TP Service sur Réseaux N°6 - MMI 2

Introduction

Ouvrez VirtualBox.

Démarrez votre VM.

Une fois la VM démarrée, ouvrez le menu et lancez Konsole.

Authentification

Il est parfois nécessaire de restreindre l'accès à une partie d'un site. Les utilisateurs légitimes doivent alors fournir un identifiant et un mot de passe pour accéder à son contenu[1].

Exemple :

```
<Directory repertoire>
      Require valid-user
      AuthName "Repertoire prive"
      AuthType Basic
      AuthUserFile /etc/apache2/users
</Directory>
```

Le fichier `users` contiendra la liste des utilisateurs et leur mots de passe. On le manipule avec la commande `htpasswd`. Cet utilitaire fait partie du paquet `apache2-utils` qu'il est par conséquent nécessaire d'installer.

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

Pour ajouter un utilisateur ou changer un mot de passe, on exécutera la commande suivante :

```
sudo htpasswd -c /etc/apache2/users <utilisateur>
New password:
Re-type new password:
Adding password for user utilisateur
```

Attention

`'-c'` permet de créer le fichier. Si vous l'utilisez de nouveau alors que le fichier existe déjà, vous allez écraser son contenu.

Ce système d'authentification (Basic) a une sécurité très faible puisque les mots de passe circulent sans protection (ils sont uniquement codés en base64 ce qui est un simple encodage et non pas un procédé de chiffrement). Il faut noter que les documents protégés par ce mécanisme circulent également de manière non chiffrée. Si la sécurité vous importe, faites appel à SSL pour chiffrer toute la connexion HTTP.

Exercice 1

Créez un dossier private à la racine de votre domaine `www.labo.fr`. Activez l'option `Indexes`, le premier fichier qu'apache devra présenter à l'utilisateur sera `index.html`.

1 http://httpd.apache.org/docs/2.4/fr/howto/auth.html

Ajoutez à la racine de ce dossier le fichier `index.html` ci-dessous.

```
<html><body>
 <h1>Bienvenue dans la zone privée !</h1>
</body></html>
```

Mettez en place une authentification, avec 2 utilisateurs de définis.

Exercice 2

Ajoutez à la racine de votre domaine `www.labo.fr` le fichier `erreur.html` ci-dessous.

```
<html><body>
 <h1>Erreur authentification !</h1>
</body></html>
```

Si l'utilisateur n'est pas authentifié, le serveur devra afficher la page `erreur.html`.

Note	La directive **ErrorDocument** permet d'indiquer le document que le serveur renvoie au client en cas d'erreur. Liste des codes HTTP : http://fr.wikipedia.org/wiki/Liste_des_codes_HTTP

Exemple :

```
ErrorDocument 404 /notfound.html
```

Logs

Apache permet la journalisation (log) des erreurs et des accès. La journalisation des erreurs se configure avec les directives **ErrorLog** et **LogLevel**. Les réglages par défaut sont fixés dans le fichier `/etc/apache2/apache2.conf` :

```
grep ErrorLog /etc/apache2/apache2.conf
# ErrorLog: The location of the error log file.
# If you do not specify an ErrorLog directive within a <VirtualHost>
ErrorLog ${APACHE_LOG_DIR}/error.log

grep LogLevel /etc/apache2/apache2.conf
# LogLevel: Control the number of messages logged to the error_log.
# "LogLevel info ssl:warn"
LogLevel warn

grep "APACHE_LOG_DIR" /etc/apache2/apache2.conf export
APACHE_LOG_DIR=/var/log/apache2$SUFFIX
```

L'ensemble des erreurs seront donc journalisées dans le fichier : `/var/log/apache2/error.log`. Les niveaux disponibles par ordre de criticité décroissante pour la directive **LogLevel** sont : `emerg, alert, crit, error, warn, notice, info, debug`. Lorsqu'un niveau particulier est spécifié, les messages de tous les autres niveaux de criticité supérieure seront aussi enregistrés[2].

Apache fournit plusieurs directives pour personnaliser la journalisation des accès. Trois directives sont fournies : **TransferLog** pour créer un fichier journal, **LogFormat** pour définir un format personnalisé, et **CustomLog** pour définir un fichier journal et le format en une seule étape.

2 http ://httpd.apache.org/docs/2.4/fr/mod/core.html#loglevel

```
# grep LogFormat /etc/apache2/apache2.conf
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%
{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-
Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer LogFormat "%{User-agent}i"
agent

# grep CustomLog /etc/apache2/sites-available/000-default.conf
CustomLog ${APACHE_LOG_DIR}/access.log combined
```

Ici, vhost_combined, combined, common, et agent sont les noms donnés aux formats définis avec la directive **LogFormat** et utilisables avec la directive **CustomLog**[3].

Actuellement, le serveur Apache est configuré pour n'utiliser qu'un seul fichier de log pour tous les hôtes virtuels.

Exercice 3

Configurez votre serveur Apache pour qu'il journalise l'ensemble des messages d'informations sauf ceux de débogage.

Configurez vos deux hôtes virtuels pour qu'ils assurent une journalisation des accès dans des fichiers de log séparés : `/var/log/apache2/www-labo-fr.log` et `/var/log/apache2/test-labo-fr.log`.

Moteur de réécriture d'URL

Apache fournit un moteur de réécriture à base de règles permettant de réécrire les URLs des requêtes à la volée[4]. Il accepte un nombre illimité de règles, ainsi qu'un nombre illimité de conditions attachées à chaque règle, fournissant ainsi un mécanisme de manipulation d'URL vraiment souple et puissant. Les manipulations d'URL peuvent dépendre de nombreux tests, des variables du serveur, des variables d'environnement, des en-têtes HTTP ou de l'horodatage.

Il faut commencer par activer le module mod_rewrite et redémarrer le serveur Apache :

```
sudo a2enmod rewrite
sudo apache2ctl graceful
```

Vous trouverez de nombreux exemples d'utilisation courante (et moins courante) dans la documentation spécifique à la réécriture.

Exemple : supposons qu'on a récemment renommé la page default.html en index.html et que l'on désire que les accès à l'ancienne URL restent compatibles. Cependant, on veut que les utilisateurs de l'ancienne URL ne puissent pas reconnaître que les pages ont été renommées. Pour cela, on utilise les directives :

– `RewriteEngine` qui active ou désactive l'exécution du moteur de réécriture.

– `RewriteRule` qui définit les règles pour le moteur de réécriture en utilisant des expressions rationnelles compatible perl[5].

3 http://httpd.apache.org/docs/2.4/fr/mod/mod_log_config.html#formats

4 http://httpd.apache.org/docs/2.4/fr/mod/mod_rewrite.html

5 http://httpd.apache.org/docs/2.4/fr/rewrite/intro.html#regex

Voici l'implémentation correspondante :

```
<Directory repertoire>
      RewriteEngine On
      RewriteRule ^default.html$ index.html
</Directory>
```

Exercice 4

En utilisant le moteur de réécriture d'URL, assurez-vous que toutes les requêtes sur `http://test.labo.fr/private` soient redirigées vers `http://www.labo.fr/private`.

Exercice 5

Mettre en oeuvre une réécriture d'URL sur votre site `test.labo.fr` qui permet de rediriger toutes les requêtes de votre voisin de droite (utilisez son IP) vers `https://www.google.fr/?q=$1`

Exercice 6

Expliquer en détails la règle de réécriture ci-dessous. Donner un exemple d'utilisation de cette règle.

```
RewriteRule ^page-([0-9]+)\.html$ /page.php?id=$1 [L]
```

HTTP sécurisé

Apache 2.4 intègre en standard le module SSL nécessaire au support du HTTP sécurisé (HTTPS). Il faut juste l'activer avec `a2enmod ssl` puis placer les directives de configuration nécessaires dans la configuration[6].

SSL offre des fonctions fondamentales nécessaires à la communication sécurisé sur Internet et sur tout réseau TCP/IP :

– L'authentification SSL du serveur permet de garantir son identité à chacun des clients utilisant ses services. Cet authentification s'appuie en particulier sur des techniques de chiffrement à clé publique/clé privée. La confirmation de l'identité d'un serveur est très importante. Notamment, si vous devez envoyer votre numéro de carte de crédit sur le réseau pour réaliser un achat électronique, il faut que vous soyez certain de l'identité du site de commerce électronique destinataire.

– L'authentification SSL du client permet au serveur de valider l'identité du client. Cette authentification mutuelle est également très importante si le serveur Web de votre banque doit vous faire parvenir des informations confidentielles relatives à vos comptes bancaires.

– Une connexion SSL permet de chiffrer l'ensemble des données échangées entre un client et un serveur, ce qui apporte un haut niveau de confidentialité. La confidentialité est importante pour les deux parties dans la plupart des transactions privées. En complément de ce chiffrement, des mécanismes de vérification d'intégrité détectent automatiquement l'altération des données lors du transfert.

Le certificat est un ensemble d'informations utilisé par SSL pour réaliser l'authentification d'un service, d'une machine ou d'un utilisateur. Le certificat contient la clé publique de son détenteur et des informations sur son identité. Le certificat est signé électroniquement par une Autorité de Certification (CA) qui atteste son authenticité.

La vérification du certificat peut être effectuée par tout service qui possède la clé publique de l'autorité de certification.

6 http://httpd.apache.org/docs/2.4/fr/mod/mod_ssl.html

L'installation d'Apache a créé un hôte virtuel par défaut utilisant SSL. Voici les principales directives permettant sa mise en place :

```
grep SSL /etc/apache2/sites-available/default-ssl.conf
```

```
// # SSL Engine Switch:
// # Enable/Disable SSL for this virtual host.
SSLEngine on

// # SSLCertificateFile directive is needed.
SSLCertificateFile /etc/ssl/certs/ssl-cert-snakeoil.pem
SSLCertificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key
```

La directive **SSLEngine** permet d'activer l'utilisation du protocole SSL. La directive **SSLCertificateFile** indique le chemin du certificat (`/etc/ssl/certs/ssl-cert-snakeoil.pem`) et la directive **SSLCertificateKeyFile** indique le chemin de la clé privée (`/etc/ssl/private/ssl-cert-snakeoil.key`). Lors de l'installation d'Apache, le paquet `ssl-cer` a été auto-installé et un certificat a été créé (`/etc/ssl/certs/ssl-cert-snakeoil.pem`).

Ce certificat est auto-signé : voir le nom de l'AC signataire du certificat (voir le champ `Issuer`).

```
openssl x509 -in /etc/ssl/certs/ssl-cert-snakeoil.pem -text
```

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 14719432862616435902 (0xcc45ee339f76bcbe)
    Signature Algorithm: sha1WithRSAEncryption
        Issuer: CN=mint
        Validity
            Not Before: May 14 07:56:36 2014 GMT
            Not After : May 11 07:56:36 2024 GMT
        Subject: CN=mint
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
                Modulus:
00:c5:7f:05:b5:0e:f3:ab:9d:ca:3a:e4:74:2a:52:
39:d8:ed:5b:26:36:b0:43:63:15:94:72:62:5c:04:
f0:38:80:13:f7:98:16:9f:49:2a:c8:e5:cc:a8:6c:
06:c4:4d:ed:c7:98:8d:b2:46:d2:f8:2d:fd:8f:4b:
dc:4a:a8:f1:a6:c9:a2:a8:3d:71:8b:24:41:44:ed:
e5:df:ff:4f:94:71:1b:d9:ba:a4:fd:11:87:0a:5c:
51:37:7a:9f:f3:33:b4:38:8c:74:cc:0d:a3:49:ca:
74:cb:2f:9a:9d:1a:30:fe:f4:b8:b3:86:7c:11:57:
e7:d7:2e:99:22:e4:22:00:57:7a:cc:bd:2b:74:bf:
1d:e1:47:53:43:86:cc:39:ce:22:1c:22:ac:8b:e1:
8e:ac:6a:a6:8a:ee:5f:c6:e3:24:6f:36:2a:c0:35:
b7:2c:06:34:04:af:e9:0d:c3:fe:f8:df:b4:f2:f3:
75:58:31:23:2c:9b:21:3b:aa:9a:43:7e:7d:5f:ae:
ff:6c:3a:95:45:a0:89:06:d6:4e:93:20:35:14:3f:
be:f3:d4:c4:38:e9:ab:87:e7:20:72:33:b8:cf:f5:
69:e9:f3:72:f2:6f:2d:c9:48:5e:fb:fb:3e:a3:af:
fe:5e:9f:10:d2:53:24:f2:85:81:64:c8:88:70:f6:
                    f4:85
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
```

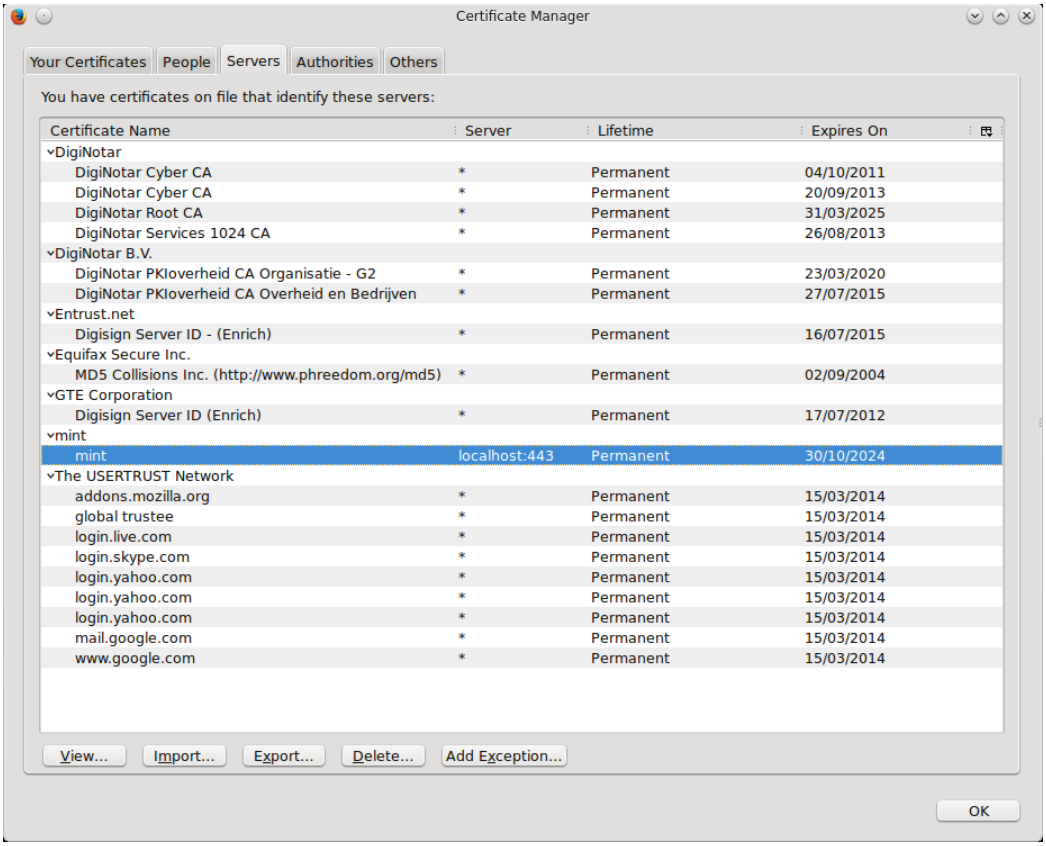
```
    Signature Algorithm: sha1WithRSAEncryption
         1c:4f:45:6f:81:e8:3b:e2:9a:96:df:48:9f:63:42:eb:90:10:
         51:07:fb:ba:2f:7e:e8:ca:0a:8f:46:83:ae:8f:79:0b:c7:14:
         24:84:0b:01:79:c6:f0:cd:de:c0:e3:66:6f:c6:3e:80:5b:dc:
         df:02:04:18:c6:9f:ec:81:1e:17:97:ed:d2:83:32:09:08:1e:
         aa:ac:36:f0:a7:00:1b:f0:b3:8a:de:fe:dc:4f:f1:72:9b:6d:
         6a:b9:e2:e7:cf:ed:84:6e:f6:35:ba:9d:5b:f7:30:16:d6:d5:
         fd:2c:be:2a:bb:c7:4b:76:dc:c6:f8:a8:10:2e:de:b9:42:27:
         34:48:1c:8e:8b:41:1a:b2:0a:a3:95:a5:48:3a:e4:17:11:cc:
         1e:da:3c:40:17:3d:ee:f8:70:84:f5:33:f1:f6:5d:27:09:9b:
         2d:17:87:2f:cc:5f:1f:ef:11:0d:2e:c0:80:60:c5:de:21:cc:
         64:c0:cd:77:3f:34:c2:d6:ab:49:f2:8a:12:10:cd:46:78:37:
         1a:a6:18:8e:8c:fa:05:8a:a7:22:04:94:6a:0d:27:cd:62:83:
         f9:54:05:93:cf:12:3b:1f:1e:5a:71:31:8c:d8:3d:cb:9d:9f:
         9c:a7:bd:20:7f:92:94:dd:4f:4d:76:7e:98:ed:8e:48:83:30:
         d1:23:7a:cb
-----BEGIN CERTIFICATE-----
MIICrjCCAZagAwIBAgIJAMxF7jOfdry+MA0GCSqGSIb3DQEBBQUAMA8xDTALBgNV
BAMTBG1pbnQwHhcNMTQwNTE0MDc1NjM2WhcNMjQwNTExMDc1NjM2WjAPMQ0wCwYD
VQQDEwRtaW50MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAxX8FtQ7z
q53KOuR0KlI52O1bJjawQ2MVlHJiXATwOIAT95gWn0kqyOXMqGwGxE3tx5iNskbS
+C39j0vcSqjxpsmiqD1xiyRBRO3l3/9PlHEb2bqk/RGHClxRN3qf8zO0OIx0zA2j
Scp0yy+anRow/vS4s4Z8EVfn1y6ZIuQiAFd6zL0rdL8d4UdTQ4bMOc4iHCKsi+GO
rGqmiu5fxuMkbzYqwDW3LAY0BK/pDcP++N+08vN1WDEjLJshO6qaQ359X67/bDqV
RaCJBtZOkyA1FD++89TEOOmrh+cgcjO4z/Vp6fNy8m8tyUhe+/s+o6/+Xp8Q0lMk
8oWBZMiIcPb0hQIDAQABow0wCzAJBgNVHRMEAjAAMA0GCSqGSIb3DQEBBQUAA4IB
AQAcT0Vvgeg74pqW30ifY0LrkBBRB/u6L37oygqPRoOuj3kLxxQkhAsBecbwzd7A
42Zvxj6AW9zfAgQYxp/sgR4Xl+3SgzIJCB6qrDbwpwAb8LOK3v7cT/Fym21queLn
z+2EbvY1up1b9zAW1tX9LL4qu8dLdtzG+KgQLt65Qic0SByOi0EasgqjlaVIOuQX
Ecwe2jxAFz3u+HCE9TPx9l0nCZstF4cvzF8f7xENLsCAYMXeIcxkwM13PzTC1qtJ
8ooSEM1GeDcaphiOjPoFiqciBJRqDSfNYoP5VAWTzxI7Hx5acTGM2D3LnZ+cp70g
f5KU3U9Ndn6Y7Y5IgzDRI3rL
-----END CERTIFICATE-----
```

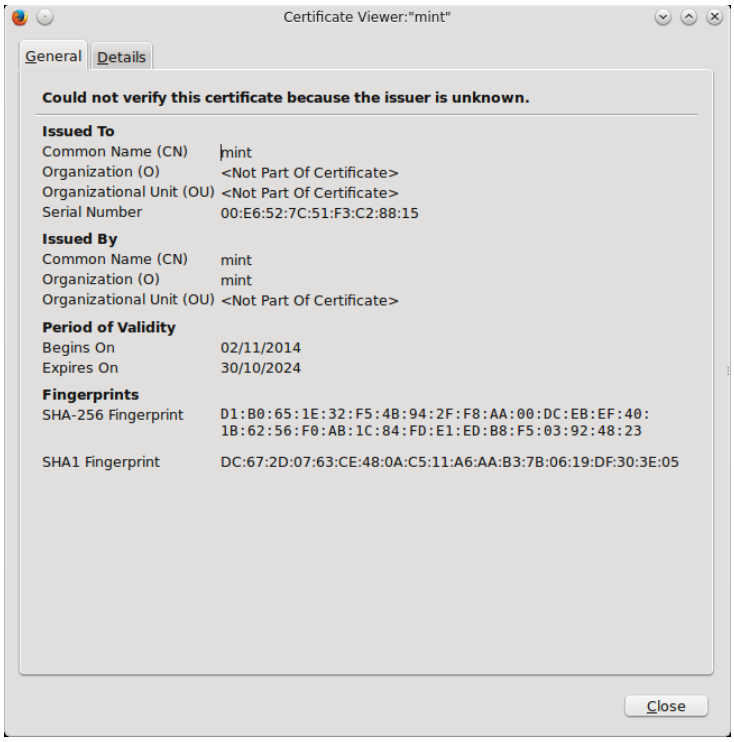
Il suffit maintenant d'activer l'hôte virtuel utilisant SSL, de recharger la configuration d'Apache et de tester l'accès HTTPS :

```
sudo a2enmod ssl
sudo a2ensite default-ssl
sudo apache2ctl graceful
```

Ouvrez la page `https://localhost` dans firefox.

À partir du navigateur firefox, il est possible d'afficher les certificats enregistrés :





Il est aussi possible de créer son certificat SSL auto signé :

// Création du certificat SSL auto signé :

```
sudo openssl req -x509 -nodes -days 365 -newkey rsa:1024 -out
/etc/apache2/server.crt -keyout /etc/apache2/server.key
```

```
Generating a 1024 bit RSA private key
.....................................................................
..............++++++
.................................++++++
writing new private key to '/etc/apache2/server.key'
-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished
Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:FR
State or Province Name (full name) [Some-State]:
Locality Name (eg, city) []:Vichy
Organization Name (eg, company) [Internet Widgits Pty
Ltd]:UNIVIC
Organizational Unit Name (eg, section) []:MMI
Common Name (e.g. server FQDN or YOUR name) []:www.labo.fr
Email Address []:contact@labo.fr
```

Exercice 8

Créez votre certificat auto-signé. Servez vous du modèle fourni par Apache (`/etc/apache2/sites-available/default-ssl.conf`) pour créer votre hôte virtuel SSL pour le site `https://www.labo.fr` défini dans un nouveau fichier : `/etc/apache2/sites-available/www-labo-fr-ssl.conf`.

Exercice 9

En utilisant la directive **`Redirect`** dans la configuration de l'hôte virtuel accessible sur le port 80, assurez-vous que les accès vers le site `http://www.labo.fr/` soit redirigés vers `https://www.labo.fr/`.