TP Supplémentaire 1 - Mise en œuvre d'un serveur web mutualisé dans une machine virtuelle

L'objectif de ce TP est d'utiliser une machine virtuelle afin d'y installer un serveur web mutualisé. Après une installation basique, nous verrons les problèmes de sécurité que cela pose et des pistes pour les résoudre. Seules les grandes lignes des commandes seront décrites, pour avoir la syntaxe complète d'une commande on vous invite à utiliser le manuel : `man [commande]` *(`[ ]` indique que c'est optionnel), exemple :* `man ls`*.*

IMPORTANT : changer le mot de passe de root !!

1 La machine virtuelle

1.1 Généralités

- L'installation d'une ou plusieurs machine(s) virtuelle(s) est la mise en œuvre du concept de virtualisation : faire fonctionner un ou plusieurs systèmes d'exploitation comme un logiciel quelconque, sur un(e) seul(e) machine physique / ordinateur.
- Une machine virtuelle peut être vue comme un ordinateur virtuel créé par un logiciel d'émulation. Ce dernier se chargeant en quelque sorte de "simuler" la présence de ressources matérielles (comme la mémoire, le processeur, etc.) et logicielles. On parle également d'environnement virtuel.
- Il existe de nombreux logiciels de virtualisation, qui mettent en œuvre différentes techniques. On peut notamment citer *Oracle VM VirtualBox* ou *KVM/QEMU*, qui sont des logiciels libres, ou encore *VMware*. Dans la suite nous utiliserons *VirtualBox* qui est, normalement, déjà installé sur vos machines.

1.2 Créer une machine virtuelle

- lancer l'outil de virtualisation grâce à la commande `virtualbox` dans un terminal, ou alors via l'interface graphique.
- Une fois le logiciel démarré, il faut créer une machine virtuelle, puis vous pourrez y installer un système d'exploitation. La création d'une machine se fait via le bouton `Nouvelle` ou l'item `Nouvelle` du menu `Machine`, il suffit ensuite de répondre aux questions posées en suivant les indications données.

ATTENTION : dans la suite on désignera par le terme machine hôte la "vraie" machine.

Les manipulations à effectuer sont :

1. lancer `virtualbox` ;
2. vérifier que l'`Extension Pack` a été correctement installé en allant au niveau du menu `Fichier` dans l'item `Outils`, puis `Extension Pack Manager` (sinon l'installer). On peut aussi accéder à cette information en cliquant sur la partie droite d'`Outils`, puis en choisissant `Extensions` ;

3. créer ensuite une machine virtuelle avec les caractéristiques suivantes :
   — comme nom mettre `Debian Web server` ;
   — le système d'exploitation sera `Linux` avec `Debian (64-bits)` pour version ;
   — pour la mémoire, prendre un quart de la taille de la mémoire physique réelle (pour avoir la quantité de mémoire installé → `cat /proc/meminfo`). Faites attention aux unités... ;
   — puis créer un disque dur de type `VDI`, `Dynamiquement` alloué avec une taille de 8 Gio ;
   — répondre aux questions restantes, si il y en a ;
4. télécharger l'image iso `debian-11.6.0-amd64-netinst.iso` sur le site `www.debian.org` via `Download` / `Téléchargement`. Il s'agit d'une image permettant de faire une installation par le réseau (*netinstall*) ;
5. procéder à l'installation d'une VM à l'image de `vm-texte` lors d'un TP précédent ;
6. redémarrer la machine virtuelle et connectez vous avec le compte `root`. Vérifier que l'installation minimale en mode texte est bien fonctionnelle et que vous atteignez bien Internet, par exemple via la commande `ping www.univ-fcomte.fr` dans un terminal.

2 Configuration réseau de la machine virtuelle

Dans un premier temps nous allons lier la machine hôte et la machine virtuelle la création d'un `Réseau privé hôte`. L'idée est de créer un réseau privé avec les adresses suivantes :
— adresse de la machine hôte → `192.168.56.1` ;
— adresse de la machine virtuelle → `192.168.56.201` ;
avec comme masque `255.255.255.0`. La configuration de l'adresse de la machine hôte se fait au niveau de l'interface de `Virtualbox`, tandis que celle de la machine virtuelle se fait via son fichier `/etc/network/interfaces` (du moins en utilisant la configuration à l'"ancienne").

ATTENTION : si dans la suite vous avez déjà cette plage d'adresse qui est utilisée pour un autre réseau, vous utiliserez `192.168.57.0/24`, puis si ça ne va toujours pas `192.168.58.0/24`, etc.

Les manipulations à effectuer sont :

1. arrêter / éteindre la machine virtuelle si celle-ci fonctionne ;
2. afin de pouvoir créer un réseau "privé hôte" et configurer son adressage cliquer sur la partie droite d'`Outils`, sélectionner l'item `Réseau`, puis choisir l'onglet `Host-only Networks` et créer un nouveau réseau privé, ce qui fera apparaître une ligne `vboxnet0` si c'est le premier (`vboxnet1` si c'est le deuxième, etc.) ;
3. après avoir sélectionné `vboxnet0`, cliquer sur `Adapter`, puis vérifier que la configuration manuelle du nouvel adaptateur est correcte. Dans l'onglet `Serveur DHCP` vous désactiverez le serveur si celui-ci est actif. **ATTENTION :** dans l'onglet `Interface`, il s'agit de configurer l'adresse qu'aura la machine hôte et non la machine virtuelle ;
4. lancer un `ip address show` dans un terminal sur la machine hôte et constater le résultat ;
5. dans la `Configuration` de la machine choisir `Réseau`, puis sélectionner `Adapter 2` et cocher `Activer la carte réseau` ;
6. choisir le mode d'accès `Réseau privé hôte` avec pour nom `vboxnet0`, valider avec `Ok` ;

7. démarrer la machine virtuelle et utiliser le `networking.service` pour lui affecter en plus l'adresse IP `192.168.56.201`. Il s'agira de définir une configuration statique (inspirez vous du fichier `interfaces` de l'énoncé du TP1 en supplément) pour une nouvelle interface réseau, dont vous déterminerez le nom en utilisant la commande `ip link` dans un terminal.

   ATTENTION : il s'agit d'ajouter une interface supplémentaire dans le fichier `/etc/network/interfaces` et non pas de modifier la configuration de l'interface réseau `enp0s3` ;
8. modifier la configuration du serveur `ssh` pour permettre à `root` de se connecter sur la machine virtuelle ;
9. vérifier que les deux machines peuvent se "pinger" via le réseau privé et qu'une connexion via `ssh` est possible depuis la machine hôte.

3 Serveur web mutualisé

3.1 C'est quoi au juste ?

Les serveurs web mutualisés sont aujourd'hui la base de nombreux sites web. C'est une infrastructure qui est basée sur du Cloud (stockage en ligne). Avant, lorsque l'on voulait mettre en place un site web, il fallait installer un service web (`IIS`, `Apache`, etc.) sur une machine physique appelée serveur web, qui elle-même se trouvait dans une infrastructure réseau (réseau d'entreprise, d'un particulier, etc.). Ce schéma est encore possible, mais à l'heure actuelle on opte souvent pour une installation via un hébergeur qui se charge de tout via le Cloud.

Ainsi, on a plus qu'à s'occuper de développer et installer son site, à savoir la couche la plus haute de ce qui permet à un site web de fonctionner. Un site web standard et qui n'induit pas nécessairement un trafic très important n'a pas besoin d'avoir une machine physique qui lui est dédié. C'est pourquoi plusieurs sites web peuvent être hébergés sur une même machine et un même service web. Un serveur web mutualisé est donc un seul et même service/serveur qui est simplement utilisé par plusieurs clients, avec chaque client qui a son propre site web. Par exemple, si un client 1 achète un emplacement avec quelques ressources pour héberger son site `www.site1.fr` et qu'un client 2 fait de même pour `www.site2.fr`, tous deux seront stockés sur la même machine et gérés par le même service web.

Dans son espace, un client peut habituellement créer une base données et déployer les fichiers de son site sur son espace web via un accès FTP. Parfois, il est possible d'avoir un accès `Shell` qui donne un accès plus avancé, ou un accès fourni par l'hébergeur qui permet de gérer ses services et son espace. En termes de caractéristiques, un serveur web mutualisé c'est :

— un espace disque qui est réservé à l'utilisateur pour héberger son site ;
— une ou plusieurs bases de données dédiées ;
— un accès FTP qui permet d'uploader les fichiers de son site ;
— des droits qui sont en principe restreints pour que tous les utilisateurs vivent ensemble de manière "harmonieuse" ;
— des quotas peuvent être fixés pour l'espace disque, les bases de données, le débit, etc. C'est l'hébergeur qui fixe ces limitations. Celles varient en fonction du prix payé. Dans tous les cas l'hébergeur assure le contrôle et le suivi de l'espace alloué.

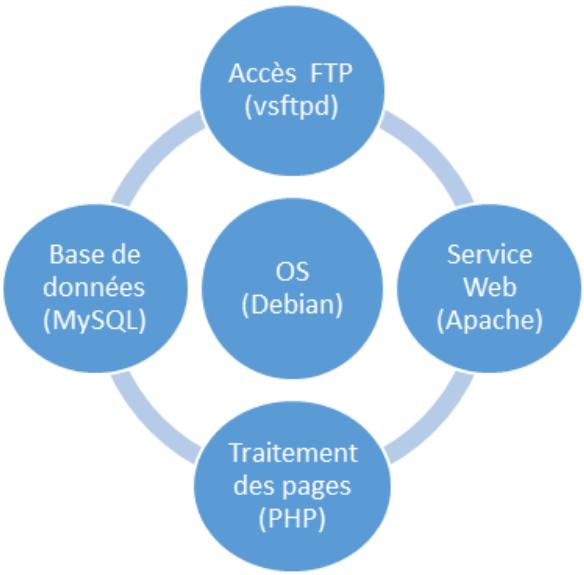
Le fait de ne pas avoir à se préoccuper de la configuration et de l'infrastructure, notamment de la mise à jour du serveur web, est avantageux. Cependant, cela signifie aussi que l'utilisateur se retrouve dépendant du bon vouloir de l'hébergeur en terme de mise à jour, par exemple pour

changer de version de PHP, ou encore en cas de découverte d'une faille de sécurité. Le besoin de modules spécifiques peut également être facturé, inversement de nombreuses fonctionnalités et modules peuvent être activés par défaut ce qui peut poser problème.

3.2 Architecture

La figure ci-après montre les services basiques que nous allons installés pour mettre en place un serveur web mutualisé. Chaque utilisateur aura ainsi un accès FTP, au minimum une base de données et un service web permettant de traiter les requêtes HTTP qui seront réceptionnées par son application web.

Dans la suite nous allons donc installer les différents services, les configurer, et créer une arborescence pour héberger les sites web de deux utilisateurs. Ainsi, à la fin du TP on aura deux utilisateurs : `user-site1` et `user-site2`, qui auront chacun un site web associé, à savoir `www.site1.fr` et `www.site2.fr` respectivement.



3.3 Installation

ATTENTION : en dehors du navigateur qui sera lancé dans la machine hôte, toutes les manipulations qui suivent sont bien entendu à faire dans la machine virtuelle avec le compte `root`.

Les manipulations à effectuer sont :

1. mettre à jour la liste des paquets disponibles depuis le dépôt

   ```
   apt update
   ```

2. mettre à jour les paquets existants

   ```
   apt upgrade
   ```

3. on installe le service web, à savoir `apache` avec `php`

   ```
   apt install apache2 php7.4
   ```

4. on installe un service de base de données, plus les fonctions qui permettront à `php` de communiquer avec une telle base de données

   ```
   apt install default-mysql-server php7.4-mysql
   ```

5. avec le navigateur sur la machine hôte se connecter sur la machine virtuelle et vérifier que vous obtenez bien l'affichage de la page par défaut du serveur Apache.

3.4 Configuration

Idéalement, dans le cas d'un serveur mutualisé, les services devraient être configurés en tenant compte des considérations suivantes :

— Apache / PHP
Le serveur web sera, a priori, utilisé par un très grand nombre de sites web qui vont produire beaucoup de requêtes à traiter. Aussi, il faut notamment vérifier le nombre de connexions simultanées par défaut qui peuvent être gérées et éventuellement modifier la valeur. Il s'agit également de cloisonner les différents sites en prévoyant un dossier web et un `virtualHost` par client / site.

— VSFTP
Idéalement, chaque utilisateur doit pouvoir accéder à son espace et pas à celui d'autres utilisateurs. Il faut donc éviter qu'un utilisateur puisse se promener dans l'arborescence avec son client FTP. Pour ce faire, l'accès de chaque utilisateur sera chrooté et la gestion de la connexion sera faite via la base des utilisateurs du serveur. Lors de la dépose des fichiers par un utilisateur les droits par défauts devront être positionnés pour permettre au service de les lire sans que l'administrateur ait besoin d'intervenir.

— MySQL
La configuration de la base de données doit être calquée sur celle d'Apache au niveau du trafic / des requêtes attendues. Le nombre de connexions maximales doit donc être adapté en conséquence. Chaque utilisateur aura un accès dédié à chacune de ses bases de données et ne devra pas pouvoir accéder à celle des autres.

— Comptes des utilisateurs, droits, groupes et arborescence
Les accès utilisateurs seront gérés via des comptes classiques en fixant leurs droits, leurs groupes d'appartenance, leur `home directory`, etc. Du point de vue du stockage, pour nous simplifier la vie, chaque utilisateur aura un emplacement pour stocker son site. Il s'agira d'un répertoire sur lequel pointera un `virtualHost` d'Apache.

Dans la suite, nous allons configurer les services, mais pas de manière nécessairement correcte, afin de montrer les problèmes de sécurité qui peuvent en découler.

3.4.1 Création de l'arboresence

Nous allons faire tout stocker dans le répertoire `/storage`, qui contiendra les sous-répertoires des utilisateurs, un sous-répertoire pour les sessions, un autre pour les uploads et un dernier pour les logs. Un dernier répertoire sera laissé vide et utilisé pour la configuration par défaut. **ATTENTION :** la quasi totalité des manipulations qui suivent sont à faire dans la machine virtuelle via le compte `root`. Le mieux est d'utiliser SSH pour se connecter à la VM.

Les manipulations à effectuer sont :

1. construction de l'arborescence

```
mkdir /storage
mkdir /storage/web
mkdir /storage/web/default
mkdir /storage/web/sessions
mkdir /storage/web/uploads
mkdir /storage/logs
```

2. on met Apache en propriétaire de quasiment toute l'arborescence, sinon il ne pourra pas accéder aux fichiers

   ```
   chown www-data:www-data /storage/web -Rf
   ```

3. on enlève les droits à `other` sur ces mêmes répertoires

   ```
   chmod 770 /storage/web -Rf
   ```

3.4.2 Configuration d'Apache

La documentation en ligne d'Apache est disponible via le lien `http://httpd.apache.org/docs/2.4/`

Les manipulations à effectuer sont :

1. consulter le contenu du répertoire `/etc/apache2` ;
2. consulter plus en détail le fichier `apache2.conf` ;
3. afficher la liste des modules qui sont chargés via

   ```
   apachectl -M
   ```

4. consulter le répertoire `/etc/apache2/mods-enabled` pour voir les fichiers de configuration ;
5. à quoi sert le module `mpm_prefork_module` (cf. la documentation en ligne), comment est-il configuré et quelles sont les alternatives ? Ajouter dans la configuration du module les paramétrages suivants :

   ```
   MaxRequestWorkers 500
   ServerLimit 500
   ```

6. consulter la section `Optimisation des performances` dans la documentation en ligne ;
7. modifier le fichier `/etc/apache2/sites-available/000-default.conf` qui contient le site par défaut pour le faire pointer sur `/storage/web/default`. Pour cela, modifier en conséquence la valeur `DocumentRoot` dans le fichier.
8. redémarrer le service http

   ```
   systemctl restart apache2
   ```

9. se connecter sur le site par défaut directement via l'adresse IP `192.168.56.201`. Quelles informations sont affichées et qui posent problème ? Aller modifier les paramètres suivants dans le fichier `security.conf` du répertoire `/etc/apache2/conf-available`

   ```
   ServerTokens Prod
   ServerSignature Off
   ```

10. redémarrer (`restart`) ou recharger (`reload`) Apache et se connecter à nouveau sur le site et voir la différence.

3.4.3 Configuration de PHP

Le fichier de configuration est dans le répertoire `/etc/php/7.4/apache2`.

Les manipulations à effectuer sont :

1. éditer le fichier de configuration `php.ini` afin de le modifier au niveau des directives indiquées comme ci-dessous, en lisant les commentaires qui précise le rôle de chacune

   ```
   memory_limit = 256M
   upload_tmp_dir = /storage/web/uploads
   session.save_path = "/storage/web/sessions"
   ```

2. redémarrer Apache.

3.4.4 Configuration de MySQL

La configuration du serveur de base de données se situe dans le répertoire `/etc/mysql`. Ici on ne va rien modifier, il s'agira juste de voir comment se fait la configuration, quels fichiers sont chargés et dans quel ordre.

Les manipulations à effectuer sont :

1. consulter le fichier `my.cnf` et en déduire comment se fait la configuration ;
2. dans quel fichier trouve-t-on les directives `max_allowed_paquet` et `max_connections` ?
3. installer `phpmyadmin` en suivant les indications disponible sur la page accessible à l'adresse `https://www.itzgeek.com/how-tos/linux/debian/how-to-install-phpmyadmin-with-apache-on-debian-10.html` ;
4. vérifier que l'on peut se connecter au serveur de bases de données avec le navigateur via `http://192.168.56.201/phpMyAdmin` en utilisant le login `app_user`.

3.4.5 Configuration des logs

On va simplement rediriger les logs vers notre répertoire personnalisé.

Les manipulations à effectuer sont :

1. éditer le fichier `/etc/logrotate.d/apache2` et remplacer le chemin `/var/log/apache2/*.log` au début du fichier par `/storage/logs/*.log` ;
2. redémarrer les services concernés par ce changement

   ```
   systemctl restart apache2
   systemctl restart rsyslog
   ```

3. vérifier que les services ont correctement redémarrés

   ```
   systemctl status apache2.service
   systemctl status rsyslog.service
   ```

3.4.6 Configuration de VSFTP

L'élément important dans la configuration de ce service est la restriction de chaque utilisateur utilisant FTP à son seul répertoire web. Pour cela, on utilise le chroot pour l'empêcher d'aller chez les voisins ou se promener dans le serveur.

Les manipulations à effectuer sont :

1. installer le service FTP

   ```
   apt-get install vsftpd
   ```

2. éditer le fichier `/etc/vsftpd.conf` et modifier, si besoin, ou ajouter, si absente, les directives comme indiquées ci-dessous. Les commentaires vous indiqueront le rôle de chacune.

```
anonymous_enable=NO
local_enable=YES
write_enable=YES
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_file=/var/log/vsftpd.log
chroot_local_user=NO
chroot_list_enable=YES
chroot_list_file=/etc/vsftpd.chroot_list
allow_writeable_chroot=YES
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
```

3. créer le fichier (le nom du fichier est donné dans une directive ci-dessus...) qui contiendra la liste des utilisateurs à chrooté en y mettant

   ```
   user-site1
   user-site2
   ```

   Comme vous l'avez sûrement compris, `user-site1` et `user-site2` seront les logins respectifs des deux utilisateurs pour lesquels nous mettons en place des sites web ;
4. ajouter le droit `x` à `other` au niveau de `/storage` et `/storage/web` ;
5. comme d'habitude, après modification, on redémarre le service

   ```
   systemctl restart vsftpd.service
   ```

3.5 Création des utilisateurs et des sites

Les manipulations à effectuer sont :

1. création des comptes des deux utilisateurs `user-site1` et `user-site2`. Vous mettrez comme mots de passe respectifs `site1` et `site2`

   ```
   adduser user-site1 --home /storage/web/www-site1
   adduser user-site2 --home /storage/web/www-site2
   chmod 770 /storage/web/www-site1 /storage/web/www-site2
   ```

2. création des répertoires des logs

   ```
   mkdir /storage/logs/www-site1
   mkdir /storage/logs/www-site2
   chmod 777 /storage/logs -Rf
   ```

3. se connecter avec le compte `root` sur `mysql`

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

4. création des bases de données. Seule la procédure pour `user-site1` est présentée (le mot de passe utilisé est `site1`), il suffit d'adapter pour faire de même pour `user-site2`

   ```
   create database sql01site1;
   create user "user-site1"@"%" identified by "site1";
   grant all privileges on sql01site1.* to "user-site1"@"%";
   flush privileges;
   ```

5. il reste à créer les deux sites au niveau d'Apache. Cela se fait sous la forme d'un fichier de configuration dans `/etc/apache2/sites-available`. Voici le fichier pour le site de `user-site1`, soit `site1.conf`. Vous vous baserez sur celui-ci pour définir la configuration du site du second utilisateur.

```
<VirtualHost *:80>
  ServerAdmin webmaster@localhost
  ServerName www.site1.fr
  DocumentRoot /storage/web/www-site1
  Alias /database "/usr/share/phpMyAdmin/"
  ErrorLog "/storage/logs/www-site1/site1-error.log"
  CustomLog "/storage/logs/www-site1/site1-access.log" common
  <Directory /storage/web/www-site1>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride none
    Require all granted
  </Directory>
</VirtualHost>
```

6. on active les 2 sites via `a2ensite site1` et `a2ensite site2` ;
7. puis on recharge Apache comme indiqué.

3.6 Test et première correction

Pour tester l'accès on va se contenter de faire un petit fichier `index.php` pour chaque utilisateur. Chaque fichier affichera un simple message et sera saisi et stocké sur la machine hôte (donc pas dans la machine virtuelle). Il faudra alors utiliser `filezilla` pour télécharger chacun des fichiers dans l'espace dédié à l'utilisateur correspondant (après l'avoir installé).

Les manipulations à effectuer sont :

1. éditer le fichier `index.php` sur la machine hôte (pas dans la virtuelle) afin qu'il contienne

```
<html>
<body>
<?php
  echo "Welcome on site 1";
?>
</body>
</html>
```

2. uploader le fichier dans le l'espace dédié à `user-site1` sur la machine virtuelle ;
3. vérifier que vous êtes bien chrooté et donc que vous ne pouvez pas remonter dans l'arborescence et vous promener ;
4. se connecter avec le navigateur sur le site `http://www.site1.fr`. Est-ce votre site ? Que faut-il mettre dans le fichier `/etc/hosts` de la machine hôte pour pouvoir se connecter ?
5. que constatez-vous ? Cela a-t-il fonctionné ou pas ?
6. refaire toutes les étapes précédentes, mais pour le site de l'utilisateur `user-site2`.

Pas de panique si cela ne marche pas, c'est normal, mais pourquoi ?

— Regarder dans la machine virtuelle les droits des fichiers `index.php` qui ont été téléchargés. Qu'en déduisez-vous ?

— Le transfert via FTP fait que les fichiers `index.php` n'ont pas pour groupe `www-data`, donc le service web n'a pas le droit de les lire...

Une solution possible est de donner les droits `rx` à `other` sur l'arborescence du site web de `user-site1` (idem pour le site du second utilisateur)

```
chmod o+rx /storage/web/www-site1 -Rf
```

Le souci est qu'il faudra remettre ces droits sur tous les fichiers que l'on upload au fur et à mesure... Pour mettre fin a cet inconvénient, on va reconfigurer le service FTP.

Les manipulations à effectuer sont :

1. on propose de résoudre le problème en modifiant / ajoutant ces directives dans le fichier de configuration du service FTP :

   ```
   local_umask=002
   file_open_mode=0777
   ```

2. redémarrer le service impacté par la modification

   ```
   systemctl restart vsftpd.service
   ```

3. supprimer ler fichiers `index.php`, puis les uploader à nouveau et constater le changement au niveau des droits ;
4. se connecter à nouveau sur un des sites web et vérifier que ça fonctionne. **ATTENTION :** dans le cas où la page du ne s'affiche pas, c'est qu'il y a peut être un souci de droit d'accès. Auquel cas il faut vérifier les droits de répertoires / fichiers et éventuellement modifier le fichier `/etc/apache2/apache2.conf`.

3.7 Problèmes de sécurité

Même si un utilisateur est confiné dans son chroot, le fait que les scripts PHP sont exécutés par Apache et donc l'utilisateur `www-data` pose problème. En effet, l'utilisateur `www-data` n'est pas coincé par le chroot et a des droits plus étendus sur le serveur. Ainsi, si un utilisateur demande l'exécution d'un script pour lire le fichier d'un voisin, cela fonctionnera car Apache dispose des droits adéquats.

Pour illustrer cela, nous allons commencer par montrer que l'on peut exécuter aisément une commande Linux pour obtenir une information qui pourrait être exploitée par un tiers qui aurait un accès non autorisé à l'espace de `user-site2`.

Les manipulations à effectuer sont :

1. éditer le fichier `shell.php` comme suit sur la machine hôte

   ```
   <?php
     $resultat=shell_exec('pwd');
     echo $resultat;
   ?>
   ```

2. l'uploader dans le site de `user-site1` ;
3. le lire dans le navigateur ;
4. modifier le script pour afficher le contenu de l'espace de `user-site2` ;
5. finalement afficher, toujours en modifiant le script, le contenu du fichier `index.php` de `user-site2` via `cat`. Vous afficherez le code source de la page via `View Page Source`.

Supposons maintenant que `user-site2` ajoute un fichier dont il protège la lecture via un `.htaccess`. Pour cela il procède comme décrit ci-dessous.

Les manipulations à effectuer sont :

1. éditer le fichier `restreint.php` comme suit sur la machine hôte

   ```
   <?php
     echo "Je suis le fichier avec un acces restreint... normalement";
   ?>
   ```

2. l'uploader dans le site de `user-site2` ;
3. afin de créer le fichier contenant le mot de passe permettant de contrôler l'accès au fichier, nous aurons besoin de la commande `htpasswd` sur la machine hôte. Vérifier qu'elle est installée, sinon procéder comme suit :

   ```
   apt-get install apache2-utils
   ```

4. créer le fichier `.htpasswd` sur la machine hôte, celui-ci nous servira à contrôler l'accès au fichier `restreint.php`

   ```
   htpasswd -c .htpasswd user-site2
   ```

   il vous faudra donner un mot de passe. À noter que l'option `-c` doit être omise si on veut ajouter un autre utilisateur ;
5. éditer un fichier `.htaccess` sur la machine hôte avec le contenu suivant :

   ```
   AuthUserFile /storage/web/www-site2/.htpasswd
   AuthType Basic
   AuthName "Protected file"
   <Files "restreint.php">
     Require valid-user
   </Files>
   ```

6. uploader les deux fichiers dans le répertoire où se trouve le fichier `restreint.php` ;
7. modifier le fichier de configuration `site2.conf` au niveau de la directive `AllowOverride`

   ```
   AllowOverride AuthConfig
   ```

8. redémarrer le service web ;
9. puis essayer d'accéder au fichier `restreint.php` grâce au navigateur via `www.site2.fr` et constater qu'une fenêtre contrôlant l'accès s'ouvre ;
10. modifier le script `shell.php` pour qu'il affiche le contenu de `restreint.php` ;
11. que constatez-vous en lisant le fichier `restreint.php` via `www.site1.fr` ?
    Vous afficherez le code source de la page via `View Page Source`.

3.8 Solutions

Pour résoudre le problème des droits liés à l'utilisateur `www-data`, nous allons tirer parti du module `mpm_itk_module` qui permet de faire s'exécuter chaque `virtualHost` sous un couple `uid` /`gid` différent (en l'occurrence ceux de l'utilisateur associé à un site).

Les manipulations à effectuer sont :

1. installer le module via

`apt-get install libapache2-mpm-itk`

2. ajouter la ligne suivante avant le bloc `<Directory> ... </Directory>` dans le fichier de configuration du site de `user-site1`

   `AssignUserId user-site1 user-site1`

3. faire de même, en adaptant la ligne, pour le site de `user-site2`
4. puis recharger Apache via `systemctl reload apache2` ;
5. modifier les droits de `restreint.php` dans la machine virtuelle pour que `other` n'ait plus aucun droit sur le fichier ;
6. vérifier qu'il n'est plus possible d'accéder aux fichiers du voisin ;
7. les droits des sous-répertoires et fichiers `web` pourront à ce moment là également être repositionnés en `750` et pour que cela soit pérenne il faut modifier le service FTP via la directive `local_umask=027`. Le service FTP devra être bien entendu redémarré.

Passons à la résolution de quelques soucis avec PHP. Le premier souci concerne le fait que l'on peut toujours lire des fichiers accessibles par tout le monde sur le système, par exemple `/etc/passwd`. Le second est l'usage de la fonction `shell_exec`. Voyons comment résoudre ces deux problèmes.

Les manipulations à effectuer sont :

1. editer le fichier `lire.php` pour qu'il contienne les lignes suivantes :

```
<?php
  $list=file_get_contents('/etc/passwd');
  echo $list;
?>
```

2. uploader le fichier dans `www.site1.fr` et le lire. Que constatez-vous ?
3. ajouter la directive suivante dans le fichier de configuration du site, juste avant `Options` dans le bloc `Directory`

   `php_admin_value open_basedir "/storage/web/www-site1/"`

   puis redémarrer le service web ;

4. relire le fichier et constater le changement. On voit donc que cette directive restreint la portée des fichiers atteignables via PHP. Le fait que le paramètre `php_admin_value` soit dans la configuration du `virtualHost` permet de régler finement les droits, contrairement à un réglage global dans un fichier de configuration de PHP ;
5. justement, pour désactiver la fonction `shell_exec`, c'est au niveau du fichier de configuration globale qu'il faut intervenir. Il faut ainsi ajouter la fonction dans la liste de celles qui sont désactivées (`disable_functions`) dans le fichier suivant :

   `/etc/php/7.4/apache2/php.ini`