ETTORI Bastien	BTS SIO 2ème année
15 Novembre 2016	Année scolaire : 2016/2017
Option : SISR	Version 1

SERVEUR SSL DEBIAN

SOMMAIRE :

I) Objectif

Dans cette procédure, nous allons voir comment installer et configurer un serveur **SSL** sous Debian.

II) Prérequis

Pour réaliser cette procédure, nous avons besoin des éléments suivants :

OS	Distribution	Version	C/S
Debian	Linux	8.5	**S**

Nom du serveur SSL	IP du serveur SSL	IP du client Windows
SSL	192.168.1.132	192.168.1.74

III) Définition

SSL (**S**ecure **S**ocket **L**ayer) est un service de boîte à outils informatiques qui permet le chiffrement, la confidentialité et l'intégrité des données avec une authentification sécurisée. De plus, il permet les échanges de données de manière sécurisée.

IV) Installation du service SSL

- Nous mettons à jour les paquets :

```
root@SSL:~# apt-get update
```

- Nous installons le service « **openssl** » :

```
root@SSL:~# apt-get install openssl
```

V) Création des dossiers et des fichiers SSL

- Nous nous connectons en tant qu'utilisateur nommé « **bastien** » (une fois créé) et créons le dossier « **/home/bastien/tpssl** » :

```
bastien@SSL:~$ mkdir /home/bastien/tpssl
bastien@SSL:~$ _
```

- Nous nous reconnectons en tant que « **root** » et faisons une copie du fichier « **/etc/ssl/openssl.cnf** » dans le répertoire « **/home/bastien/tpssl** » :

```
root@SSL:~# cp /etc/ssl/openssl.cnf /home/bastien/tpssl/
root@SSL:~# _
```

- Nous nous reconnectons en tant qu'utilisateur « **bastien** » et créons les dossiers suivants dans « **/home/bastien** » :

```
bastien@SSL:~$ mkdir /home/bastien/tpssl/private
bastien@SSL:~$ mkdir /home/bastien/tpssl/certs
bastien@SSL:~$ mkdir /home/bastien/tpssl/crl
bastien@SSL:~$ mkdir /home/bastien/tpssl/newcerts
bastien@SSL:~$ _
```

Description des répertoires :

- ⇨ « **private** » : Ce dossier représente le contenu des clés privées.
- ⇨ « **certs** » : Ce répertoire permet d'enregistrer les certificats.
- ⇨ « **crl** » : Celui-ci contient la liste des certificats n'étant plus valides.
- ⇨ « **newcerts** » : Celui-ci concerne la copie de nouveaux certificats avec un numéro de série.

- Ensuite, nous créons les fichiers suivants :

```
bastien@SSL:~$ touch /home/bastien/tpssl/index.txt
bastien@SSL:~$ touch /home/bastien/tpssl/serial
bastien@SSL:~$ _
```

- ⇨ « **index.txt** » : Ce fichier est utilisé pour le stockage des données sur les certificats signés.
- ⇨ « **serial** » : Celui-ci contient le numéro de série du certificat suivant.
- ⇨ Le numéro de série pour les certificats **SSL** permet d'identifier un certificat de manière unique et de faire autorité de certification (**CA** : **C**ertificate **A**uthority).

- Nous pouvons lister le contenu du dossier « **/home/bastien/tpssl** » :

```
bastien@SSL:~$ ls /home/bastien/tpssl/
certs  crl  index.txt  newcerts  openssl.cnf  private  serial
bastien@SSL:~$ _
```

- Maintenant, nous éditons le fichier « **/home/bastien/tpssl/serial** » et attribuons un numéro de série qui est « **01** » :

```
  GNU nano 2.2.6       Fichier : /home/bastien/tpssl/serial

01
```

VI) Création des certificats SSL

- Nous nous rendons dans le dossier « **/home/bastien/tpssl** » et créons le certificat de l'autorité de certification :

```
bastien@SSL:~/tpssl$ openssl req -new -x509 -extensions v3_ca -keyout private/ca
key.pem -out cacert.pem -days 3650 -config openssl.cnf _
```

- ⇨ « **cakey.pem** » représente la clé privée protégée par un mot de passe.
- ⇨ « **cacert.pem** » représente la demande de certificat numérique étant valable 3650 jours.

- Nous saisissons un message (« **bonjour** » par exemple non visible) au niveau du champ de saisie « **Enter PEM pass phrase** » :

```
Generating a 2048 bit RSA private key
.....................................+++
......+++
writing new private key to 'private/cakey.pem'
Enter PEM pass phrase:_
```

- Nous saisissons les données suivantes pour le certificat (en jaune) :

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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]:14
Locality Name (eg, city) []:caen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Techrom
Organizational Unit Name (eg, section) []:Service reseau
Common Name (e.g. server FQDN or YOUR name) []:CA Techrom
Email Address []:bastien.ettori@gmail.com
bastien@SSL:~/tpssl$ _
```

- Nous vérifions la présence des fichiers « **cakey.pem** » et « **cacert.pem** » :

```
bastien@SSL:~/tpssl$ ls -l
total 36
-rw-r--r-- 1 bastien bastien  1440 nov.  15 09:19 cacert.pem
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:01 certs
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:01 crl
-rw-r--r-- 1 bastien bastien     0 nov.  15 09:03 index.txt
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:01 newcerts
-rw-r--r-- 1 root    root    10835 nov.  15 08:53 openssl.cnf
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:12 private
-rw-r--r-- 1 bastien bastien     3 nov.  15 09:08 serial
bastien@SSL:~/tpssl$ _
```

```
bastien@SSL:~/tpssl$ ls -l private/
total 4
-rw-r--r-- 1 bastien bastien 1834 nov.  15 09:19 cakey.pem
bastien@SSL:~/tpssl$ _
```

- Nous devons extraire le certificat racine :

```
bastien@SSL:~/tpssl$ openssl x509 -text -in cacert.pem
```

- Et, nous visualisons le contenu du certificat :

```
-----BEGIN CERTIFICATE-----
MIID+TCCAuGgAwIBAgIJAPk92AUcWUb1MA0GCSqGSIb3DQEBCwUAMIGSMQswCQYD
VQQGEwJGUjELMAkGA1UECAwCMTQxDTALBgNVBAcMBGNhZW4xEDAOBgNVBAoMB1Rl
Y2hyb20xFzAVBgNVBAsMDlNlcnZpY2UgcmVzZWF1MRMwEQYDVQQDDApDQSBUZWNo
cm9tMScwJQYJKoZIhvcNAQkBFhhiYXNOaWVuLmVOdG9yaUBnbWFpbC5jb20wHhcN
MTYxMTE1MTAwMDI2WhcNMjYxMTEzMTAwMDI2WjCBkjELMAkGA1UEBhMCRlIxCzAJ
BgNVBAgMAjEOMQ0wCwYDVQQHDARjYWVuMRAwDgYDVQQKDAdUZWNocm9tMRcwFQYD
VQQLDA5TZXJ2aWNlIHJlc2VhdTETMBEGA1UEAwwKQOEgVGVjaHJvbTEnMCUGCSqG
SIb3DQEJARYYYmFzdGllbi5ldHRvcmlAZ21haWwuY29tMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEA5vVUhl+eV6MfgtrBp/sGbJ534w6I5GEp1IaDJU1+
mW9fq48AekykXDKGEPhO41eCO2INKQlixoLXpX3ah7dL9rsngcOEOFqkC7AbOxhu
e3GznmdkfSpFW+E5xJnUulxAVcIllKBqvL11Ghbgy7JvDX7CXl26PUlgI9+Tsj7R
TplqZ1B3KBOTFO2JZ5oKtk9G4vHDGG4AFhglUt7VSGqyHPhjA1GhLdvzGerDKEnR
Io8ysmsNIlDssLynokaMe+EQXSqrg83HY6T4qu+6Z1VVhXcBX2NFuhoGtYC1OgCl
o4LTVRYJILKIPNBfL4c3z7Zil5OQ6mvPhkqsvO95t9z4nwIDAQABo1AwTjAdBgNV
HQ4EFgQUdo7Y4kw2gLKJt2CmHAaGOBz5Te8wHwYDVR0jBBgwFoAUdo7Y4kw2gLKJ
t2CmHAaGOBz5Te8wDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQsFAAOCAQEAL0Ti
lHOk8N+zp3wmVRaWlINDBB41SZvwSXltKnc2KAPefogIVgod2m8HO+OjQvFpolu3
MByZPLUCSXbXgOifecZMNGKeeT/BwnT3ixdJsDITIoY+S+db306Gar90XsiSX726
L8Ja+qrNMeCpiwsA4qCsmffudEIgOI8kySb2sedUUv2EZ+DpXLZIcroayXOemfij
bGMgMb9W2VAY6WwLPnX2KBLyDG8SZtRZtJq1ZJAtjp0dKfGbejyFwMCAQhlICQGw
97SmTTOyvZU8Cg6D/MKafwHP1DTkIUTpwsl/phdFR46wb1TLe8BU8/mjioNiqsf7
J/7OVwEH4FEhNDpsBQ==
-----END CERTIFICATE-----
bastien@SSL:~/tpssl$ _
```

- Nous archivons les fichiers :

```
bastien@SSL:~/tpssl$ tar -czf rootca.tar.gz private/cakey.pem cacert.pem
bastien@SSL:~/tpssl$ _
```

- Nous créons les 2 clés, demandons le certificat et saisissons un message (« **bonjour** » par exemple non visible comme pour le premier) :

```
bastien@SSL:~/tpssl$ openssl req -config ./openssl.cnf -new -keyout private/webk
ey.pem -out certs/newreq.pem
Generating a 2048 bit RSA private key
.............+++
...............................+++
writing new private key to 'private/webkey.pem'
Enter PEM pass phrase:_
```

- Nous saisissons les données suivantes pour le certificat (en jaune) :

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
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]:14
Locality Name (eg, city) []:caen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Techrom
Organizational Unit Name (eg, section) []:service reseau
Common Name (e.g. server FQDN or YOUR name) []:techrom.fr
Email Address []:bastien.ettori@gmail.com
```

- Nous saisissons un mot de passe pour la demande de certificat :

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:root
An optional company name []:
bastien@SSL:~/tpssl$ _
```

- Nous listons les données pour vérifier si tous les fichiers sont présents et constatons que c'est le cas :

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

```
bastien@SSL:~/tpssl$ ls private/
cakey.pem  webkey.pem
bastien@SSL:~/tpssl$ ls certs/
newreq.pem
bastien@SSL:~/tpssl$ _
```

  - ⇨ « **webkey.pem** » représente la paire de clé publique/privée.
  - ⇨ « **newreq.pem** » représente la nouvelle demande de certificat contenant la clé publique.

- Nous nous déconnectons pour se reconnecter en tant que « **root** » :

```
bastien@SSL:~/tpssl$ su
Mot de passe :
root@SSL:/home/bastien/tpssl#
```

- Nous attribuons les droits au fichier « **openssl.cnf** » pour l'utilisateur « **bastien** » :

```
root@SSL:/home/bastien/tpssl# chown bastien.bastien openssl.cnf
root@SSL:/home/bastien/tpssl# _
```

- Nous nous déconnectons de la session « **root** » pour se reconnecter en tant que « **bastien** » et vérifions les droits sur ce fichier :

```
root@SSL:/home/bastien/tpssl# exit
exit
bastien@SSL:~/tpssl$ ls -l
total 40
-rw-r--r-- 1 bastien bastien  1440 nov.  15 09:19 cacert.pem
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:32 certs
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:01 crl
-rw-r--r-- 1 bastien bastien     0 nov.  15 09:03 index.txt
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:01 newcerts
-rw-r--r-- 1 bastien bastien 10835 nov.  15 08:53 openssl.cnf
drwxr-xr-x 2 bastien bastien  4096 nov.  15 09:29 private
-rw-r--r-- 1 bastien bastien  2570 nov.  15 09:25 rootca.tar.gz
-rw-r--r-- 1 bastien bastien     3 nov.  15 09:08 serial
bastien@SSL:~/tpssl$ _
```

VII) Configuration du service SSL

- Nous éditons le fichier « **openssl.cnf** » et indiquons le dossier où se situe le fichier :

```
  GNU nano 2.2.6            Fichier : openssl.cnf

default_ca      = CA_default            # The def

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

dir             = /home/bastien/tpssl   # Where e
```

- Ensuite, nous signons le certificat pour le déployer et ressaisissons un message comme avant :

```
bastien@SSL:~/tpssl$ 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/bastien/tpssl/private/cakey.pem:_
```

- Nous répondons « **y** » pour **yes** afin d'accepter la signature du certificat et de mettre à jour la base de données (BDD) :

```
Certificate is to be certified until Nov 15 08:48:55 2017 GMT (365 days)
Sign the certificate? [y/n]:y


1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
bastien@SSL:~/tpssl$ _
```

- Nous vérifions si le chemin du dossier est correct et constatons que c'est le cas :

```
bastien@SSL:~/tpssl$ openssl verify -CAfile cacert.pem certs/webcert.pem
certs/webcert.pem: OK
bastien@SSL:~/tpssl$ _
```

VIII) Installation du certificat SSL

Pour prendre en charge les certificats **SSL**, nous devons posséder les éléments suivants :

- ⇨ « **webcert.pem** » représente le certificat pour le serveur Web **Apache2**.
- ⇨ « **webkey-clair.pem** » représente la clé privée non cryptée du serveur Web « **apache2** ».

- Nous installons le service Web « **apache2** » en tant que « **root** » :

```
bastien@SSL:~/tpssl$ su
Mot de passe :
root@SSL:/home/bastien/tpssl#
```

```
root@SSL:~# apt-get install apache2
```

- Nous nous reconnectons en tant que « **bastien** », créons un fichier avec la clé privée non cryptée et saisissons un nouveau message :

```
bastien@SSL:~/tpssl$ openssl rsa -in private/webkey.pem -out private/webkey-clai
r.pem
Enter pass phrase for private/webkey.pem:
writing RSA key
bastien@SSL:~/tpssl$ _
```

- Nous nous déconnectons pour se reconnecter en tant que « **root** » :

```
bastien@SSL:~/tpssl$ su
Mot de passe :
root@SSL:/home/bastien/tpssl#
```

- Nous nous rendons dans le dossier « **/etc/ssl** » et attribuons les droits au dossier « **/etc/apache2** » à l'utilisateur « **bastien** » :

```
root@SSL:~# cd /etc/ssl/
root@SSL:/etc/ssl# chown bastien.bastien /etc/apache2/
root@SSL:/etc/ssl# _
```

- Nous créons un dossier « **ssl** » dans « **/etc/apache2** » et c'est dans ce répertoire que nous allons copier les fichiers :

```
root@SSL:~# mkdir /etc/apache2/ssl
root@SSL:~# _
```

- Nous nous reconnectons avec l'utilisateur « **bastien** » et copions le fichier de la clé non cryptée « **webkey-clair.pem** » dans le répertoire « **/etc/apache2/ssl** » :

```
bastien@SSL:~/tpssl/private$ cp webkey-clair.pem /etc/apache2/ssl/webkey-clair.p
em
bastien@SSL:~/tpssl/private$ _
```

- Nous copions le fichier du certificat « **webcert.pem** » dans ce même dossier :

```
bastien@SSL:~/tpssl/certs$ cp webcert.pem /etc/apache2/ssl/webcert.pem
bastien@SSL:~/tpssl/certs$ _
```

- Nous nous déconnectons de nouveau du compte utilisateur « **bastien** », allons dans le « **/etc/ssl** » et attribuons les droits aux dossiers « **/etc/apache2/mods-available** » et « **/etc/apache2/mods-enabled** » :

```
root@SSL:/etc/ssl# chown bastien.bastien /etc/apache2/mods-available/
root@SSL:/etc/ssl# chown bastien.bastien /etc/apache2/mods-enabled/
root@SSL:/etc/ssl# _
```

- Nous créons les liens symboliques des répertoires « **/etc/apache2/mods-available** » et « **/etc/apache2/mods-enabled** » en tant qu'utilisateur « **bastien** » :

```
bastien@SSL:~$ ln -s /etc/apache2/mods-available/ /etc/apache2/mods-enabled/
bastien@SSL:~$ _
```

- Nous activons le mode **SSL** en tant que « **root** » :

```
root@SSL:/etc/ssl# a2enmod ssl
```

- Nous redémarrons le service « **apache2** » :

```
root@SSL:~# systemctl restart apache2.service
root@SSL:~# _
```

- Nous nous rendons dans le fichier « **/etc/apache2/sites-available/default-ssl.conf** » et modifions les 2 lignes suivantes (en jaune) :

```
GNU nano 2.2.6 Fichier : ...apache2/sites-available/default-ssl.conf

#   If both key and certificate are stored in the same fil
#   SSLCertificateFile directive is needed.
SSLCertificateFile      /etc/apache2/ssl/webcert.pem
SSLCertificateKeyFile   /etc/apache2/ssl/webkey-clair.pem
```

- Nous activons le fichier **SSL** par défaut :

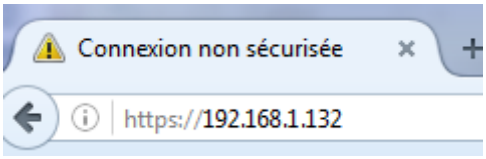
```
root@SSL:/etc/apache2/ssl# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@SSL:/etc/apache2/ssl# _
```

- Nous redémarrons le service « **apache2** » :

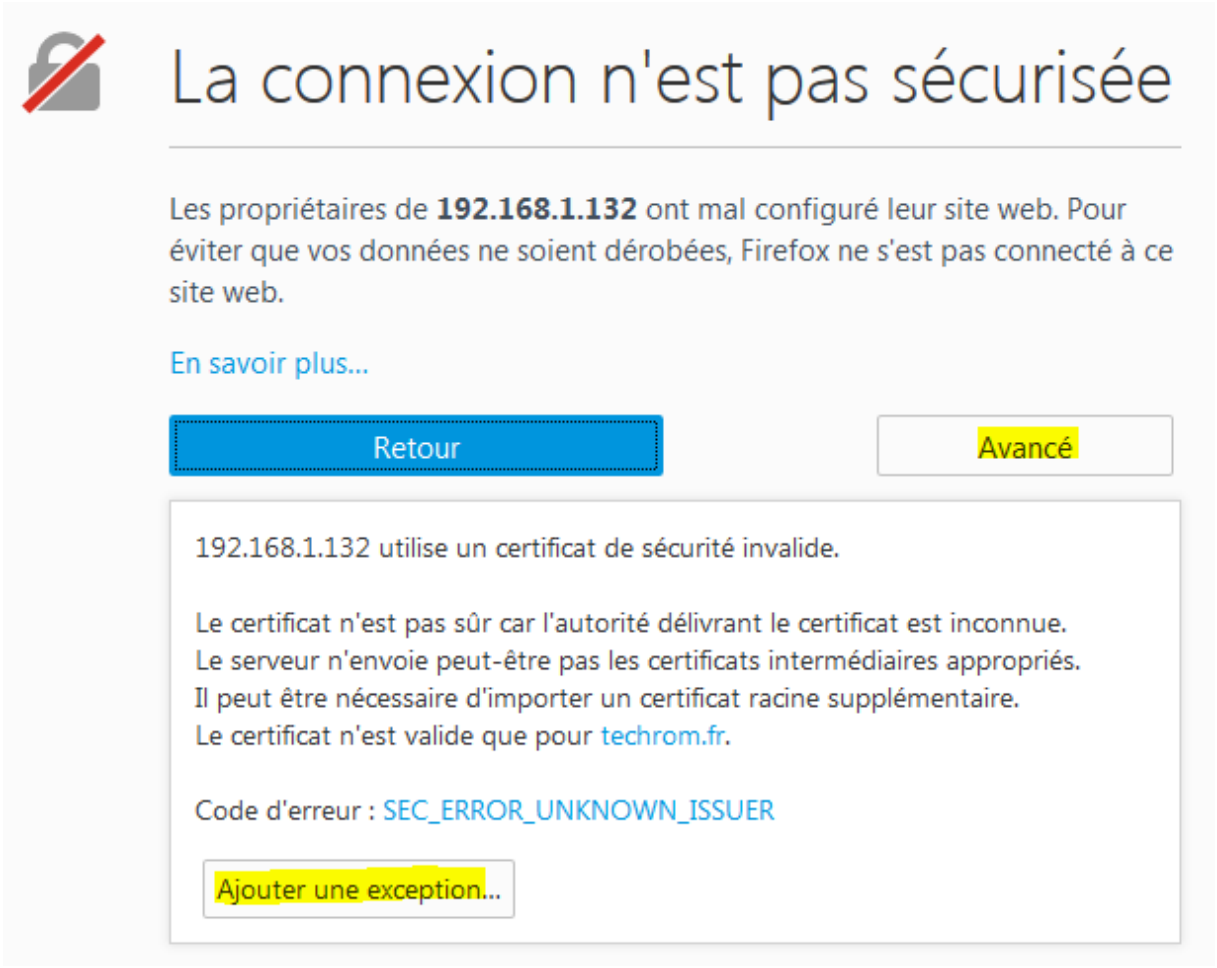
```
root@SSL:~# systemctl restart apache2.service
root@SSL:~# _
```

- Maintenant, nous testons le service « **apache2** » via un navigateur :

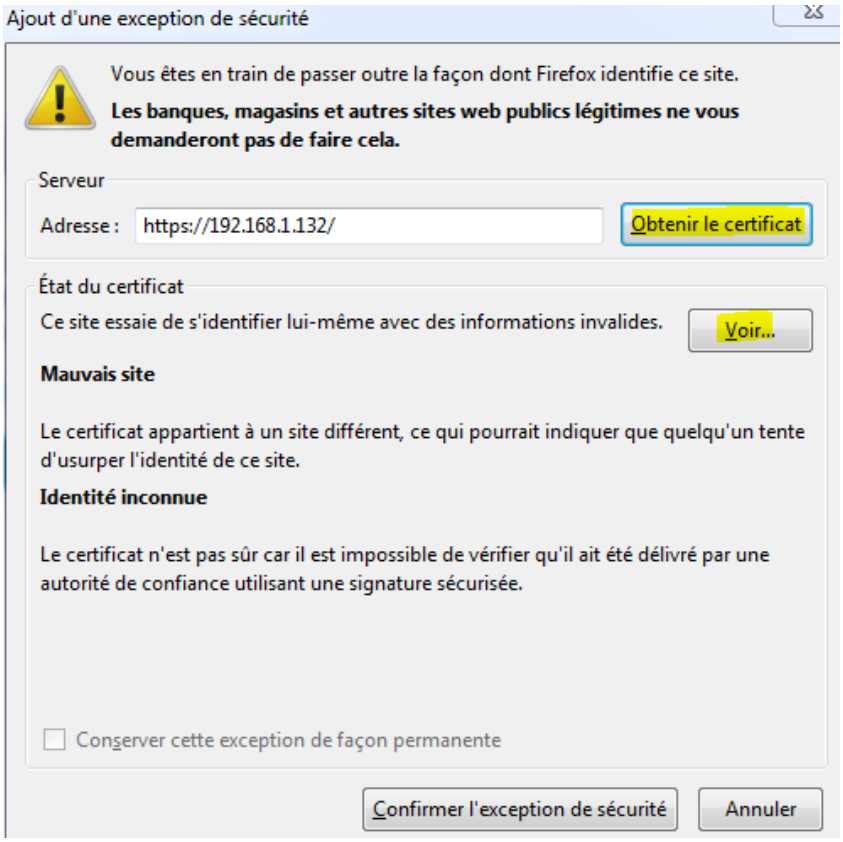
- Nous testons le service « **apache2** » en **HTTPS** :



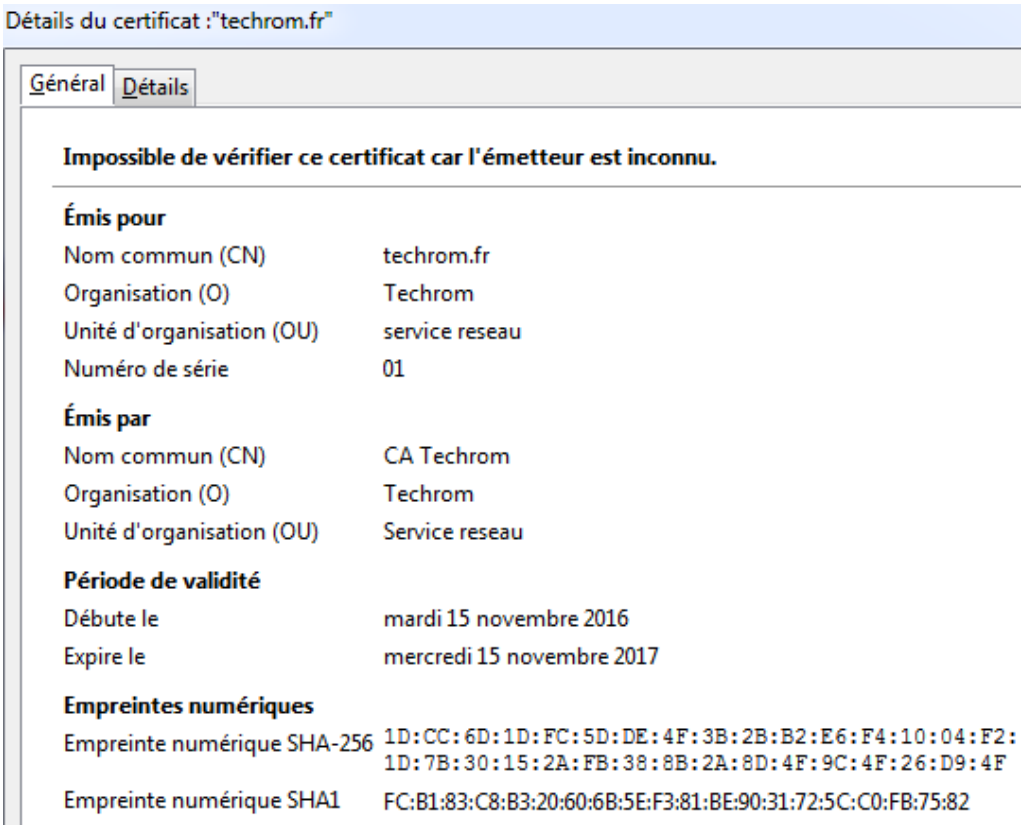
- Nous cliquons sur « **Avancé** » et « **Ajouter une exception** » :



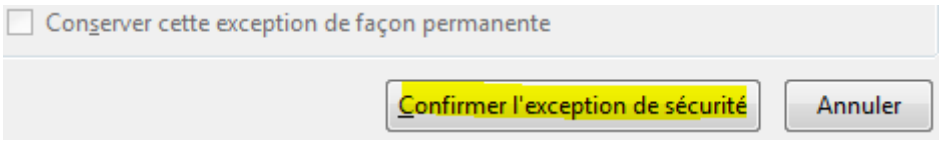
- Nous cliquons sur « **Obtenir le certificat** » et « **Voir** » :



- Nous visualisons les détails du certificat « **techrom.fr** » :



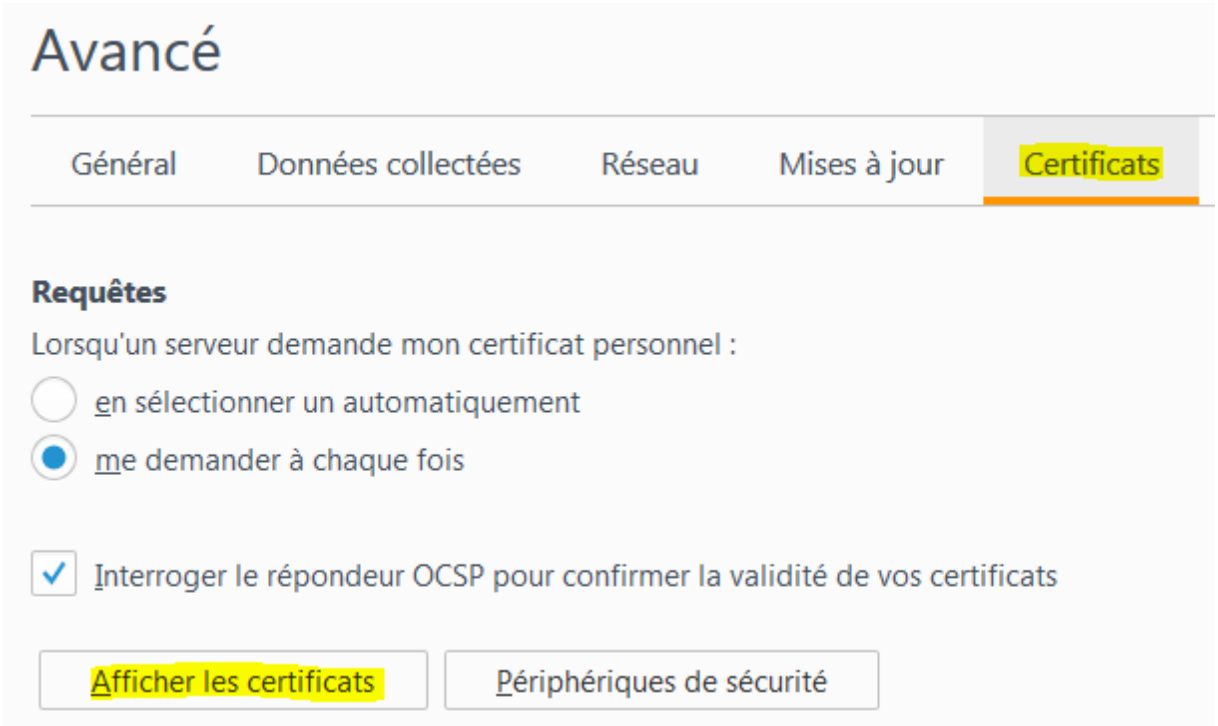
- Nous cliquons sur « **Confirmer l'exception de sécurité** » :



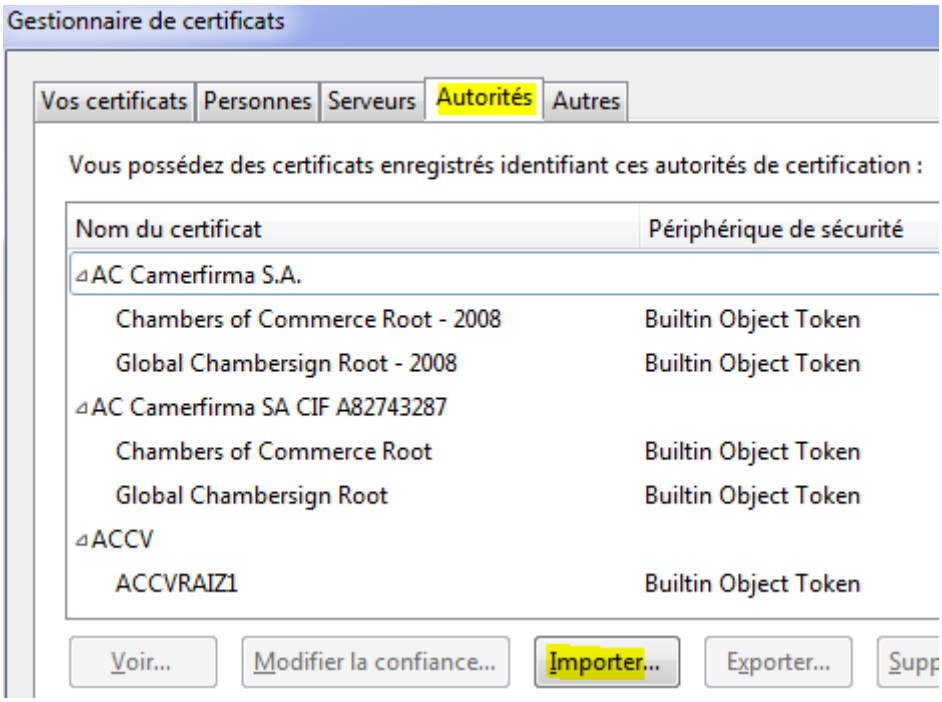
- Nous constatons que le service « **apache2** » est en **HTTPS** :

IX) Importation du fichier « cacert.pem » sur un navigateur

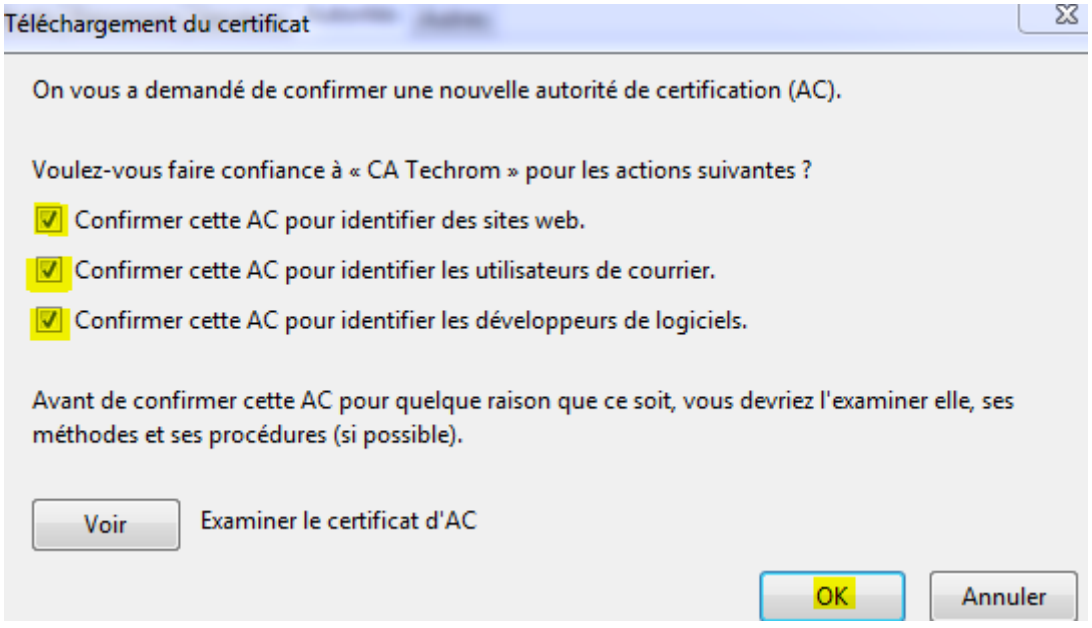
- D'abord, nous cliquons sur « **Certificats** » et « **Ajouter les certificats** » :



- Nous allons dans l'onglet « **Autorités** » et « **Importer** » :



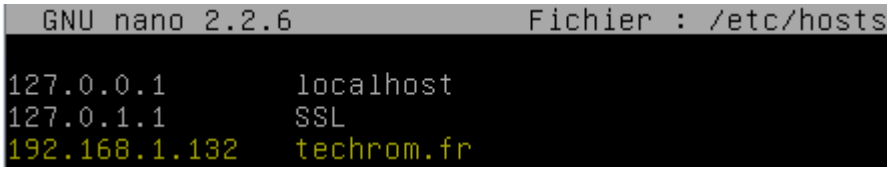
- Nous cochons les 3 cases pour confirmer l'autorité de certification et « **OK** » :



- Et, nous constatons que cela continue de fonctionner :

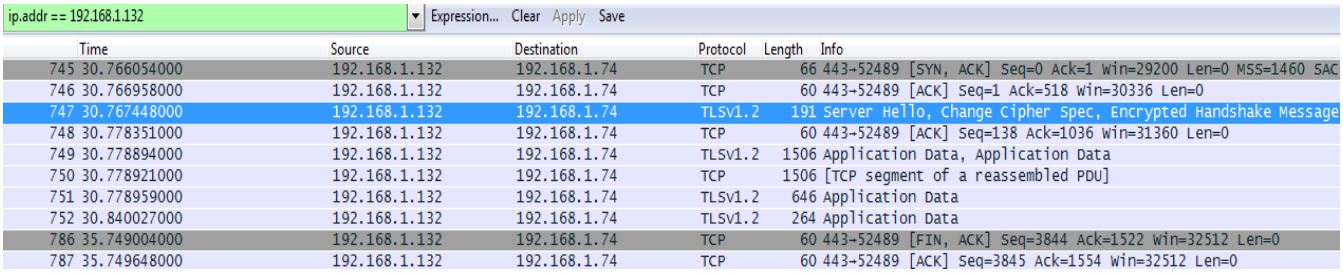
X) Configuration du nom DNS

- Nous éditons le fichier « **/etc/hosts** » et saisissons l'adresse IP du serveur **SSL** avec le nom de l'organisation :



XI) Test de la connexion sécurisée via une capture de trame

- Pour visualiser et vérifier que la connexion sécurisée s'est bien réalisée, nous pouvons effectuer une analyse de trame via le logiciel **Wireshark** :



Nous constatons que la connexion est bien sécurisée entre la machine cliente et le serveur **SSL** via le protocole de sécurisation **TLSv1.2**.

XII) Conclusion

En conclusion, nous pouvons constater que le serveur **SSL** est fonctionnel car il utilise le protocole sécurisé **TLSv1.2** et que le serveur Web **Apache** est donc bien sécurisé.