TP5 - Configuration d'un serveur web mutualisé

Tips and tricks pour sécuriser Apache

Ce TP s'inscrit dans le cadre du TP2 de SAE et constitue en quelques sorte un TP3. Vous verrez comment ajouter un service FTP pour permettre à un utilisateur de déployer les fichiers de son site web et des aspects de configuration. Pour ce dernier point, vous verrez plus particulièrement comment pallier certains problèmes de sécurité, dont certains vus lors du TP2. La machine virtuelle est dans l'état qui permet de démarrer avec la Section 4. Activation de TLS de l'énoncé de TP2.

Comme d'habitude, il peut être nécessaire d'être connecté en super-utilisateur (`root`). 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 1 : changer le mot de passe de root!!

IMPORTANT 2 : les manipulations sont à faire dans la machine virtuelle

1 Création de l'arborescence

Pour ce TP, 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`.

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
   ```

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. désactiver les deux sites existants, définis par `site1.conf` et `site2.conf`, puis réactiver le site par défaut (cf. la Section 3.3 de l'énoncé de TP2 de SAE) ;
9. redémarrer le service http

   ```
   systemctl restart apache2
   ```

10. se connecter sur le site par défaut directement via l'adresse IP `192.168.62.201`. 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
    ```

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

3 Configuration de PHP

Le fichier de configuration est dans le répertoire `/etc/php/8.2/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.

4 Configuration des logs d'Apache

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
   ```

5 Configuration du service 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.

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 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
   ```

5.1 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 `www-site1.conf`. Vous vous baserez sur celui-ci pour définir la configuration du site du second utilisateur.

   ```
   <VirtualHost *:80>
     ServerAdmin webmaster@localhost
     ServerName www.user-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 www-site1` et `a2ensite www-site2` ;
7. puis on recharge Apache comme indiqué.

5.2 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 website of user-site1";
?>
</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 les 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`.

6 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`.

ATTENTION 1 : la suite reprend la Section 5 du TP2 qui a été modifiée par rapport à l'arborescence des sites.

ATTENTION 2 : les fichiers PHP sont en principe déjà présents dans le compte *tpsae* de la machine hôte (la "vraie" machine).

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 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 `www-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.user-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.user-site1.fr` ?
    Vous afficherez le code source de la page via `View Page Source`.

6.1 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. ajouter la directive suivante, qui est celle pour le site de `user-site1`, dans les fichiers de configuration des différents sites en l'adaptant, juste avant `Options` dans le bloc `Directory`

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

   puis redémarrer le service web ;
2. 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/8.2/apache2/php.ini`

7 Contenu d'un répertoire et liens symboliques

Les manipulations à effectuer sont :

1. changer le nom du fichier `index.php` dans le site `www.user-site1.fr` en `old.index.php` ;
2. recharger le site `www.user-site1.fr` dans le navigateur et constater le problème ;
3. pour empêcher cela, modifier les `Options` dans le fichier de configuration du site comme suit

   ```
   Options -Indexes -FollowSymLinks +MultiViews
   ```
4. quel est le rôle de l'option `MultiViews` et à quel module est-elle liée ?
5. remettre le nom du fichier comme au départ (`index.php`) et vérifier que le message *Welcome on website of user-site1* s'affiche à nouveau.

8 Prévenir des attaques via les en-têtes de sécurité HTTP

Les navigateurs implémentent de nombreux mécanismes de défense contre des attaques qui sont rarement mis en œuvre. Vous allez donc en apprendre un peu plus sur les en-têtes HTTP que peut renvoyer un serveur en réponse à des requêtes.

Liens utiles :

- Une page de Saas Production qui explique différents en-têtes.
- `securityheaders.io` donne une note à un site web en analysant les en-têtes de sécurité.
- Les ressources pour développeurs de Mozilla - consulter l'item *Web Technology*, puis *Security* de l'item de menu `References` en choisissant la langue English (US) ; dans l'item de menu `Tools`, tester l'*HTTP Observatory*.