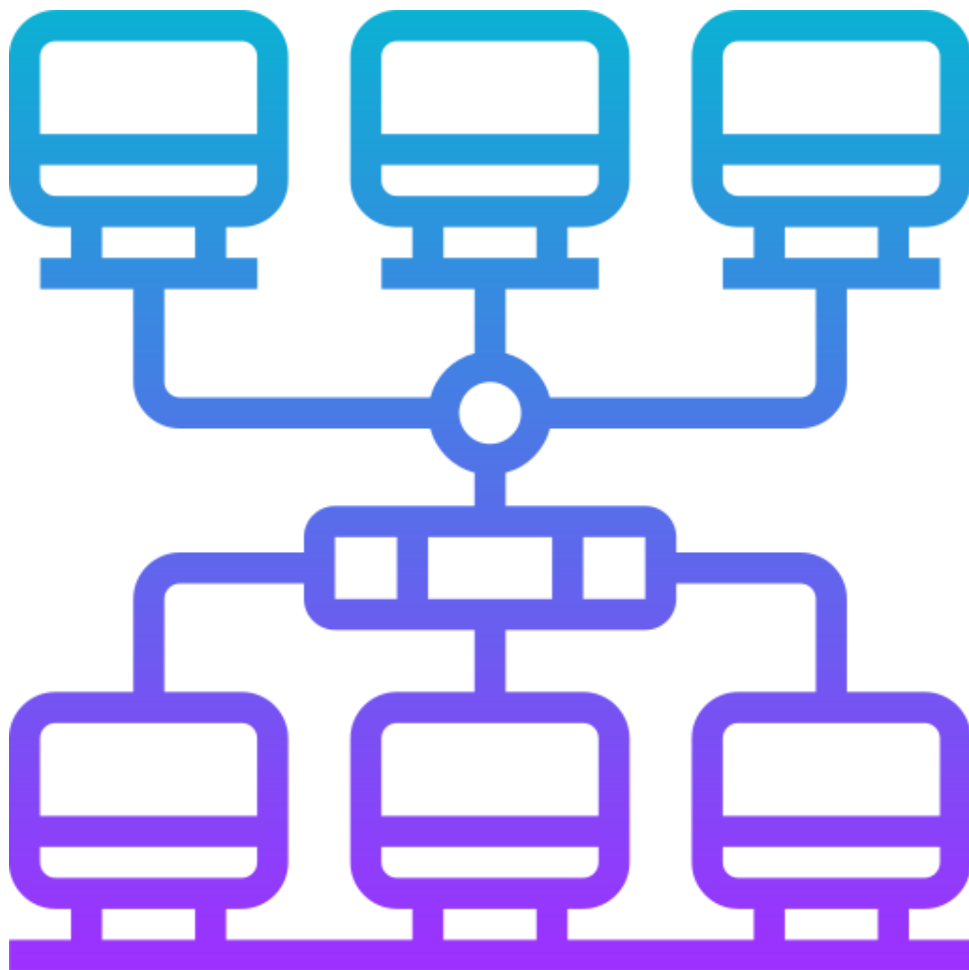


Mise en place d'une Passerelle Linux et d'une DMZ sous Debian 12



Cédric Le Meur
Mars 2024

Sommaire

Contexte.....	p.3
Pré-Configuration.....	p.5
Création du Serveur Web.....	p.6
Création du Routeur.....	p.7
Configuration du Client.....	p.9
Configuration du Serveur Web.....	p.9
Installation d'Apache2.....	p.9
Installation ProFTPD.....	p.11
Installation SSH.....	p.11
Création des VirtualHost.....	p.12
Sécurisation HTTPS.....	p.13
Configuration du Routeur.....	p.16
Activation NAT.....	p.16
Installation Bind9 et mise en place DNS.....	p.16
Mise en Place du Firewall.....	p.19
Tests Client.....	p.20
Test d'accès au réseau Administration.....	p.20
Test d'accès au réseau Formation.....	p.24
Possibilités d'amélioration.....	p.26

Contexte

- Votre 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 : maquetter le nouveau réseau et filtrer les flux

- La solution qui a été retenue est de créer un sous-réseau nommé DMZ pour héberger les services partagés par le personnel et les stagiaires (sous-réseau Formation). A termes, ce réseau DMZ devrait être accessible depuis Internet.
- 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.

Travail à réaliser

Cahier des charges :

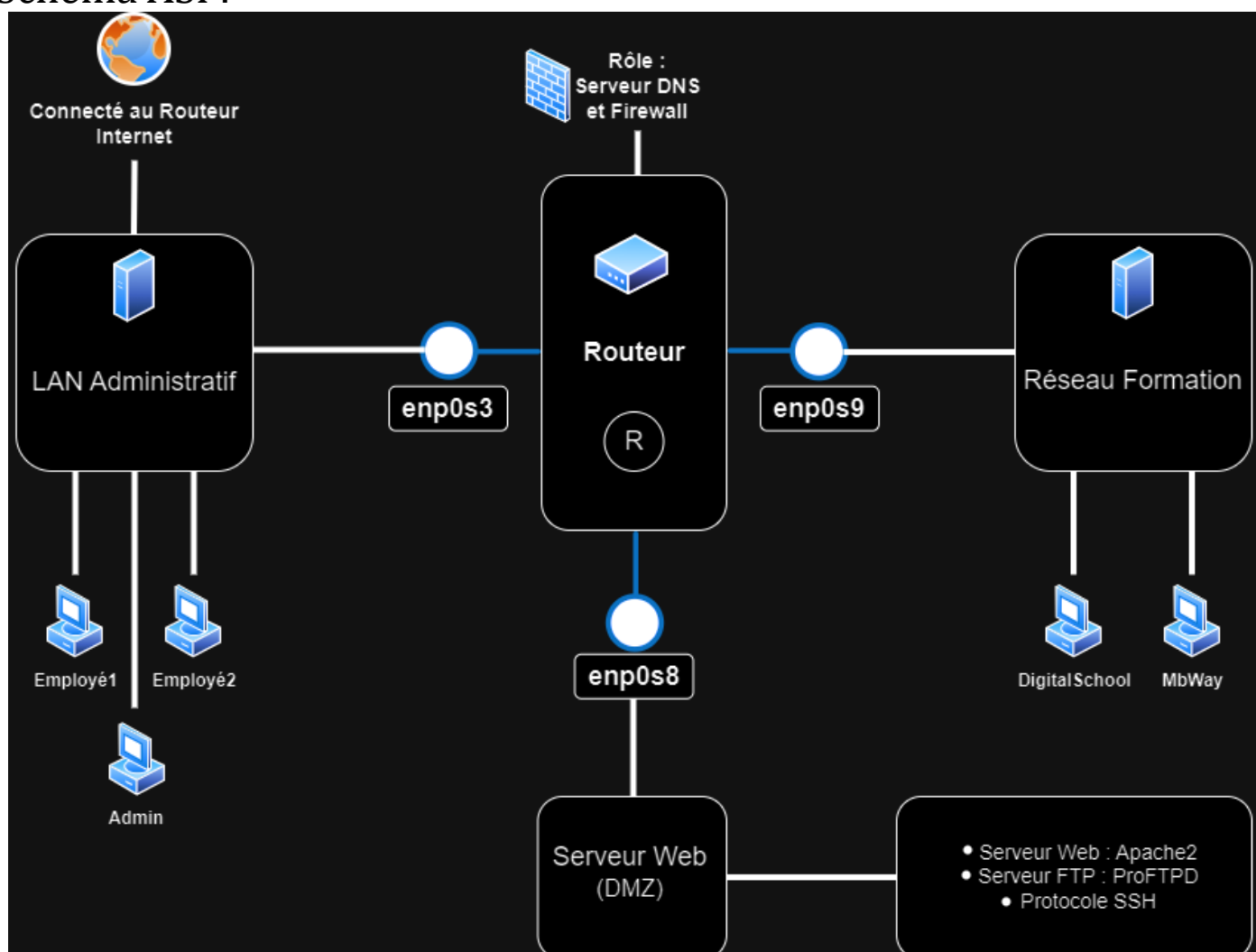
- Permettre l'accès au serveur Web dans la DMZ pour tous, LAN Administratif et Formation.
- Permettre l'accès à internet pour tous en utilisant le Routeur Debian (R) comme passerelle. Ce routeur hébergera aussi les service DNS et fera office de Firewall pour filtrer les accès à la DMZ.
- Permettre l'accès 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

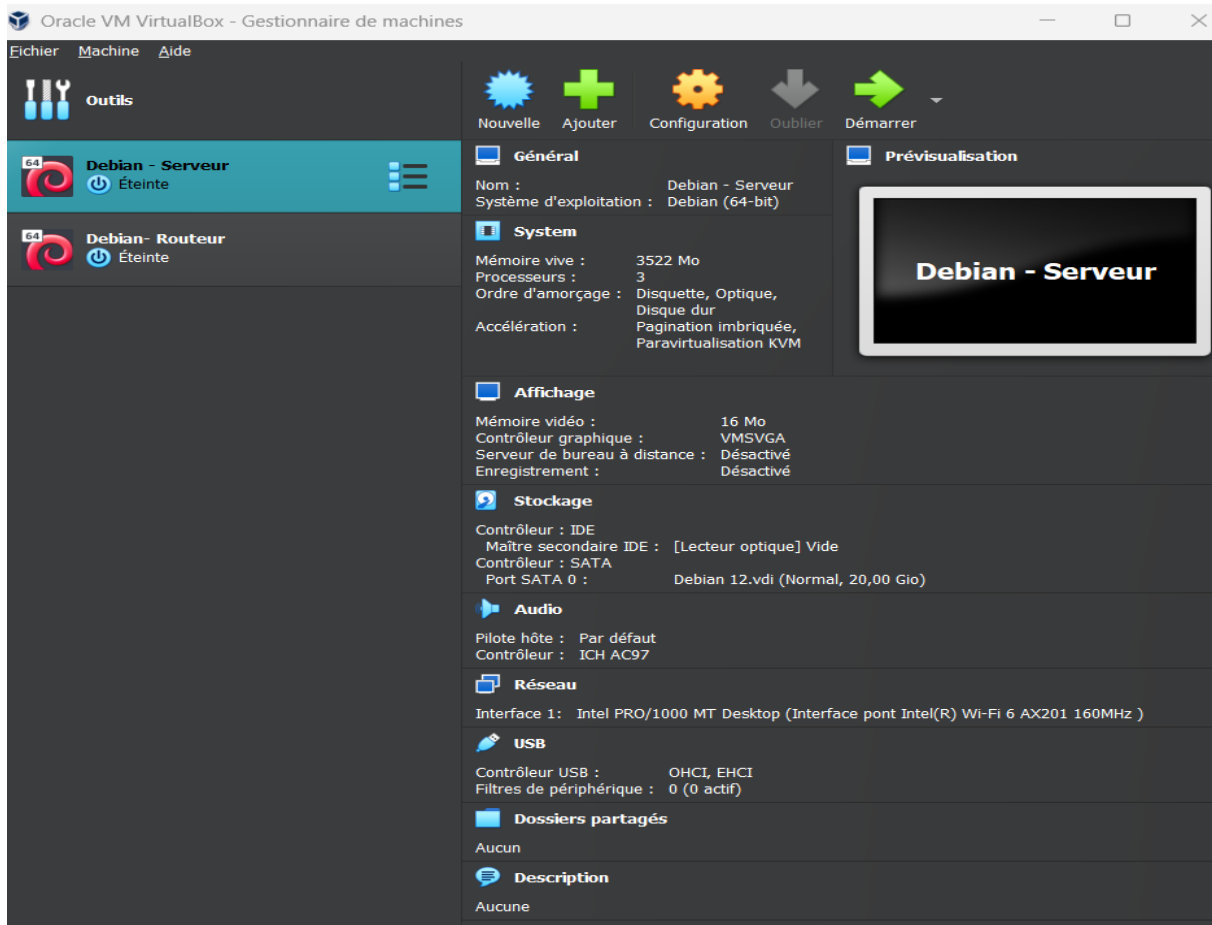
Pour répondre aux objectifs, j'ai opté pour la création d'une zone démilitarisée (DMZ). Cette DMZ abrite mon serveur WEB Apache, qui héberge deux intranets distincts, ainsi qu'un routeur Linux assurant les fonctions de serveur DNS et de pare-feu. De plus, deux machines clientes sont intégrées dans ce système : l'une fournissant l'accès à Internet, tandis que l'autre est utilisée exclusivement pour les tâches administratives et de formation.

Schéma ASI :

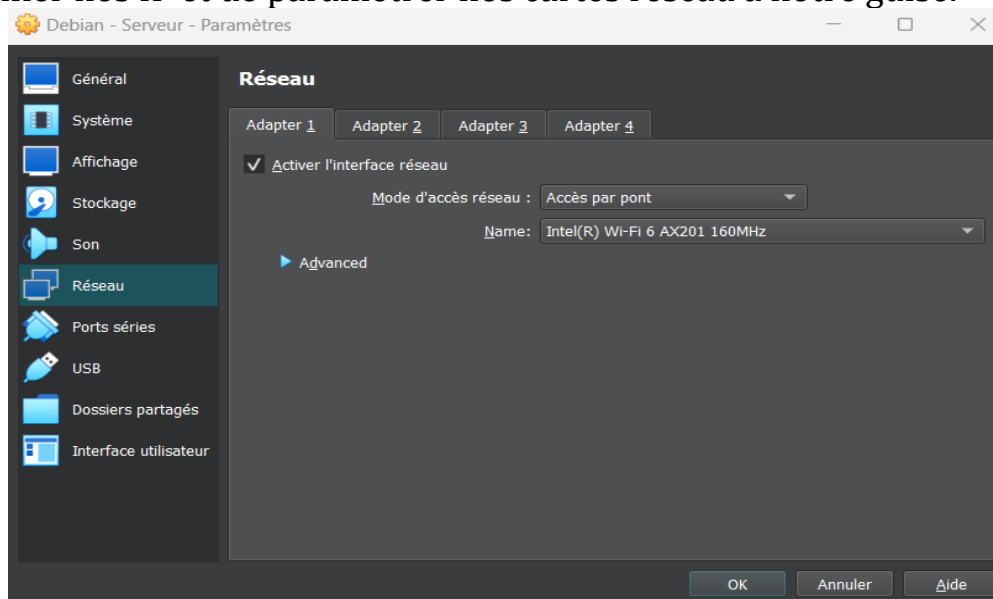


Pré-configuration

Avant toute chose, nous allons paramétrer nos 2 machines virtuels sous Debian 12, une qui nous servira de serveur, et une qui nous servira de routeur. J'utilise ici VirtualBox pour la création des VM. Pour le poste client, j'utiliserais ma propre machine sous Windows 11.



Il faut penser à paramétrer le mode d'accès réseau de nos VM en accès par pont afin de pouvoir modifier nos IP et de paramétrer nos cartes réseau à notre guise.



Création du Serveur Web

Pour commencer, nous allons créer et configurer notre Serveur Web sur une de nos VM Debian 12. Avant de débiter toute installation, nous allons passer notre utilisateur dans le fichier des “sudoers”, afin de ne pas avoir à utiliser le mode Super User de linux, qui est une pratique à éviter dans une utilisation professionnelle.

Pour se faire, après être entré en mode Super User avec la commande “su -” dans le terminal, nous pouvons ajouter notre utilisateur au fichier de la manière suivante :

```
nox44@Debian:~$ su -  
Mot de passe :  
root@Debian:~# nano /etc/sudoers
```

Puis, dans le fichier des sudoers, ajouter la ligne suivante sous “root” :

```
# User privilege specification  
root    ALL=(ALL:ALL) ALL  
nox44   ALL=(ALL) NOPASSWD:ALL
```

J'utilise ici le paramètre NOPASSWD, qui évite d'avoir à entrer le mot de passe à chaque commande, dans un suffit d'efficacité de réalisation, cependant l'utilisation de ce paramètre est à éviter en situation professionnelle.

Cela étant fait, nous pouvons commencer la configuration de la carte réseau “enp0s3” du serveur web, qui sera par la suite connecté à une des interfaces du routeur :

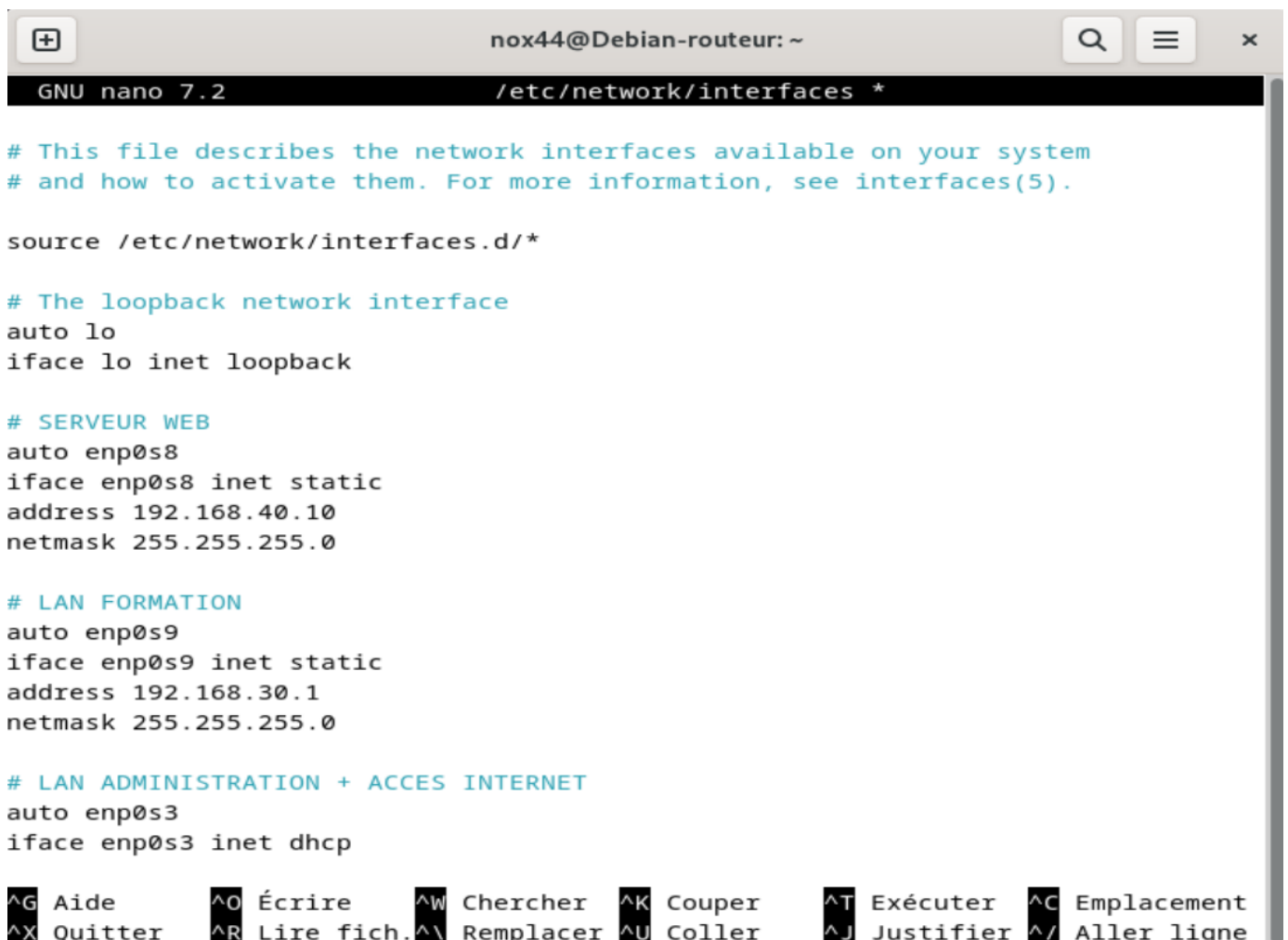


```
nox44@Debian: ~  
GNU nano 7.2 /etc/network/interfaces  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
auto enp0s3  
iface enp0s3 inet static  
address 192.168.40.1  
gateway 192.168.40.10  
netmask 255.255.255.0  
|  
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement  
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^/ Aller ligne
```

Nous accédons au fichier de configuration des interfaces réseau avec la commande “`sudo nano etc/network/interfaces`”, nous allons utiliser l’ip **192.168.40.1/24** pour cette carte réseau, en n’oubliant pas de la configurer sur “static” et non sur “DHCP”. Après cela, nous redémarrons la carte réseau avec la commande “`sudo systemctl restart networking.interfaces`” pour prendre en compte les changements.

Création du Routeur

Nous répétons l’étape d’ajout d’utilisateur dans le fichier des sudoers, puis nous allons configurer les 3 cartes réseaux de notre routeur. Une interface sera reliée au serveur web, une autre sera relié au Lan Formation, et la dernière sera relié au Lan Administration et aura également un accès à internet. Nous répétons donc les commandes citées à l’étape précédente pour modifier notre fichier de configuration de carte réseau, puis de redémarrage du networking de la VM.



```
no44@Debian-router: ~  
GNU nano 7.2 /etc/network/interfaces *  
  
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# SERVEUR WEB  
auto enp0s8  
iface enp0s8 inet static  
address 192.168.40.10  
netmask 255.255.255.0  
  
# LAN FORMATION  
auto enp0s9  
iface enp0s9 inet static  
address 192.168.30.1  
netmask 255.255.255.0  
  
# LAN ADMINISTRATION + ACCES INTERNET  
auto enp0s3  
iface enp0s3 inet dhcp  
  
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement  
^X Quitter   ^R Lire fich.^_ Remplacer  ^U Coller    ^J Justifier ^/ Aller ligne
```


Configuration du client

La machine cliente remplira deux fonctions distinctes : elle servira à la fois d'outil pour l'administration du site et de support de formation pour les élèves. Pour modifier les paramètres réseau du poste client nous allons accéder aux paramètres ipv4.

The image displays three side-by-side screenshots of the Windows 'Propriétés de : Protocole Internet version 4 (TCP/IPv4)' window. Each window is set to 'Général' and has the 'Utiliser l'adresse IP suivante' option selected. The configurations are as follows:

- Configuration Administration:** Adresse IP: 192.168.1.10, Masque de sous-réseau: 255.255.255.0, Passerelle par défaut: 192.168.1.179.
- Configuration Administrateur:** Adresse IP: 192.168.1.20, Masque de sous-réseau: 255.255.255.0, Passerelle par défaut: 192.168.1.179.
- Configuration Formation:** Adresse IP: 192.168.30.10, Masque de sous-réseau: 255.255.255.0, Passerelle par défaut: 192.168.30.1.

In all three configurations, the DNS settings are set to 'Utiliser l'adresse de serveur DNS suivante' with preferred and auxiliary DNS servers left blank. The 'Valider les paramètres en quittant' checkbox is unchecked, and the 'Avancé...' button is visible at the bottom right of each window.

Configuration Administration

Configuration Administrateur

Configuration Formation

Voici donc les différentes configurations d'IP de nos différents postes clients.

La première et la seconde est pour un accès au LAN de l'administration du campus, avec l'IP 192.168.1.10 pour les employés, et 192.168.1.20 pour l'administrateur. Nous passons par l'interface enp0s3, qui est en DHCP, puisque ce réseau possède un accès à internet. La troisième a pour IP 192.168.30.1, et donc pour passerelle 192.168.30.10, qui est l'IP de l'interface enp0s9 sur notre routeur.

Configuration du Serveur Web

Installation Apache2 :

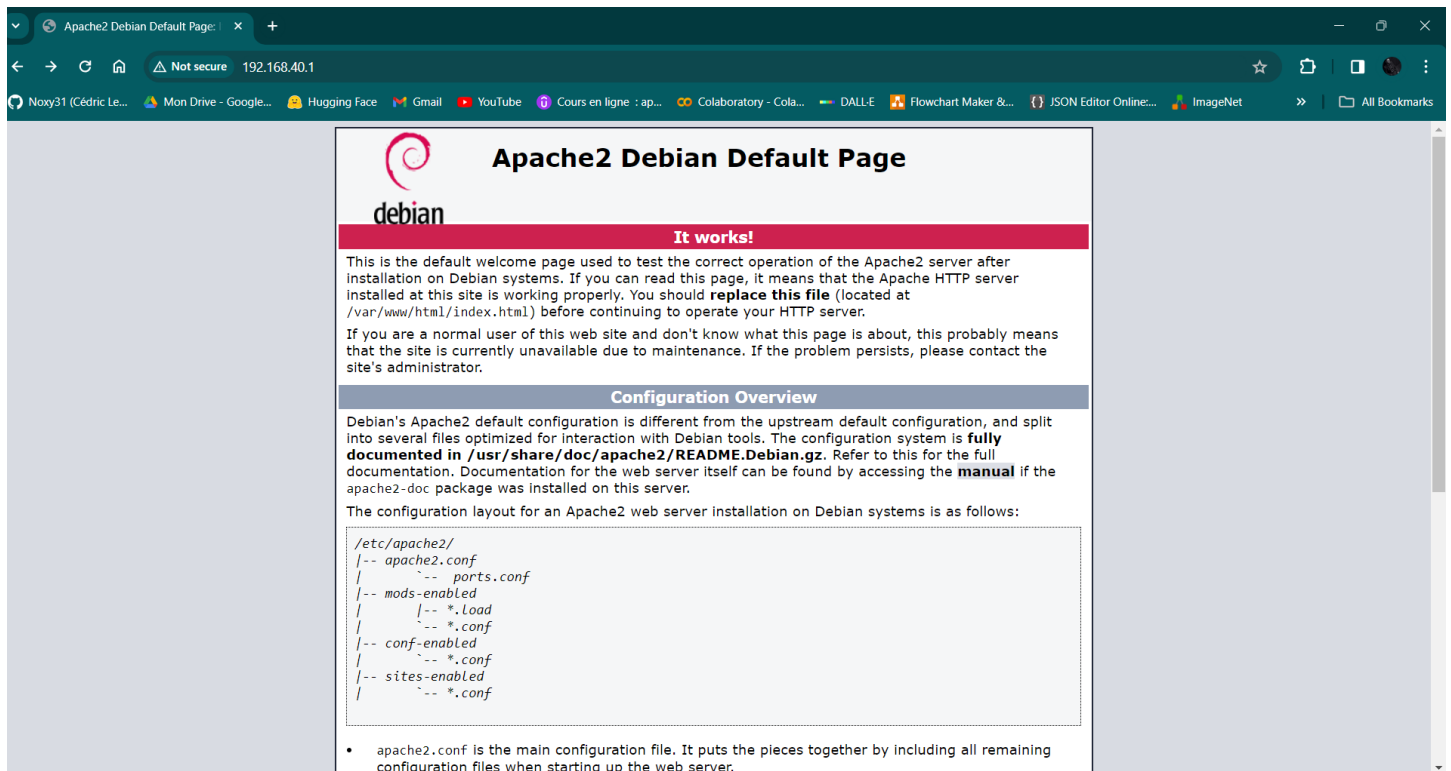
Pour commencer, nous allons installer Apache2 pour héberger les intranets de nos deux écoles. Apache est le serveur web HTTP. Son rôle est d'écouter les requêtes émises par les navigateurs (qui demandent des pages web), de chercher la page demandée et de la renvoyer.

Pour se faire nous allons utiliser la commande `sudo apt install apache2`. Une fois l'installation effectuée, nous allons redémarrer et afficher l'état d'apache pour s'assurer de son bon fonctionnement. Pour redémarrer les services d'Apache nous allons utiliser la commande `sudo systemctl restart apache2` puis `sudo systemctl status apache2`.

```
nox44@Debian:~$ sudo systemctl restart apache2
nox44@Debian:~$ sudo systemctl status apache2
• apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; preset: enab>
   Active: active (running) since Tue 2024-03-12 11:26:11 CET; 11s ago
     Docs: https://httpd.apache.org/docs/2.4/
  Process: 2208 ExecStart=/usr/sbin/apachectl start (code=exited, status=0/SU>
 Main PID: 2213 (apache2)
    Tasks: 55 (limit: 4021)
   Memory: 16.9M
      CPU: 66ms
   CGroup: /system.slice/apache2.service
           └─2213 /usr/sbin/apache2 -k start
             └─2214 /usr/sbin/apache2 -k start
               └─2215 /usr/sbin/apache2 -k start

mars 12 11:26:11 Debian systemd[1]: Starting apache2.service - The Apache HTTP >
mars 12 11:26:11 Debian apachectl[2212]: AH00558: apache2: Could not reliably d>
mars 12 11:26:11 Debian systemd[1]: Started apache2.service - The Apache HTTP S>
lines 1-17/17 (END)
```

Les services d'Apache2 sont donc bien installés et fonctionnels. Afin d'effectuer une vérification supplémentaire nous allons accéder au serveur Apache via le client avec l'IP du serveur `192.168.40.1`.



L'affichage de la page d'Apache nous confirme que tout fonctionne. Passons donc à la suite sur notre serveur Web, avec l'installation de proFTPD.

Installation proFTPD :

Afin d'avoir un serveur de transfert de fichier nous allons utiliser proFTPD qui est un serveur FTP gratuit. Pour l'installation, le procédé est le même que pour Apache, avec la commande `sudo apt install proftpd`. Une fois l'installation effectué, comme pour Apache, nous allons restart puis afficher le status de proFTPD.

```
no44@Debian:~$ sudo systemctl restart proftpd
no44@Debian:~$ sudo systemctl status proftpd
● proftpd.service - ProFTPD FTP Server
   Loaded: loaded (/lib/systemd/system/proftpd.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-03-19 09:54:28 CET; 8s ago
     Docs: man:proftpd(8)
  Process: 2162 ExecStartPre=/usr/sbin/proftpd --configtest -c $CONFIG_FILE $OPTIONS (code=exited, status=0/SUCCESS)
  Process: 2164 ExecStart=/usr/sbin/proftpd -c $CONFIG_FILE $OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 2165 (proftpd)
    Tasks: 1 (limit: 4021)
   Memory: 1.9M
      CPU: 27ms
  CGroup: /system.slice/proftpd.service
          └─2165 "proftpd: (accepting connections)"

mars 19 09:54:28 Debian systemd[1]: Starting proftpd.service - ProFTPD FTP Server...
mars 19 09:54:28 Debian proftpd[2162]: Checking syntax of configuration file
mars 19 09:54:28 Debian systemd[1]: Started proftpd.service - ProFTPD FTP Server.
no44@Debian:~$
```

Installation SSH :

Nous pouvons répéter les étapes précédentes pour l'installation de SSH, le protocole de communication sécurisé. Une fois l'installation terminée, nous pouvons que SSH est bien installé.

```
no44@Debian:~$ sudo systemctl status ssh
● ssh.service - OpenBSD Secure Shell server
   Loaded: loaded (/lib/systemd/system/ssh.service; enabled; preset: enabled)
   Active: active (running) since Tue 2024-03-19 09:44:36 CET; 58min ago
     Docs: man:sshd(8)
           man:sshd_config(5)
  Process: 627 ExecStartPre=/usr/sbin/sshd -t (code=exited, status=0/SUCCESS)
 Main PID: 663 (sshd)
    Tasks: 1 (limit: 4021)
   Memory: 3.8M
      CPU: 106ms
  CGroup: /system.slice/ssh.service
          └─663 "sshd: /usr/sbin/sshd -D [listener] 0 of 10-100 startups"

mars 19 09:44:36 Debian systemd[1]: Starting ssh.service - OpenBSD Secure Shell server...
mars 19 09:44:36 Debian sshd[663]: Server listening on 0.0.0.0 port 22.
mars 19 09:44:36 Debian sshd[663]: Server listening on :: port 22.
mars 19 09:44:36 Debian systemd[1]: Started ssh.service - OpenBSD Secure Shell server.
no44@Debian:~$
```

Création des intranets (VirtualHosts) :

Pour créer les sites respectifs de chaque école, nous allons créer un dossier par école dans le dossier `/var/www/html` de notre VM. Nous utilisons donc les commandes `cd /var/www/html` pour accéder au dossier, `sudo mkdir "nom du dossier"` pour les créer, puis `ls -l` pour en afficher

```

no44@Debian: /var/www/html
no44@Debian:~$ cd /var/www/html
no44@Debian:/var/www/html$ mkdir MyDigitalSchool
mkdir: impossible de créer le répertoire « MyDigitalSchool »: Permission non accordée
no44@Debian:/var/www/html$ sudo mkdir MyDigitalSchool
no44@Debian:/var/www/html$ ls -l
total 16
-rw-r--r-- 1 root root 10701 16 janv. 11:07 index.html
drwxr-xr-x 2 root root 4096 19 mars 11:06 MyDigitalSchool
no44@Debian:/var/www/html$ sudo mkdir Mbwat
no44@Debian:/var/www/html$ ls -l
bash: ls-l : commande introuvable
no44@Debian:/var/www/html$ ls -l
total 20
-rw-r--r-- 1 root root 10701 16 janv. 11:07 index.html
drwxr-xr-x 2 root root 4096 19 mars 11:07 Mbwat
drwxr-xr-x 2 root root 4096 19 mars 11:06 MyDigitalSchool
no44@Debian:/var/www/html$

```

Nous allons maintenant configurer les adresses d'accès à nos intranets, dans les fichiers de configuration d'apache, pour donner un ServerName et changer le chemin d'accès par lequel Apache récupère les fichiers de nos sites (Html, Css et JS ect...). Nous allons donc nous rendre dans le

dossier /etc/apache2/sites-available et entrer la commande `cp 000-default.conf mbway.conf` (qui va copier le fichier de configuration d'host par défaut d'apache et le recréer en le renommant mbway.conf). Nous réalisons l'opération deux fois, pour chaque école.

```

no44@Debian: /etc/apache2/sites-available
GNU nano 7.2 mydigitalschool.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.example.com

    ServerAdmin webmaster@localhost
    ServerName mds.lan
    ServerAlias www.mds.lan
    DocumentRoot /var/www/html/MyDigitalSchool

```

Sécurisation avec protocole HTTPS avec OpenSSL :

Afin de sécuriser nos intranets, il convient de ne pas laisser les connexions en http. Nous allons donc générer des certificats via OpenSSL, une librairie libre de création de certificats et de clés.

Avant toute chose, nous allons donc installer OpenSSL grâce à la commande `apt install proftpd openssl`.

Ensuite, nous allons créer un nouveau certificat grâce à la commande suivante : `openssl req -new -x509 -keyout /etc/ssl/apache.key -days 365 -nodes -out /etc/ssl/apache.crt`

Ici, -keyout suivi du chemin indique que clé sera créée dans le dossier /etc/ssl, -x509 indique que nous voulons que le certificat soit auto-signé, -days 365 indique la durée de validité du certificat auto-signé, -nodes indique que la clé ne sera pas protégée par un mot de passe lorsqu'elle sera créée, puis -out donne le chemin de sortie /etc/ssl/.

Lors de la création de certificat, certaines informations seront demandées. Nous pouvons les renseigner, ou entrer "." pour laisser l'information vide.

[illegible]

Pour terminer, nous allons lier notre certificat ainsi que notre clé à nos Virtual Host, en retournant dans les fichiers de configuration `/etc/apache2/sites-available/`.

!!! Faites attention aux majuscules et minuscules dans les chemins DocumentRoot de vos dossiers de sites, ils sont sensibles à la casse. Également, avant l'activation de vos sites, penser à activer SSL avec la commande `sudo a2enmod ssl`, afin de ne pas rencontrer d'erreur avec apache !!!


```
no44@Debian: ~
GNU nano 7.2 /etc/apache2/sites-available/mydigitalschool.conf *
# 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
</VirtualHost>

# Penser a ouvrir la balise VirtualHost et a rentrer le port 443
<VirtualHost *:443>
    ServerName www.mds.lan:443
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/MyDigitalSchool
# Nous paramétrons l'accès au certificat et a la clé
    SSLEngine on
    SSLCertificateFile      /etc/ssl/apache.crt
    SSLCertificateKeyFile   /etc/ssl/apache.key
# Nous pouvons rentrer a nouveau l'accès aux log d'erreurs
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
|
```

^G Aide	^O Écrire	^W Chercher	^K Couper	^T Exécuter	^C Emplacement	M-U Annuler	M-A Marquer
^X Quitter	^R Lire fich.	^I Remplacer	^U Coller	^J Justifier	^_ Aller ligne	M-E Refaire	M-G Copier

Pour terminer il suffit de rentrer les commandes suivantes pour activer nos intranets, puis pour redémarrer apache2 :

- sudo a2ensite mbway.conf
- sudo a2ensite mydigitalschool.conf
- sudo systemctl restart apache2

Configuration du Routeur

Nous allons maintenant passer à la configuration de notre routeur, gérer les droits d'entrée et de sortie de connexions, mettre en place notre service DNS et configurer notre firewall.

Activation du NAT et paramétrage IPTABLES :

Pour commencer, nous devons installer iptables-persistent grâce à la commande `sudo apt install iptables-persistent`, qui permet de sauvegarder nos règles iptables dans des scripts qui s'exécuteront au démarrage de notre routeur.

Pour s'assurer de la bonne sauvegarde de nos règles Iptables, et afin de pouvoir les modifier plus aisément à l'avenir, nous pouvons également sauvegarder nos règles dans un fichier spécifique grâce à la commande `sudo iptables-save > /etc/iptables_rules.save`, puis il faudra récupérer le fichier sauvegardé dans notre fichier `/etc/network/interfaces` pour y ajouter la ligne suivant : `post-up iptables-restore < /etc/iptables_rules.save`.

Pour activer notre routeur, nous devons également nous rendre dans le fichier `sysctl` : `sudo nano /etc/sysctl.conf`, puis décommenter la ligne `net.ipv4.ip_forward=1` en supprimant le #.

Nous allons également activer le NAT sur la carte réseau qui permettra l'accès à internet avec la commande `sudo iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE`.

Configuration du DNS avec Bind9 :

Pour commencer, nous devons modifier le fichier `/etc/resolv.conf` pour y entrer : `nameserver 192.168.40.10` ; (Ce qui correspond à l'adresse de la carte réseau `enp0s8` menant vers notre serveur.)

Ensuite, nous allons installer Bind9 avec la commande `sudo apt install bind9`.

Nous allons ensuite nous rendre dans le dossier `/etc/bind`, puis nous allons, comme pour `apache2`, copier et renommer les fichiers de configuration par défaut.

Ce fichier est nommé "db.local", il suffit donc d'entrer les commandes suivantes :

- `sudo cp db.local db.mbway.lan`
- `sudo cp db.local db.mds.lan`

Dans ce fichier, nous allons paramétrer les informations et IP de notre DNS pour `mbway`.

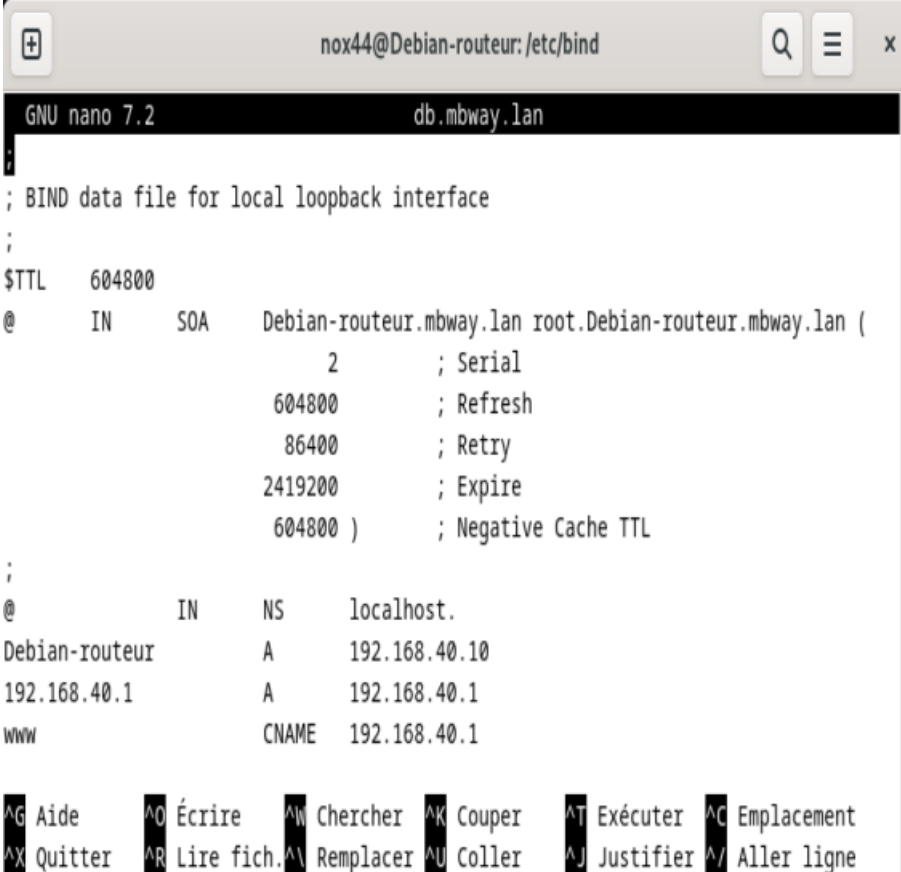
Nous allons rentrer le nom de notre serveur : Debian-routeur.mbway.lan

La ligne comprenant “root” indique l’administrateur du serveur.

Les informations situées en dessous (Serial, Refresh, Retry ect..) n’ont pas besoin d’être modifiés.

L’ip 192.168.40.10 est l’IP de l’interface connecté au serveur, et permet d’identifier la machine.

L’ip 192.168.40.40 permet d’identifier le DNS sur le serveur web.



```
no44@Debian-routeur:/etc/bind
GNU nano 7.2 db.mbway.lan
; BIND data file for local loopback interface
;
$TTL 604800
@      IN      SOA      Debian-routeur.mbway.lan root.Debian-routeur.mbway.lan (
                                2          ; Serial
                                604800     ; Refresh
                                86400      ; Retry
                                2419200    ; Expire
                                604800 )   ; Negative Cache TTL
;
@      IN      NS       localhost.
Debian-routeur      A      192.168.40.10
192.168.40.1        A      192.168.40.1
www                CNAME  192.168.40.1

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

Il suffit ensuite de répéter l’opération avec le fichier de configuration de MyDigitalSchool. Ensuite, nous allons devoir créer deux “zones” pour nos deux intranets. La première modification se fera dans le fichier named.conf.local, qui se situe également dans le dossier /etc/bind.

```
no44@Debian-routeur:/etc/bind
GNU nano 7.2 named.conf.local
//
// Do any local configuration here
//

// Consider adding the 1918 zones here, if they are not used in your
// organization
//include "/etc/bind/zones.rfc1918";

zone "mbway.lan" {
    type master;
    file "/etc/bind/db.mbway.lan";
};

zone "mds.lan" {
    type master;
    file "/etc/bind/db.mds.lan";
};

[ 17 lignes écrites ]
^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement M-U Annuler
^X Quitter   ^R Lire fich. ^\ Remplacer ^U Coller    ^J Justifier ^/ Aller ligne M-E Refaire
```

Nos zones sont maintenant configurées, et les chemins d'accès y sont correctement spécifiés. Pour terminer, il nous reste à indiquer les IP "forwarders" dans le fichier `named.conf.options`, qui se situe toujours dans le même dossier.

```
no44@Debian-routeur:/etc/bind
GNU nano 7.2 named.conf.options *
options {
    directory "/var/cache/bind";

    // If there is a firewall between you and nameservers you want
    // to talk to, you may need to fix the firewall to allow multiple
    // ports to talk. See http://www.kb.cert.org/vuls/id/800113

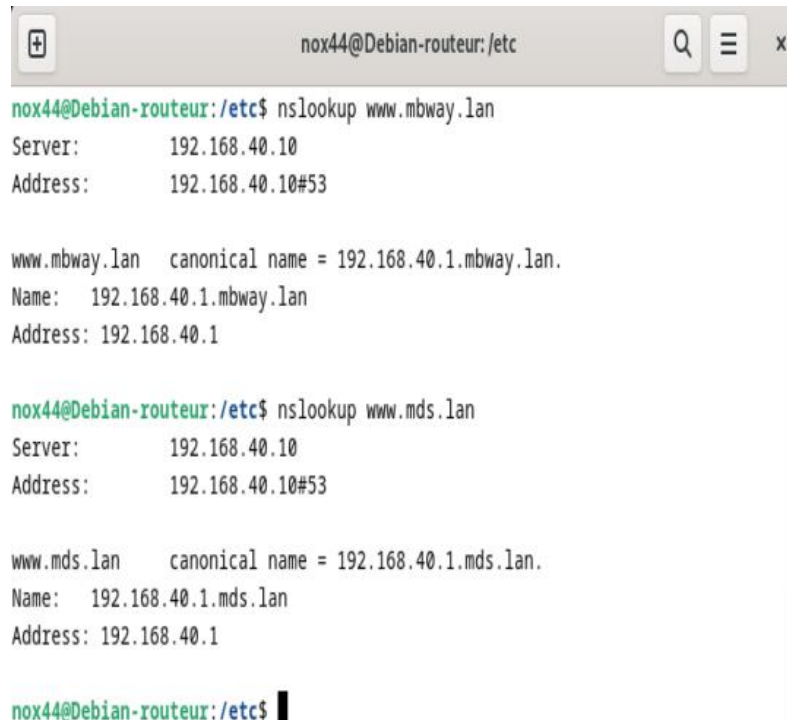
    // If your ISP provided one or more IP addresses for stable
    // nameservers, you probably want to use them as forwarders.
    // Uncomment the following block, and insert the addresses replacing
    // the all-0's placeholder.

    forwarders {
        192.168.40.10;
        8.8.8.8;
        8.8.4.4;
    };
};

^G Aide      ^O Écrire    ^W Chercher  ^K Couper    ^T Exécuter  ^C Emplacement M-U Annuler
^X Quitter   ^R Lire fich. ^\ Remplacer ^U Coller    ^J Justifier ^/ Aller ligne M-E Refaire
```

Nous avons donc ajouté l'adresse IP de notre serveur DNS, ainsi que les DNS de Google. Nous pouvons maintenant relancer les services de bind9 avec la commande `sudo systemctl restart bind9`.

Pour vérifier la bonne configuration de notre DNS, il nous suffit d'entrer les commandes `nslookup www.mds.lan` et `nslookup www.mbway.lan`. Nous pouvons constater sur la capture d'écran ci-dessous que le nslookup nous retourne bien l'IP de notre serveur web, 192.168.40.1.



```
no44@Debian-routeur: /etc
no44@Debian-routeur: /etc$ nslookup www.mbway.lan
Server:      192.168.40.10
Address:     192.168.40.10#53

www.mbway.lan canonical name = 192.168.40.1.mbway.lan.
Name:   192.168.40.1.mbway.lan
Address: 192.168.40.1

no44@Debian-routeur: /etc$ nslookup www.mds.lan
Server:      192.168.40.10
Address:     192.168.40.10#53

www.mds.lan canonical name = 192.168.40.1.mds.lan.
Name:   192.168.40.1.mds.lan
Address: 192.168.40.1

no44@Debian-routeur: /etc$
```

Création du Firewall avec Iptables :

Nous pouvons maintenant passer à la création du Firewall sur notre Routeur, qui permettra de gérer les accès de connexions entrantes sur notre Serveur, et également de gérer les permissions de communications entre nos réseaux Administration et Formation.

Nous allons donc créer nos règles d'accès avec Iptables de la manière suivante :

Accès aux intranets :

- `iptables -A FORWARD -s 192.168.30.0/24 -p tcp --dport 80:443 -j ACCEPT`
- `iptables -A FORWARD -s 192.168.1.0/24 -p tcp --dport 80:443 -j ACCEPT`

Accès FTP et SSH pour l'administrateur uniquement :

- `iptables -A FORWARD -s 192.168.1.20 -p tcp --dport 21 -j ACCEPT`
- `Iptables -A FORWARD -s 192.168.1.20 -p tcp --dport 22 -j ACCEPT`

Nous devons aussi bloquer les accès FTP et SSH à nos réseaux Administration et Formation grâce aux commandes suivantes :

- iptables -A FORWARD -s 192.168.30.0/24 -p tcp --dport 21 -j DROP
- iptables -A FORWARD -s 192.168.1.0/24 -p tcp --dport 21 -j DROP
- iptables -A FORWARD -s 192.168.30.0/24 -p tcp --dport 22 -j DROP
- iptables -A FORWARD -s 192.168.1.0/24 -p tcp --dport 22 -j DROP

Comme évoquer plus haut, n'oublions pas la commande `iptables-save > /etc/iptables_rules.save` afin de sauvegarder nos règles dans notre fichier de sauvegarde.

Tests Client

Nous allons maintenant configurer notre poste client Windows avec notre DNS (192.168.40.10), et effectuer les tests d'accès à nos intranets ainsi qu'à notre serveur FTP et à notre connexion via SSH.

Tests d'accès au réseau Administration :

Pour commencer testons les accès du client ayant l'IP 192.168.1.10.

Sur l'image qui suit, nous pouvons constater via les pings du DNS de google 8.8.4.4 que le client a bien accès à internet. Également, nous pouvons constater que notre DNS est bien configuré, puisque les pings sur nos intranets sont également concluant. Nos deux intranets sont également accessibles via un navigateur, et les sites n'ont plus qu'à être développé.

```
Invite de commandes
C:\Users\Nox44>ping 8.8.4.4

Envoi d'une requête 'Ping' 8.8.4.4 avec 32 octets de données :
Réponse de 8.8.4.4 : octets=32 temps=17 ms TTL=115
Réponse de 8.8.4.4 : octets=32 temps=18 ms TTL=115
Réponse de 8.8.4.4 : octets=32 temps=15 ms TTL=115
Réponse de 8.8.4.4 : octets=32 temps=15 ms TTL=115

Statistiques Ping pour 8.8.4.4:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 15ms, Maximum = 18ms, Moyenne = 16ms

C:\Users\Nox44>ping www.mds.lan

Envoi d'une requête 'ping' sur 192.168.40.1.mds.lan [192.168.40.1] avec 32 octets de données :
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=2 ms TTL=63

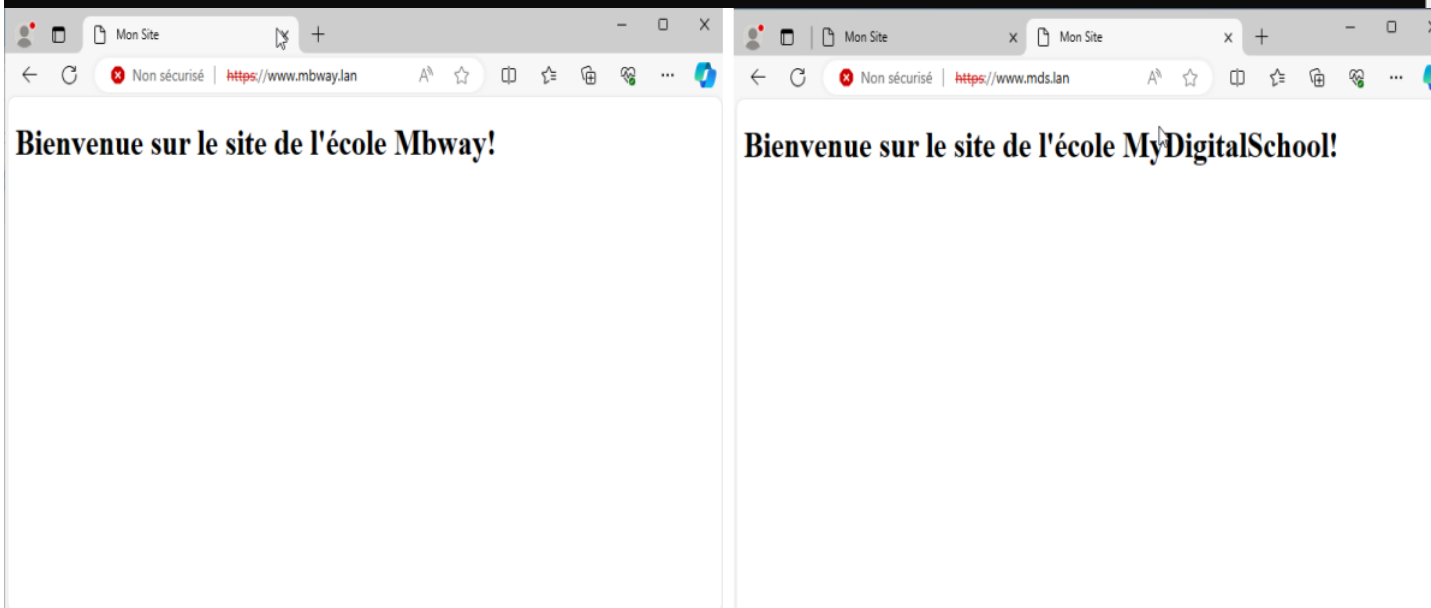
Statistiques Ping pour 192.168.40.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 1ms, Maximum = 2ms, Moyenne = 1ms

C:\Users\Nox44>ping www.mbway.lan

Envoi d'une requête 'ping' sur 192.168.40.1.mbway.lan [192.168.40.1] avec 32 octets de données :
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=4 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=2 ms TTL=63

Statistiques Ping pour 192.168.40.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
    Durée approximative des boucles en millisecondes :
        Minimum = 1ms, Maximum = 4ms, Moyenne = 2ms

C:\Users\Nox44>
```



En revanche, lorsque notre utilisateur de l'administration essaye de se connecter au serveur ftp ou via SSH, il n'en a pas l'accès, ce qui confirme la bonne prise en compte de nos règles iptables sur les adresses en 192.168.1.0/24, excepté pour la .20 qui est celle du poste de l'administrateur.

Ici, le certificat créé avec openssl doit être mis dans le répertoire du site, mais pour une utilisation réelle, il faudra un certificateur agréé comme ZeroSSL, ACM, ou Let's Encrypt.

```
Invite de commandes - ftp 192.168.40.1
C:\Users\Nox44>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . :
    Adresse IPv6. . . . . : 2a01:e0a:44e:5030:e905:eeb3:bf1a:ec04
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:44e:5030:1049:a70f:a3b5:a4aa
    Adresse IPv6 de liaison locale. . . . : fe80::11a2:b051:185c:2835%4
    Adresse IPv4. . . . . : 192.168.1.10
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : fe80::de00:b0ff:fe67:9a2%4
                                      192.168.1.179

C:\Users\Nox44>ftp 192.168.40.1
> ftp: connect :Délai de connexion dépassé
ftp>

Invite de commandes
Microsoft Windows [version 10.0.19045.3803]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Nox44>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . :
    Adresse IPv6. . . . . : 2a01:e0a:44e:5030:e905:eeb3:bf1a:ec04
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:44e:5030:91b1:20db:7464:545f
    Adresse IPv6 de liaison locale. . . . : fe80::11a2:b051:185c:2835%4
    Adresse IPv4. . . . . : 192.168.1.10
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : fe80::de00:b0ff:fe67:9a2%4
                                      192.168.1.179

C:\Users\Nox44>ssh nox44@192.168.40.1
ssh: connect to host 192.168.40.1 port 22: Connection timed out

C:\Users\Nox44>
```

Alors que l'adresse de l'administrateur (192.168.1.20), en plus d'avoir accès aux intranets, comme pour les autres adresses de son réseau, peut également se connecter au serveur FTP, ainsi que via SSH.

```
Sélection Invite de commandes - ftp 192.168.40.1
Microsoft Windows [version 10.0.19045.3803]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Nox44>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . :
    Adresse IPv6. . . . . : 2a01:e0a:44e:5030:e905:eeb3:bf1a:ec04
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:44e:5030:1049:a70f:a3b5:a4aa
    Adresse IPv6 de liaison locale. . . . : fe80::11a2:b051:185c:2835%4
    Adresse IPv4. . . . . : 192.168.1.20
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : fe80::de00:b0ff:fe67:9a2%4
                                      192.168.1.179

C:\Users\Nox44>ftp 192.168.40.1
Connecté à 192.168.40.1.
220 ProFTPD Server (NoxySERVER) [::ffff:192.168.40.1]
200 UTF-8 activÃ©
Utilisateur (192.168.40.1:(none)) :

nox44@Debian: ~
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\Nox44>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . :
    Adresse IPv6. . . . . : 2a01:e0a:44e:5030:e905:eeb3:bf1a:ec04
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:44e:5030:91b1:20db:7464:545f
    Adresse IPv6 de liaison locale. . . . : fe80::11a2:b051:185c:2835%4
    Adresse IPv4. . . . . : 192.168.1.20
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : fe80::de00:b0ff:fe67:9a2%4
                                      192.168.1.179

C:\Users\Nox44>ssh nox44@192.168.40.1
nox44@192.168.40.1's password:
Linux Debian 6.1.0-18-amd64 #1 SMP PREEMPT_DYNAMIC Debian 6.1.76-1 (2024-02-01) x86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Mar 23 18:10:26 2024 from 192.168.1.20
nox44@Debian:~$
```

Les tests sur notre réseau Administration sont donc tous concluants, par rapport au cahier des charges.

Passons désormais aux tests sur notre réseau Formation.

Tests d'accès au réseau Formation :

Pour ce qui est des accès aux utilisateurs du réseau Formation, pour lequel nous avons l'utilisateur dont l'ip est 192.168.30.10, nous pouvons voir qu'ils ont bien accès aux sites des deux écoles, ainsi qu'à internet.

```
Invite de commandes

C:\Users\Nox44>ipconfig

Configuration IP de Windows

Carte Ethernet Ethernet :

    Suffixe DNS propre à la connexion. . . :
    Adresse IPv6. . . . . : 2a01:e0a:44e:5030:e905:eeb3:bf1a:ec04
    Adresse IPv6 temporaire . . . . . : 2a01:e0a:44e:5030:91b1:20db:7464:545f
    Adresse IPv6 de liaison locale. . . . : fe80::11a2:b051:185c:2835%4
    Adresse IPv4. . . . . : 192.168.30.10
    Masque de sous-réseau. . . . . : 255.255.255.0
    Passerelle par défaut. . . . . : fe80::de00:b0ff:fe67:9a2%4
                                      192.168.30.1

C:\Users\Nox44>ping 8.8.4.4

Envoi d'une requête 'Ping' 8.8.4.4 avec 32 octets de données :
Réponse de 8.8.4.4 : octets=32 temps=21 ms TTL=115
Réponse de 8.8.4.4 : octets=32 temps=15 ms TTL=115
Réponse de 8.8.4.4 : octets=32 temps=19 ms TTL=115
Réponse de 8.8.4.4 : octets=32 temps=16 ms TTL=115

Statistiques Ping pour 8.8.4.4:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 15ms, Maximum = 21ms, Moyenne = 17ms

C:\Users\Nox44>ping www.mds.lan

Envoi d'une requête 'ping' sur 192.168.40.1.mds.lan [192.168.40.1] avec 32 octets de données :
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63

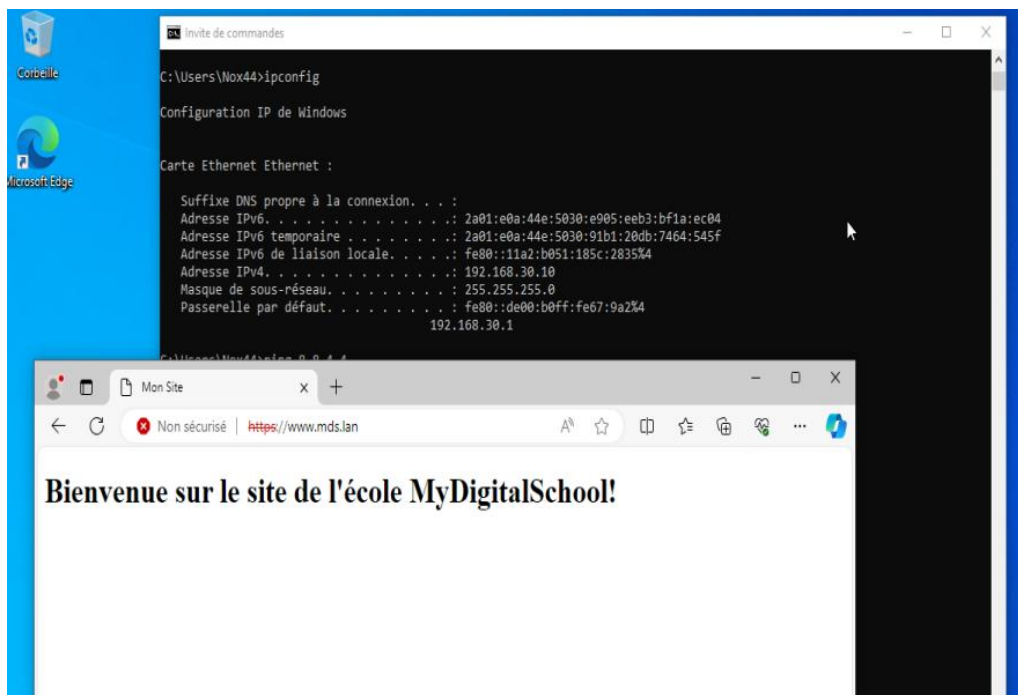
Statistiques Ping pour 192.168.40.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 1ms, Moyenne = 1ms

C:\Users\Nox44>ping www.mbway.lan

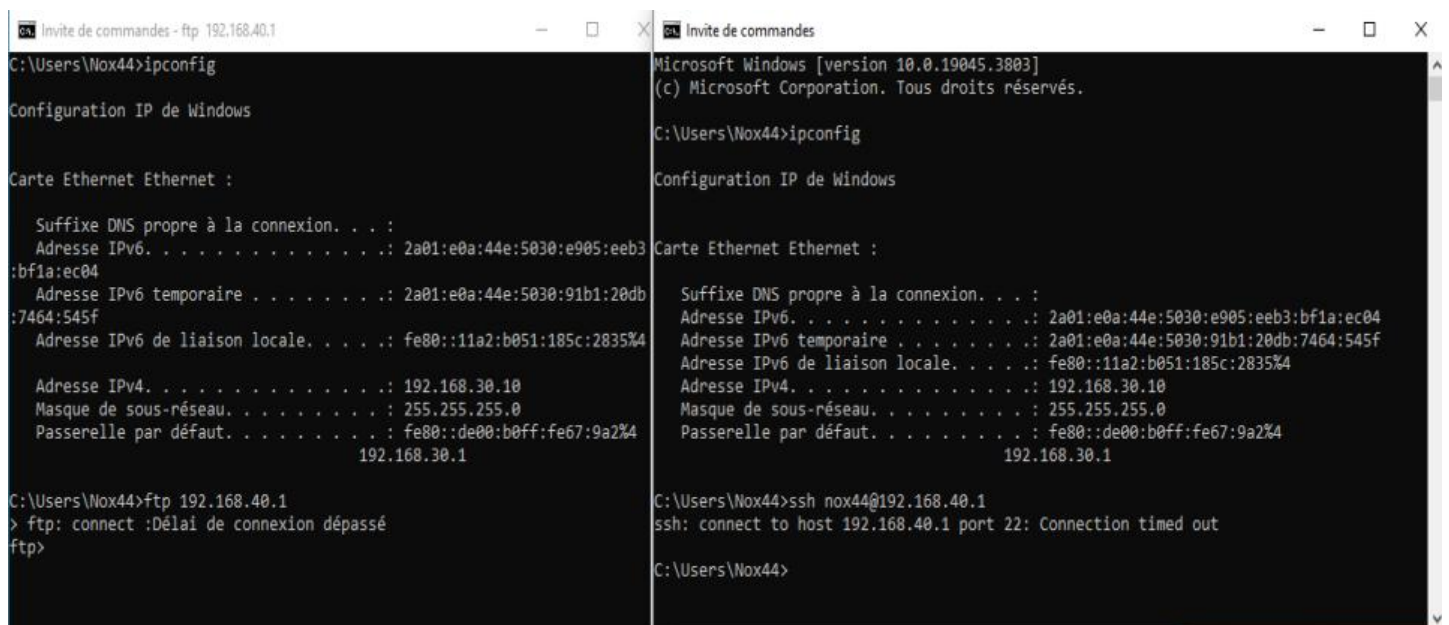
Envoi d'une requête 'ping' sur 192.168.40.1.mbway.lan [192.168.40.1] avec 32 octets de données :
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=1 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=3 ms TTL=63
Réponse de 192.168.40.1 : octets=32 temps=2 ms TTL=63

Statistiques Ping pour 192.168.40.1:
    Paquets : envoyés = 4, reçus = 4, perdus = 0 (perte 0%),
Durée approximative des boucles en millisecondes :
    Minimum = 1ms, Maximum = 3ms, Moyenne = 1ms

C:\Users\Nox44>
```

Également, de même que pour les utilisateurs du réseau Administration, à l'exception de l'Administrateur du campus, nous pouvons aussi confirmer qu'ils n'ont pas accès au serveur FTP, ni à une connexion via SSH.



Les règles sur notre réseau Formation sont donc également correctement appliquées, et l'intégralité des demandes du cahier des charges ont été respectées.

Possibilités d'amélioration

Voici quelques exemples d'amélioration possibles pour notre passerelle et pour notre DMZ :

- Créer des comptes utilisateurs pour mieux gérer les accès de chaque utilisateur, avec des rôles ayant des autorisations spécifiques
- Permettre une modification définitive du fichier `/etc/resolv.conf`, fichier permettant la configuration du DNS, qui ne se sauvegarde pas au redémarrage de la machine.
- Renforcer la sécurité du compte administrateur, avec une double authentification par exemple (app authenticator ou envoi d'un sms à la connexion).

Cédric Le Meur
Mars 2024