BTS SIO – Option SLAM

Documentation d'épreuve

Rédacteur	Version	Date	Nb pages
Johann DIETRICH	1.2	16/03/2023	18

Installation et sécurisation d'un serveur web Debian

SOMMAIRE

1 Cahier des charges

1.1 Introduction

Type de mission
Mission effectuée en classe, puis accentuée pendant le temps personnel
Contexte
TP durant lequel nous était demandé de sécuriser un serveur.
Demande du client
Besoin d'avoir un serveur Debian sécurisé et d'un serveur web apache avec celui-ci. Bien évidemment, également sécurisé du mieux que possible.
Budget disponible
Aucun réel budget disponible
Outils disponibles
VMware, virtualbox

1.2 Expression fonctionnelle du besoin

Liste des fonctionnalités attendues :

Front office
- Page principale apache fonctionnelle

1.3 Contraintes

Générales
Aucune réel contrainte de temps
Juridiques
Aucunes
Techniques
Debian 10

1.4 Gestion des droits d’accès

Administrateur
Peux modifier le serveur

2 Description des environnements

Environnement de développement

L’environnement choisi est une Debian 10, car il s’agit du modèle sur lequel je suis le plus apte et habitué à manipuler. Il s’agit de l’environnement de développement qui fut utilisé durant toute la durée de notre BTS.

3 Méthodologie

3.1 Méthodologie et versioning

Utilisations de snapshots sur VMWare pour garder des sauvegardes de différents stades de la Machine Virtuelle.

3.2 Gestion des tests de la solution

unitaires

Test tout simple de la sécurisation une par une en essayant la faille qui est supposé avoir été sécurisé

Par exemple le test d'un mot de passe n'ayant pas le nombre de charactères nécessaire

4 Mise en oeuvre

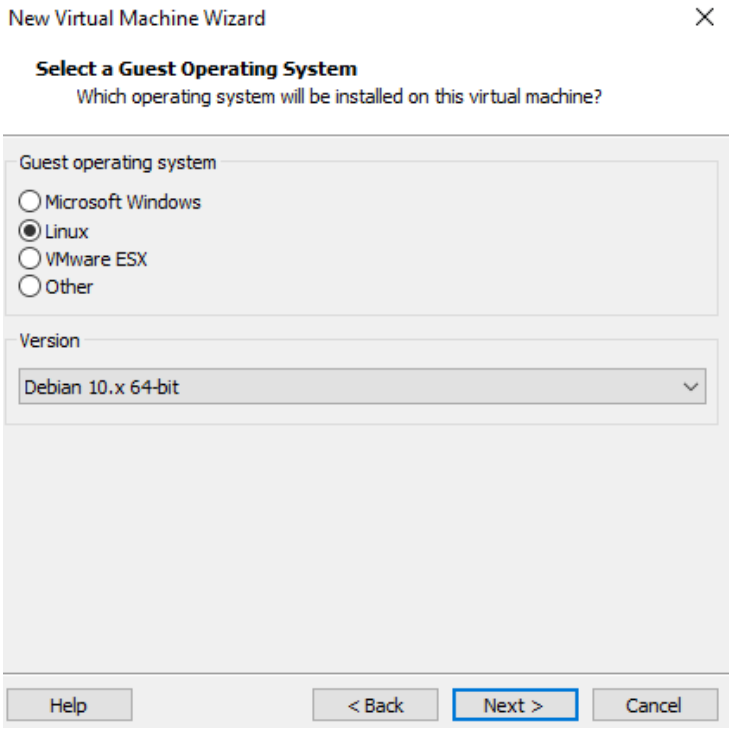
Pour commencer ce projet, il nous faut déjà commencer avec l'installation de la machine Debian qui celle-ci peut être effectué de différentes façon selon le support. Mais dans notre cas, toute la mission a été effectuée sur Machine Virtuelle donc la documentation sera présentée de tel manière.

4.1 Installation de Debian sur une machine virtuelle VMWare

Dans cette partie, nous allons passer en revue les étapes nécessaires pour installer Debian sur une machine virtuelle VMWare. L'utilisation d'une machine virtuelle offre de nombreux avantages, notamment la possibilité de tester différentes configurations système sans affecter le système hôte, ainsi que la possibilité de cloner facilement des environnements pour une configuration rapide.

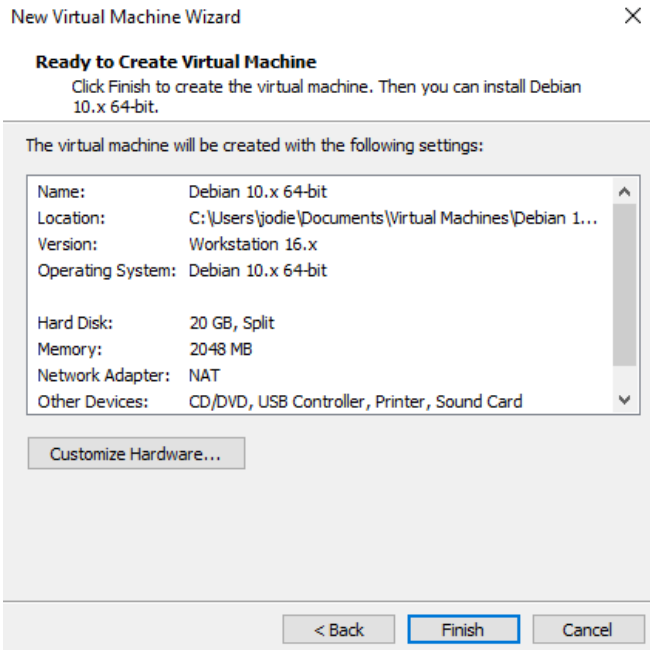
Pour commencer, il faut se manier d'un ISO Debian qui va servir pour l'installation.

- Ensuite, ouvrez VMWare Workstation ou VMWare Player.
- Sélectionnez "Nouvelle machine virtuelle" dans le menu "Fichier".
- Sélectionnez "Installation personnalisée".
- Choisissez "Linux" comme système d'exploitation et "Debian 64-bit" comme version.



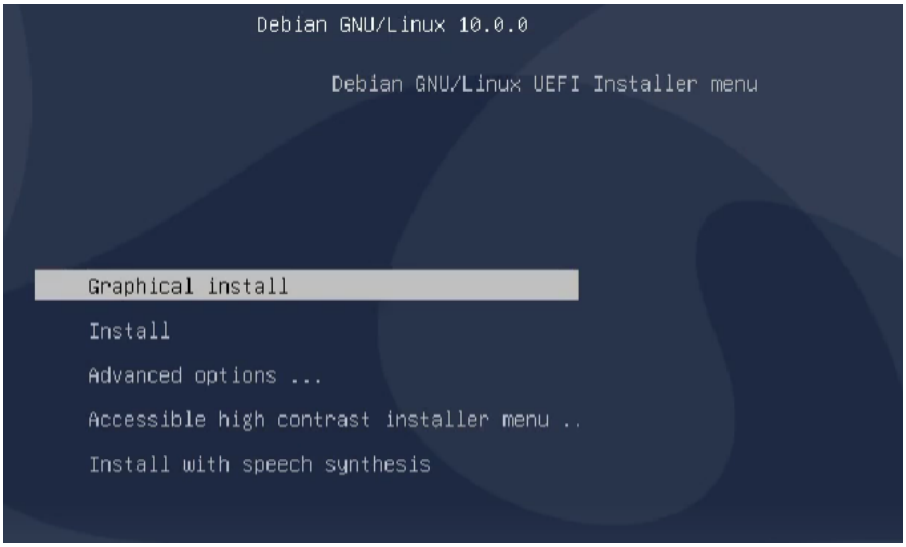
- Sélectionnez la quantité de mémoire vive et la taille du disque dur que vous souhaitez allouer à la machine virtuelle.
- Configurez les paramètres réseau en fonction de vos besoins. Pour une utilisation locale, vous pouvez sélectionner "Réseau privé" ou "Réseau hôte". (réseau hôte reste préférable)

Terminez la configuration de la machine virtuelle en suivant les instructions à l'écran.

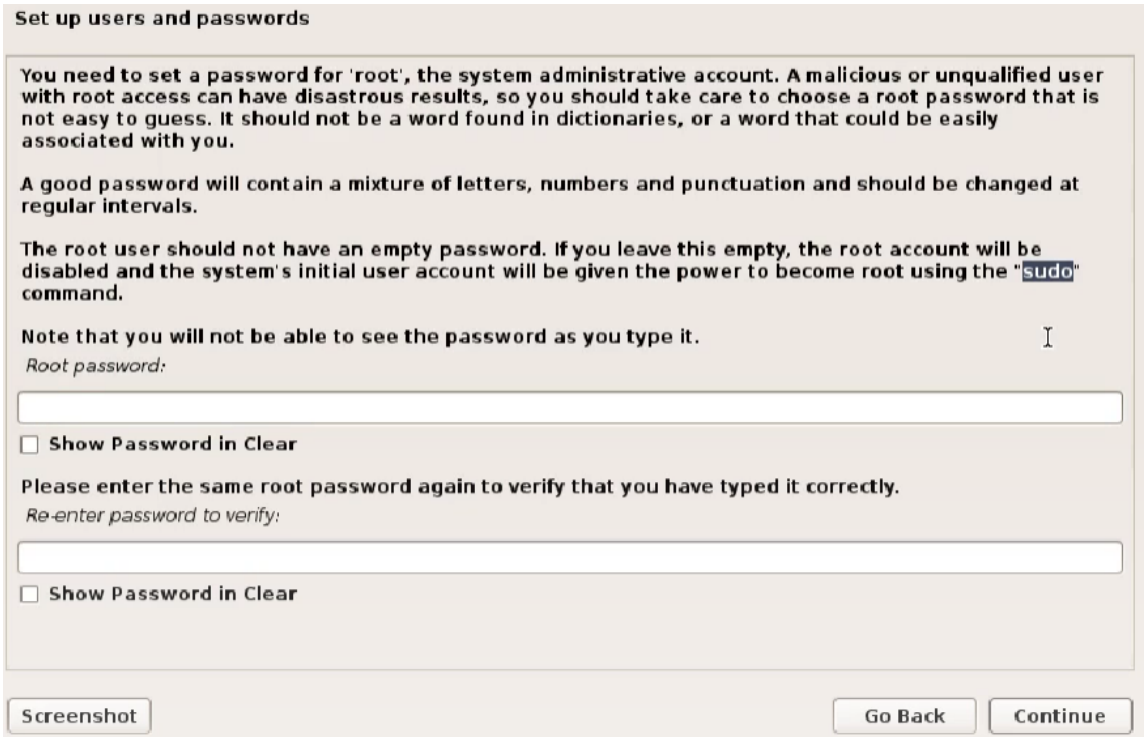


Après cela, il nous reste a installer la machine elle-même. Pour cela :

- Démarrez la machine virtuelle.
- Insérez l'image ISO de Debian dans le lecteur virtuel de la machine virtuelle.
- Démarrez l'installation de Debian en sélectionnant "Installer" dans le menu de démarrage.



- Suivez les instructions à l'écran pour installer Debian. Vous devrez sélectionner votre langue préférée, votre emplacement géographique et les paramètres de clavier. Vous devrez également configurer le réseau, le stockage et les partitions.



Une fois l'installation terminée, redémarrez la machine virtuelle et celle-ci devrait fonctionner parfaitement.

4.2 Mise en place de SSH

SSH (Secure Shell) est un protocole de communication sécurisé qui permet d'accéder à distance à un serveur. Dans cette partie, nous allons passer en revue les étapes nécessaires pour installer SSH sur une Debian en version non graphique et le sécuriser pour limiter les risques d'attaques.

Installation de SSH :

- Ouvrez une session en ligne de commande en tant qu'utilisateur root.
- Installez le paquet OpenSSH-server en saisissant la commande suivante : apt-get install openssh-server

Sécurisation de SSH :

- Ouvrez le fichier de configuration de SSH en saisissant la commande suivante : nano /etc/ssh/sshd_config

Modifiez les paramètres suivants pour renforcer la sécurité de SSH :

```
# Désactiver la connexion root
PermitRootLogin no

# Changer le port de connexion par défaut (22) pour un port plus élevé
Port 2222

# Désactiver l'authentification par mot de passe
PasswordAuthentication no

# Autoriser uniquement l'authentification avec clé publique
PubkeyAuthentication yes

# Désactiver l'authentification avec GSSAPI
GSSAPIAuthentication no
```

Redémarrez le service SSH pour appliquer les modifications en saisissant la commande suivante :

systemctl restart ssh

4.3 Sécurisation des ports avec iptables

Iptables est un pare-feu intégré à Debian qui permet de contrôler le trafic réseau entrant et sortant sur le serveur. Dans cette partie, nous allons voir comment sécuriser les ports de votre serveur en utilisant iptables pour bloquer tous les ports entrants et sortants, sauf ceux nécessaires au fonctionnement de votre site web. Nous verrons également comment autoriser l'accès SSH uniquement à partir d'adresses IP autorisées.

Pour commencer, on va configurer iptables :

- Ouvrez une session en ligne de commande en tant qu'utilisateur root.
- Vérifiez que iptables est installé en saisissant la commande suivante : which iptables
- Si iptables n'est pas installé, installez-le en saisissant la commande suivante : apt-get install iptables
- Commencez par bloquer tous les ports entrants et sortants en saisissant les commandes suivantes :

```
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP
```

- Autorisez ensuite les ports nécessaires au fonctionnement de votre site web en saisissant les commandes suivantes en remplaçant X.X.X.X par l'adresse IP de votre serveur web :

```
iptables -A INPUT -p tcp -s X.X.X.X --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp -d X.X.X.X --sport 22 -j ACCEPT
iptables -A INPUT -p tcp -s X.X.X.X --dport 80 -j ACCEPT
iptables -A OUTPUT -p tcp -d X.X.X.X --sport 80 -j ACCEPT
iptables -A INPUT -p tcp -s X.X.X.X --dport 443 -j ACCEPT
iptables -A OUTPUT -p tcp -d X.X.X.X --sport 443 -j ACCEPT
```

Ces commandes permettent d'autoriser les ports SSH, HTTP et HTTPS.

- Enfin, autorisez l'accès SSH uniquement à partir des adresses IP autorisées en saisissant les commandes suivantes en remplaçant Y.Y.Y.Y par l'adresse IP autorisée :

```
iptables -A INPUT -p tcp -s Y.Y.Y.Y --dport 22 -j ACCEPT
iptables -A OUTPUT -p tcp --sport 22 -j ACCEPT
```

Ces commandes permettent d'autoriser l'accès SSH uniquement à partir de l'adresse IP autorisée.

- Vérifiez la configuration d'iptables en saisissant la commande suivante : iptables -L -n
- Enregistrez la configuration iptables pour qu'elle soit persistante en saisissant la commande suivante : iptables-save > /etc/iptables/rules.v4

4.4 Sécurisation de mot de passe

Pour sécuriser du mieux notre machine, on va devoir sécuriser le mot de passe. Et ceci de tel sorte qu'il doit être changé tout les 90 jours et que celui-ci doit avoir une taille minimum afin de pouvoir être appliqué. Voici comment faire :

- Installer le paquetage "libpam-modules" s'il n'est pas déjà installé:

```
sudo apt-get update
sudo apt-get install libpam-modules
```

- Modifier le fichier de configuration /etc/pam.d/common-password en ajoutant les lignes suivantes en haut du fichier :

```
password requisite pam_pwquality.so retry=3 minlen=12
password required pam_unix.so use_authtok remember=5
```

Ces lignes imposent des exigences de qualité de mot de passe, en plus d'exiger un minimum de 12 caractères, et limitent le nombre de tentatives de connexion.

- Modifier le fichier de configuration /etc/login.defs en ajoutant ou en modifiant les lignes suivantes :

```
PASS_MAX_DAYS 90
PASS_MIN_DAYS 0
PASS_WARN_AGE 14
```

Ces lignes définissent la durée maximale (90 jours) et minimale de validité du mot de passe ainsi que la durée de préavis (14 jours) avant l'expiration du mot de passe.

- Pour forcer le changement de mot de passe tous les 90 jours (ou toute autre durée définie dans /etc/login.defs), utilisez la commande "chage" :

```
sudo chage -M 90 <nom_utilisateur>
```

Cela définit la durée maximale de validité du mot de passe pour l'utilisateur spécifié.

En suivant ces étapes, vous avez configuré votre système pour exiger un mot de passe d'au moins 12 caractères lors de la création d'un nouveau mot de passe, ainsi que pour exiger un changement de mot de passe lors de la première connexion et tous les 90 jours.

4.5 Installation et sécurisation du site web avec Apache

- Installez Apache avec la commande suivante :

```
sudo apt install apache2
```

- Activez le module SSL d'Apache en exécutant la commande suivante :

```
sudo a2enmod ssl
```

- Redémarrez Apache pour appliquer les modifications :

```
sudo systemctl restart apache2
```

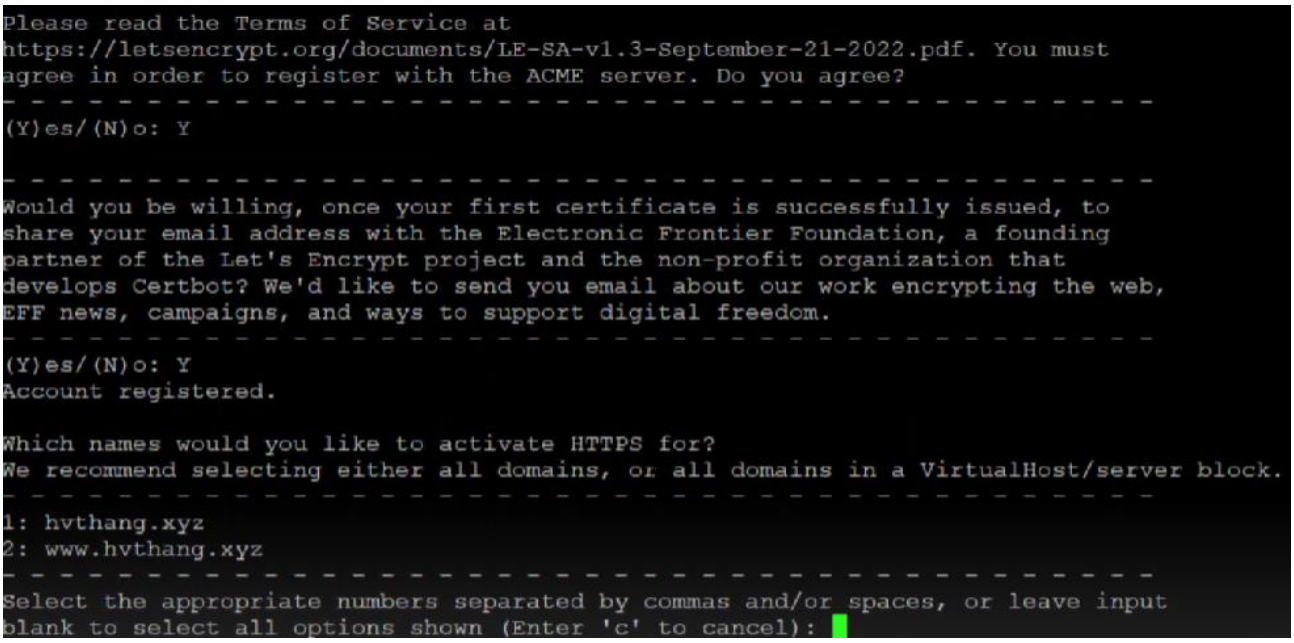
- Installez Certbot à l'aide de la commande suivante :

```
sudo apt install certbot python3-certbot-apache
```

- Obtenez le certificat SSL pour votre site web en utilisant Certbot :

```
sudo certbot --apache
```

- Suivez les instructions à l'écran pour générer un certificat SSL pour votre site web.



Une fois cela fait, tout les sites sélectionnés devraient être mis en https grâce a Certbot. Il nous reste plus qu'a limiter l'accès apache de notre machine.

- Pour limiter l'accès au serveur Apache, vous pouvez configurer des autorisations d'accès strictes dans les fichiers de configuration. Par exemple, vous pouvez restreindre l'accès à un répertoire spécifique en utilisant la directive "Require" dans le fichier de configuration Apache :

```
<Directory /var/www/html/secure>
  Options Indexes FollowSymLinks
  AllowOverride None
  Require all denied
  Require ip 192.168.0.0/24
</Directory>
```

- Dans cet exemple, l'accès au répertoire "/var/www/html/secure" est limité aux adresses IP dans la plage "192.168.0.0/24".
- Redémarrez Apache pour appliquer les modifications :

```
sudo systemctl restart apache2
```

Voilà, Apache est maintenant installé et sécurisé avec un certificat SSL valide généré par Certbot sur Debian 10.

5 Gestion de la maintenance (corrective / évolutive)

5.1 Mise à jour de la documentation du SI

Pour mettre a jour le système, il suffit de faire la commande :

```
sudo apt update && sudo apt upgrade
```

Afin de tenir le système et tout les programmes installés à jour.

5.2 Procédure de correction d'un dysfonctionnement

Analyser et corriger un dysfonctionnement

Les problèmes liés seraient généralement liés à l'utilisation d'un mauvais réseau, soit a la mise en place du https.

- Dans le cas d'un problème de réseau, il suffit de vérifier la connectivité internet et/ou rechanger le type de réseau (de réseau local a NAT par exemple).

- Dans le cas de la mise en place du https, il suffit de faire attention au fichier en etc/apache2/sites-available/nomdusite.conf et de vérifier si celui-ci a bien indiqué le nom de domaine voulu approprié au site.

6 Bilan du projet

6.1 Validation des exigences point par point

- Création d'une machine virtuelle
- Installation d'un OS Debian 10 dessus
- Installation et sécurisation de la connexion SSH
- Sécurisation des ports
- Installation d'Apache et sécurisation de son accès
- Sécurisation du site en https

6.2 Axes d'amélioration

Potentiellement installer d'autre outils utiles au développement d'un site web tel que PhpMyAdmin et de le sécuriser également.

6.3 Compétences acquises

Installation d'une machine virtuelle, d'un OS dans celle-ci. De sa sécurisation basique et de la sécurisation et installation d'un site avec Apache. Tout cela bien évidemment sur Debian 10