I. Création des systèmes

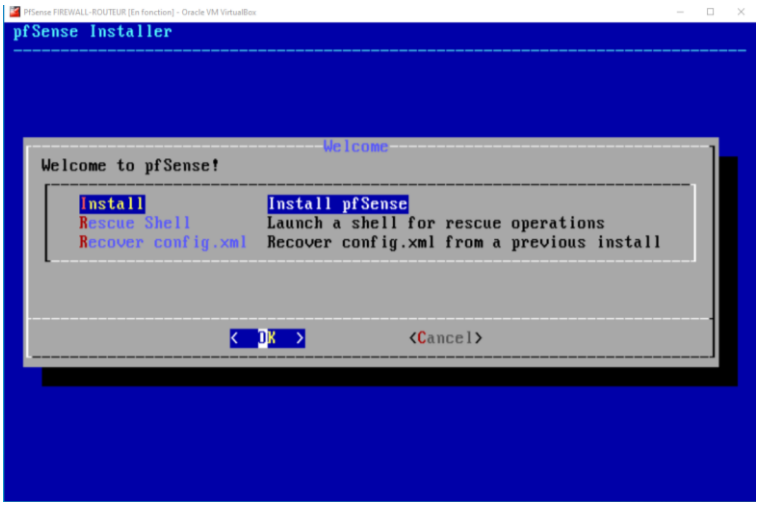
1. PfSense

Nous traiterons dans cette première partie, de la création d'une machine virtuelle simulant un pare-feu physique PfSense. La VM nécessite de simples paramétrages de base. Il convient cependant de configurer trois interfaces réseaux : la première en **bridge**, la seconde et la troisième en **réseau interne** (une appelée **LAN** et l'autre **DMZ**).

Installation

Il s'agit ici de lancer la VM, et d'attendre jusqu'à arriver à cette fenêtre :

Faire Entrée pour accepter, puis Entrée à nouveau pour installer le système :



Après cela, suivre les étapes : choisir la langue du clavier (Français pour nous), puis partionner automatiquement (débutant). Une fois l'installation terminée, faire « NO » à la demande d'ouverture d'un **SHELL**. Enfin, faire « Yes » à la demande de reboot. Nous pouvons ensuite arrêter la VM.

2. Serveur WEB

Dans cette partie, nous verrons comment héberger un site WEB et le configurer, grâce à des outils tels que le serveur *Apache HTTPD*, *PHP, MySQL*. Ainsi, notre objectif sera de pouvoir accéder via notre navigateur à une page WEB générique, qui nous permettra d'interagir avec notre base de données, par une authentification. On appelle cela un serveur LAMP : Linux, Apache, MySQL, PHP.

Nous utiliserons une machine virtuelle sous Debian pour héberger tous les services nécessaires à cette réalisation. Il est fortement conseillé d'installer Debian en non-graphique pour optimiser les performances de la machine. Nous pouvons configurer deux interfaces réseaux : une en **bridge**, l'autre en **réseau interne** appelée **DMZ**. Nous désactiverons la première une fois l'installation et le paramétrage du serveur terminé.

a. Création du site web sous apache

Dans cette partie, nous verrons comment installer et configurer Apache pour l'hébergement d'un site WEB rudimentaire.

Installation & configuration d'Apache

Passons en utilisateur racine :

```
su
```

Debian possède un package d'Apache, qu'il faut télécharger & installer via le gestionnaire de package :

```
apt-get install apache2
```

Apache est capable de gérer plusieurs sites web sur la même machine, c'est ce qu'on appelle des hôtes virtuels ou virtual hosts. Dans la requête que fera le client sur la machine, un entête HTTP précisera si le client veut plutôt consulter *toto.com* ou *tata.org* qui sont tous les deux hébergés sur la machine.

Configuration des hôtes virtuels

La configuration de chaque virtual host se fait dans le ficher : **/etc/apache2/sites-available/01-www.monsite.com.conf**. Créons ce fichier :

```
cd /etc/apache2/sites-available/
nano 01-www.monsite.com.conf
```

Tout d'abord, on remarque que la configuration d'un virtual host est comprise dans des balises "type HTML" c'est-à-dire **<VirtualHost *:80></VirtualHost>**. Le ***:80** de la balise ouvrante indique que le virtual host peut être utilisé sur le port 80 sur toutes les interfaces.

Détail des directives

Voici ensuite le détail des directives utilisées :

ServerName	Précise l'hôte pour lequel cette configuration sera utilisée. Il ne peut y en avoir qu'un seul par hôte virtuel.
ServerAlias	Il ne peut y avoir qu'un **ServerName** mais par contre, nous pouvons compléter par autant de **ServerAlias** que nous voulons séparés par des espaces
ServerAdmin	Indique une adresse mail de contact et qui peut être affichée sur certains messages d'erreur.
DocumentRoot	C'est la racine de l'arborescence de notre site web, là où nous mettrons tous nos fichiers.
CustomLog	Demande de stocker les fichiers de logs au format **combined** dans un fichier à part plutôt que dans le fichier de logs général.
ErrorLog	Demande de stocker aussi les logs d'erreur dans un fichier à part mais leur format est fixe.
<Directory />	Dans la balise Directory, nous pouvons définir des règles qui s'appliquent uniquement au contenu d'un répertoire. Ici, nous activerons toutes les options. Voici une partie des options activées :
	ExecCGI : l'exécution de scripts à l'aide du module CGI est permise.
	FollowSymlinks : le serveur va suivre les liens symboliques. Cette option est la seule active par défaut.
	Includes : les inclusions côté serveur à l'aide du module **mod_include** sont autorisées
	Indexes : si aucun fichier par défaut type index.html n'est présent, le module **mod_autoindexes** va présenter une liste des fichiers et répertoires formatée par Apache
AllowOverride None	Cette directive placée dans **<Directory />** indique qu'aucune option ne peut être surchargée par les options qui seraient contenues dans un fichier appelé **.htaccess** placé dans cette arborescence. Nous aurions pu préciser une à une les options qui peuvent être surchargées ou utiliser **All** pour autoriser la surcharge de toutes les options : dans ce cas, les options du fichier **.htaccess** auront la priorité sur celles du Virtual Host.

Entrons donc ces paramètres :

```
<VirtualHost *:80>
    ServerName www.monsite.com
    ServerAlias monsite.com
    ServerAdmin webmaster@ monsite.com
    DocumentRoot /var/www/html/www.monsite.com

    CustomLog ${APACHE_LOG_DIR}/www.monsite.com-access.log combined
    ErrorLog ${APACHE_LOG_DIR}/www.monsite.com-error.log

    <Directory /var/www/html/www.monsite.com>
        Options All
        AllowOverride None
    </Directory>
</VirtualHost>
```

Premier test

Pour pouvoir tester notre site web, il nous reste à :

- Créer la racine de notre arborescence et y déposer un premier fichier **.html** :

```
mkdir /var/www/html/www.monsite.com/
cp /var/www/html/index.html /var/www/html/www.monsite.com
```

L’utilisateur Apache (par défaut www-data) doit avoir les droits de lecture sur ce répertoire et ses fichiers. C’est le cas par défaut.

- Activer la configuration de notre VirtualHost (si la commande n’est pas trouvée, quitter le compte racine via **exit** et retournons-y avec un tiret, pour permettre la pris en compte de variables environnement des autres comptes :

```
exit
su -
a2ensite 01-www.monsite.com
```

Cette commande crée automatiquement le lien symbolique du répertoire **sites-enabled** vers le répertoire **sites-available**.

De la même façon, pour supprimer le lien symbolique, Nous pouvons utiliser la commande **a2dissite**. De même, pour activer/désactiver des modules, nous pouvons utiliser **a2enmod** et **a2dismod**.

- Recharger la configuration d’Apache pour prendre en compte nos modifications :

```
systemctl reload apache2
```

Notre serveur est prêt. Maintenant si nous ajoutons **www.monsite.com** en alias à l’adresse IP de notre serveur, dans le fichier **C:\Windows\System32\drivers\etc\hosts** de notre machine physique sous Windows 10, nous pouvons admirer la page d’accueil de notre site dans notre navigateur. Le logiciel gratuit **HostMan** peut nous aider à modifier un tel fichier système ; il est nativement protégé par des règles très strictes.

b. Sécurisation du serveur WEB

Dans ce chapitre, nous verrons deux façons de renforcer la sécurité d’un site web :

- En évitant de donner des informations sur notre architecture
- En restreignant l’accès à certaines ressources

Rendre Apache opaque

En matière de sécurité, il est toujours préférable de donner le moins d'informations possibles sur son système à de potentiels attaquants. Or, par défaut, Apache renvoie des entêtes HTTP contenant le nom et la version du serveur web ainsi que le nom du système d'exploitation.

Pour éviter ça, Debian nous facilite la vie en regroupant les directives sensibles en termes de sécurité dans le fichier **/etc/apache2/conf-available/security.conf**. Dans ce fichier, configurons les directives comme ceci afin de donner le moins d'informations :

```
ServerTokens Prod
ServerSignature Off
```

Nous pouvons également modifier la page d'accueil par défaut d'Apache, en remplaçant le fichier **init.html** par le classique **«** Hello World ! ». Pour cela, il faut créer un fichier caché appelé **.htaccess :**

```
nano /var/www/html/www.monsite.com/.htaccess
```

Ajoutons-y seulement la ligne :

```
DirectoryIndex acceuil.html
```

Ensuite, créons **acceuil.html** dans le même répertoire, via **nano**, et insérons-y ces paramètres :

```
<!doctype html>
<html lang="fr">
<head><meta charset="utf-8"><meta http-equiv="X-UA-Compatible"
content="IE=edge"><title>Mon site de test</title></head>
<body>Hello world</body>
</html>
```

Après quoi modifions le ficher **apache2.conf**, situé *dans /etc/apache2* :

```
nano /etc/apache2/apache2.conf
```

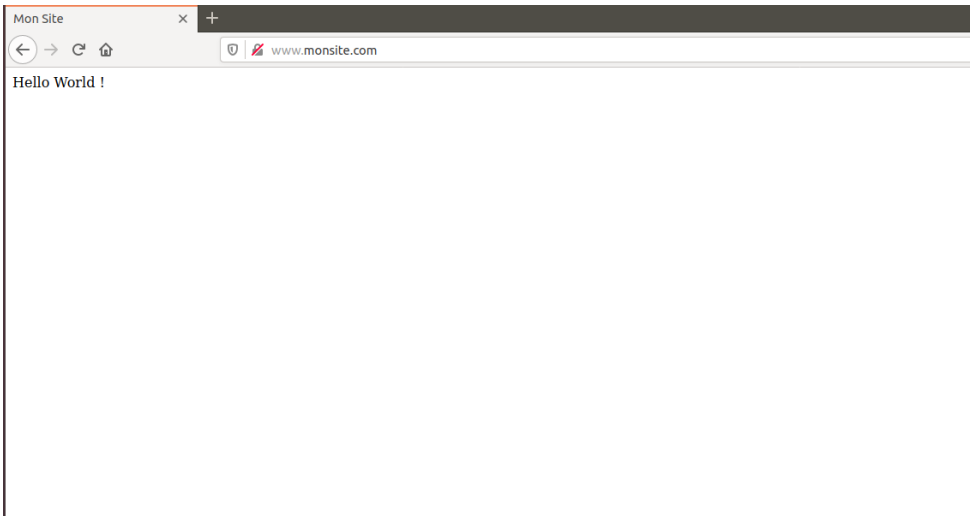
A la ligne **<Directory /var/www/>**, modifions les paramètres ainsi :

```
<Directory /var/www/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride None
        Require all granted
</Directory>
```

Rechargeons ensuite la configuration d'Apache :

```
systemctl reload apache2
```

Désormais, notre page d'accueil ressemble à cela :



Restriction d'accès par authentification

Apache permet de restreindre l'accès à des ressources en fonction d'un grand nombre de critères. Les modes d'authentification les plus courants sont par IP et par login/mot de passe.

Commençons par créer les données à protéger :

```
mkdir /var/www/html/www.monsite.com/top_secret
echo "voici mon secret" > /var/www/html/www.monsite.com/top_secret/index.html
```

Pour protéger notre répertoire **top_secret**, ajoutons la section suivante à la configuration de notre **<VirtualHost />** dans **/etc/apache2/sites-available/01-www.monsite.com.conf** et rechargeons la configuration Apache :

```
<Directory /var/www/html/www.monsite.com/top_secret>
AuthType Basic
    AuthName "Accès restreint aux utilisateurs authentifiés"
    AuthBasicProvider file
    AuthUserFile "/etc/apache2/passwords"
    Require ip 192.168.0.3
    Require valid-user
</Directory>
```

Après cela, nous pouvons nous rendre à l'adresse **www.monsite.com/top_secret** via le navigateur de notre machine physique.

Directives d'authentification

La directive qui permet de restreindre l'accès à une ressource est **Require**.

Par défaut, les directives sont évaluées dans l'ordre : dès qu'une condition est valide, l'accès est autorisé, sinon on passe à la suivante. Quand aucune directive ne correspond, l'accès est refusé.

Ici, toutes les requêtes venant de l'IP 192.168.0.3 sont automatiquement acceptées. Une fenêtre d'authentification est proposée aux requêtes venant d'autres IP. Un couple login/mot de passe valide doit être entré pour que l'accès soit autorisé.

Les directives **Auth*** précisent le fonctionnement de cette autorisation par mot de passe :

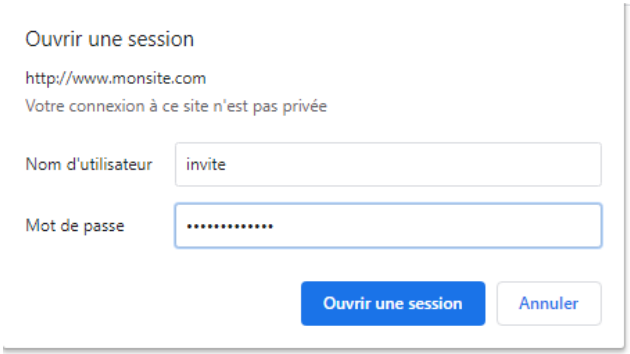
AuthType Basic	Indique une authentification par login/mot de passe simple.
AuthName	Permet de définir le titre de la fenêtre d'authentification qui sera affiché aux clients.
AuthBasicProvider file	Indique que les comptes utilisateurs sont définis dans un fichier.
AuthUserFile	Précise le fichier où sont enregistrés les comptes utilisateurs.

Il nous faut donc maintenant créer ce fameux fichier de mots de passes grâce à la commande **htpasswd** :

```
htpasswd -c /etc/apache2/passwords invite
```

On nous demande de définir un mot de passe et voilà, nous venons de créer l'utilisateur "invite". L'option **-c** n'est à utiliser que pour le premier utilisateur car elle crée le fichier s'il n'existe pas.

Nous pouvons tester en nous connectant à **http://www.monsite.com/top_secret/**, en ayant au préalable mis en commentaire la ligne « **Require ip 192.168.0.3** ».



Une autre méthode consiste à utiliser un annuaire LDAP pour l'authentification (nécessite un Active Directory). C'est utile pour la création de portails internes, mais ce n'est pas notre cas. (Voir Documentation Serveur LAMP)

Nous pouvons remarquer que les authentifications par mot de passe que nous avons mises en place transmettent le mot de passe en clair. Ce n'est pas une bonne pratique de sécurité car toute personne ayant accès au réseau pourrait lire ces mots de passe. Ce problème est réglable en mettant en place un protocole de chiffrement de la connexion tel que HTTPS, mais nous ne verrons pas cela ici (par soucis de simplicité et de mis en place rapide). (Voir Documentation Serveur LAMP).

c. Installation & configuration de PHP

De manière générale, il y a deux manières d'exécuter des programmes avec Apache :

- Sous forme d'un module intégré à Apache, ce qui permet aux processus Apache de gérer eux-mêmes l'exécution du programme.
- Sous forme d'un script CGI. Dans ce cas, Apache délègue l'exécution du programme à un processus tiers.

Nous verrons ici comment installer PHP sous forme de module Apache, par soucis de simplicité et rapidité (voir Documentation Serveur LAMP pour la méthode script CGI).

Installer PHP sous forme de module Apache

La manière la plus classique et la plus simple d'installer PHP est de l'installer sous forme d'un module Apache. Pour cela, il nous suffit d'installer le module **libapache2-mod-php** :

```
apt-get install libapache2-mod-php
```

PHP est alors installé ainsi que le module **mod-php** qui permet à Apache d'utiliser ce langage.

Pour tester l'installation de PHP sur notre système, dans le répertoire */var/www/html/www.monsite.com/top_secret* de notre site web, supprimons le fichier **index.html** et créons le fichier **index.php** contenant :

```
<?php
phpinfo();
?>
```

Après s'être authentifiés, nous voyons s'afficher beaucoup d'informations sur notre système, notre configuration Apache et notre configuration PHP. C'est pourquoi il ne faut jamais laisser une page phpinfo() accessible publiquement.

On trouve la configuration de PHP dans le répertoire */etc/php/7.2* où 7.2 est notre version de PHP :

```
$ ls /etc/php/7.2
apache2 cli mods-available
```

On y trouve 3 répertoires :

- **Apache2** est la configuration utilisée par le module PHP d'Apache
- **Cli** est la configuration utilisée par le client en ligne de commandes PHP
- **Mods-available** est le répertoire où sont installés tous les modules PHP

```
$ ls /etc/php/7.2/apache2
conf.d php.ini
```

Dans le répertoire **apache2** (et pareil pour **cli**), nous trouverons le fichier de configuration général **php.ini** et la configuration propre à chaque module dans le répertoire **conf.d.**

*Lisons bien les commentaires au début de notre fichier **php.ini** qui indiquent quelles valeurs sont préconisées pour un environnement de production. Le reste de la configuration est à ajuster en fonction des applications PHP que nous désirerons installer.*

d. Installation & configuration de MySQL

Dans cette partie, nous verrons comment installer une base de données rudimentaire, afin de compléter notre serveur LAMP. Nous ne nous en servirons pas vraiment, mais cela peut être utile dans notre compréhension du fonctionnement du site WEB. Cela permettra également de se rapprocher d'une situation réaliste.

Notre première installation a été particulièrement compliquée. En cas de difficultés, consulter la Documentation Complète Serveur LAMP.

Installation de MySQL

Commençons par vérifier que notre système est à jour :

```
apt update && apt -y upgrade
```

Pour installer MySQL, il nous suffit d'installer le paquet **mysql-server** :

```
apt-get install mysql-server
```

La première chose à faire ensuite, c'est sécuriser l'installation par défaut de MySQL qui n'est pas adaptée à un environnement de production. MySQL nous fournit l'utilitaire **mysq_secure_installation** pour nous faciliter la vie :

```
mysql_secure_installation
```

Le jeu avec cet utilitaire, c'est de répondre oui à toutes les questions. Ça va notamment nous permettre de définir un mot de passe administrateur (root) pour notre base de données, de n'autoriser que sa connexion en local, d'interdire les connexions anonymes et de supprimer la base de test. Nous verrons les détails de cette procédure dans la partie Serveur Base de Données.

Maintenant que MySQL est installée, nous pouvons faire un premier test de connexion en utilisant le compte root.

Première connexion

Le compte root n'est maintenant autorisé à se connecter à MySQL qu'en local depuis le serveur. Nous pouvons nous connecter par la commande :

```
mysql -h localhost -u root -p
```

mysql	C'est le nom du client texte qui permet de se connecter à MySQL.

-h localhost	Précise le nom de l'hôte auquel nous voulons nous connecter. localhost est l'hôte par défaut donc nous pouvons nous passer de cette option pour les connexions locales.
-u root	Indique le nom de l'utilisateur avec lequel nous voulons nous connecter.
-p	Demande d'afficher un prompt pour entrer le mot de passe de manière interactive.

Un fois connecté, nous pouvons par exemple afficher les bases de données existantes par la commande **show databases** :

```
mysql> show databases;
+----------------------------+
| Database                   |
+----------------------------+
| information_schema         |
| mysql                      |
| performance_schema         |
| sys                        |
+----------------------------+
mysql> quit;
```

Ici les 4 bases affichées sont toutes des bases de données gérées par MySQL pour assurer son fonctionnement.

Pou

r aller plus loin si besoin est, voir la documentation Serveur LAMP, partie MySQL.

3. Serveur Base de données

Il existe plusieurs moteurs de base de données relationnelle : MySQL, MariaDB, PostgreSQL … Dans le cadre de notre exercice, nous utiliserons MySQL. Nous l'installerons sur une VM sous Debian, comme notre serveur LAMP ; il n'y a d'ailleurs pas de configuration particulière pour sa création. Nous pouvons configurer deux interfaces réseaux : une en **bridge**, l'autre en **réseau interne** appelée **LAN**. Nous désactiverons la première une fois l'installation et le paramétrage du serveur terminé.

Notre première installation a été particulièrement compliquée. En cas de difficultés, consulter la Documentation Complète Serveur LAMP.

a. Installation du serveur de base de données : MySQL Server

Installation et paramétrage de MySQL

Chaque commande est entrée avec le compte root.

Mettre à jour le système :

```
Apt update && apt -y upgrade
```

Installer le paquet **gnupg** :

```
apt install gnupg
```

Télécharger directement le paquet de configuration du paquet **mysql-server.** Pour cela, se rendre sur https://dev.mysql.com/downloads/repo/apt/ et noter la version du paquet proposé. Ensuite, entrer la commande :

```
wget https://dev.mysql.com/get/mysql-apt-config_0.8.15-1_all.deb
```

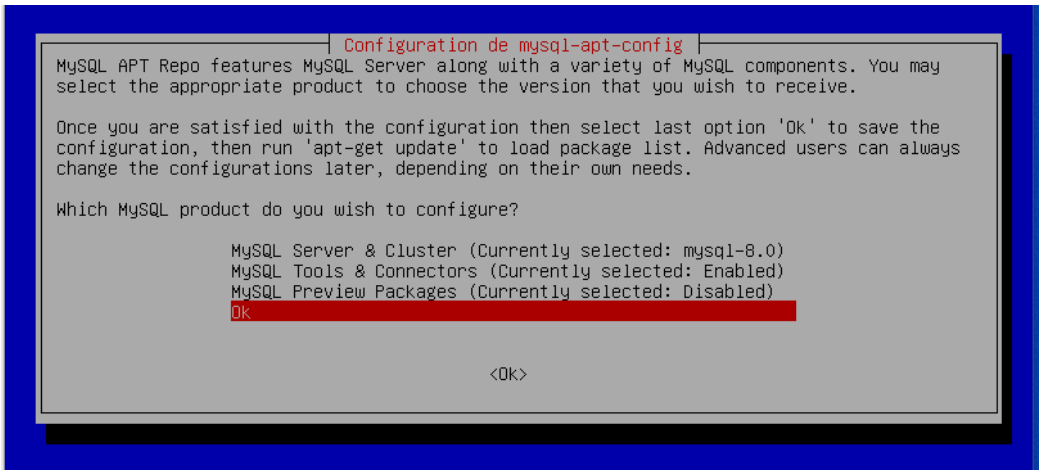
Avec bien-sûr, le nom du paquet que vous avez relevé à la place de celui mit en exemple dans la commande précédente.

Ensuite, extraire ce paquet :

```
dpkg -i mysql-apt-config_0.8.15-1_all.deb
```

Toujours avec le nom du paquet que vous avez relevé à la place de celui mit en exemple.

La fenêtre suivante devrait s'ouvrir : (aucun paramétrage particulier à entrer ; après avoir fait « Ok », le prompt devrait nous renvoyer « OK » si tout s'est passé correctement.



Après cela, on peut remettre à jour le système :

```
apt update && apt -y upgrade
```

Et lancer l'installation de MySQL :

```
Apt install mysql-server
```

Il se peut que le prompt nous demande de rentrer la commande suivante, ce qu'il faudra faire avant l'installation :

```
Apt --fix-broken install
```

La commande **apt install mysql-server** permet l'installation brute de MySQL ; pour apporter une touche de configuration, faire :

```
mysql_secure_installation
```

MySQL nous propose d'installer un plugin de gestion de mots de passe. Dans notre simulation de déploiement professionnel, nous l'installons en entrant « yes ». Ensuite, il nous est proposé de choisir la complexité du mot de passe du compte root pour le serveur MySQL : 0 pour une complexité basse, 1 pour moyenne, 2 pour forte. Nous choisirons 0 pour aller plus vite. Mais dans le cadre professionnel, l'option 2 est conseillée. Choisissons un mot de passe, et confirmons-le.

```
VALIDATE PASSWORD PLUGIN can be used to test passwords
and improve security. It checks the strength of password
and allows the users to set only those passwords which are
secure enough. Would you like to setup VALIDATE PASSWORD plugin?

Press y|Y for Yes, any other key for No: Yes

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: domino13

Invalid option provided.

There are three levels of password validation policy:

LOW    Length >= 8
MEDIUM Length >= 8, numeric, mixed case, and special characters
STRONG Length >= 8, numeric, mixed case, special characters and dictionary                  file

Please enter 0 = LOW, 1 = MEDIUM and 2 = STRONG: 0
Please set the password for root here.

New password:

Re-enter new password:

Estimated strength of the password: 50
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : Yes
```

Ensuite, le prompt nous demande si nous souhaitons désactiver la connexion via utilisateur anonyme. Dans notre situation, nous la laisserons activée (en saisissant « no »). Elle est cependant à désactiver dans un cadre de mise en fonction réelle.

```
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) : No

 ... skipping.
```

Après cela, on nous propose de désactiver l'accès au compte root par connexion distante. Comme pour le serveur SSH, il faut effectivement le désactiver, par sécurité.

```
Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) : No

 ... skipping.
```

Maintenant, le prompt nous demande si nous souhaitons supprimer une base de données par défaut, appelée « test ». Elle peut nous être utile ; saisissons « no ».

```
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.


Remove test database and access to it? (Press y|Y for Yes, any other key for No) : No

 ... skipping.
```

Enfin, nous pouvons recharger les tables de privilèges, pour appliquer tous les paramètres que nous avons saisis.

```
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) : Yes
Success.

All done!
root@oscar-VirtualBox:/home/sshuser#
```

MySQL est configuré. Nous pouvons désormais créer un utilisateur, avec les droits nécessaires sur le serveur MySQL.

Création d'un utilisateur MySQL

Pour accéder au serveur MySQL en utilisateur basique (toujours via le compte root) :

```
mysql
```

Pour accéder au serveur en compte racine (seulement en local, comme définit dans le paramétrage) :

```
mysql –u root -p
```

Le -u permet le choix du compte auquel on accède ; le –p demande un accès via mot de passe.

À partir de là, créons un nouvel utilisateur appelé « administrateur » et attribuons-lui un mot de passe :

```
CREATE USER 'administrateur'@'localhost' IDENTIFIED BY 'motdepasse' ;
```

Ensuite, donnons à **administrateur** les privilèges appropriés. Par exemple, nous pouvons lui accorder accès à toutes les tables au sein de la base de données ainsi que le pouvoir d'ajouter, changer ou supprimer des privilèges d'utilisateurs :

```
GRANT ALL PRIVILEGES ON *.* TO 'administrateur'@'localhost' WITH GRANT OPTION;
```

Attention : ici, **administrateur** n'aura accès au MySQL uniquement par la machine qui héberge le service (soit notre machine SRV-BDD). Pour permettre l'accès au service à partir d'une autre

machine, il faudra réitérer les étapes précédentes, en précisant l'adresse ip de la machine avec laquelle on souhaite se connecter. Ainsi, si l'adresse ip de notre PC-ADMIN est 192.168.50.6, il faudra entrer :

```
CREATE USER 'administrateur'@'192.168.50.6/29' IDENTIFIED BY 'motdepasse' ;
```

Et donc également :

```
GRANT ALL PRIVILEGES ON *.* TO 'administrateur'@'192.168.50.6/29' WITH GRANT
OPTION;
```

On peut désormais quitter le shell de MySQL, en entrant « exit ».

Gérer le serveur MySQL

Le serveur MySQL est supposé s'être lancé automatiquement après son installation. Pour vérifier s'il est actif ou non, faisons :

```
systemctl status mysql.service
```

Voici ce qui en ressort si le serveur est actif :

```
root@oscar-VirtualBox:/home/sshuser# systemctl status mysql.service
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Sat 2020-01-11 18:01:12 CET; 20h ago
 Main PID: 4532 (mysqld)
    Tasks: 29 (limit: 4915)
   CGroup: /system.slice/mysql.service
           └─4532 /usr/sbin/mysqld --daemonize --pid-file=/run/mysqld/mysqld.pid

janv. 11 18:01:12 oscar-VirtualBox systemd[1]: Starting MySQL Community Server...
janv. 11 18:01:12 oscar-VirtualBox systemd[1]: Started MySQL Community Server.
root@oscar-VirtualBox:/home/sshuser#
```

Si le secteur est inactif, lançons-le avec :

```
systemctl start mysql.service
```

Pour l'arrêter, faire :

```
systemctl stop mysql.service
```

4. Poste admin

Il s'agit ici de créer une machine virtuelle avec un paramétrage basique, qui servira de poste administrateur pour notre serveur de base de données. Nous installerons le système d'exploitation Ubuntu avec interface graphique, ainsi que le logiciel MySQL Workbench. Nous pouvons configurer deux interfaces réseaux : une en **bridge**, l'autre en **réseau interne** appelée **DMZ**. Nous désactiverons la première une fois l'installation et le paramétrage du serveur terminé.

Sur le PC-ADMIN, installer MySQL Workbench via **Logiciels Ubuntu** :

mysql workbench
MySQL Workbench
MySQL Workbench est un outil de modélisation qui permet de concevoir et de générer des bases de données MySQL graphiquement. Il a également des modules ...
Installée

II. Mise en relation des systèmes

1. Mise en place de PfSense

a. Configuration IP des VMs

Dans cette partie, nous allons voir comment fixer de nouvelles adresses ip statiques à nos machines, et les configurer pour qu'elles puissent communiquer entre elles. Nous n'avons pas encore configuré les interfaces de PfSense, cela viendra tout de suite après ; inutile donc de s'inquiéter si nos systèmes ne peuvent ping leur passerelle, puisqu'elle n'existe pas encore.

> *Pour s'assurer du bon déroulement de ces opérations, il convient d'activer uniquement les interfaces réseaux en **réseau interne** sur chacune de nos VMs (et les créer si elles n'existent pas). Ainsi nous pourrons vérifier que les connexions se font directement entre elles, sans passer par notre machine physique.*

Configuration du PC-ADMIN

Nous devons maintenant fixer l'adresse ip correspondante à notre machine (celle que l'on a choisi). Dans *Paramètres > Réseau,* cliquer sur l'icône de configuration du réseau filaire. Dans l'onglet IPv4, choisir la méthode **Manuel**, puis fixer l'adresse et le masque correspondant (`192.168.50.2 – 255.255.255.248`). Puis insérer dans « passerelle » l'adresse de passerelle que l'on a fixé précédemment (soit *192.168.50.1*). Faire ensuite valider.

Configuration de SRV-BDD

Le changement d'adresse ip est plutôt rapide sous Debian. Allons dans le fichier **interfaces**, qui stocke les paramètres des interfaces réseaux :

```
Nano /etc/network/interfaces
```

Repérons la ligne de notre interface réseau principale. Rentrons les paramètres suivants (en remplacement l'option **DHCP** par **static** si nécessaire) :

```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
        address 192.168.50.6
        netmask 255.255.255.248
        gateway 192.168.50.1
```

Redémarrons l'interface réseau :

```
etc/init.d/networking restart
```

Faire un **ip a** pour vérifier que votre interface a bien été modifiée ; si ce n'est pas le cas, redémarrer le serveur.

Configuration de SRV-WEB

Ici, il faudra appliquer la même procédure que précédemment. La différence réside dans l'adresse, le masque et la passerelle :

```
# The primary network interface
allow-hotplug enp0s3
iface enp0s3 inet static
        address 172.31.0.2
        netmask 255.255.255.252
        gateway 172.31.0.1
```

b. Configuration des interfaces de PfSense

Lors de son premier lancement, PfSense nous demande si nous souhaitons configurer nos VLANs ; ce n'est pas ce que l'on veut. Faire non, et configurer ensuite l'interface WAN, selon ce que PfSense nous demande. Le seul principe, c'est de ne jamais activer un service DHCP lorsque ça nous est proposé, et d'activer les interfaces dans l'ordre : `em0` pour WAN et `em1` pour LAN. Nous configurerons `em2` plus tard.

Configuration de la première interface : WAN

On arrive donc sur ce menu :

```
                http://192.168.50.1/

Press <ENTER> to continue.
VirtualBox Virtual Machine - Netgate Device ID: c6546780b47c7807db1d

*** Welcome to pfSense 2.4.4-RELEASE (amd64) on pfSense ***

 WAN (wan)       -> em0        -> v4/DHCP4: 192.168.0.36/24
 LAN (lan)       -> em1        ->

 0) Logout (SSH only)                  9) pfTop
 1) Assign Interfaces                 10) Filter Logs
 2) Set interface(s) IP address       11) Restart webConfigurator
 3) Reset webConfigurator password    12) PHP shell + pfSense tools
 4) Reset to factory defaults         13) Update from console
 5) Reboot system                     14) Enable Secure Shell (sshd)
 6) Halt system                       15) Restore recent configuration
 7) Ping host                         16) Restart PHP-FPM
 8) Shell

Enter an option:
```

Pour configurer nos interfaces, faisons **2**, pour **Set Interface(s) IP address** :

```
Available interfaces:

1 - WAN (em0 - static)
2 - LAN (em1 - static)
3 - OPT1 (em2 - static)
```

Faisons 1 pour choisir l'interface WAN, puis rentrons l'adresse ip et le CIDR que l'on souhaite assigner à notre interface (doit être sur le même réseau que notre box) :

```
Enter the new WAN IPv4 address.  Press <ENTER> for none:
> 192.168.0.2

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8

Enter the new WAN IPv4 subnet bit count (1 to 31):
> 24
```

Ensuite, entrer ladresse ip de notre box :

```
For a WAN, enter the new WAN IPv4 upstream gateway address.
For a LAN, press <ENTER> for none:
> 192.168.0.1
```

Refuser les configurations d'IPv6 qui arrivent ensuite. Faire **y** si on nous demande d'activer le retour en HTTP du configurateur WEB.

Notre interface WAN est maintenant configurée. Dans un shell, nous pouvons désormais faire un test de ping vers **google.fr :**

```
[2.4.4-RELEASE][root@pfSense.localdomain]/root: ping google.fr
PING google.fr (216.58.198.195): 56 data bytes
64 bytes from 216.58.198.195: icmp_seq=0 ttl=54 time=17.882 ms
64 bytes from 216.58.198.195: icmp_seq=1 ttl=54 time=17.810 ms
```

Configuration de la deuxième interface : LAN

Pour configurer nos interfaces, faisons **2**, pour **Set Interface(s) IP address**, puis **2** :

```
Available interfaces:

1 - WAN (em0 - static)
2 - LAN (em1 - static)
3 - OPT1 (em2 - static)
```

Entrons ensuite l'adresse et le CIDR que l'on a choisi pour l'interface LAN dans notre définition de l'environnement (soit **`192.168.50.1/29`**) :

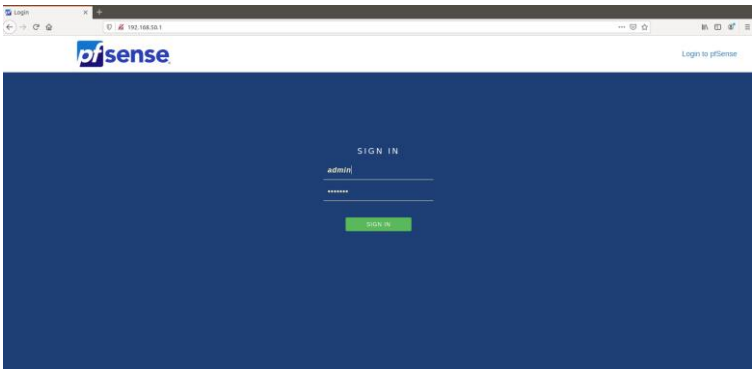
```
Enter the new LAN IPv4 address.  Press <ENTER> for none:
> 192.168.50.1

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8

Enter the new LAN IPv4 subnet bit count (1 to 31):
> 29
```

On nous demande d'entrer une adresse de passerelle, seulement pour interfaces WAN. Laisser vide et faire **Entrée**. Faire **Entrée** pour ne pas fixer d'adresse IPv6. Faire **n** pour ne pas activer le DHCP le réseau de notre interface.

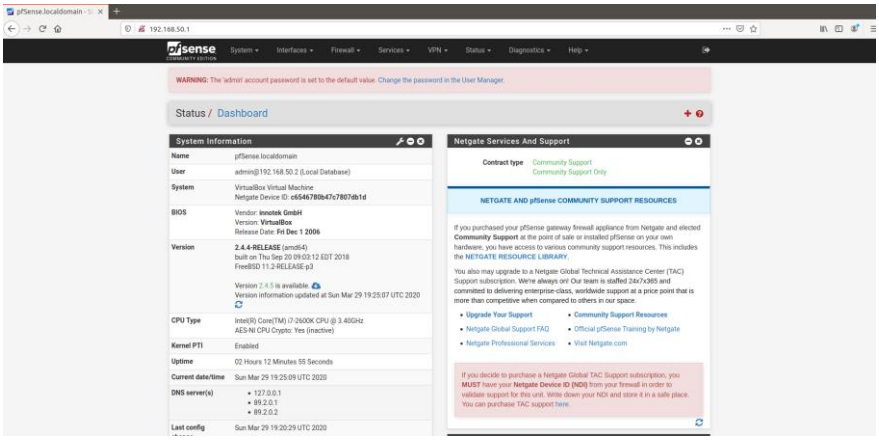
Notre interface est configurée, on peut donc accéder au configurateur WEB via l'URL affichée :

```
The IPv4 LAN address has been set to 192.168.50.1/29
You can now access the webConfigurator by opening the following URL in your web
browser:
                http://192.168.50.1/
```

Faisons donc un test de connexion au configurateur web via le PC-ADMIN (login : **admin** et mot de passe : **pfsense**) :



On devrait pouvoir arriver sur cette page :



Configuration de la troisième interface : DMZ

Pour configurer nos interfaces, faisons **2**, pour **Set Interface(s) IP address**, puis **3** :

```
Available interfaces:

1 - WAN (em0 - static)
2 - LAN (em1 - static)
3 - OPT1 (em2 - static)
```

Entrons ensuite l'adresse et le CIDR que l'on a choisi pour l'interface DMZ dans notre définition de l'environnement (soit **`172.31.0.2/30`**) :

```
Enter the new OPT1 IPv4 address.  Press <ENTER> for none:
> 172.31.0.2

Subnet masks are entered as bit counts (as in CIDR notation) in pfSense.
e.g. 255.255.255.0 = 24
     255.255.0.0   = 16
     255.0.0.0     = 8

Enter the new OPT1 IPv4 subnet bit count (1 to 31):
> 30
```

On nous demande d'entrer une adresse de passerelle, seulement pour interfaces WAN. Laisser vide et faire **Entrée**. Faire **Entrée** pour ne pas fixer d'adresse IPv6. Faire **n** pour ne pas activer le DHCP le réseau de notre interface.

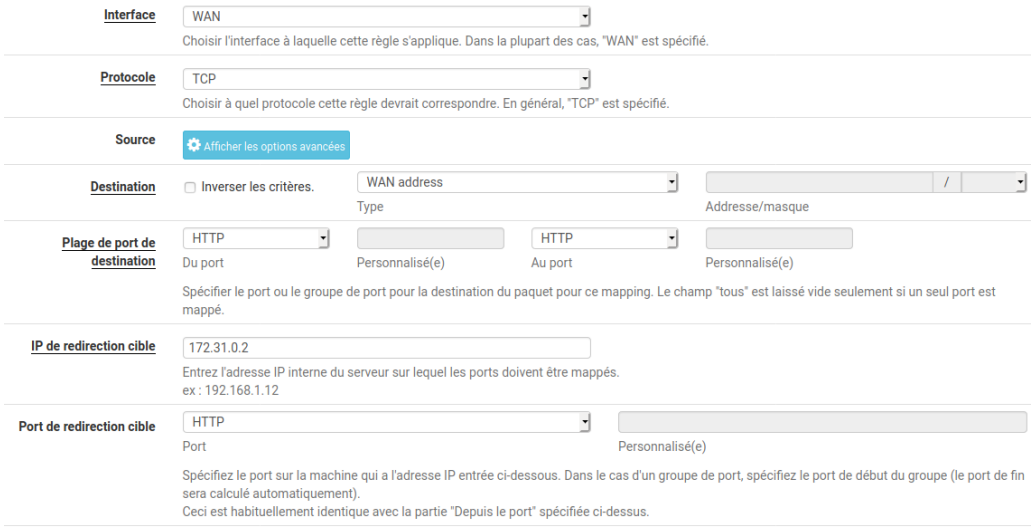
Nos systèmes sont désormais connectés à leurs passerelles respectives.

Etablissement du NAT

Enfin, nous allons pouvoir configurer notre PfSense pour qu'il puisse permettre l'accès à notre site depuis l'extérieur (donc ici, la machine physique).

Dans l'interface web de PfSense, allons dans **Pare-Feu > NAT**. Ici, nous allons créer une règle de transfert de port, qui va donc permettre la redirection, quand un client se connecte via l'adresse de l'interface WAN ( ) par protocole http (port 80). Faire **Ajouter**, puis configurer ainsi :

- **Interface :** WAN
- **Protocole :** TCP
- **Sources :** Toutes
- **Destination :** Wan address
- **Plage de port de destination :** du port http au port http
- **IP de redirection cible :** 172.31.0.2
- **Port de redirection cible :** http



Puis enregistrer. On peut ainsi envisager de créer une règle de transfert de port similaire dans les paramètres de notre box, pour que le site soit accessible depuis Internet, via l'adresse IP publique de notre box et le port 80 (ex : 165.84.251.38 :80).

Il nous manque plus qu'à établir quelques règles de pare-feu pour sécuriser nos flux, et permettre l'accès au site web par notre machine physique (et pourquoi pas Internet).

c. Etablissement des règles de Pare-Feu

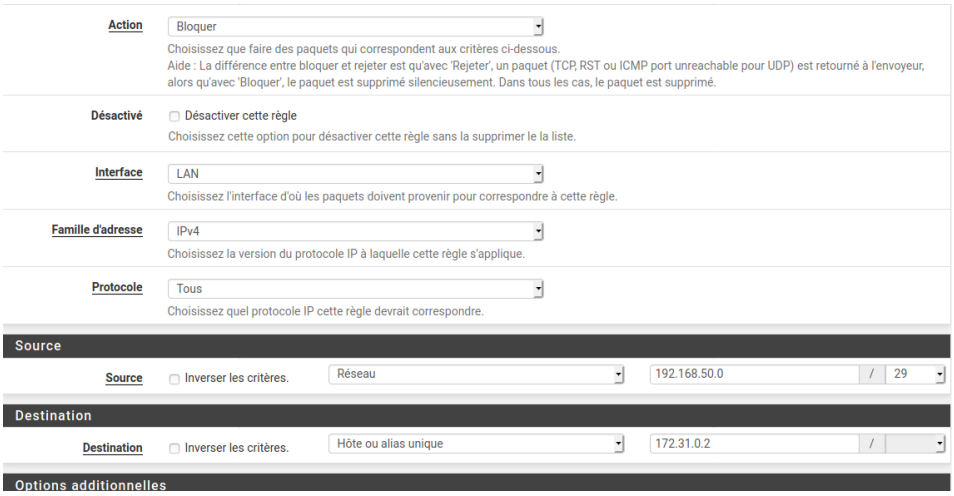
Nous avons pu, dans la partie précédente, établir une première règle de Pare-Feu, autorisant les flux http de notre serveur à l'interface WAN. Il s'agira, dans cette partie, de définir une politique de sécurité pour notre Pare-Feu.

La politique de sécurité définit l'ensemble des règles de filtrage à implémenter sur le pare-feu. Cela consiste à spécifier l'ensemble des services et protocoles TCP/IP pouvant être accessibles par des utilisateurs internes ou externes au site. Elle dépend bien sûr des besoins de sécurité de l'entreprise mais aussi des capacités du pare-feu à pouvoir satisfaire ces besoins.=

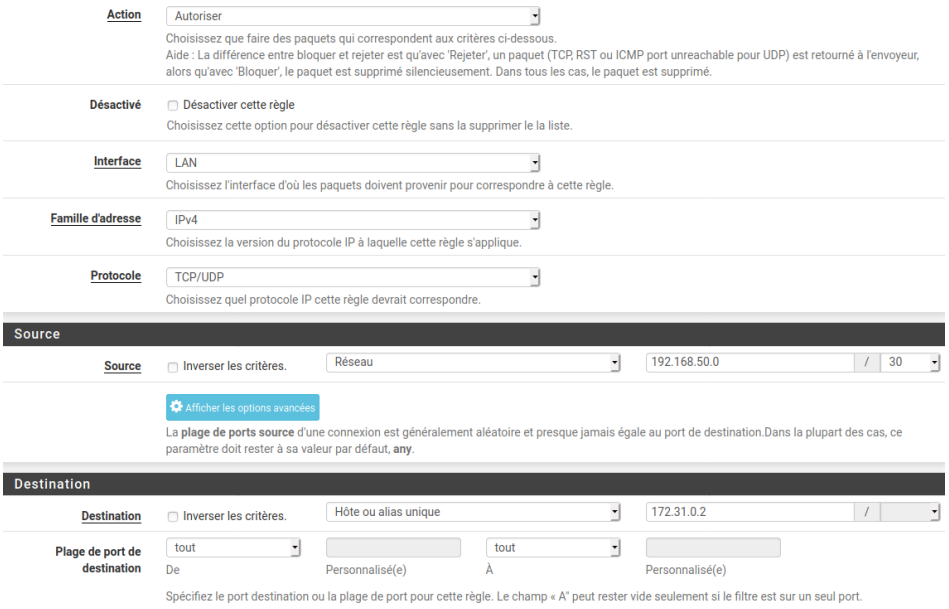
Règles du LAN

Par défaut, pour l'interface LAN, les règles autorisent l'accès à l'interface web et les connexions vers l'extérieur. Pour que le serveur de base de données en LAN puisse communiquer de manière sécurisée avec le serveur web, il nous faut créer une règle autorisant seulement le trafic TCP + UDP depuis le réseau LAN vers la DMZ.

Créons une règle dans l'onglet LAN (**Pare-Feu > Règles)**, et saisissons les paramètres suivants :



Cette règle permet de bloquer tous les échanges du LAN à la DMZ. Ensuite, créons une autre règle avec ces paramètres :



Ainsi, seul le trafic TCP/UDP est autorisé à circuler ; la page s'affiche correctement sur le PC-ADMIN, mais les pings ne circulent plus.

Voilà à quoi ressemble nos règles pour le LAN :

Règles (Faire glisser pour changer l'ordre)

	États	Protocole	Source	Port	Destination	Port	Passerelle	File d'attente	Ordonnancement	Description	Actions
	1 /1.18 MiB	*	*	*	LAN Address	80	*	*		Règle anti-blocage	
	0 /13 KiB	IPv4 TCP/UDP	192.168.50.0/30	*	172.31.0.2	*	*	aucun			
	0 /1 KiB	IPv4 *	192.168.50.0/29	*	172.31.0.2	*	*	aucun			
	0 /219 KiB	IPv4 *	LAN net	*	*	*	*	aucun		Default allow LAN to any rule	
	0 /0 B	IPv6 *	LAN net	*	*	*	*	aucun		Default allow LAN IPv6 to any rule	

Règles de la DMZ

Du côté de la DMZ, nous n'avons plus de règles à appliquer ; l'essential est déjà filtré par Apache (nous ne pouvons plus envoyer de ping à partir du serveur WEB). Nous pouvons toutefois établir une règle n'autorisant seulement les trafics TCP/UDP, comme pour le LAN :

Action : Autoriser
Choisissez que faire des paquets qui correspondent aux critères ci-dessous.
Aide : La différence entre bloquer et rejeter est qu'avec 'Rejeter', un paquet (TCP, RST ou ICMP port unreachable pour UDP) est retourné à l'envoyeur, alors qu'avec 'Bloquer', le paquet est supprimé silencieusement. Dans tous les cas, le paquet est supprimé.

Désactivé : ☐ Désactiver cette règle
Choisissez cette option pour désactiver cette règle sans la supprimer le la liste.

Interface : OPT1
Choisissez l'interface d'où les paquets doivent provenir pour correspondre à cette règle.

Famille d'adresse : IPv4
Choisissez la version du protocole IP à laquelle cette règle s'applique.

Protocole : TCP/UDP
Choisissez quel protocole IP cette règle devrait correspondre.

Source

Source : ☐ Inverser les critères. Hôte ou alias unique 172.31.0.2 /

Afficher les options avancées
La **plage de ports source** d'une connexion est généralement aléatoire et presque jamais égale au port de destination.Dans la plupart des cas, ce paramètre doit rester à sa valeur par défaut, **any**.

Destination

Destination : ☐ Inverser les critères. tout Destination Address /

Plage de port de destination : (autre) De — Personnalisé(e) — (autre) À — Personnalisé(e)
Spécifiez le port destination ou la plage de port pour cette règle. Le champ « A" peut rester vide seulement si le filtre est sur un seul port.

Voilà à quoi ressemble nos règles pour la DMZ :

Règles (Faire glisser pour changer l'ordre)

	États	Protocole	Source	Port	Destination	Port	Passerelle	File d'attente	Ordonnancement	Description	Actions
	0 /2 KiB	IPv4 TCP	OPT1 net	*	*	*	*	aucun		LAN-DMZ to any	
	0 /0 B	IPv6 TCP	*	*	*	*	*	aucun			

Règles du WAN

Dans l'onglet **WAN**, créer une règle avec ces paramètres :

Action — Autoriser

Choisissez que faire des paquets qui correspondent aux critères ci-dessous.
Aide : La différence entre bloquer et rejeter est qu'avec 'Rejeter', un paquet (TCP, RST ou ICMP port unreachable pour UDP) est retourné à l'envoyeur, alors qu'avec 'Bloquer', le paquet est supprimé silencieusement. Dans tous les cas, le paquet est supprimé.

Désactivé — Désactiver cette règle

Choisissez cette option pour désactiver cette règle sans la supprimer le la liste.

Règle de filtrage associée — This is associated with a NAT rule.
Editing the interface, protocol, source, or destination of associated filter rules is not permitted.
Visionner la règle NAT

Interface — WAN

Choisissez l'interface d'où les paquets doivent provenir pour correspondre à cette règle.

Famille d'adresse — IPv4

Choisissez la version du protocole IP à laquelle cette règle s'applique.

Protocole — TCP

Choisissez quel protocole IP cette règle devrait correspondre.

Source

Source — Inverser les critères. tout — Source Address /

Afficher les options avancées

La **plage de ports source** d'une connexion est généralement aléatoire et presque jamais égale au port de destination.Dans la plupart des cas, ce paramètre doit rester à sa valeur par défaut, **any**.

Destination

Destination — Inverser les critères. Hôte ou alias unique — 172.31.0.2 /

Plage de port de destination — HTTP (80) De — Personnalisé(e) — HTTP (80) À — Personnalisé(e)

Spécifiez le port destination ou la plage de port pour cette règle. Le champ « A" peut rester vide seulement si le filtre est sur un seul port.

Puis une autre avec ceux-ci :

Action — Bloquer

Choisissez que faire des paquets qui correspondent aux critères ci-dessous.
Aide : La différence entre bloquer et rejeter est qu'avec 'Rejeter', un paquet (TCP, RST ou ICMP port unreachable pour UDP) est retourné à l'envoyeur, alors qu'avec 'Bloquer', le paquet est supprimé silencieusement. Dans tous les cas, le paquet est supprimé.

Désactivé — Désactiver cette règle

Choisissez cette option pour désactiver cette règle sans la supprimer le la liste.

Interface — WAN

Choisissez l'interface d'où les paquets doivent provenir pour correspondre à cette règle.

Famille d'adresse — IPv4

Choisissez la version du protocole IP à laquelle cette règle s'applique.

Protocole — Tous

Choisissez quel protocole IP cette règle devrait correspondre.

Source

Source — Inverser les critères. tout — Source Address /

Destination

Destination — Inverser les critères. tout — Destination Address /

Options additionnelles

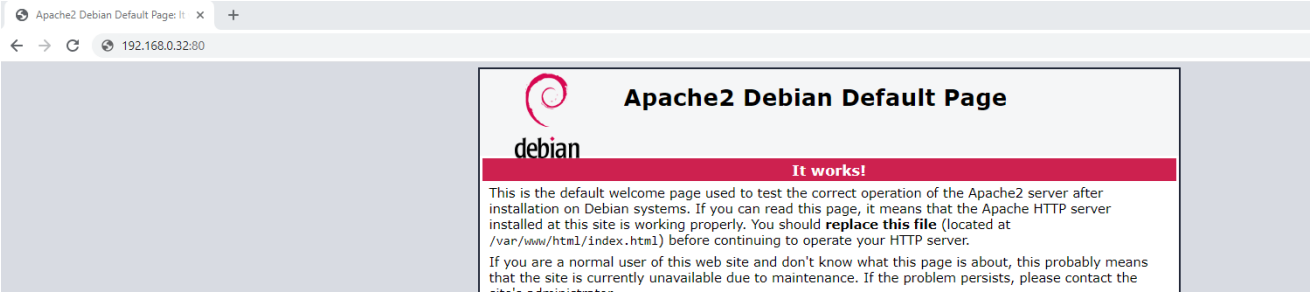
Ces paramètres refusent tous types de flux venant du WAN, vers n'importe quel réseau.

Voilà à quoi ressemble nos règles pour le WAN :

Règles (Faire glisser pour changer l'ordre)

	États	Protocole	Source	Port	Destination	Port	Passerelle	File d'attente	Ordonnancement	Description	Actions
✔	1 /105 KiB	IPv4 TCP	*	*	172.31.0.2	80 (HTTP)	*	aucun		NAT	
✖	0 /2 KiB	IPv4 *	*	*	*	*	*	aucun			

Notre site WEB est désormais accessible via notre machine physique, si l'on rentre l'adresse WAN de PfSense, avec le port 80 :



2. Communication entre poste de travail et serveur BDD via SSH

Cette partie consiste à mettre en marche des serveurs ssh, avec des paramétrages de base. Les opérations d'installation, démarrage, configuration et sécurisation côté serveurs sont à effectuer sur notre serveur BDD. Pour ce qui est de la mise en relation entre notre base de données et le PC-ADMIN, nous utiliserons MySQL Workbench, logiciel gratuit permettant des manipulations sur les bases de données. Nous connecterons ces deux-là par SSH.

a. Installation & démarrage du serveur ssh

Ainsi, toutes les commandes entrées ici se feront via le compte root (qui possède tous les droits sur le serveur), accessible via la commande **su** .

Passons sur le compte root :

```
su -
```

Installons le service sshd :

```
apt-get install openssh-server
```

Après l'installation, un fichier de configuration nommé **sshd_config** est créé ; c'est principalement ce fichier-là qui sera manipulé. Etant sensible, il est nécessaire d'en effectuer une copie, que l'on nommera **sshd_config_sauvegarde** pour éviter des conflits.

```
cp /etc/ssh/sshd_config /etc/ssh/sshd_config_sauvegarde
```

Nous pouvons maintenant démarrer le serveur ssh. En revanche, il sera inaccessible pour notre machine. Il faudra apporter certaines modifications au fichier **sshd_config**, afin d'assurer la connexion de nos machines, ainsi que la sécurisation du serveur lui-même. A chaque modification apportée, il ne faudra pas oublier de redémarrer le serveur.

Démarrer le serveur :

```
systemctl start ssh
```

Arrêter le serveur :

```
systemctl stop ssh
```

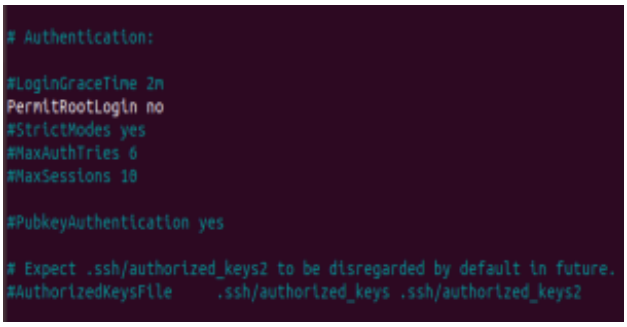
Redémarrer le serveur :

```
systemctl restart ssh
```

Configuration du serveur ssh : configuration basique

Dans une optique de sécurisation du serveur, nous allons voir comment changer le port de connexion (par défaut, 22) ainsi qu'interdire la connexion au compte root. Nous choisirons comme port de connexion le port 1564. Il s'agit donc d'ouvrir le fichier **sshd_config** via **nano**, et à la ligne « #Port 22 », changer 22 pour 1564 et supprimer le dièse (qui marque le texte comme un commentaire, et définit la valeur de la commande comme étant celle par défaut. Ainsi, si l'on change 22 par 1564, sans supprimer le #, le système gardera le paramètre 22).

Plus bas, à la ligne « #PermitRootLogin », changeons la valeur en no (qui devrait être en « prohibit-password » par défaut), et supprimons le dièse.



La configuration minimale pour pouvoir se connecter au serveur est en place ; n'oublions pas de sauvegarder les modifications et de redémarrer le serveur. Nous pouvons désormais effectuer un test de connexion, de notre PC-ADMIN vers notre serveur BDD. Dans un invité de commande sur le PC-ADMIN, nous devons renseigner dans la commande de connexion le compte sur le serveur auquel on souhaite se connecter, son adresse ip, et le numéro de port que l'on a modifié dans le fichier config :

```
ssh oscar@192.168.50.6 -p 1564
```

Un message nous avertissant sur l'authenticité du serveur apparaît. Dans notre cas, étants sûrs qu'il s'agit bien de notre serveur, nous pouvons nous connecter les yeux fermés.

```
oscar@PC-ADMIN:~$ ssh oscar@192.168.50.6 -p 1564
The authenticity of host '[192.168.50.6]:1564 ([192.168.50.6]:1564)' can't be es
tablished.
ECDSA key fingerprint is SHA256:bZ62mibK3sjA9WJ2et1p3FdCiZUKecq2bEcViziMjkc.
Are you sure you want to continue connecting (yes/no)?
```

Après avoir entré le mot de passe du compte auquel nous essayons d'accéder, nous voilà connecté au serveur ssh.

Nous pouvons également paramétrer le mode d'authentification par clé. Cette opération étant plutôt fastidieuse et nécessitant les add-ons invités de VirtualBox, nous passerons outre cette étape. Voir Documentation Connexion A Distance Via SSH.

```
Linux SRV-BDD 4.19.0-8-amd64 #1 SMP Debian 4.19.98-1 (2020-01-26) 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: Mon Mar 30 19:32:43 2020
oscar@SRV-BDD:~$
```

Ajout d'un utilisateur avec accès restreints

La prochaine étape consiste à créer un profil utilisateur avec peu de droits sur le serveur. Pour ce faire, comme Debian ne gère pas les droits par comptes utilisateurs, il faut modifier les autorisations de certains fichiers du serveur. Ensuite, nous devrons désactiver l'historisation des commandes pour cet utilisateur.

Commençons par créer le nouvel utilisateur « sshuser » :

```
adduser sshuser
```

Dans le fichier **sshd_config**, autorisons-le à se connecter au serveur (cette ligne peut déjà exister, à vérifier). Attention, n'oublions pas de rajouter notre propre compte à la ligne précédente, sinon il sera considéré comme non-autorisé :

```
AllowUsers sshuser
```

Maintenant, nous pouvons limiter les accès de cet user (toujours via le compte root). Dans un premier temps on va modifier les droits du répertoire */home/oscar*, afin que l'user ne soit pas autorisé à accéder aux fichiers du compte **oscar**.

```
chmod 700 /home/oscar
```

Enfin, nous allons autoriser **sshuser** à seulement lire ses fichiers (tout ce qui est contenu dans son répertoire /home/sshuser) :

```
chmod 704 /home/sshuser
```

Ainsi, l'accès au fichier **sshd_config** est en lecture seule pour **sshuser** :

```
# To disable tunneled clear text passwords, change to no here!
PasswordAuthentication no
#PermitEmptyPasswords no

# Change to yes to enable challenge-response passwords (beware issues with
# some PAM modules and threads)
                                [ File 'sshd_config' is unwritable ]
^G Get Help     ^O Write Out    ^W Where Is     ^K Cut Text     ^J Justify      ^C Cur Pos      M-U Undo
^X Exit         ^R Read File    ^\ Replace      ^U Uncut Text   ^T To Spell     ^_ Go To Line   M-E Redo
```

La dernière étape consiste à supprimer le fichier **.bash_history**, où les commandes entrées dans le terminal du serveur sont enregistrées. Chaque user à son propre historique, donc chaque user à son propre fichier. Supprimons celui de **sshuser** :

```
rm /home/sshuser/.bash_history
```

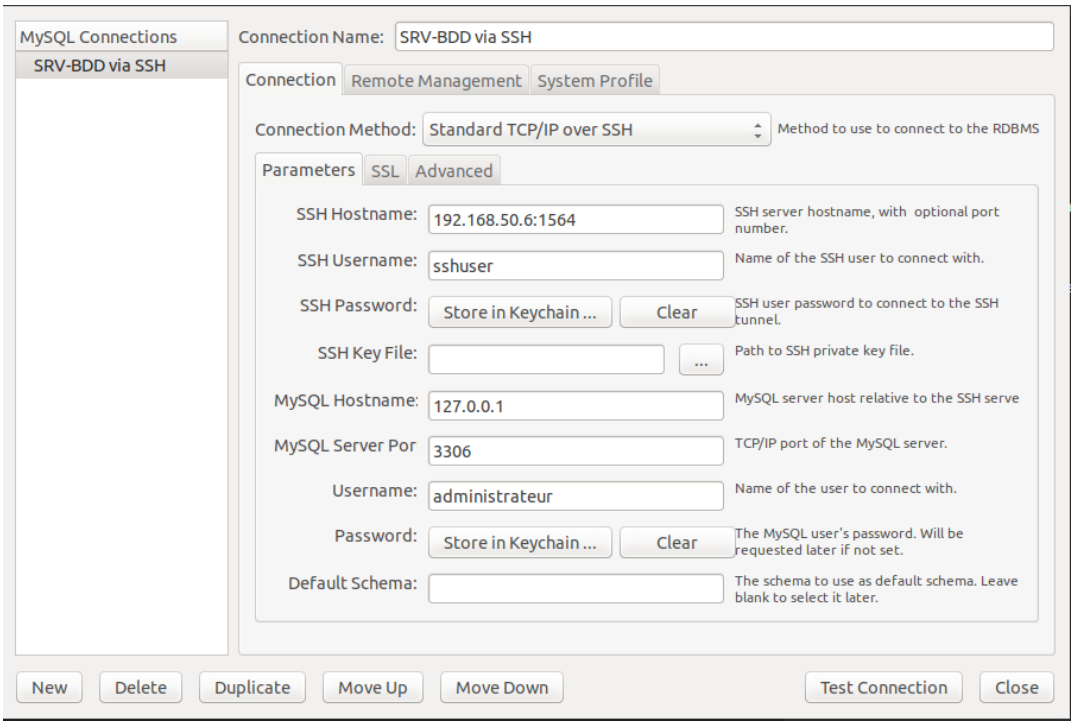
Désormais, le compte sshuser est créé, et le serveur est sécurisé.

b. Connecter MySQL Workbench au serveur MySQL, via SSH

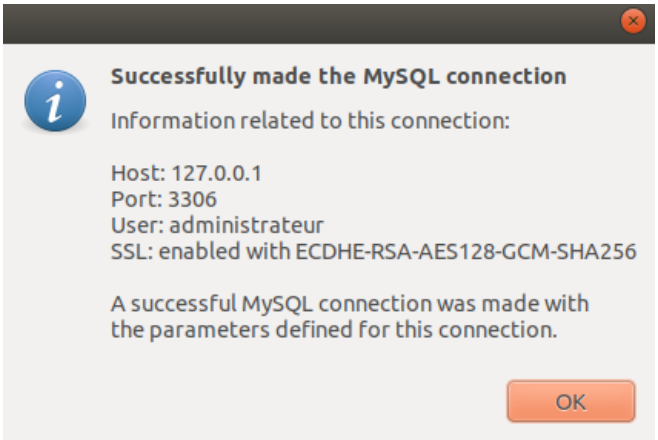
Nos serveurs SSH sont configurés. Nous pouvons maintenant lancer MySQL Workbench sur notre PC-ADMIN pour le configurer, afin qu'il accède au serveur BDD via SSH.

Dans le menu d'accueil de Workbench, cliquons sur le petit « + » à droite de « MySQL Connections ». La page de configuration s'ouvre.

- Dans **Connection Name**, renseignons le nom de notre configuration. Nous choisirons ici « Serveur BDD via SSH ».
- Dans **Connection Method**, Choisir « Standard TCP/IP over SSH »
- Dans **SSH Hostname**, renseigner l'adresse ip du serveur SSH avec le port.
- Dans **SSH Username**, mettre le compte utilisateur SSH auquel on souhaite se connecter : **sshuser**.
- Laisser les champs **MySQL Hostname** et **MySQL Server Port** par défaut.
- Dans **Username**, renseigner le compte utilisateur MySQL auquel nous nous connectons : **administrateur**.



Nous pouvons désormais faire un test de connexion, via le bouton « Test Connection ». Si tout fonctionne bien, faisons « OK ».



Notre Workbench est désormais connecté au serveur BDD, grâce à SSH.

3. Communication entre serveur LAMP et BDD

Dans cette partie, nous allons voir comment configurer notre serveur LAMP, pour qu'il puisse communiquer convenablement avec notre base de données SRV-BDD. C'est le module PHP qui s'occupe d'effectuer ces interactions, et il y a plusieurs moyens pour y parvenir, via plusieurs scripts ou sous-modules. Nous utiliserons ici le sous-module **mysqli**, qui est fournit de base avec le paquet de PHP7 (le paquet s'appelle **libapache2-mod-php**, comme vu précédemment).

a. Configuration côté SRV-BDD

Du côté de notre SRV-BDD, nous allons devoir exécuter quelques opérations pour amorcer notre démarche. Toutes ces opérations sont à effectuer via le compte root.

Création d'un nouvel utilisateur

Dans notre base MySQL, créons un nouvel utilisateur « **phpuser** » ayant pour mot de passe « **motdepasse** », et permettons-lui de se connecter depuis l'adresse ip de notre serveur WEB :

```
CREATE USER 'phpuser'@'172.31.0.2' IDENTIFIED WITH mysql_native_password BY
'motdepasse' ;
```

Ici, l'option **mysql_native_password** permet à PHP de comprendre la méthode d'authentification de MySQL.

Ensuite, donnons à **phpuser** les privilèges appropriés.

```
GRANT ALL PRIVILEGES ON *.* TO 'phpuser'@'172.31.0.2' WITH GRANT OPTION;
```

Puis rechargeons les privilèges de MySQL :

```
Flush privileges ;
```

b. Configuration côté SRV-WEB

Nous verrons ici les dernières étapes permettant la mise en place d'interactions entre nos deux serveurs. Toutes ces opérations sont à effectuer via le compte root.

Création d'une nouvelle page

Créons le nouveau répertoire, d'où nous pourrons stocker notre fichier de configuration. Ce répertoire devra se trouver dans */var/www/html/www.monsite.com,* pour qu'il soit accessible via le navigateur. Appelons le « **Database** » :

```
mkdir /var/www/html/www.monsite.com/database
```

Plaçons-nous dans database et créons le contenu de cette page :

```
cd /var/www/html/www.monsite.com/database
nano index.php
```

Configuration de la connexion

Dans le fichier **index.php** précédemment créé, entrons ces paramètres :

```
<?php
$link = mysqli_connect("p:"."192.168.50.6", "phpuser", "motdepasse", "mysql");

if (!$link) {
```

```
    echo "Erreur : Impossible de se connecter à MySQL." . PHP_EOL;
    echo "Errno de débogage : " . mysqli_connect_errno() . PHP_EOL;
    echo "Erreur de débogage : " . mysqli_connect_error() . PHP_EOL;
    exit;
}

echo "Succès : Une connexion a pu être établie à la base." . PHP_EOL;
echo "Information d'hôte : " . mysqli_get_host_info($link) . PHP_EOL;

mysqli_close($link);
?>
```

Voici le détail des paramètres :

mysqli_connect("192.168.0.10 ", "account", "password", "my_db") ;	Lance une connexion à la base **my_db**, hébergée sur la machine **192.168.0.10**, via le compte **account** et son mot de passe **password**. L'option **"p:"** permet l'établissement d'une connexion permanente.

Les autres options ne sont là que pour nous indiquer si la connexion a bien été établie, ou non. C'est utile en cas d'échec de connexion ; on peut relever le message d'erreur et ainsi agir en conséquence.

Rechargeons la configuration d'Apache :

```
systemctl reload apache2
```

Dans notre navigateur, entrons l'URL de notre fichier **index.php** créé précédemment. Si tout a bien été configuré, voici ce qui est affiché :

Succès : une connection a pu être établie à la base. Information d'hôte : 192.168.50.6 via TCP/IP

Nous avons pu établir une connexion entre notre serveur LAMP et notre serveur de base de données.