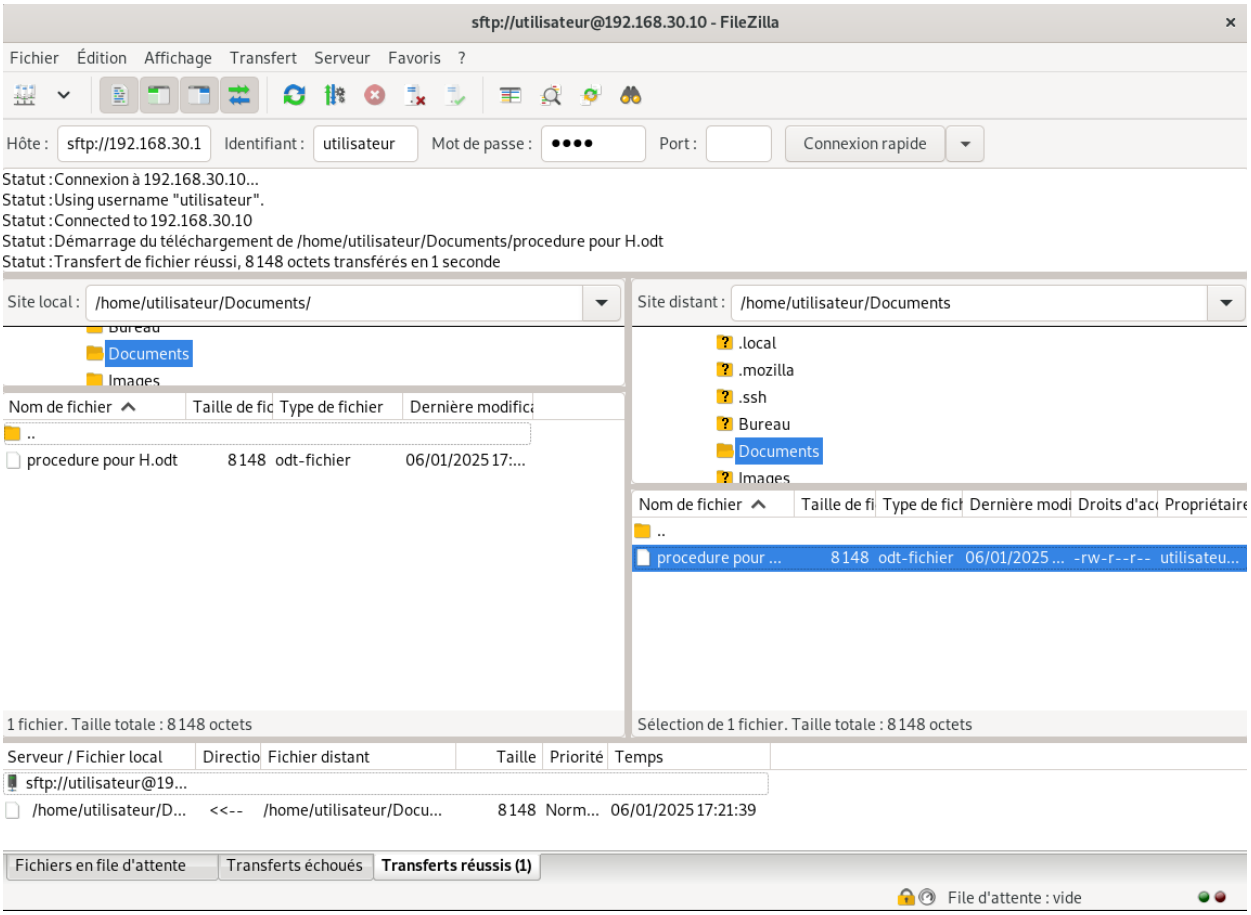
On va utiliser le protocole SFTP afin de se connecter au serveur FTP. On utilisera donc le protocole SHH que l'on a autoriser plus haut.

Nous allons maintenant pouvoir passer au teste de connexions.

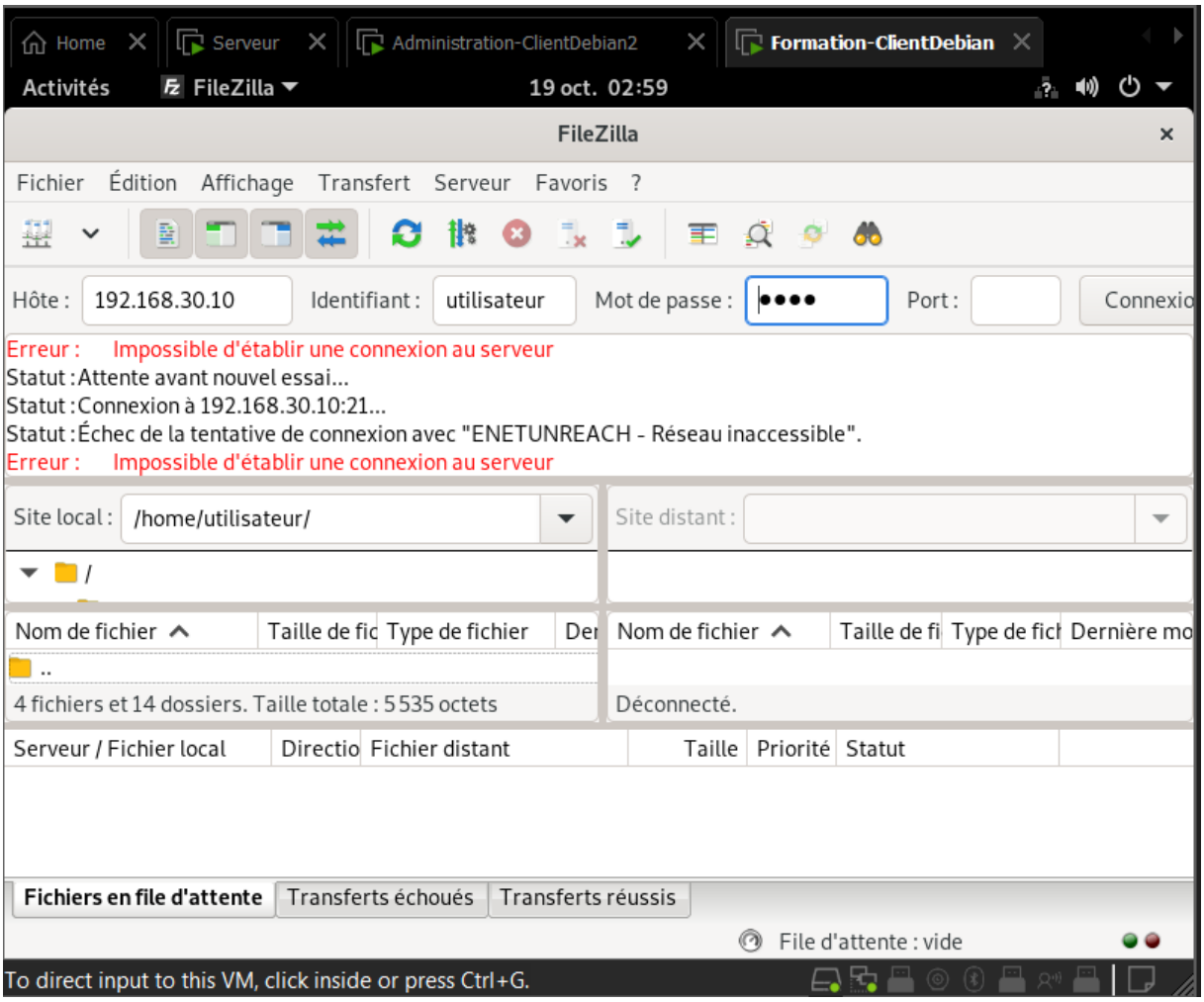
Phase d'expérimentation et de test du parc informatique :

Test de connexion avec notre Serveur :

Depuis le poste administrateur :

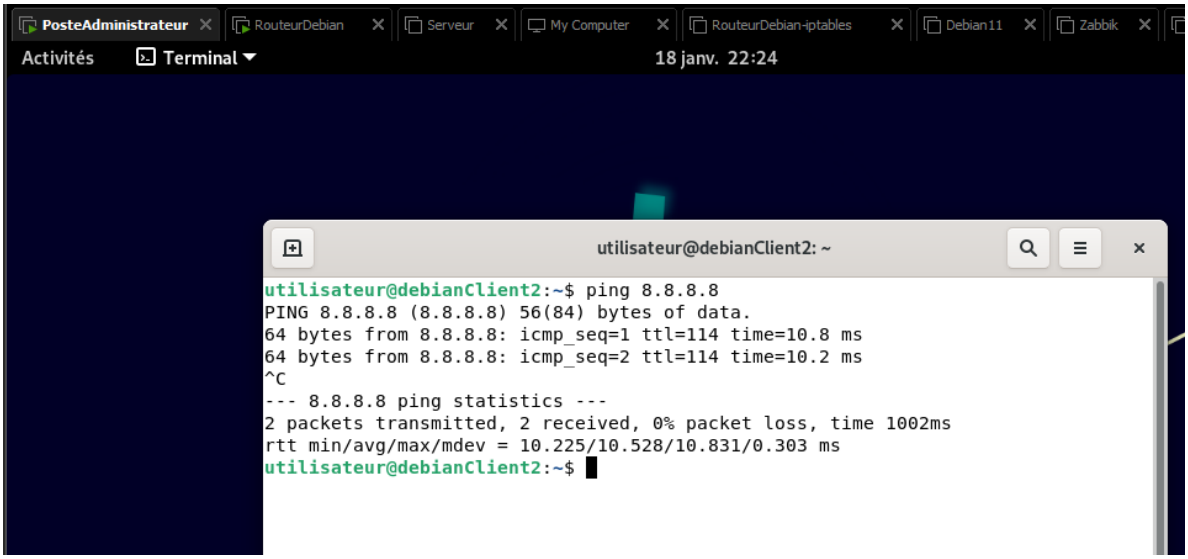


Depuis le poste formation :

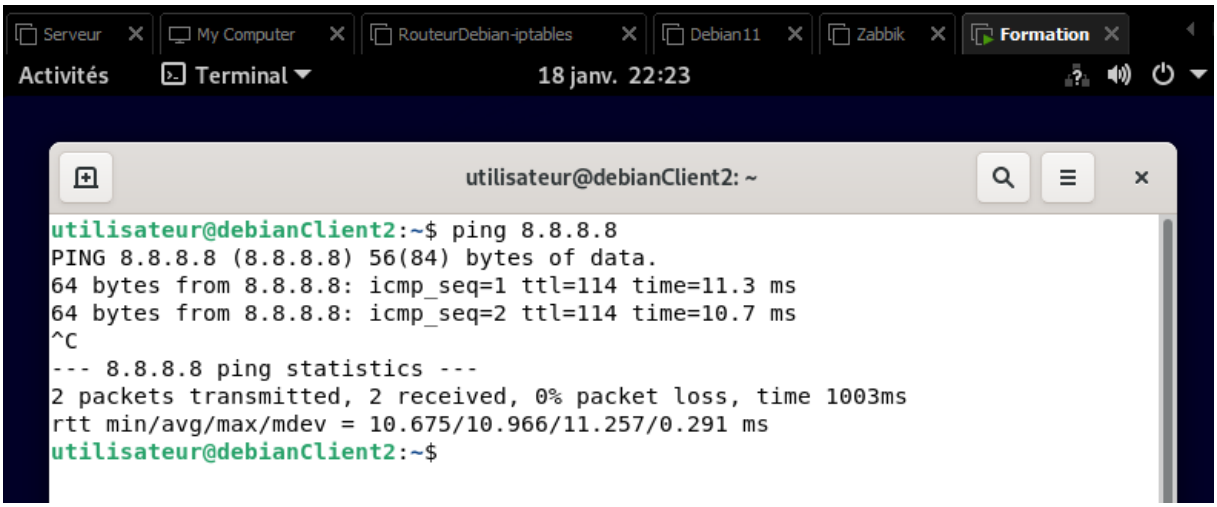


Accès Internet :

Depuis le poste administrateur :



Depuis le poste formation :



Conclusion :

En conclusion, tous les critères du cahier des charges ont été respectés. Le serveur Debian fonctionne dans un réseau unique, incluant un serveur web, FTP et DNS.

Le serveur web héberge deux sites accessibles à tous les utilisateurs.

Le serveur FTP est exclusivement réservé aux administrateurs, et le réseau administratif est isolé du réseau de formation.

L'administrateur a un accès SSH sécurisé au serveur.

Nos tests confirment que les règles de filtrage ont été configurées correctement, garantissant que tout fonctionne comme prévu.