DETROZ Sébastien Version 1 24/11/2016

Sommaire :

Introduction

Objectif : L'objectif de ce TP est de créer sa propre autorité de certification afin de créer des certificats pour sécuriser les communications entre les services et les clients.

Pré-requis : Il faut des connaissances en sécurité TLS/SSL et en service web.

Norme : Toutes les commandes issues d'une machine avec un système d'exploitation Debian ou Windows sont écrites ***en gras et en italique***.

1 – Le fichier openssl.cnf :

Il faut organiser les différents certificats de façon à ce que les clients et serveurs puissent les trouver et les tester. Il est possible de profiter de l'organisation proposée dans le fichier de configuration d'openssl (fichier openssl.cnf). Celui-ci utilise l'organisation suivante :

```
sebastien@debianseb:~$ tree
.
└── tpssl
    ├── certs
    ├── crl
    ├── index.txt
    ├── newcerts
    ├── openssl.cnf
    ├── private
    └── serial

5 directories, 3 files
```

Le répertoire tpssl devra vous appartenir.
Le certificat racine est placé directement dans tpssl.
La clé privée du certificat racine est dans tpss/private.
Le fichier index.txt ne doit rien contenir, tandis que le fichier serial doit contenir « 01 ».

Il faut modifier la variable dir dans le fichier openssl.cnf, qu'il faut préalablement copier depuis /etc/ssl :

```
####################################################################
[ CA_default ]

dir             = /home/sebastien/tpssl         # Where everything is kept
certs           = $dir/certs            # Where the issued certs are kept
crl_dir         = $dir/crl              # Where the issued crl are kept
database        = $dir/index.txt        # database index file.
#unique_subject = no                    # Set to 'no' to allow creation of
```

2 – Création des certificats :

2.1 – Création du certificat de l'autorité de certification :

Cette étape consiste à créer la paire de clés privée/publique puis un certificat racine auto-signé (signifie qu'une signature numérique a été ajoutée. Cette signature a été créée à partir du certificat lui-même). A l'issue de cette étape, nous aurons :

- Une clé privée protégée par un mot de passe (***cakey.pem***).
- Une demande de certificat numérique valable 3650 jours (***cacert.pem***).

Les renseignements du tableau devront être fournis impérativement avec la commande :

openssl req –new –x509 –extensions v3_ca –keyout private/cakey.pem –out cacert.pem –days 3650 –config ./openssl.cnf

Vérifiez la présence des deux fichiers ***cakey.pem*** et ***cacert.pem***. Observez l'en-tête du fichier ***cakey.pem***. La clé privée est protégée avec une variante de l'algorithme 3DES. Le mot de passe saisi sera indispensable pour lire la clé.

2.2 – Extraction du certificat racine :

L'extraction consiste à afficher une sortie écran d'un certificat. On peut alors vérifier que le certificat est conforme aux attentes, avec la commande ***openssl x509 –text –in cacert.pem*** :

Pour sauvegarder vos fichiers, procédez à leur archivage :

```
sebastien@debianseb:~/tpssl$ tar -czf rootca.tar.gz private/cakey.pem cacert.pem

sebastien@debianseb:~/tpssl$ ls
cacert.pem  crl         newcerts     private         serial
certs       index.txt   openssl.cnf  rootca.tar.gz
```

3 – Création d'un certificat SSL pour un serveur web :

Un certificat SSL peut être utile afin de sécuriser les échanges entre un serveur Web et des clients potentiels. Ce certificat, installé sur un serveur Web, est transmis au client lorsqu'un échange sécurisé est demandé. Le certificat SSL, associé à une paire de clés publique/privée, permet au server d'échanger des données cryptées avec le navigateur du client.

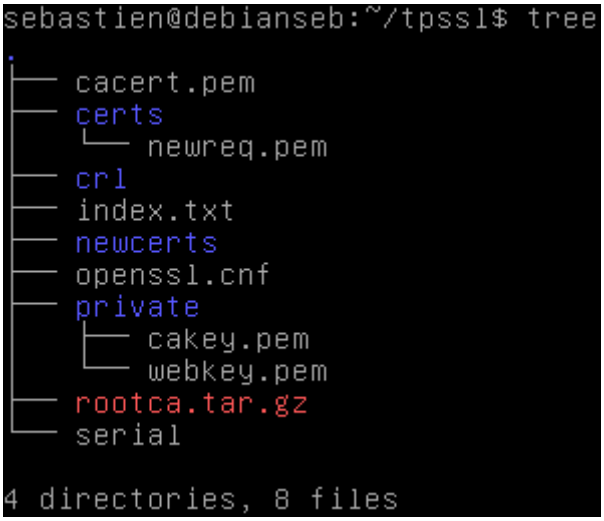
Il faut donc :

- Créer une nouvelle paire de clé publique/privée (***webkey.pem***).
- Créer une nouvelle demande de certificat pour le serveur qui contiendra la clé publique (***newreq.pem***).
- Signer cette demande de certificat avec le certificat de l'autorité (***cacert.pem***) et obtenir un nouveau certificat (***webcert.pem***).

3.1 – Création de la paire de clé et de la demande de certificat :

Les renseignements du tableau devront être fournis impérativement avec la commande :

openssl req –config ./openssl.cnf –new –keyout private/webkey.pem –out certs/newreq.pem



3.2 – Signature de la demande de certificat par l'autorité :

Il faut maintenant signer ce certificat afin qu'il puisse être déployé sur le serveur Web. Pour cela, la clé privée de l'autorité de certification sera nécessaire puisqu'elle est la seule à pouvoir créer la signature numérique, avec la commande :

openssl ca –config ./openssl.cnf –policy policy_anything –out certs/webcert.pem –infiles certs/newreq.pem

```
Using configuration from ./openssl.cnf
Enter pass phrase for /home/sebastien/tpssl/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Nov 15 08:19:37 2016 GMT
            Not After : Nov 15 08:19:37 2017 GMT
        Subject:
            countryName               = FR
            stateOrProvinceName       = 14
            localityName              = Caen
            organizationName          = Techrom
            organizationalUnitName    = Service r\C3\A9seau
            commonName                = techrom.fr
            emailAddress              = sebastien.detroz@sts-sio-caen.info
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            Netscape Comment:
                OpenSSL Generated Certificate
            X509v3 Subject Key Identifier:
                92:C5:76:28:4B:51:DC:EE:DC:6C:38:3E:2B:9C:91:74:E0:BC:1B:E3
```

3.3 – Vérification du chemin de certification :

L'objectif est de vérifier que la signature du certificat a bien été effectuée par notre autorité de certification. Cela prouve que le chemin de certification est correct. Pour cela, on utilise la commande « ***verify*** » d'openssl :

openssl verify –CAfile cacert.pem certs/webcert.pem

```
sebastien@debianseb:~/tpssl$ openssl verify -CAfile cacert.pem certs/webcert.pem

certs/webcert.pem: OK
```

4 – Installation du certificat SSL :

4.1 Export des certificats et de la clé privée :

Les éléments nécessaires à Apache2 pour prendre en charge SSL sont les suivants :

- Le certificat du serveur (***webcert.pem***).
- La clé privée non cryptée du serveur.

Le fait d'accéder à la clé privée du serveur pose un sérieux problème de sécurité. En effet, si quelqu'un s'empare de cette clé, il pourra décrypter tous les échanges entre le serveur et ses clients. Il est possible de maintenant un cryptage de la clé privée grâce à un mot de passe. Dans ce cas, dès que le serveur Apache2 démarre, il demandera le mot de passe. Dans ce TP, nous allons laisser la clé privée non-cryptée.

Décryptage de la clé privée du serveur web

La commande suivante permet de générer un nouveau fichier contenant la clé privée non cryptée (***webkey-clair.pem***) :

openssl rsa –in private/webkey.pem –out private/webkey-clair.pem

```
sebastien@debianseb:~/tpssl$ openssl rsa -in private/webkey.pem -out private/web
key-clair.pem
Enter pass phrase for private/webkey.pem:
writing RSA key
```

Copie des fichiers dans le répertoire d'Apache2

Copier les fichiers ***webcert.pem***, ***webkey-clair.pem*** dans le répertoire SSL d'Apache (à créer si nécessaire : ***/etc/apache2/ssl***) :

```
root@debianseb:/etc/apache2/ssl# ls -l
total 12
-rw-r--r-- 1 root root 4771 nov.  15 09:41 webcert.pem
-rw-r--r-- 1 root root 1679 nov.  15 09:40 webkey-clair.pem
```

4.2 – Configuration d'Apache :

Il faut d'abord configurer le serveur Web pour qu'il utilise SSL. Le module doit donc être activé. Il s'agit de créer un lien symbolique entre les 2 répertoires suivants :

- /etc/apache2/mod-available/
- /etc/apache2/mod-enabled/

Avec ***a2enmod ssl***.

Nous devons créer un hôte virtuel (VirtualHost) pour qu'Apache soit capable de répondre aux requêtes SSL (https) :

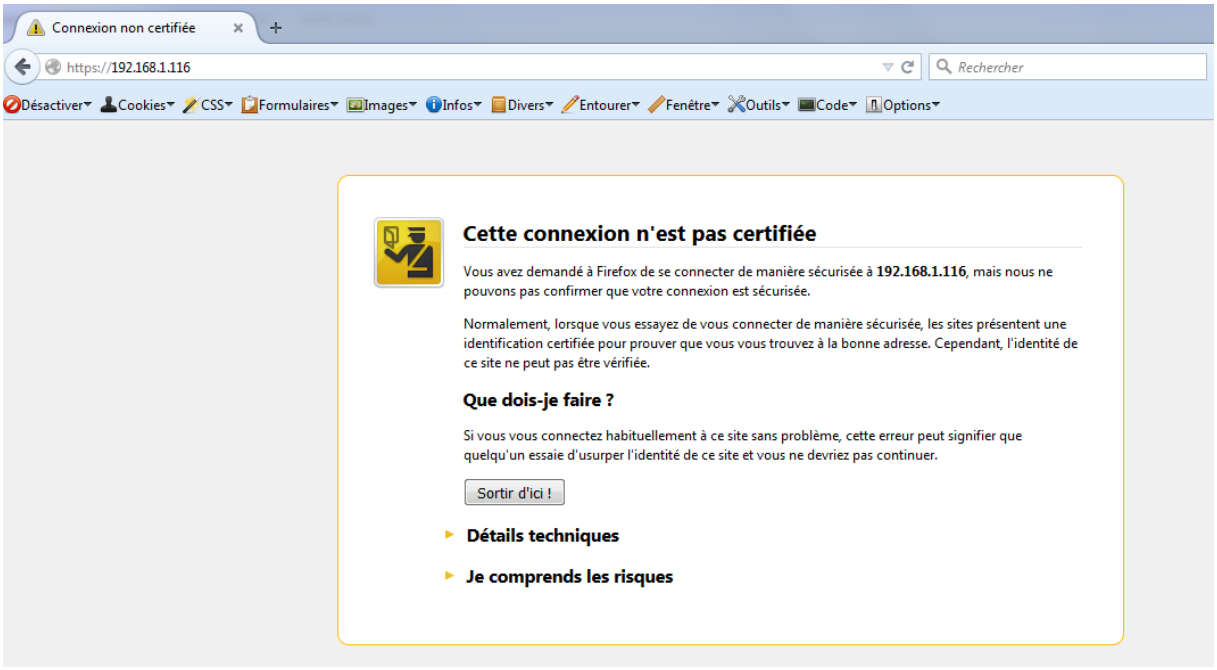
- Editer le fichier /etc/apache2/sites-available/default-ssl et modifier le chemin des certificats.
- Activer cet hôte virtuel puis redémarrer Apache2.

```
#   A self-signed (snakeoil) certificate can be created by inst$
#   the ssl-cert package. See
#   /usr/share/doc/apache2/README.Debian.gz for more info.
#   If both key and certificate are stored in the same file, on$
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/apache2/ssl/webcert.pem
SSLCertificateKeyFile /etc/apache2/ssl/webkey-clair.pem
```

Avec ***a2ensite default-ssl***.

```
root@debianseb:/etc/apache2/sites-available# a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@debianseb:/etc/apache2/sites-available# service apache2 reload
root@debianseb:/etc/apache2/sites-available# _
```

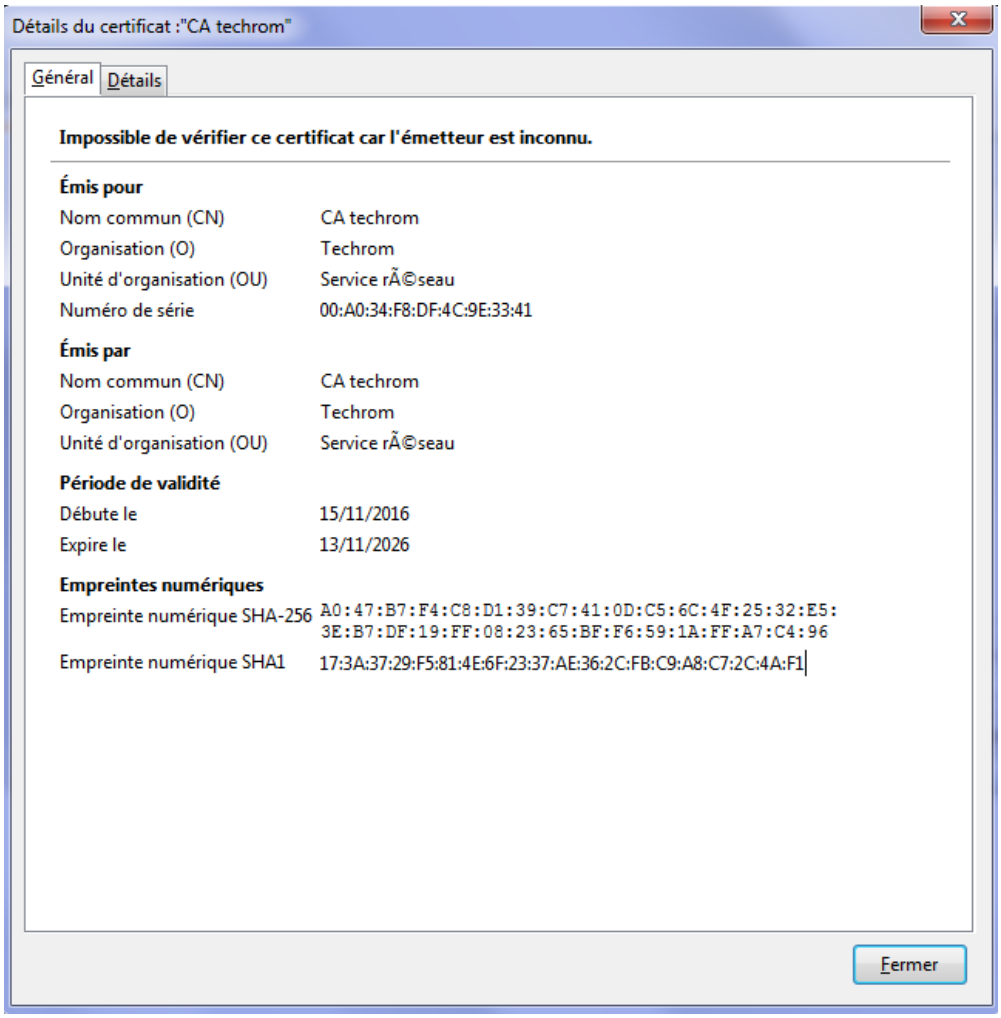
Lancer le navigateur Firefox avec l'url https://localhost ou https://@IP. Un message apparaît, **n'acceptez pas ce certificat** qui n'a pas été vérifié par une autorité de certification de confiance.



4.3 – Ajout de notre autorité de certification dans le navigateur Firefox :

Afin d'éviter le message d'acceptation du certificat, il est possible de configurer le navigateur pour qu'il accepte tous les certificats venant de notre autorité de certification. Pour cela, il faut absolument copier le certificat racine (***cacert.pem***) sur le poste du client et l'importer dans la configuration du navigateur.

/options/options/avancé [chiffrement] – afficher les certificats – importer



Il reste un problème de résolution de noms DNS. En effet, la valeur du champ « Common Name » du certificat crée précédemment est « techrom.fr ». Si vous n'accédez pas au serveur web avec une URL basée sur le même nom, la plupart des navigateurs affichent un message d'avertissement. Ce problème pourra être résolu lorsque le nom de domaine du serveur sera configuré correctement.

4.4 – Résolution du problème DNS :

Afin de résoudre ce problème sans pour autant modifier le système de résolution DNS, nous allons installer une résolution statique DNS par l'intermédiaire du fichier /etc/hosts sur Linux et C://Windows/System32/drivers/etc/hosts sur Windows. Ajouter la ligne suivante :

192.168.1.116 techrom.fr

Lancer de nouveau votre navigateur favori Firefox et rendez-vous sur la page d'accueil du serveur Web en utilisant le protocole HTTPS :

- Réaliser une capture de trame à l'aide du sniffeur Wireshark.
- Que constatez-vous ?
- Conclure.