

TP2 Sécurité (SSL / HTTPS)

Référence

Pour rédiger ce TP, je me suis inspiré des informations fournies par JF Moreau sur le site <http://netsafe.free.fr/index.php>.

1. But

Le but de ce TP est mettre en place un serveur sécurisé HTTPS.

2. Outils

- 2.1. *Serveur Apache prêt pour faire du SSL*
- 2.2. *OpenSSL (Module permettant de faire faire du SSL)*

3. Préambule

Dans ce TP, nous n'allons pas utiliser *EasyPHP* pour la mise en place du HTTPS. En effet la version *EasyPHP* telle qu'elle est fournie ne permet pas de le faire de façon « native » car cette version n'est pas compilée pour les supports SSL.

Nous allons donc utiliser une distribution « light » de Apache 1.3 qui contient et les modules nécessaires.

Aussi plusieurs manipulations seront nécessaires. Je vous invite donc à les suivre étape par étape et de façon rigoureuse.

4. Préparation

4.1. *Votre adresse IP*

Prenez connaissance de votre adresse IP avec la commande *ipconfig* à partir d'une commande DOS.

4.2. *Fichier hosts*

Le certificat étant généré pour un nom de domaine (ou une adresse IP), vous devez modifier votre fichier *hosts* en conséquence.

- Editez le fichier *C:\WINDOWS\system32\drivers\etc\hosts* (WindowsXP)
- Rajoutez la ligne suivante à la fin de ce fichier

Votre_Adresse_IP www.mpe.com

- Sauvegardez ce fichier avec la nouvelle modification.

5. Installation d'Apache

- Téléchargez le fichier <http://www.waolany.com/mpe/ApacheSSL.zip>
- Dézipper ce fichier dans votre « home directory », dans le répertoire *ApacheSSL*.
- Créez un répertoire *www* dans *ApacheSSL*. Ce répertoire sera le « Document Root » de votre site web.
- Créez y un fichier *index.html* contenant juste la ligne « Bonjour les Master MPE »
- Allez dans le répertoire *ApacheSSL/conf* et éditez le fichier *httpd.conf*
- Remplacez la ligne `ServerName localhost` par `ServerName www.mpe.com`
- Remplacez la ligne `Port 80` par `Port 443`
- Remplacez la ligne `Listen 12.34.56.78:80` par `Listen 443`
- Remplacer la ligne `DocumentRoot "c:/apache/htdocs"` par `DocumentRoot "<lecteur_disque>:/ApacheSSL/www"`

6. Configuration du module de support PHP

La version Apache que vous venez d'installer ne supporte pas le module PHP. Vous devez alors le configurer vous-mêmes. Pour cela :

- Copiez le répertoire *php* de votre EsyPHP dans *ApacheSSL*
- Editez le fichier *httpd.conf* de *ApacheSSL*
- Rajoutez la ligne suivante à la fin de la section *LoadModule*.

```
LoadModule php4_module "php/php4apache.dll"
```

- Rajoutez la ligne suivante à la fin de la section *AddModule*.

```
AddModule mod_php4.c
```

- Ajoutez la ligne suivante au dessus `<IfModule mod_mime.c>`.

```
AddType application/x-httpd-php .phtml .phtml .phtml .php3 .php4 .php .php2 .inc
```

7. Test

Ouvrez une console DOS et allez dans le répertoire où vous avez installé la nouvelle version d'Apache (*ApacheSSL*).

- Exécutez la commande suivante : `apache start`.
- Essayez l'URL <http://www.mpe.com>. Expliquez pourquoi ça ne marche pas.
- Essayez maintenant l'URL <http://www.mpe.com:443>. Et là, bingo ça marche.

A ce stade, les données transmises ne seront pas encore chiffrées (cryptées), mais si cela fonctionne, c'est que la configuration du port SSL est correcte.

8. Installation des outils permettant la génération des certificats

- Téléchargez le fichier <http://www.waolany.com/mpe/OpenSSL.zip>
- Dézippez ce fichier à la racine du répertoire Apache (*ApacheSSL*)
- Copiez les fichiers *ssleay32.dll* et *libeay32.dll* du répertoire *OpenSSL* dans le répertoire *System32* de windows

9. Création d'un certificat temporaire

Important :

Le certificat que nous allons générer ici n'a aucune valeur auprès des autorités compétentes. Il a juste pour but de vous montrer comment cela se fait.

Allez dans le répertoire *OpenSSL*, créez un fichier *ssl.bat* contenant les lignes suivantes

```
openssl req -config openssl.cnf -new -out mon-server.csr
openssl rsa -in privkey.pem -out mon-server.key
openssl x509 -in mon-server.csr -out mon-server.cert -req -signkey mon-server.key -days 365
openssl x509 -in mon-server.cert -out mon-server.der.crt -outform DER
```

ATTENTION:

Remplacez toutes les occurrences de `mon-server` par www.mpe.com

9.1. Descriptifs des commandes utilisées dans le fichier *ssl.bat*

- `openssl req -config openssl.cnf -new -out mon-serveur.csr`

Cela crée un CSR (Certificat Signing Request) et une clef privée. Lorsque l'on vous demande votre "nom de domaine", donnez le nom de domaine exact de votre serveur Web (par exemple **www.mon-serveur.com**, **195.125.204.24**). Le certificat appartient à ce nom de serveur et les navigateurs se "plaignent" si le nom ne correspond pas.

- `openssl rsa -in privkey.pem -out mon-serveur.key`

Cela enlève la phrase/motdepasse de la clef privée. Vous DEVEZ comprendre ce que cela signifie; `mon-serveur.key` doit être lisible seulement par le serveur apache et l'administrateur. Vous devez supprimer le fichier `.rnd` parce qu'il contient l'information d'entropie pour créer la clef et pourrait être employé pour des attaques cryptographiques contre votre clef privée.

- `openssl x509 -in mon-serveur.csr -out mon-serveur.cert -req -signkey mon-serveur.key -days 365`

Cela crée un certificat signé que vous pouvez employer avant que vous n'en obteniez un "réel" d'une autorité de certification. (Le certificat réel est facultatif; si vous connaissez vos utilisateurs, vous pouvez leur dire d'installer le certificat dans leur(s) navigateur(s).) Notez que ce certificat expire après un an, vous pouvez modifier le délai (`-days 365`) si vous le désirez.

- `openssl x509 -in mon-serveur.cert -out mon-serveur.der.crt -outform DER`

Permet aux utilisateurs naviguant avec MS Internet Explorer 4.x d'installer le certificat dans leur navigateur (en le téléchargeant et en l'ouvrant).

9.2. Génération du certificat

- Ouvrez une console DOS et allez dans le répertoire *OpenSSL*
- Exécutez la commande *ssl.bat* et répondez aux différentes questions posées
- Si tout c'est bien passé, vous devez avoir les 4 fichiers suivants: www.mpe.com.key, www.mpe.com.der.crt, www.mpe.com.cert, et www.mpe.com.csr dans le répertoire *OpenSSL*. Sinon, il y a eu une erreur.
- Créez un répertoire *ssl* dans le répertoire *conf* de *ApacheSSL*
- Copiez les fichiers www.mpe.com.cert et www.mpe.com.key dans le répertoire *ssl* crée ci-dessus.

10. Configuration de Apache pour supporter du SSL

- Arrêtez apache en faisant ^C (Control C) à partir de la console DOS.
- Editez le fichier `httpd.conf` se trouvant dans le répertoire `conf` de `ApacheSSL`.
- Rajoutez la ligne suivante à la fin de la section `LoadModule`.

```
LoadModule ssl_module modules/mod_ssl.so
```

- Rajoutez la ligne suivante à la fin de la section `AddModule`.

```
AddModule mod_ssl.c
```

- Rajoutez les lignes suivantes à la fin du fichier `httpd.conf`.

```
##### Gestion SSL
SSLMutex sem
SSLRandomSeed startup builtin
SSLSessionCache none
SSLLog logs/SSL.log
SSLLogLevel info

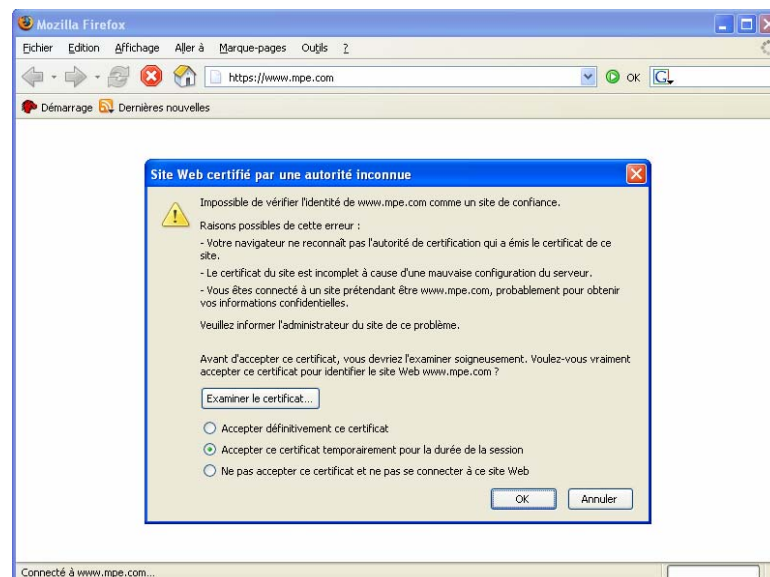
<VirtualHost mon-server:443>
    SSLEngine On
    SSLCertificateFile conf/ssl/mon-server.cert
    SSLCertificateKeyFile conf/ssl/mon-server.key
    DocumentRoot <Chemin_install_apache>/www
</VirtualHost>
```

N'oubliez pas de :

- Remplacez toutes les occurrences de `mon-server` par www.mpe.com
- Remplacer `<Chemin_install_apache>` par celui du répertoire `ApacheSSL`.
- Repérez la ligne `#NameVirtualHost *:80` et remplacez la par `NameVirtualHost *`
- Sauvegardez ce fichier
- Redémarrez Apache via la console DOS avec la commande `apache start`

11. Test

- Testez l'URL <https://www.mpe.com>. Si tout est OK, vous obtiendrez :



- Copiez tous vos projets PHP dans le répertoire `www` de `ApacheSSL`. Vous veillerez à prendre les répertoires si vous les avez structurés par nom de répertoire (`TPPHP1`, ... `TPPHPN`)

Vous pouvez désormais accéder à vos sites et projets PHP en HTTPS. « Plus personne ne pourra alors intercepter vos données car elles seront désormais cryptées ».

C'est pas beau tout ça ?