

TP Cybersecurity:

Dans le cadre de ce TP, nous avons entrepris la création et la configuration d'un serveur web sous Debian 12, hébergé sur une machine virtuelle via VirtualBox. L'objectif principal était de mettre en place un environnement sécurisé pour héberger deux sites web. Ce rapport détaille les étapes, les configurations, et les résultats que j'ai obtenu à la fin.

Installation et Configuration de la Machine Virtuelle

La première étape consiste à installer Debian 12 sur une machine virtuelle créée avec VirtualBox. Cette installation a été réalisée en suivant les étapes standard, incluant la configuration des paramètres réseau et la mise à jour du système pour garantir la stabilité de l'environnement.

Mise en Place du Serveur Web Apache2

Une fois Debian 12 installé, nous avons procédé à l'installation du serveur web Apache2. Cette étape a inclus :

Installation d'Apache2 : Utilisation de la commande `sudo apt-get install apache2` pour installer le serveur web.

Mise à jour du système : Exécution de `sudo apt-get update` et `sudo apt-get upgrade` pour s'assurer que toutes les dépendances et paquets étaient à jour.

Configuration des fichiers et dossiers : Création des répertoires et fichiers nécessaires pour héberger les sites web et modification des permissions pour garantir l'accès.

Création et Configuration des Sites Web

Deux sites web distincts ont été créés et configurés pour être hébergés sur le serveur Apache2. Les étapes suivies incluent :

Création des répertoires des sites : Utilisation de la commande `mkdir` pour créer les répertoires `/var/www/site1` et `/var/www/site2`.

Modification des fichiers HTML par défaut : Remplacement des fichiers HTML par défaut d'Apache2 par nos propres fichiers HTML personnalisés pour chaque site.

Configuration des hôtes virtuels : Modification des fichiers de configuration d'Apache2 pour définir les hôtes virtuels pour chaque site. Cela a inclus la création de fichiers de configuration dans `/etc/apache2/sites-available/` et l'activation des sites avec `a2ensite`.

Mise en Ligne des Sites Web

Après la configuration, les deux sites web ont été mis en ligne. Les URL par défaut ont été modifiées pour refléter les noms des sites, offrant ainsi une navigation plus intuitive et professionnelle. Les étapes spécifiques incluent :

Activation des sites : Utilisation de la commande `a2ensite site1.conf` et `a2ensite site2.conf` pour activer les sites.

Redémarrage du serveur Apache2 : Exécution de `sudo systemctl restart apache2` pour appliquer les modifications et mettre les sites en ligne.

Installation VM/Serveur web

Id

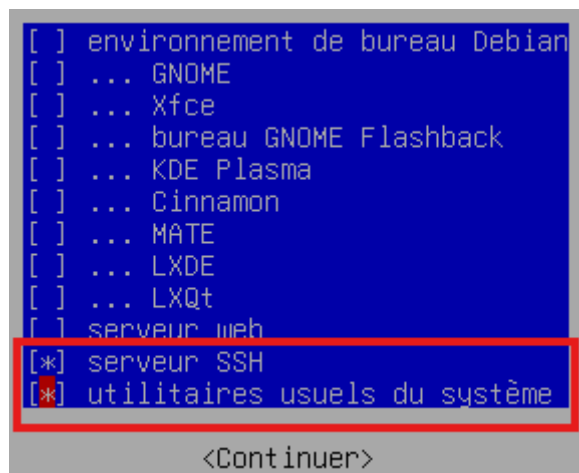
Perso: batman

Mot de passe: toor

Setting install ssh

Suivre l'installation par défauts de l'installation de l'iso de debian

Bien sélectionner « serveur SSH » et « utilitaires usuels du système » pour notre serveur web Debian12



Connexion à mon serveur + config réseau

Lancer CMD (admin) use: `ssh user@ip`

Se mettre en root : `su -`

Première commande complémentaire :

`Apt update` : update du system

`Apt install sudo` : sudo commande importante pour la suite de la procédure

```
apt update
apt install sudo
```

Config carte réseau

sudo nano /etc/network/interfaces (interface carte réseau)

Ajouter la ligne de commande du screen pour configurer la carte réseau :

```
auto enp0s8
iface enp0s8 inet static
    address 192.168.1.10/24
```

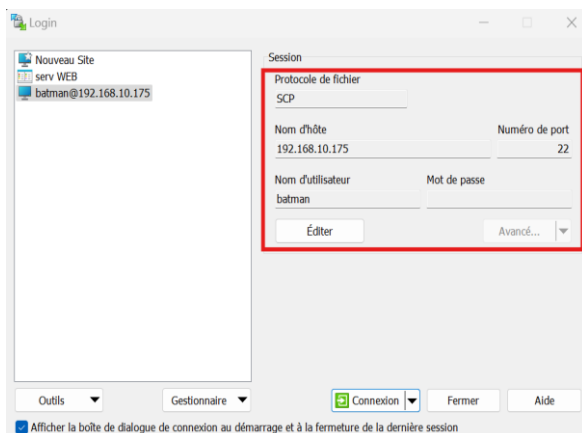
Recharger le serv avec : **sudo systemctl restart networking.service**

Mettre paquet à jour : **sudo apt-get update**

Installation Apache2 + WinSCP + quelques commandes

Installer apache 2 avec la commande suivante : **sudo apt install apache2**

Et installer aussi le logiciel : **WinSCP (pour modif pratique)** Faire les commandes pour perm WinSCP par la suite: username= mon id



Pour les perm si marche pas, avec WinSCP utiliser les commandes screen, revaloriser les droits

```
sudo chmod 777 /var/www/site1/
```

3. Changez les permissions du fichier (si nécessaire) :

- Si vous avez besoin de modifier les permissions pour permettre l'écriture, utilisez la commande suivante :

```
sudo chmod 664 /var/www/html/index.html
```

- Cette commande donne les permissions d'écriture au propriétaire et au groupe.

4. Changez le propriétaire du fichier (si nécessaire) :

- Si vous devez changer le propriétaire du fichier pour votre utilisateur, utilisez la commande suivante :

```
sudo chown your_username:your_username /var/www/html/index.html
```

Chemin + commande :


Tips des commandes pour accéder aux chemins plus rapidement

- cd /var/www/html/index.html = **aller au site**
- cd /etc/apache2/sites-available = **aller au fichier conf**
- cd /etc/hosts = **conf host site**
- rm (-f pour forcer) = **supprimer**
- mv = **rename**
- mkdir = **crée dossier**
- touch = **crée fichier**

Crée les dossiers et fichiers sites + config.conf

On va pour commencer aller dans : **cd /var/www**

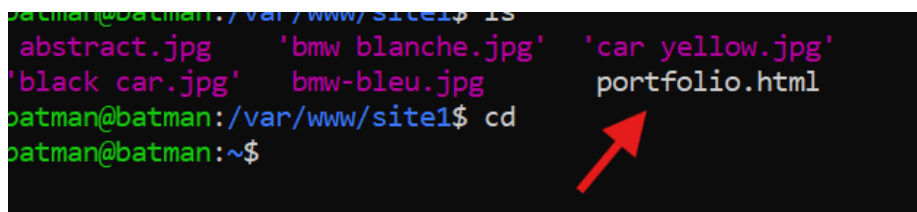
Ensuite crée 2 dossier différent pour nos 2 sites et vérif avec « ls »



```
batman@batman: /var/www$ ls
site1  site2
batman@batman: /var/www$
```

A red arrow points from the text 'crée dossier' in the previous block to the 'site1' and 'site2' directories in the terminal output.

mkdir site1 & **mkdir site2** après aller dans chaque **cd site1** & **cd site2** et crée fichier site=
touch portfolio.html



```
batman@batman: /var/www/site1$ ls
abstract.jpg  'bmw blanche.jpg'  'car yellow.jpg'
'black car.jpg'  bmw-bleu.jpg      portfolio.html
batman@batman: /var/www/site1$ cd
batman@batman: ~$
```

A red arrow points from the text 'crée fichier' in the previous block to the 'portfolio.html' file in the terminal output.

Faire un nano portfolio.html pour y accéder, utiliser WinSCP pour aller changer le code et y mettre des images, plus pratique d'utilisation.

Une fois fait on va s'occuper des fichier conf aller dans= `cd /etc/apache2/ sites-available`

Faire un `touch maconfigsite.conf` après faire un `nano maconfigsite.conf` et y ajouter le texte suivant :

```
<VirtualHost *:80>

    ServerAdmin webmaster@portfolio.batman

    ServerName portfolio.batman

    ServerAlias pbatman

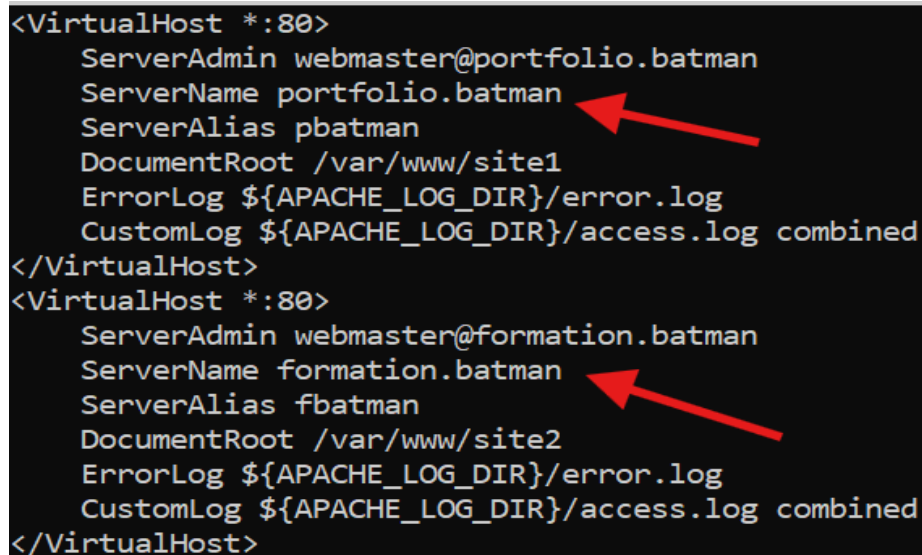
    DocumentRoot /var/www/site1

    ErrorLog ${APACHE_LOG_DIR}/error.log

    CustomLog ${APACHE_LOG_DIR}/access.log combined

</VirtualHost>
```

Résultat : (j'en est 2 pour mes 2 sites)



```
<VirtualHost *:80>
    ServerAdmin webmaster@portfolio.batman
    ServerName portfolio.batman
    ServerAlias pbatman
    DocumentRoot /var/www/site1
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
<VirtualHost *:80>
    ServerAdmin webmaster@formation.batman
    ServerName formation.batman
    ServerAlias fbatman
    DocumentRoot /var/www/site2
    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

Remplacer le nom du site du script par le nôtre, le script peut être copier-coller 2 fois dans le même fichier pour conf les 2 sites dans le même fichier, plus simple, ça évite plusieurs fichiers

Après pour activer on fait : `sudo a2ensite monsite.conf` et un petit `sudo systemctl reload apache2` pour relancer apache2 (à garder on va souvent le faire)

On n'est pas trop mal normalement, après faire un `sudo nano /etc/hosts` on va aller config l'URL site

```

GNU nano 7.2 /e
127.0.0.1    localhost
127.0.1.1    batman
127.0.0.1    portfolio.batman
127.0.0.1    pbatman
127.0.0.1    formation.batman
127.0.0.1    fbatman

# The following lines are desirable for IPv6 capable hosts
::1    localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters

```

On peut même rajouter une ligne identique en dessous mais pour l'Alias donc **127.0.0.1 pbatman** si on veut, faut que l'alias mis soit le même quand dans le fichier **.conf** d'avant

Après faire **sudo systemctl reload apache2**

Après ça reste une petite subtilité à savoir, quand on a plusieurs sites sur la même IP pour y accéder on avec un host name on use la technique bloc note à modif sur Windows system32

c:\windows\system32\drivers\etc\hosts : Ajouter l'IP de la machine virtuelle avec le server-name et l'alias associer

```

# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#       102.54.94.97    rhino.acme.com    # source server
#       38.25.63.10    x.acme.com       # x client host
#
# localhost name resolution is handled within DNS itself.
#
#       127.0.0.1    localhost
#       ::1          localhost
#       172.20.10.8  portfolio.batman
#       172.20.10.8  pbatman
#       172.20.10.8  formation.batman
#       172.20.10.8  fbatman

```

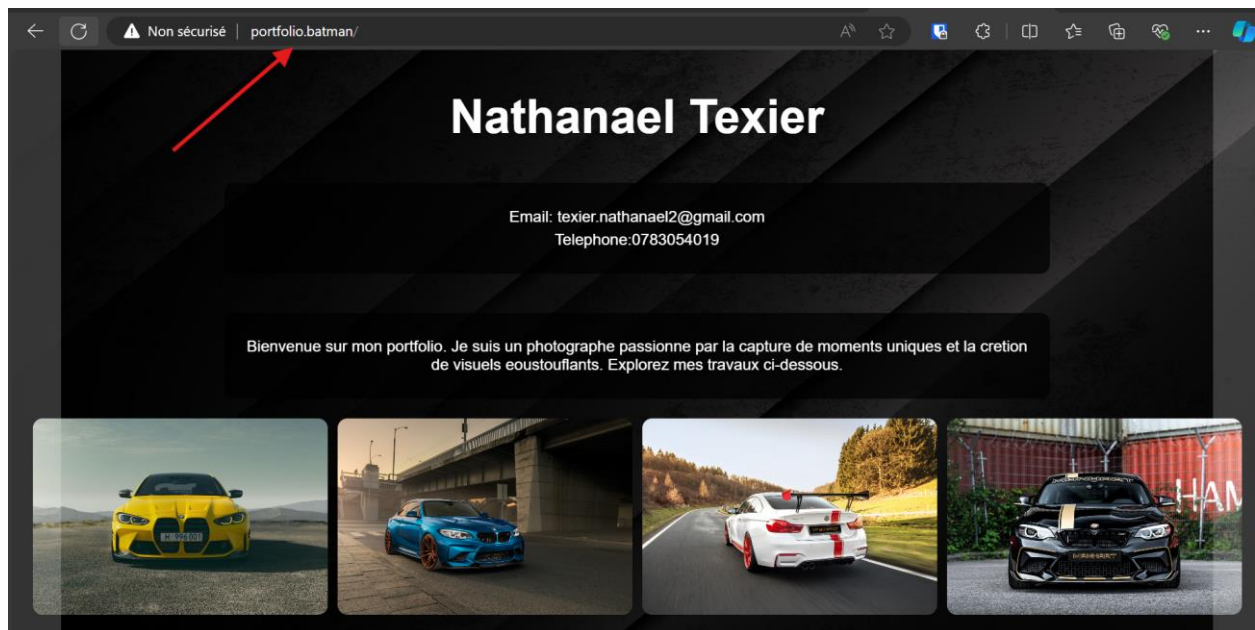
Enregistre le fichier bloc-notes

IMPORTANT : la technique du bloc note marche en interne (local) donc faut changer l'IP des url / alias quand je change de réseaux, mais sinon ça marche

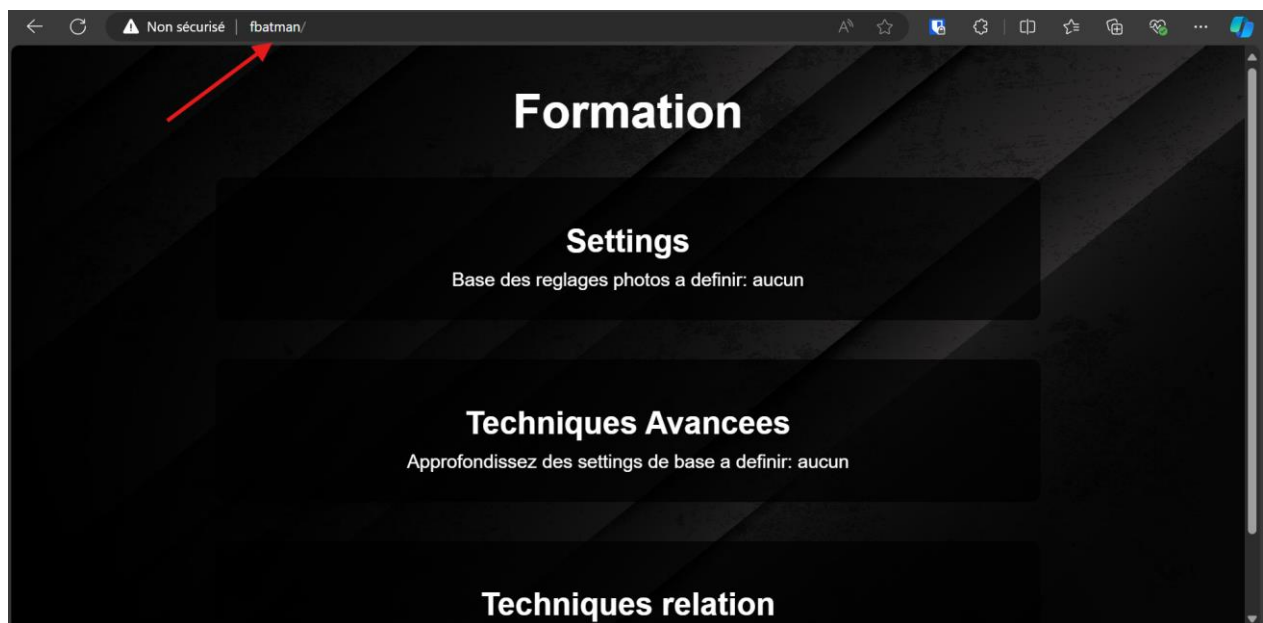
Vérification et Visualisation du TP

On va vérifier que nos sites marchent, et que tout est bien mis en œuvre

Normalement si on fait <http://portfolio.batman> (dans mon cas) je tombe bien sur mon site



Dans le cas du 2ème sites, je me connecter avec l'Alias et ça marche correctement



Problème redirection site = INDEX OF/ clic

Dans le cas où il y aura un problème de redirection (chemin) concernant le site web, suivre la procédure du screen pour résoudre le problème :

```
sudo nano /etc/apache2/sites-available/000-default.conf
```

◦ Modifiez la directive `DocumentRoot` pour qu'elle pointe vers le répertoire de votre site :

```
<VirtualHost *:80>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/site1

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>
```

3. Ajoutez la directive `DirectoryIndex` pour spécifier le fichier d'index par défaut :

◦ Toujours dans le fichier de configuration, ajoutez la ligne suivante pour définir `index.html` comme fichier d'index par défaut :

```
<Directory /var/www/site1>
    DirectoryIndex index.html
</Directory>
```

Faut ajouter le dernier encadrer, ça permet juste de bien rediriger si problème pour aller direct au site SANS AVOIR **LE INDEX OF// CLIC CLIC CLIC** (erreur de redirection)

Conclusion

Conclusion

Ce TP permet la mise en pratique et en œuvre de Apache2 sur une machine virtuel tel que VirtualBox, nous avons vu comment configurer de A-Z notre serveur pour que notre environnement puisse accueillir au mieux nos 2 sites web. La création et la configuration d'un serveur web sous Debian 12, ainsi que la mise en ligne de deux sites web, ont été réalisées avec succès grâce à cette procédure. Les connaissances acquises lors de ce TP sont essentielles pour la gestion de serveurs web et la mise en place d'environnements de développement.