Université Abdelmalek Essaadi
Département Génie Informatique
Master SecuIT & BigData

Module : Crypto & Sécurité Services
Année Universitaire 2023-2024
Pr. A. GHADI

Certificat SSL auto-signé pour la redirection http vers https

TLS, également connu sous le nom de "transport layer security", ainsi que son prédécesseur SSL, sont des protocoles établis dans le domaine des technologies de sécurité informatique. Ils sont conçus pour fournir un canal de communication sécurisé en enveloppant le trafic réseau dans une couche de protection et de chiffrement. Cette approche permet aux serveurs d'échanger des informations avec leurs clients de manière confidentielle, réduisant ainsi les risques d'interception ou de lecture par des tiers non autorisés.
Dans le cadre de ce guide, nous examinerons les étapes nécessaires à la création et à l'utilisation d'un certificat SSL auto-signé avec le serveur web Apache, spécifiquement dans le contexte du système d'exploitation Debian 12.

Un certificat auto-signé est un certificat SSL/TLS où l'entité qui génère le certificat est également celle qui le signe. Contrairement aux certificats émis par une autorité de certification (CA) reconnue, un certificat auto-signé n'a pas été validé par une tierce partie de confiance. Cela signifie qu'il peut ne pas être aussi fiable dans certains contextes, car les navigateurs et les applications peuvent ne pas le reconnaître comme étant émanant d'une source de confiance.

Dans un contexte de développement ou de test, un certificat auto-signé peut être utilisé pour fournir une couche de chiffrement et de sécurité sans les coûts associés à l'acquisition d'un certificat valide émis par une CA. Cependant, pour les déploiements en production ou lorsque la confiance des utilisateurs est cruciale, il est recommandé d'utiliser des certificats émis par des CAs reconnues.

1. Installation de apache2 et activation du module ssl

Installer Apache en utilisant `apt`. Tout d'abord, mettez à jour l'index local des paquets afin de refléter les derniers changements en amont :

$ sudo apt update
$ sudo apt install apache2

2. Création de certificat SSL auto signé

À présent que le serveur Apache a été configuré pour utiliser le chiffrement, nous sommes en mesure de procéder à la génération d'un nouveau certificat SSL. Ce certificat contiendra des informations essentielles relatives à votre site, et sera associé à un fichier clé qui autorisera le serveur à manipuler les données cryptées de manière sécurisée.

Les fichiers clés et certificats SSL peuvent être créés à l'aide de la commande OpenSSL :

$ sudo openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout /etc/ssl/private/apache-selfsigned.key -out /etc/ssl/certs/apache-selfsigned.crt

Une fois la commande saisie, vous serez dirigé vers une invite où vous pourrez fournir des informations relatives à votre site web. La ligne la plus importante est celle qui demande le `Common Name`. Vous devez entrer soit le nom d'hôte que vous utiliserez pour accéder au serveur, soit l'adresse IP publique du serveur. Il est important que ce champ corresponde à ce que vous allez mettre dans la barre d'adresse de votre navigateur pour accéder au site, car une mauvaise correspondance entraînera davantage d'erreurs de sécurité. Exemple :

```
Country Name (2 letter code) [XX]:MA
State or Province Name (full name) []: Tanger-Asila
Locality Name (eg, city) [Default City]:Tanger
Organization Name (eg, company) [Default Company Ltd]:FSTT
Organizational Unit Name (eg, section) []: MasterSecuIT-BigData
Common Name (eg, your name or your server's hostname) []: 192.168.11.111
Email Address []: votre adresse email
```

- `req -x509` : Cette spécification indique notre intention d'utiliser le processus de gestion des requêtes de signature de certificat (CSR) conformément à la norme X.509. Cette dernière représente une infrastructure de clé publique standard, à laquelle SSL et TLS adhèrent pour la gestion des clés et des certificats.
- `-nodes` : Cette option directive demande à OpenSSL de ne pas sécuriser le certificat par l'utilisation d'une phrase de passe. Cela est nécessaire pour permettre à Apache de lire le fichier sans intervention de l'utilisateur lors du démarrage du serveur. L'ajout d'une phrase de passe empêcherait cette opération, car elle nécessiterait une saisie à chaque redémarrage.
- `-days 365` : Fixe la durée de validité du certificat à un an.
- `-newkey rsa:2048` : Spécifie la création d'une clé RSA de 2048 bits.
- `-keyout` : Emplacement où mettre le fichier de clé privée nouvellement généré.
- `-out` : Emplacement où mettre le certificat nouvellement créé.
-

3. Configuration d'Apache pour utiliser SSL

Désormais équipés de notre certificat et de notre clé auto-signés, nous devons procéder à la mise à jour de notre configuration Apache afin de les intégrer. Les nouveaux fichiers de configuration Apache (qui doivent être dotés de l'extension .conf) peuvent être placés dans le répertoire /etc/apache2/sites-available/. Ils seront chargés lors du prochain rechargement ou redémarrage du service Apache."

$ sudo nano /etc/apache2/sites-available/secuit.conf

```
<VirtualHost *:443>
  ServerName 192.168.11.111
  DocumentRoot /var/www/secuit

  SSLEngine on
  SSLCertificateFile /etc/ssl/certs/apache-selfsigned.crt
  SSLCertificateKeyFile /etc/ssl/private/apache-selfsigned.key
</VirtualHost>
```

Assurez-vous de mettre à jour la ligne ServerName en fonction de la façon dont vous prévoyez de nommer votre serveur. Cela peut inclure un nom d'hôte, un nom de domaine complet ou une adresse IP. Veillez à ce que votre choix corresponde au Common Name (nom commun) que vous avez spécifié lors de la création du certificat."

Nous allons maintenant créer notre `DocumentRoot` et y insérer un fichier HTML à des fins de test :

$ sudo mkdir /var/www/secuit
$ sudo echo "<h1> Site de SecuIT </h1> " > /var/www/secuit/index.html

Activer le fichier de configuration (le site) avec l'outil `a2ensite` :

```
$ sudo a2ensite secuit.conf
```

Ensuite, effectuons un test à la recherche d'éventuelles erreurs de configuration :

$ sudo apache2ctl configtest
Si tout fonctionne correctement, vous obtiendrez un résultat qui ressemble à ceci :

```
AH00558: apache2: Could not reliably determine the server's fully qualified domain name, using 127.0.1.1.
Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

$ sudo systemctl reload apache2

4. Test de fonctionnement

"Maintenant, chargez votre site dans un navigateur en utilisant https://192.168.11.111 au début.

Vous devriez rencontrer une erreur, ce qui est normal pour un certificat auto-signé ! Le navigateur vous avertira qu'il ne peut pas vérifier l'identité du serveur, car notre certificat n'est pas signé par une autorité de certification connue. Pour des tests ou une utilisation personnelle, cela peut être parfaitement acceptable. Vous devriez pouvoir accéder à des informations avancées ou complémentaires en cliquant pour choisir de poursuivre.

Une fois que vous aurez effectué cette action, votre navigateur chargera votre site en https.

5. Redirection de http vers https

Actuellement, notre configuration ne traite que les requêtes HTTPS sur le port 443. Il est recommandé de répondre également aux requêtes sur le port 80, même si l'intention est de forcer le chiffrement de tout le trafic. Pour ce faire, mettons en place un VirtualHost pour répondre à ces requêtes non sécurisées et les rediriger vers HTTPS.

Dans votre fichier /etc/apache2/sites-available/secuit.conf, ajouter à la fin :

```
<VirtualHost *:80>
        ServerName 192.168.11.111
        Redirect /  https://192.168.11.111/
</VirtualHost>
```

Enregistrez et fermez ce fichier, puis testez à nouveau la syntaxe de votre configuration, et rechargez Apache :
$ sudo apachectl configtest
$ sudo systemctl reload apache2

Vous pouvez tester la nouvelle fonctionnalité de redirection en visitant votre site avec le simple `http://192.168.11.11`. Vous devriez être redirigé automatiquement vers `https://192.168.11.111`

Si vous envisagez d'utiliser SSL pour un site web public, vous devriez envisager d'acheter un nom de domaine et d'utiliser une autorité de certification largement reconnue telle que Let's Encrypt.