Sécurisation SSL d'un Serveur Web

Mise en place de SSL avec certificats auto-signés sur un serveur Apache2

ANATOLE BILLET

09 novembre 2015
Version 1.00

Sécurisation SSL d'un Serveur Web

Mise en place de SSL avec certificats auto-signés sur un serveur Apache2

Avant de commencer…

Objectif :

L'objectif principal du TP est la création d'une autorité de certification et création de certificats SSL. Le certificat SSL sera ensuite déployé sur un serveur Web Apache.

Prérequis :

-Debian 8.2

Code couleur :

-Bleu pour les commandes Debian

-Vert pour les noms des fichiers de configurations

-Italic pour les descriptions et anecdotes.

Table des matières

[Rappel] Installation d'apache2

Pour installer apache2 on utilise la commande :

Apt-get install apache2

On peut lui ajouter php5 et MySQL selon la configuration de notre serveur Web (recommandé) :

Apt-get install apache2 php5 mysql-server

On teste le fonctionnement d'apache2 en entrant sont IP dans un navigateur sur le même réseau.

1. Installation de openSSL

Pour installer openSSL : **apt-get install openSSL** mais il également inclut dans le paquet d'apache2 que nous utiliserons pour ce tutoriel.

2. Générer le certificat

Dans le dossier **/etc/ssl** ont créé un dossier pour notre domaine :

domaine=gsb.local

cd /etc/ssl

mkdir $domaine

cd $domaine

On crée la clé privée avec l'algorithme RSA 2048 bits.

openssl genrsa -out $domaine.key 2048

Ensuite il faut générer un fichier de « demande de signature de certificat », en anglais CSR

openssl req -new -key $domaine.key -out $domaine.csr

On répond à un certain nombre de questions.
Il faut bien mettre le nom (ou l'ip) du serveur tel qu'il est appelé de l'extérieur dans le champ « Common Name » (CN).

Ensuite, on génère le certificat signé au format x509 (ici pour 365jours auto-signé):

openssl x509 -req -days 365 -in $domaine.csr -signkey $domaine.key -out $domaine.crt

Ce certificat n'est authentifié par aucune autorité, vous aurez donc un message d'avertissement quand vous vous connectez au serveur.
C'est le fichier **gsb.local.crt** qu'on ajoute au besoin dans les navigateurs internet pour ne pas accepter le certificat à chaque fois.

Ou au lieu d'auto-signer le certificat on peut envoyer le fichier CSR à une autorité de certification reconnue.

3. Éditer le "virtualhost" SSL dans apache

Il nous faut ensuite éditer le fichier suivant pour que apache utilise les certificats :

/etc/apache2/sites-available/default-ssl

Éditer les lignes suivantes comme cela :

ServerName gsb.local

SSLCertificateFile /etc/ssl/gsb.local/gsb.local.crt

SSLCertificateKeyFile /etc/ssl/gsb.local/gsb.local.key

Suite aux dernières vulnérabilités découvertes au sein du protocole SSL en 2015, il est recommandé également d'effectuer la configuration suivante dans Apache pour plus de sécurité :

SSLProtocol -ALL +TLSv1 +TLSv1.1 +TLSv1.2

SSLHonorCipherOrder On

SSLCipherSuite ECDHE-RSA-AES128-SHA256:AES128-GCM-SHA256:HIGH:!MD5:!aNULL:!EDH:!RC4

SSLCompression off

Enregistrez puis quittez le fichier de configuration du site SSL. Ensuite, activez le module SSL et le site SSL :

a2enmod ssl

a2ensite default-ssl

service apache2 reload

Accédez à votre site en utilisant le préfixe HTTPS dans l'URL, cela devrait fonctionner.

4. Désactiver le site HTTP

Si vous souhaitez qu'on accède à votre site web uniquement via le protocole HTTPS, il est intéressant de désactiver le site accessible sur le port 80 c'est-à-

dire le site « **default** ». Pour cela on utilise la commande « **a2dissite** » qui permet de désactiver des sites dans Apache 2.

a2dissite default

Vous pouvez ensuite essayer d'accéder à votre site en HTTP et vous verrez qu'il n'est plus accessible.

5. Rediriger le HTTP vers HTTPS automatiquement

Plutôt que de désactiver le site **HTTP**, on peut le laisser activer sauf qu'on va le configurer de façon à rediriger de manière permanente les requêtes **HTTP** vers **HTTPS** autrement dit les requêtes sur le port 80 vers le port 443.

Pour cela, modifiez le fichier suivant :

/etc/apache2/sites-available/default

Dans le virtualhost, ajoutez la ligne suivante :

Redirect permanent / https://[IP ou Domaine]

Adaptez la ligne ci-dessus avec votre nom de domaine. Ensuite, il ne vous reste plus qu'à recharger la configuration d'Apache puis de tester la redirection :

service apache2 reload