ETTORI Bastien	BTS SIO 2ème année
21 mars 2016	Année scolaire : 2015/2016
Option : SISR	Version 1.0

OPENSSL DEBIAN

SOMMAIRE :

I) Objectif.....2

II) Prérequis.....2

III) Définition.....2

IV) Installation OpenSSL.....2

V) Connexion utilisateur / création des dossiers et fichiers SSL.....2-3

VI) Création des certificats SSL.....4-7

VII) Configuration OpenSSL.....7-12

VIII) Installation du service FTP.....12-13

IX) Configuration du service FTP.....13-14

X) Vérification de connexion avec un client FTP.....14-15

XI) Importation du certificat « cacert.pem » sur un navigateur.....15-17

XII) Configuration du nom DNS..... 17

XIII) Visualisation de la connexion sécurisée.....18

XIV) Conclusion..... 18

I) Objectif

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

II) Prérequis

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

Nombre de machines	SE serveur SSH	Nom serveur SSH	Adresse IP serveur SSH	Adresse IP de la machine cliente Windows
1	**Debian 7.7**	**debian**	**192.168.1.108**	**192.168.1.73**

III) Définition

Open SSL (**Open S**ecure **S**ocket **L**ayer) est une boîte à outils informatiques qui permet de chiffrer et d'échanger des données entre 2 ou plusieurs ordinateurs à distance de manière sécurisée.

IV) Installation OpenSSL

- Tout d'abord, nous mettons à jour les paquets :

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

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

```
root@debian:~# apt-get install openssl_
```

V) Connexion avec un utilisateur et création des dossiers et fichiers SSL

- Nous nous connectons avec un autre utilisateur nommé « **bastien** » et nous créons le dossier « **/home/bastien/tpssl** » :

```
debian login: bastien
Password:
Linux debian 3.2.0-4-amd64 #1 SMP Debian 3.2

The programs included with the Debian GNU/Li
the exact distribution terms for each prograr
individual files in /usr/share/doc/*/copyrigh

Debian GNU/Linux comes with ABSOLUTELY NO WAF
permitted by applicable law.
bastien@debian:~$ mkdir /home/bastien/tpssl_
```

- Nous nous reconnectons avec l'utilisateur « **root** », nous nous rendons dans le dossier « **/etc/ssl** » et nous faisons une copie du fichier « **openssl.cnf** » dans le répertoire « **/home/bastien/tpssl** » :

```
root@debian:~# cd /etc/ssl
root@debian:/etc/ssl# ls
certs  openssl.cnf  private
root@debian:/etc/ssl# cp openssl.cnf /home/bastien/tpssl/openssl.cnf
root@debian:/etc/ssl# _
```

- Nous nous reconnectons avec l'utilisateur « **bastien** », nous nous rendons dans le dossier « **/home/bastien/tpssl** » et nous créons les dossiers suivants :

  - « **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 pour nom de fichier.

```
bastien@debian:~$ cd /home/bastien/tpssl/
bastien@debian:~/tpssl$ ls
openssl.cnf
bastien@debian:~/tpssl$ mkdir private
bastien@debian:~/tpssl$ mkdir certs
bastien@debian:~/tpssl$ mkdir crl
bastien@debian:~/tpssl$ mkdir newcerts
bastien@debian:~/tpssl$ ls
certs  crl  newcerts  openssl.cnf  private
```

- Nous créons les fichiers suivants :

  - « **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).

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

- Nous éditons le fichier « **serial** » et nous attribuons un numéro de série :

```
bastien@debian:~/tpssl$ nano serial
```

- Ici, ce numéro est « **01** » :

```
  GNU nano 2.2.6                Fichier : serial

01
```

VI) Création des certificats SSL

- Nous créons le premier certificat :

```
bastien@debian:~/tpssl$ openssl req -new -x509 -extensions v3_ca -keyout private
/cakey.pem -out cacert.pem -days 3650 -config openssl.cnf_
```

- Ensuite, nous saisissons un message (« **bonjour** » par exemple) au niveau de la zone « **Enter PEM pass phrase** » :

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

- Nous introduisons les données suivantes pour le certificat :

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorp
into your certificate request.
What you are about to enter is what is called a Distinguished Name
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@debian:~/tpssl$ _
```

- Nous vérifions que les fichiers sont bien présents :

```
bastien@debian:~/tpssl$ ls -l
total 36
-rw-r--r-- 1 bastien bastien  1440 nov.   9 09:36 cacert.pem
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:27 certs
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:27 crl
-rw-r--r-- 1 bastien bastien     0 nov.   9 09:28 index.txt
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:27 newcerts
-rw-r--r-- 1 root    root    10835 nov.   9 09:26 openssl.cnf
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:32 private
-rw-r--r-- 1 bastien bastien     3 nov.   9 09:29 serial
bastien@debian:~/tpssl$ cd private/
bastien@debian:~/tpssl/private$ ls -l
total 4
-rw-r--r-- 1 bastien bastien 1834 nov.   9 09:36 cakey.pem
bastien@debian:~/tpssl/private$ _
```

- Nous nous rendons dans le dossier « **/private** » et nous vérifions la présence du certificat **SSL** :

```
bastien@debian:~/tpssl$ cd private/
bastien@debian:~/tpssl/private$ ls -l
total 4
-rw-r--r-- 1 bastien bastien 1834 nov.   9 09:36 cakey.pem
bastien@debian:~/tpssl/private$ _
```

- Nous devons extraire le certificat racine :

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

```
-----BEGIN CERTIFICATE-----
MIID+TCCAuGgAwIBAgIJAOpBZwI+KXcEMA0GCSqGSIb3DQEBBQUAMIGSMQswCQYD
VQQGEwJGUjELMAkGA1UECAwCMTQxDTALBgNVBAcMBGNhZW4xEDAOBgNVBAoMB1Rl
Y2hyb20xFzAVBgNVBAsMDlNlcnZpY2UgcmVzZWF1MRMwEQYDVQQDDApDQSBUZWNo
cm9tMScwJQYJKoZIhvcNAQkBFhhiYXN0aWVuLmV0dG9yaUBnbWFpbC5jb20wHhcN
MTUxMTA5MDgzNjMzWhcNMjUxMTA2MDgzNjMzWjCBkjELMAkGA1UEBhMCRlIxCzAJ
BgNVBAgMAjE0MQ0wCwYDVQQHDARjYWVuMRAwDgYDVQQKDAdUZWNocm9tMRcwFQYD
VQQLDA5TZXJ2aWNlIHJlc2VhdTETMBEGA1UEAwwKQ0EgVGVjaHJvbTEnMCUGCSqG
SIb3DQEJARYYYmFzdGllbi5ldHRvcmlAZ21haWwuY29tMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAu5CC/Gnh7XJL1gVpyLNXZqArNxeGQMOEb9Ettgjs
aObyZAW5J6TWfOkVHMpBjMeD/hcEDOoxczAQTdlSp8rfC/2jet/PBuOXHX+ette8
MmFqdDZapngmIASk4mjbXe7oSk4anPPfrUcRHRN8jDEwZ3XFbw37WUcqrbzhX7p6
w+w5y8WvygOqo9BMlu2z99A6ZkiOCzu6D1vQDkT193Ubur9nLWtV9fVt5OfOYfGl
GTMDoeh3VxVKHfP1jOsEO7He4Xk9ufIQAGarxcDf5OmlRet64/lOyT4A7Y/n4QHS
T3O2OosycYdvhBsea+T90WGnzVESmbhwZcjm9Vf9bRJBWwIDAQABo1AwTjAdBgNV
HQ4EFgQUQn6Eip2DO/5mhWGOwz/UbDOtB9wwHwYDVR0jBBgwFoAUQn6Eip2DO/5m
hWGOwz/UbDOtB9wwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAYqpA
/ZAktXpKNmDU1xcR3NCphajIOWDxEI4jZ5X/ME4aQiwE41Z4akj1zIUR8CrNy63Q
9p4Ez3CAOxLYHJFkdr6KxMJYbULIzCPwOByrOlO4aLVWc5IocORPNAcEuqsjfpls
jdiaKeKqNvfC/pfkitHVWPEyEGQZJPTOWvyyMQC/Ow7nMdacvOgcw9TaewlgSQx7
O8fG5N3ulHAevRWx8AOApaK1oSXB9VKm6hEKtNw/O1Dm2ZkoOIcMFNb9YAThGkOO
YsF/Z7IM8tjz4FEeo25R72nf5CGt+z581ZbIUxbZg+06B5cdIXoOAlhWo61gSTqc
SOMKFstNGgbspNi9PA==
-----END CERTIFICATE-----
bastien@debian:~/tpssl$ _
```

Nous voyons bien le contenu du certificat.

- Nous devons archiver les fichiers :

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

- Nous créons les 2 clés et nous demandons le certificat **SSL** et Ensuite, nous saisissons un message (« **bonjour** » par exemple comme pour le premier) au niveau de la zone « **Enter PEM pass phrase** » :

```
bastien@debian:~/tpssl$ openssl req -config ./openssl.cnf -new -keyout private/w
ebkey.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 introduisons les données suivantes pour le certificat :

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorp
into your certificate request.
What you are about to enter is what is called a Distinguished Name
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 listons les dossiers pour visualiser si tous les fichiers sont présents et nous

```
bastien@debian:~/tpssl$ ls
cacert.pem  crl         newcerts     private         serial
certs       index.txt   openssl.cnf  rootca.tar.gz
bastien@debian:~/tpssl$ cd private/
bastien@debian:~/tpssl/private$ ls
cakey.pem  webkey.pem
bastien@debian:~/tpssl/private$ cd ..
bastien@debian:~/tpssl$ cd certs/
bastien@debian:~/tpssl/certs$ ls
newreq.pem
bastien@debian:~/tpssl/certs$ cd ..
```

- Nous nous déconnectons pour se reconnecter avec l'utilisateur « **root** » :

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

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

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

- Nous nous déconnectons de l'utilisateur « **root** » pour se reconnecter avec l'utilisateur« **bastien** » et nous vérifions que celui-ci possède les droits sur le fichier :

```
root@debian:/home/bastien/tpssl# exit
exit
bastien@debian:~/tpssl$ ls -l
total 40
-rw-r--r-- 1 bastien bastien  1440 nov.   9 09:36 cacert.pem
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:47 certs
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:27 crl
-rw-r--r-- 1 bastien bastien     0 nov.   9 09:28 index.txt
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:27 newcerts
-rw-r--r-- 1 bastien bastien 10835 nov.   9 09:26 openssl.cnf
drwxr-xr-x 2 bastien bastien  4096 nov.   9 09:44 private
-rw-r--r-- 1 bastien bastien  2568 nov.   9 09:40 rootca.tar.gz
-rw-r--r-- 1 bastien bastien     3 nov.   9 09:29 serial
bastien@debian:~/tpssl$ _
```

VII) Configuration OpenSSL

- Nous allons éditer le fichier de configuration d'**OpenSSL** dans le dossier « **/home/bastien/tpssl** » :

```
bastien@debian:~/tpssl$ nano openssl.cnf
```

- Au niveau de la ligne « **dir** », nous mettons le dossier où se situe ce fichier :

```
  GNU nano 2.2.6              Fichier : openssl.cnf


#####################################################
[ ca ]
default_ca      = CA_default            # The defa

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

dir             = /home/bastien/tpssl_          #
```

- Maintenant, nous signons le certificat pour le déployer :

```
bastien@debian:~/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 mettre à jour la base de données (BDD) :

```
Certificate is to be certified until Nov  8 08:58:55 2016 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@debian:~/tpssl$ _
```

- Nous vérifions le chemin du répertoire :

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

- Maintenant, nous créons un fichier avec clé privée non cryptée et nous un message non visible comme auparavant :

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

- Nous nous déconnectons pour se reconnecter avec l'utilisateur « **root** » :

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

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

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

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

```
bastien@debian:~/tpssl/private$ cp webkey-clair.pem /etc/apache2/ssl/webkey-clai
r.pem
bastien@debian:~/tpssl/private$ _
```

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

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

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

```
bastien@debian:/etc/apache2/ssl$ su
Mot de passe :
root@debian:/etc/apache2/ssl# cd /etc/ssl/
root@debian:/etc/ssl# chown bastien.bastien /etc/apache2/mods-available/
root@debian:/etc/ssl# chown bastien.bastien /etc/apache2/mods-enabled/
root@debian:/etc/ssl# _
```

- Nous créons les liens symboliques des dossiers « **/etc/apache2/mods-available** » et « **/etc/apache2/mods-enabled** » :

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

- Nous activons le mode **SSL** pour le serveur Web :

```
root@debian:/etc/ssl# a2enmod ssl
Enabling module ssl.
See /usr/share/doc/apache2.2-common/README.Debian.gz on how to configure SSL and
 create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
root@debian:/etc/ssl# _
```

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

```
root@debian:/etc/ssl# service apache2 restart
[....] Restarting web server: apache2apache2:
erver's fully qualified domain name, using 127
 ... waiting apache2: Could not reliably deter
domain name, using 127.0.1.1 for ServerName
. ok
root@debian:/etc/ssl# _
```

- Nous nous connectons de nouveau du compte utilisateur « **bastien** » et nous éditons le fichier « **default-ssl** » qui se situe dans le dossier « **/etc/apache2/sites-available** » :

```
bastien@debian:~$ nano /etc/apache2/sites-available/default-ssl
```

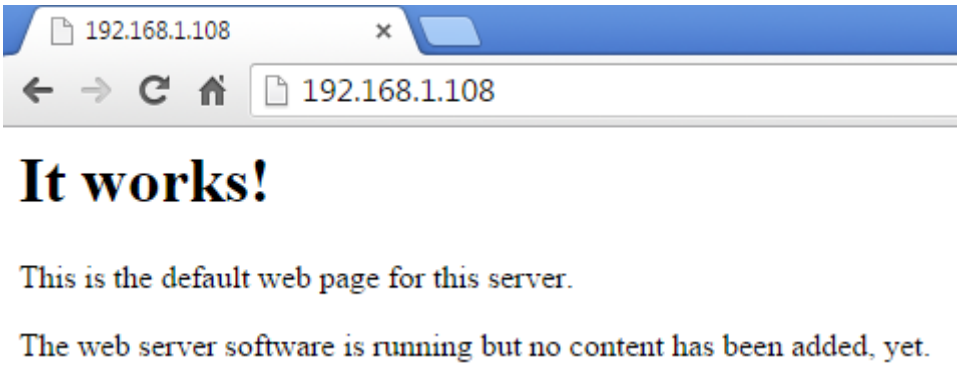
- Nous devons le modifier en tant que « **root** » et nous modifions les 2 lignes suivantes en jaune :

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

      #   Enable/Disable SSL for this virtual host.
      SSLEngine on

      #   A self-signed (snakeoil) certificate can be created by in
      #   the ssl-cert package. See
      #   /usr/share/doc/apache2.2-common/README.Debian.gz for more
      #   If both key and certificate are stored in the same file,
      #   SSLCertificateFile directive is needed.
      SSLCertificateFile    /etc/apache2/ssl/webcert.pem
      SSLCertificateKeyFile /etc/apache2/ssl/webkey-clair.pem_
```

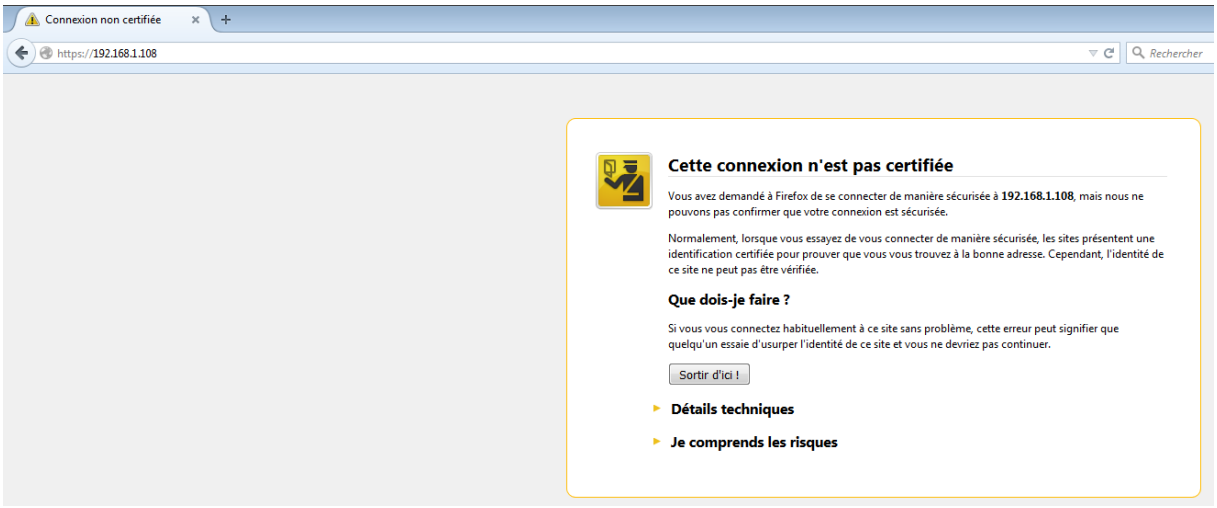
- Nous activons le fichier **SSL** par défaut :

```
root@debian:/etc/apache2/ssl# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run
  service apache2 reload
root@debian:/etc/apache2/ssl# service apache2 rest
[....] Restarting web server: apache2apache2: Coul
erver's fully qualified domain name, using 127.0.1
 ... waiting apache2: Could not reliably determine
domain name, using 127.0.1.1 for ServerName
. ok
root@debian:/etc/apache2/ssl# _
```

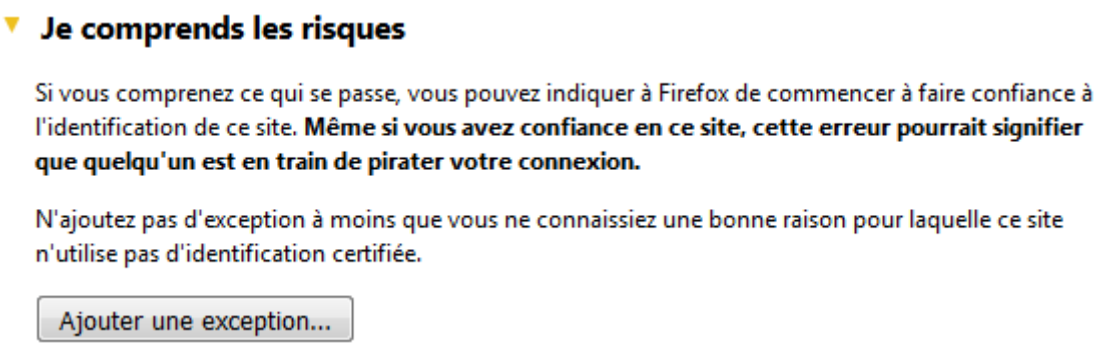
- Nous testons le service « **apache2** » dans un navigateur Web :



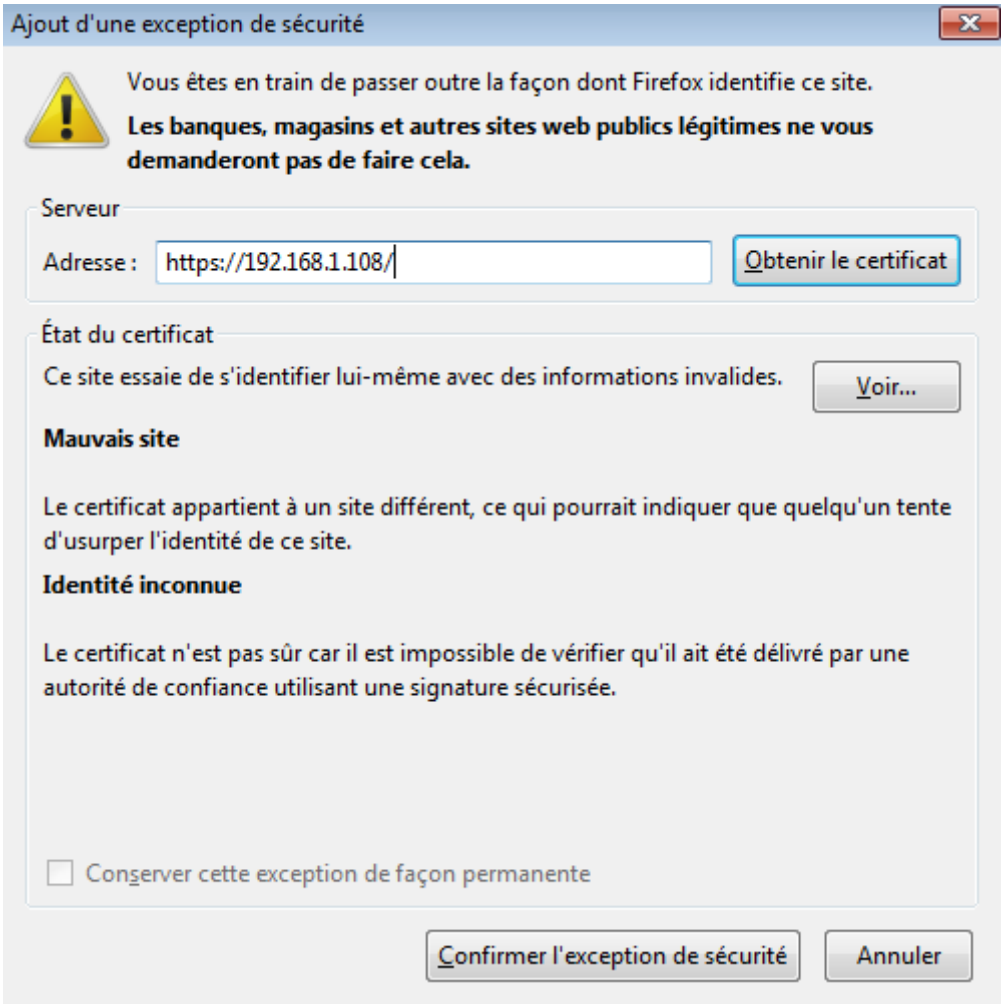
- Nous cliquons sur « **Je comprends les risques** » :



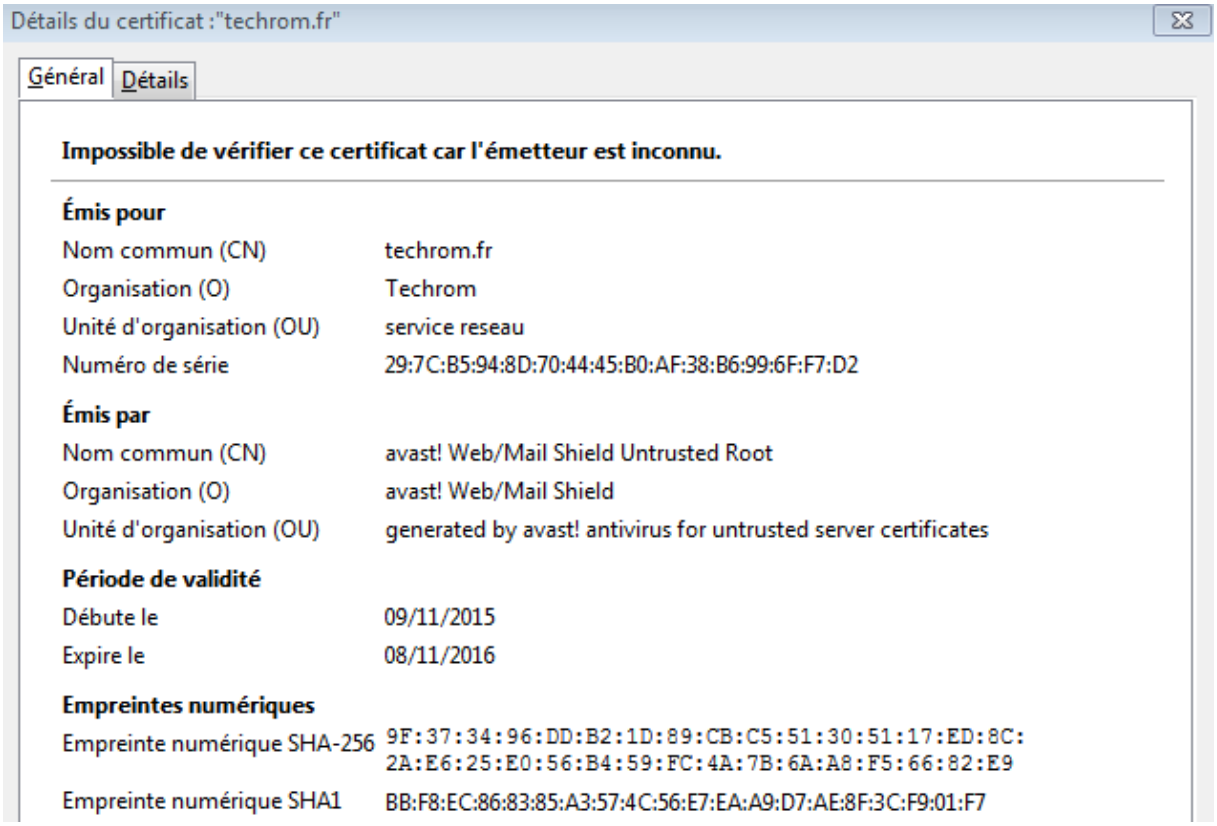
- Nous cliquons sur « **Ajouter une exception** » :



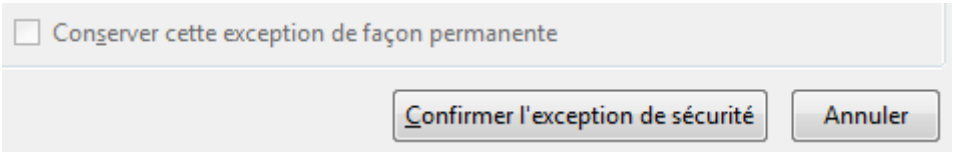
- Dans la zone « **Adresse** », nous renseignons l'adresse IP du serveur **SSL** et nous cliquons sur « **Obtenir le certificat** » :



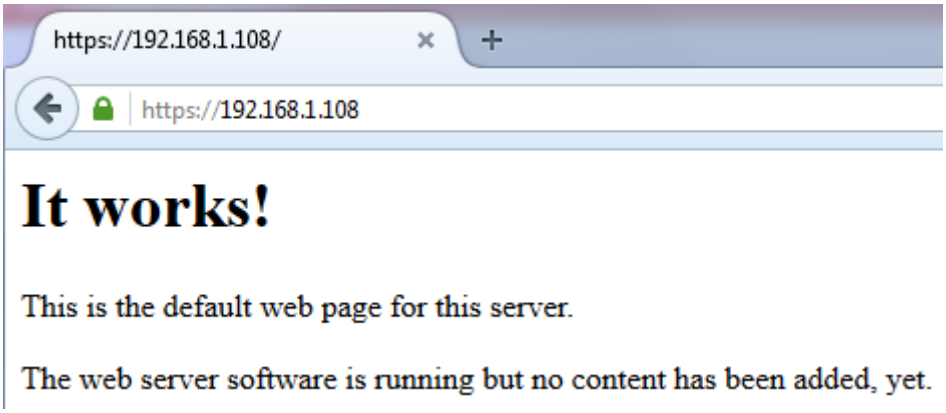
- Nous pouvons visualiser les détails du certificat « **techrom.fr** » :



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



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

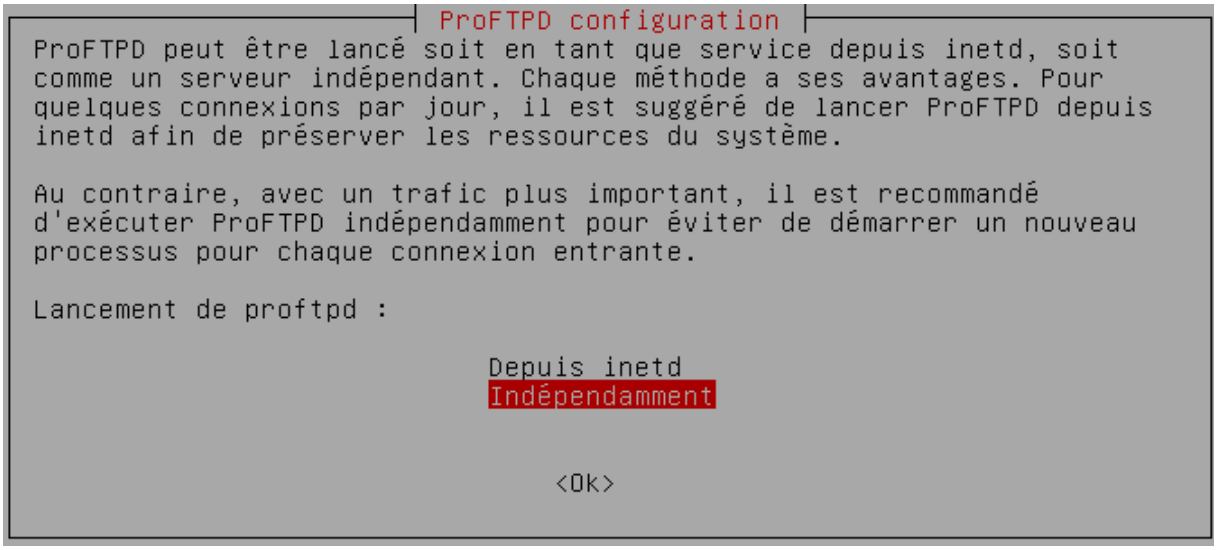


VIII) Installation du service FTP

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

```
root@debian:~# apt-get install proftpd
```

- Nous sélectionnons l'option « **Indépendamment** » :



IX) Configuration du service FTP

- Nous éditons le fichier « **proftpd.conf** » situé dans le dossier dans « **/etc/proftpd** » :

```
root@debian:/etc/proftpd# nano proftpd.conf
```

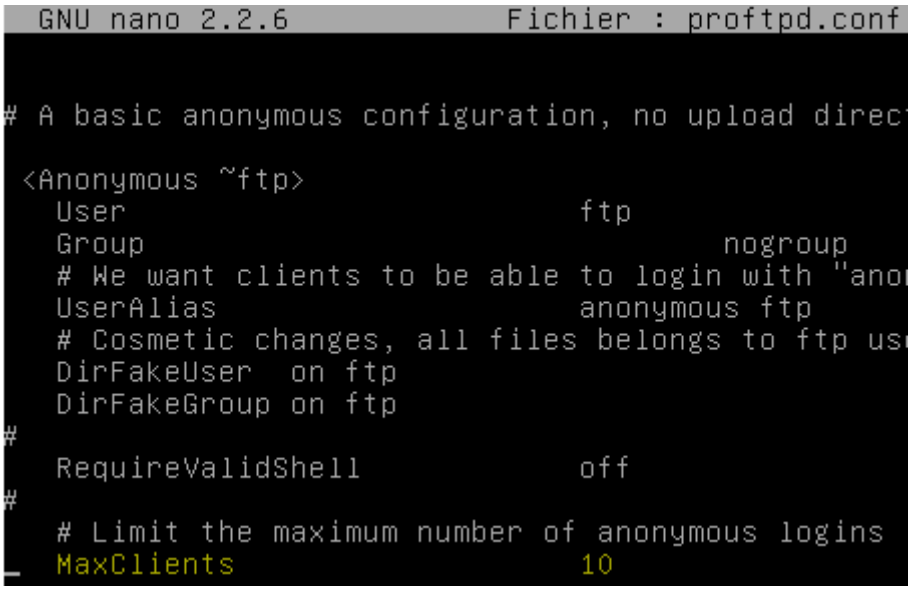
- Nous changeons le nom du serveur à la ligne « **ServerName** » et à la ligne « **DeferWelcome** », nous mettons « **on** » :

```
  GNU nano 2.2.6               Fichier : proftpd.conf

# If set on you can experience a longer connection
IdentLookups                    off

ServerName                      "Debian"
ServerType                      standalone
DeferWelcome                    on
```

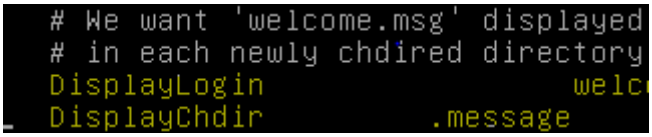
- Nous décommentons la ligne « **RequireValidShell** » :

```
# Use this to jail all users in their homes
 DefaultRoot                    ~

# Users require a valid shell listed in /etc/shells to login.
# Use this directive to release that constrain.
RequireValidShell               off
```

- Nous décommentons la ligne « **MaxClients** » :



- Nous décommentons les 2 lignes « **DisplayLogin** » et « **DisplayChdir** » :

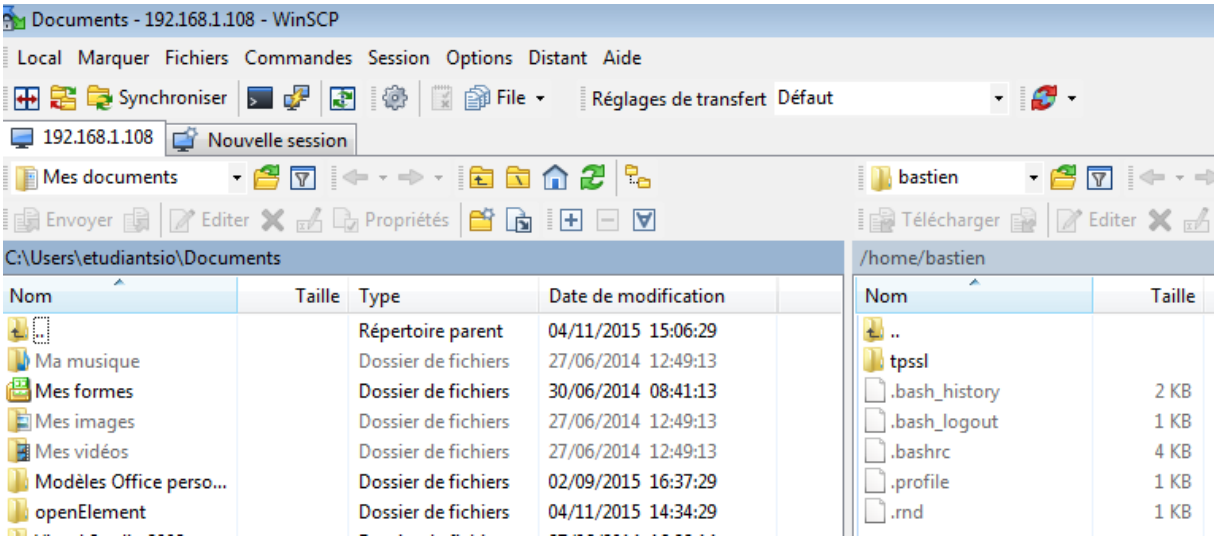


- Nous décommentons la partie entière « **<Anonymous>** » jusqu'à la fin :

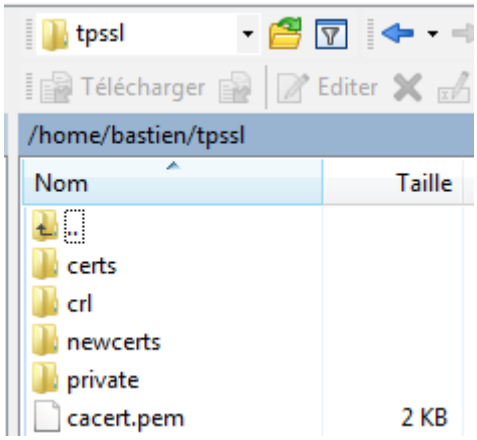
- Nous redémarrons le service « **proftpd.conf** » :

X) Vérification de connexion avec un client FTP

- Nous allons tester la connexion en tant qu'utilisateur via « **WinSCP** », nous nous nous connectons avec l'utilisateur « bastien » et nous visualisons la connexion au serveur :

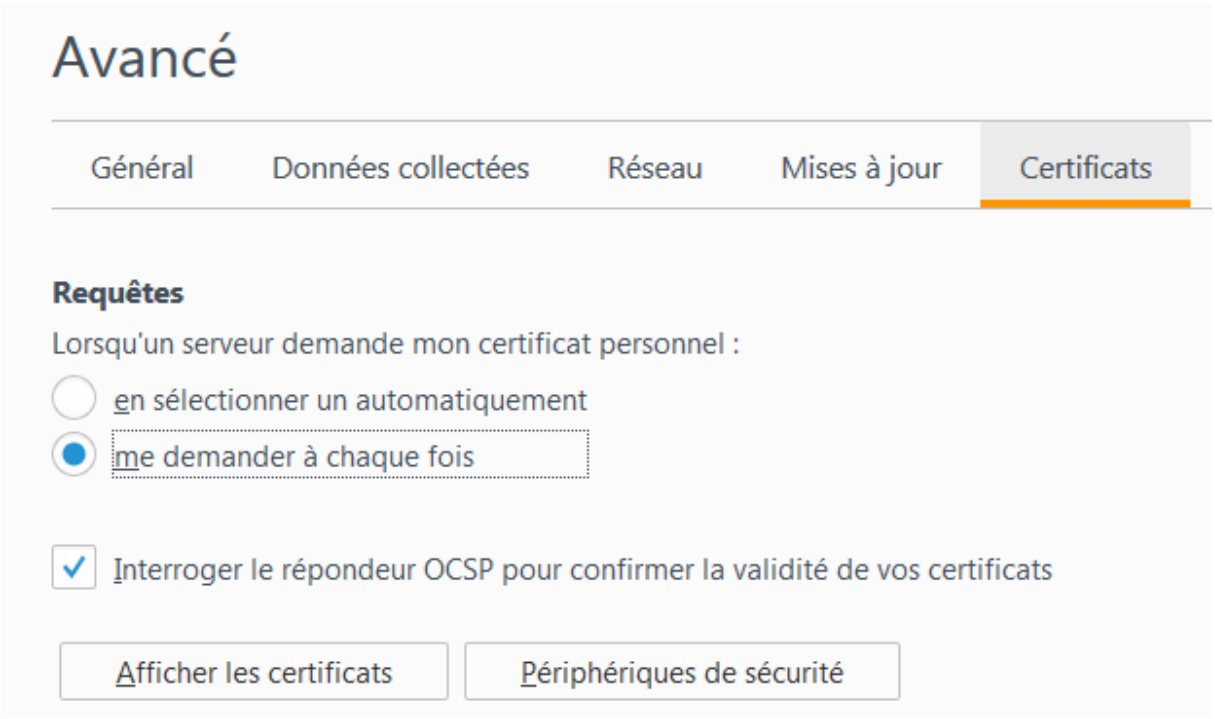


- Nous nous rendons dans le dossier « **tpssl** » créé précédemment :

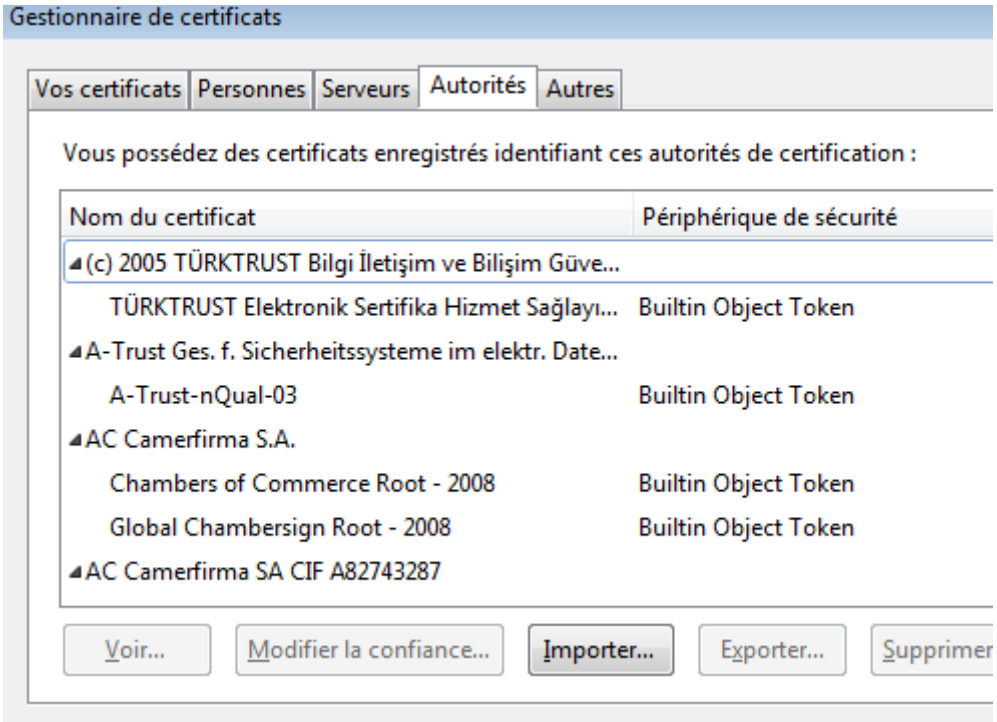


XI) Importation du certificat « cacert.pem » sur un navigateur

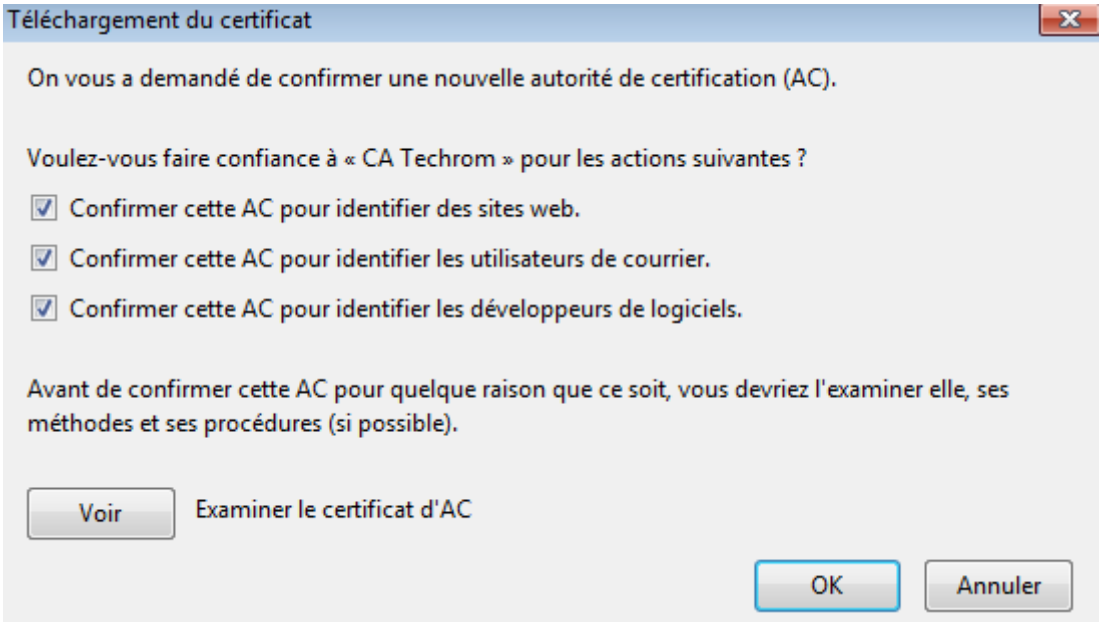
- Ensuite, nous nous rendons sur un navigateur Web (**Mozilla Firefox** par exemple), nous allons dans « **Options** », « **Avancé** », l'onglet « **Certificats** » et nous cliquons sur « **Afficher les certificats** » :



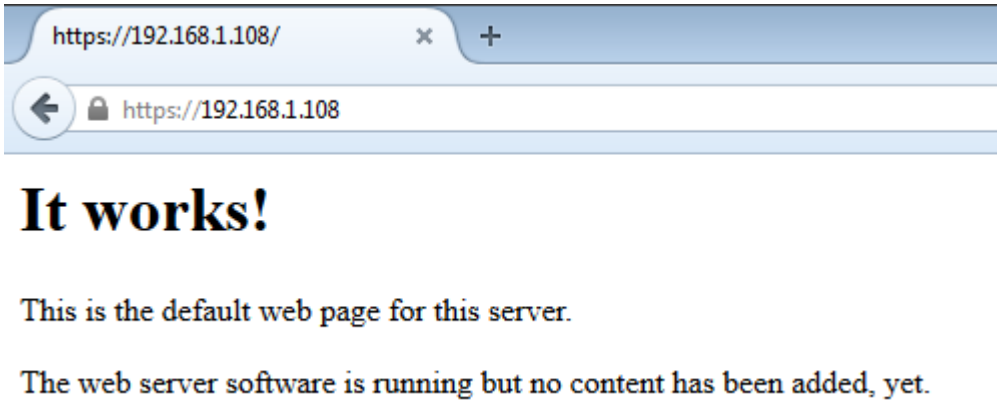
- Ensuite, nous cliquons sur l'onglet « **Autorités** » et le bouton « **Importer** » pour choisir et importer le certificat « **cacert.pem** » :



- Nous cochons les 3 cases et nous cliquons sur « **OK** » :



- Après avoir cliqué sur « **OK** », nous pouvons constater que cela continue de fonctionner :



XII) Configuration du nom DNS

- Pour configurer le nom **DNS**, nous éditons le fichier « **hosts** » qui se situe dans le répertoire « **/etc** » :

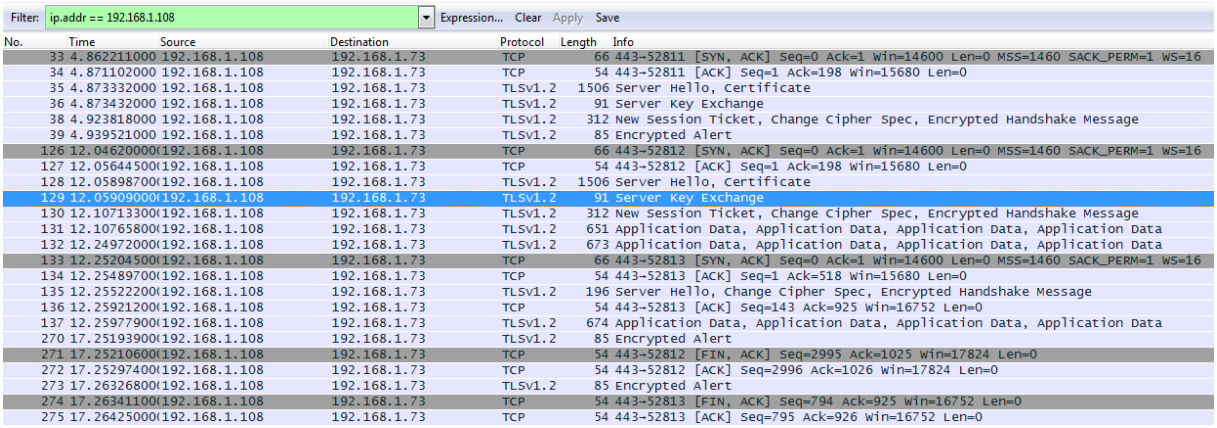
```
root@debian:~# nano /etc/hosts_
```

- Nous saisissons l'adresse IP du serveur **SSL** et le nom de l'organisation :

```
  GNU nano 2.2.6              Fichier : /etc/hosts

127.0.0.1       localhost
127.0.1.1       debian
192.168.1.108   techrom.fr_
```

XIII) Visualisation de la connexion sécurisée

- 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**.

XIV) Conclusion

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