Travaux Pratiques n°1

Etudes et configuration de serveurs pour le Web
Serveur Web : Apache

But du TP

Dans ce TP nous allons étudier la configuration d'un serveur Web Apache2 pour plusieurs sites, voir comment activer ses modules et crypter l'accès à ces sites.

Mise en place

Cette manipulation peut être effectuée sur chacun des postes des salles 410, 405 voire 102 sans notion de poste client et poste serveur. Tous les PC peuvent être à la fois clients et serveurs. Comme vous devez être administrateur sur le système, vous devrez travailler sur une machine virtuelle. Ce TP suppose l'utilisation du disque virtuel `Debian1102-20220323-pm…`, Il faut donc créer une machine virtuelle comme indiqué en adaptant le nom de votre vm.

```
createvm
la commande sans paramètre vous donne la syntaxe à utiliser.
createvm Debian1102-20220323-pm…
```

Il faut changer l'adresse MAC de l'interface en1 si elle commence par 9201 en 9202 !!!!

Faites démarrer votre VM, logez-vous sur la VM en root ou rt. Vérifiez que l'adresse IP de en1 finit bien (son dernier octet) par 2xx (par ex 201)

```
ip a s
```

Dans tout le TP il va être question d'adresses IP et de noms de machines. Les exemples utilisent soit la salle 405 soit la 410, soit la 102 : il convient de tout ramener à la salle dans laquelle vous effectuez la manipulation.

I. Apache multi-site

Nous allons faire en sorte que votre serveur Web réagisse différemment suivant le nom ou le port demandé. C'est la configuration la plus fréquente d'un serveur multi-site, mais nous garderons le site par défaut pour prévenir la personne qui navigue, qu'elle s'est fourvoyée.

1. Configuration initiale

Sur votre machine virtuelle apache 2.4 est déjà installé avec la configuration par défaut.

Tous les fichiers de configuration sont situés dans `/etc/apache2`

- Le fichier principal est `apache2.conf` : il contient tous les paramètres généraux qui affectent le fonctionnement du logiciel lui-même. Le reste du paramétrage, en particulier celui des sites, est obtenu par inclusion de fichiers partiels.
- Le fichier `envvars` contient des variables d'environnement qui modifient le démarrage du logiciel
- La gestion des ports est dans `ports.conf`
- Les modules d'extension installés sont dans le répertoire `mods-available` et sont activés par un lien dans `mods-enabled` au moyen de la commande `a2enmod`
- Les différents sites configurés sont dans le répertoire `sites-available` et sont configurés par un lien dans `sites-enabled` au moyen de la commande `a2ensite`
- Le répertoire `conf.d` n'existe plus en 2.4 mais est remplacé par le même mécanisme de `conf-available / conf-enabled` qui sert à mettre les configurations des logiciels installés qui ont besoin de `apache2` : nous y verrons, par exemple, plus tard le lien pour `phpmyadmin`. Elles sont rendues visibles au moyen de la commande `a2enconf` comme pour les modules et les sites.

Jetez un coup d'œil sur les différents fichiers de configuration pour en comprendre l'architecture. Elle est décrire au début du fichier `/etc/apache2/apache2.conf`.

Le site par défaut est accessible avec un navigateur à l'url `http://127.0.0.1` : le vérifier.

2. Sites virtuels distingués par noms

Nous allons distinguer plusieurs noms pour votre serveur web et nous devons connaître son adresse IPv4. J'appellerai dans la suite `addrIpWeb` l'adresse IP de votre serveur web qui devrait être du style 10.4.1{05,10}.1(*numéro du pc*). Votre serveur web aura plusieurs noms qui correspondent à ce qui a été défini dans le serveur de noms de la zone `gtr.tp`. Le véritable nom de votre pc est « `rt4{10,05}p1`(*numéro du pc sur deux chiffres*)`.gtr.tp` ». Par exemple la première machine virtuelle de la salle 405 a pour nom `rt405p101.gtr.tp` et pour adresse 10.4.105.101. Cette machine dispose en plus de 3 alias : ils sont obtenus en changeant « rt » par a, b ou c dans le nom. Sur l'exemple précédent il s'agirait des alias a405p101 b405p101 et c405p101.

Vérifier l'adresse ip de votre serveur web, son nom et ses alias, en adaptant la commande à votre VM.

```
host a405p101
host b405p101
```

a) La partie site

On veut qu'une interrogation à l'adresse `http://a405p1xx` donne le `site-a`, à l'adresse `http://b405p1xx` donne le `site-b` et que sur le port 8080 on obtienne le `site-8080`.

On va créer l'architecture de répertoires et de fichiers nécessaires à la distinction des trois sites.

Dans le répertoire `/var` il existe le répertoire `www` dans lequel réside le site par défaut. Créer les répertoires `/var/www-site-a`, `/var/www-site-b` et `/var/www-site-8080` dans le répertoire `/`.

Créer le fichier `index.html` dont le contenu pourrait être celui du listing 1 page 3 dans chacun des trois nouveaux répertoires et modifier son contenu pour qu'ils fassent apparaître respectivement « Site A », « Site B » et « Site 8080 ».

```
<html>
  <head>
  </head>
  <body>
    <h1> Bonjour !</h1>
    <h2> Site A </h2>
  </body>
</html>
```

Listing 1: Exemple de fichier index.html pour le site A

b) La partie configuration

Dans le répertoire `/etc/apache2/sites-available/` éditez un fichier `site-a.conf` dont le contenu correspond au listing n°2 page 4. Il convient bien évidemment d'adapter le numéro de la salle et celui du pc à **votre** contexte. Tous les fichiers de configuration d'apache2 ont un nom qui finit par « `.conf` ».

L'interprétation des trois premières lignes de cette définition peut se faire ainsi :

- pour tous les clients qui arrivent sur le port 80, et qui utilisent le nom de serveur « `a410p108.gtr.tp` », le serveur montre comme racine du site le répertoire « `/var/www-site-a` ».

Définir de la même manière le site-b.

```
<VirtualHost *:80>
        ServerName a410p108.gtr.tp
        DocumentRoot /var/www-site-a
        ServerAlias a410p108
            <Directory /var/www-site-a>
                    Require all granted
            </Directory>
</VirtualHost>
```

Listing 2: Contenu du fichier site-a définissant le site virtuel A du pc rt410p108

La commande `a2ensite` vous permet d'activer ces deux nouveaux sites

Recharger la nouvelle configuration

```
# service apache2 reload
```

Observer ce que vous obtenez avec une url pointant vers l'adresse IP, vers le nom en rt..., le nom en a..., le nom en b...

Conclusion ?

Vérifiez à quoi sert la directive `ServerAlias`.

Si vous constatez un fonctionnement bizarre, ne pas hésiter à vider les caches voire redémarrer le navigateur ou passer en navigation privée.

3. Site distingué par le port TCP

Nous avons construit plus haut un répertoire `/var/www-site-8080` qui contient un fichier d'index.

Il faut qu'il soit vu au moyen d'une requête pointant sur le port 8080.

Dans le fichier `ports.conf`

- ajouter la directive « `Listen 8080` » pour qu'apache écoute ce qui vient sur le port 8080

Editez un fichier de site virtuel « `site-8080.conf` » adapté à la configuration souhaitée (port 8080), c'est-à-dire qui commence par `<VirtualHost *:8080>`

L'activer, recharger la configuration apache

Tester le nouveau site en utilisant l'adresse IP, les différents noms mais toujours sur le port 8080.

Conclusion ?

Le nom (`ServerName`) ne sert ici à rien car si apache ne trouve pas de site dont le nom correspond, il utilise le premier serveur virtuel dont l'adresse et le port correspondent, et comme il n'y en a qu'un…

II. Quelques limites d'accès

Nous allons limiter les accès aux sites de manières différentes. Une première est de limiter en fonction de l'origine des clients (adresses IP, domaine…)

1. Interdictions d'adresse IP

Les versions précédentes d'apache utilisaient les trois directives `Order`, `Allow` et `Deny`. Depuis la version 2.4 du serveur apache, elles sont obsolètes même si beaucoup de sites les utilisent encore. Il est fortement déconseillé de continuer à les utiliser car leur développement n'est plus assuré et elles vont disparaître d'où un sérieux problème de sécurité. Il faut maintenant utiliser la directive `Require`.

Cette directive peut interdire ou autoriser l'accès en fonction de l'adresse ip, du nom ou du domaine du client, d'une variable d'environnement, une méthode d'accès http (GET, POST…), un utilisateur, des utilisateurs validés, des groupes, et plein d'autres qui dépendent des modules additionnels `mod_authz_xxx`. Les plus usuels étant `mod_authz_core`, `mod_authz_user`, `mod_authz_host`, `mod_authz_groupefile`.

Prenons un exemple : on veut autoriser 8 machines de la salle à accéder à votre site A mais l'interdire à toutes les autres. Dans le fichier de définition du site A, qui correspond au listing 2 page 4, modifier l'option `Require` comme indiqué dans le listing n°3 page 5. Bien sûr remplacer « `addrIpWeb` » par l'adresse IP du serveur Web.

```
<Directory /var/www-site-a >
        Require ip addrIpWeb/29
</Directory>
```

Listing 3: Autorisations du site A modifiées

Les diverses directives `Require` peuvent être englobées dans des `<RequireAll> </RequireAll>` pour faire un ET logique et `<RequireAny> </RequireAny>` pour faire un OU logique.

Quelles sont les adresses autorisées, et les adresses interdites avec cette règle ? Le vérifier. Vous pouvez utiliser `sipcalc` pour facilement calculer les adresses qui sont autorisées.

Je vous rappelle qu'il est conseillé de **redémarrer** le serveur web après chaque modification des fichiers de configuration, le `reload` est plus rapide et n'interrompt pas le service, mais toutes les modifications ne sont pas prises en compte.

2. Autorisation avec un compte

Dans ce cas il nous faut authentifier l'utilisateur, il va donc devoir donner un nom et un mot de passe. Pour cela en Web, il est fortement conseillé de passer par `https` ou `tls`. Avant de voir comment effectuer cette authentification, il nous faut donc un site avec du ssl. Sans rentrer dans la cryptographie qui sera traitée ailleurs, nous allons devoir activer des certificats, des clés et toute une PKI… ? Non !

a) Activation du ssl

Un des paquets préinstallés a déjà presque tout mis en place. Malheureusement dans la vraie vie ce ne sera pas si simple : il faudra avoir un vrai certificat issu d'une vraie autorité de certification…

Il nous suffit donc d'activer (a2ensite) le site ssl par défaut pour pouvoir tester.

Vérifier que cela ne fonctionne pas ! Aviez-vous relancé apache ?

Vérifier que cela ne fonctionne toujours pas ! Aviez-vous vérifié que le module ssl est activé ? L'activer (a2enmod) et re-relancer apache.

Maintenant vous pouvez vérifier que le site par défaut est aussi accessible en https. Que se passe-t-il si l'on accède avec les différents noms (rt…, a…, b…) ? Expliquez !

b) On crypte les accès au site B

Nous allons transformer le site B en site uniquement accessible par https. Le répertoire du site BS reste le même que celui du site B. Seule la définition change.

Copiez la définition du site-B, site-b.conf en site-bs.conf et ajoutez au bon endroit du fichier site-bs.conf les 3 directives SSL non commentées qui sont présentes dans le fichier default-ssl.conf. Il faut aussi modifier le port de l'hôte virtuel de *:80 en *:443.

Désactivez les sites site-b.conf et default-ssl.conf, activer le site-bs.conf et redémarrer apache ! Ça marche ?

Que voyez-vous si vous utilisez l'url en https et l'url en http ? Expliquez.

Vérifiez que comme pour le site-8080.conf la définition du port est prépondérante par rapport aux noms de serveur.

```
<IfModule mod_authnz_external.c>
      AddExternalAuth pwauth /usr/sbin/pwauth
      SetExternalAuthMethod pwauth pipe
</IfModule>
<Directory /var/www-site-b >
      AuthType Basic
      AuthName "Acces restreint"
      Require valid-user
      AuthBasicProvider external
      AuthExternal pwauth
</Directory>
```

Listing 4: Limitation d'accès au site B aux utilisateurs déclarés

c) On limite l'accès à certains utilisateurs

L'authentification va se faire au moyen de PAM, le système d'authentification de Linux par défaut.

Il nous faut ajouter quelques modules[1]

```
# apt-get install libapache2-mod-authnz-external pwauth
```

On active le nouveau module

```
# a2enmod authnz_external
```

Les modules en `authn` sont des modules d'authentification (vérification user/pass) et les modules en `authz` sont des modules d'autorisation (donne accès à une zone du site). Le module `authnz_external` fait les deux.

Dans la définition du site BS, ajoutez les directives du listing n°4 page 6, en éliminant la précédente directive `Directory`.

Vérifiez que vous devez maintenant vous authentifier. Créer plusieurs utilisateurs supplémentaires par `adduser` pour pouvoir faire plusieurs tests.

Que se passe-t-il si vous indiquez le nom de machine en a ou en rt ? Expliquez.

d) Utilisation des répertoires utilisateurs

Nous allons maintenant utiliser le troisième site sur le port 8080. Dans l'URL la partie « ~rt/ » indique que l'on veut le contenu web proposé par l'utilisateur `rt`.

Vous pouvez tester que l'url `http://votreServeurWeb:8080` répond mais pas `http://votreServeurWeb:8080/~rt/`.

Dans la définition du site-8080 **ajouter** le contenu du listing n°5 page 7.

Il faut aussi activer le module `userdir`.

Si cela ne fonctionne pas pour l'url `http://votreServeurWeb:8080/~rt/`, c'est probablement que le répertoire unix `~rt/www` n'existe pas : le créer et re-vérifiez. À moins que le module `userdir` ne soit pas actif ?

Que ce passe-t-il, il faut le voir avec

```
# tail -f /var/log/apache2/error.log
```

```
<IfModule mod_userdir.c>
Userdir www
Userdir disabled root
<Directory /home/*/www>
   Require all granted
</Directory>
</IfModule>
```

Listing 5: Déclaration du nom du répertoire utilisateur

Qu'indique le fichier de log ?

1 Il faut probablement effectuer un `apt-get update` au paravant.

Ajoutez l'option « Options +Indexes » pour que l'index soit généré, ou laisser la définition du site telle quelle est, et ajoutez un fichier d'index dans le répertoire. Vérifiez que cela est mieux.

e) Limitation avec un compte dynamique

Il est facile de limiter l'accès à une personne ou à un groupe avec des identifiants dynamiques. J'entends par là qu'il ne s'agit pas d'un véritable compte informatique sur le système mais d'un accès avec des identifiants choisis par un simple utilisateur.

Dans le répertoire `~rt/www` , créer un répertoire « `pourTitus` » : ce répertoire ne sera accessible qu'au pseudo utilisateur titus avec un mot de passe « `emoiemoiemoi` ».

Dans le répertoire `~rt/www/pourTitus` créez un fichier `.htaccess` qui contient le listing n°6 page 8. Je pense qu'il n'y a besoin d'explications que pour la ligne `AuthUserFile`.

Le lien identifiant ↔ mot de passe est dans le fichier `/home/rt/.htUnPass-titus`. Nous devons le créer avec le compte de rt (pas root) :

```
htpasswd -c /home/rt/.htUnPass-titus titus
```

En pratique le fichier `.htaccess` n'est pas lu, car les répertoires au-dessus interdisent l'usage de ces `.htaccess`. Pour les autoriser ajouter à la configuration du site, dans les règles d'accès aux répertoires `/home/*/www` la directive « `AllowOverride All` ».

Après avoir redémarré apache2, vérifier qu'il est possible d'accéder à l'url `http://votreServeurWeb:8080/~rt/pourTitus` sous le nom de `titus` avec le bon mot de passe. **Attention :** comme vous l'avez peut-être lu dans la bannière l'envoi du mot de passe est **en clair** ! Il aurait fallu passer par https.

```
AuthType Basic
AuthName "Acces restreint"
Require user titus

AuthUserFile /home/rt/.htUnPass-titus
```

Listing 6: Autorisation du répertoire `~rt/www/pourTitus`

III. Visualisation des variables d'environnement

Pour pouvoir afficher facilement les variables qu'utilise apache2.4, le plus simple est d'utiliser le langage de script `php7`. Nous allons les afficher en utilisant le protocole crypté.

1. Configuration et installation php

Le module qui ajoute la fonctionnalité php est normalement déjà installé et actif. Le vérifier avec la commande :

```
# a2enmod php7.4
```

au besoin relancez apache2.

```
# systemctl restart apache2
```

Pour faire le test, le mieux est d'utiliser le site b sécurisé au paragraphe II. 2. b) page 6.

Dans le répertoire /var/www-site-b renommer le fichier index.html en index.php.

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

Listing 7: Pour faire afficher toutes les variables du site b sécurisé

Dans le fichier index.php, juste avant la balise </body> ajouter le contenu de la figure 7 page 9.

2. Affichages des informations

a) Parcours rapide

Dans votre navigateur afficher la page du site b en https.

Vous devriez avoir beaucoup d'informations visibles sur :

- le module php7
- les variables d'environnement du serveur apache
- la configuration de php7
- les cookies en particulier la session
- la configuration du moteur ZEND (le moteur d'interprétation de php)
- les variables accessibles en php
- les extensions intégrées et utilisables avec php
- et beaucoup d'autres informations

b) Vérification des variables

Revenons sur les variables dans la partie « Apache Environment »

L'interprétation est facile à obtenir sur votre site à l'url /manual/fr/mod/mod_ssl.html

On va en vérifier quelques-unes, les générales :

- SERVER_ADDR : l'adresse du serveur web
- REMOTE_ADDR : l'adresse de votre client (navigateur)

- DOCUMENT_ROOT : la racine du site b
- HTTP_USER_AGENT : l'identité de votre navigateur
- HTTP_ACCEPT : ce qu'accepte de lire votre navigateur

Vous voyez que celui qui accède à ces informations sait tout sur votre site web, et sait donc comment l'attaquer.

Fichier *TP-1_Web-2223.odt* imprimé le 22 mars 2024