

ETTORI Bastien	BTS SIO 2 ^{ème} 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

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

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 Secure Socket Layer) 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/Linux system
the exact distribution terms for each program.
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY,
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** » :

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

```
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 : Certificate Authority**).

```
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
```

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

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** :

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

```
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+KXcEMAOGCSqGSIb3DQEEBBQUAMIGSMQswCQYD
VQQGEwJGUjELMAKGA1UECAwCMTQxDALBgNVBACMBGNhZW4xEDAOBgNVBAoMB1Rl
Y2hyb20xZmFzAVBgNVBAsMD1NlcnZpY2UgcmlvZzZWF1HRMwEQYDVQQDDApDQSB1Z
cm9tMScwJQYJKoZIhvcNAQkBFhhYXN0aWVudG9yaUBnbWVpbC5jb20wHhcN
MTUxMTA5MDgzNjMzWWhcNMjUxMTA5MDgzNjMzWWhcNjUxMTA5MDgzNjMzWWhc
BgNVBAGMAjEOMQowCwYDVQQHDARjYVUuMRAwDgYDVQQKDAUZWnocm9tMRcwFQYD
VQQLEA5TZXJ2aWNlIHJlc2VhdTETMBEGA1UEAwKQ0EgVGVjaHJvbnTenMCUGCSqG
SIb3DQEJARYYYmFzdG11bi5ldHRvcmlAZ21haWwY29tMIIBIjANBgkqhkiG9w0B
AQEFAAOCAQ8AMIIBCgKCAQEAAu5CC/Gnh7XJL1gVpyLNxZqArNxeGQM0Eb9Ettgjs
a0by2AW5J6TWf0kVHMpBjMeD/hcED0oxczAQtd1Sp8rFC/2jet/PBU0XHX+ette8
MmFqdDZapngmIASK4mjbXe7oSk4anPPfrUcRHRN8jDEw23XFbw37WUcqrzbzhX7p6
w+w5y8WvygDqp9BMLu2z99A6Zki0Czu6D1vQDKT193Ubur9nLWtV9fVt50f0YfG1
GTMDoeh3VxVKHfP1j0sE07He4Xk9ufIQAGarxcDf50m1Ret64/10yT4A7Y/n4QHS
T3020osycYdvhBsea+T90WgnzVESmbhwZcjm9Vf9bRJBWIDAQABo1AwTjAdBgNV
HQ4EFgQUQn6Eip2D0/5mhWG0wz/UbD0tB9wwHwYDVR0jBBgwFoAUQn6Eip2D0/5m
hWG0wz/UbD0tB9wwDAYDVR0TBAAUwAwEB/zANBgkqhkiG9w0BAQUFAAOCAQEAYqpA
/ZAktXpKNmDU1xcr3NCphajI0WDXEI4jZ5X/ME4aQiwE4124akj1zIUR8CrNy63Q
9p4Ez3CA0xLYHJFKdr6KxMJYbULIzCPw0Byr0104aLVWc5Ioc0RPNAcEuqsjfp1s
jdiaKeKqNvfC/pfkithVWPEyEGQZJPT0WvyyMQC/0w7nMdacv0gcw9TaewlgSQx7
08fG5N3u1HAevRWx8A0ApaK1oSXB9VKm6hEKtNw/D1Dm22ko0IcMFNB9YATHGk00
YsF/27IM8tjz4FEeo25R72nf5CGt+z5812bIUxbZg+06B5cdIXo0A1hWo61gSTqc
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 :

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

```

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
DN. There are quite a few fields but you can leave some blank.
For some fields there will be a default value, and for others you must
enter an entry. 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  crt          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 :

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

```
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 default CA configuration file
#####
[ 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 **y** 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 :

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

```
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-clear.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-clear.pem** » dans le répertoire « **/etc/apache2/ssl** » :

```
bastien@debian:~/tpssl/private$ cp webkey-clear.pem /etc/apache2/ssl/webkey-clear.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 :

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

```
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:
server's fully qualified domain name, using 127.0.1.1 for ServerName
... waiting apache2: Could not reliably determine
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

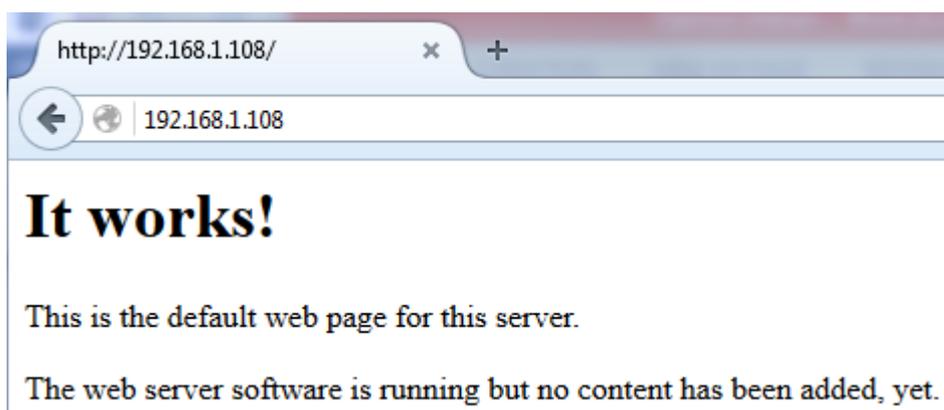
# A self-signed (snakeoil) certificate can be created by in
# the ssl-cert package. See
# /usr/share/doc/apache2.2-common/README.Debian.gz for more
# If both key and certificate are stored in the same file,
# SSLCertificateFile directive is needed.
SSLCertificateFile /etc/apache2/ssl/webcert.pem
SSLCertificateKeyFile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

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

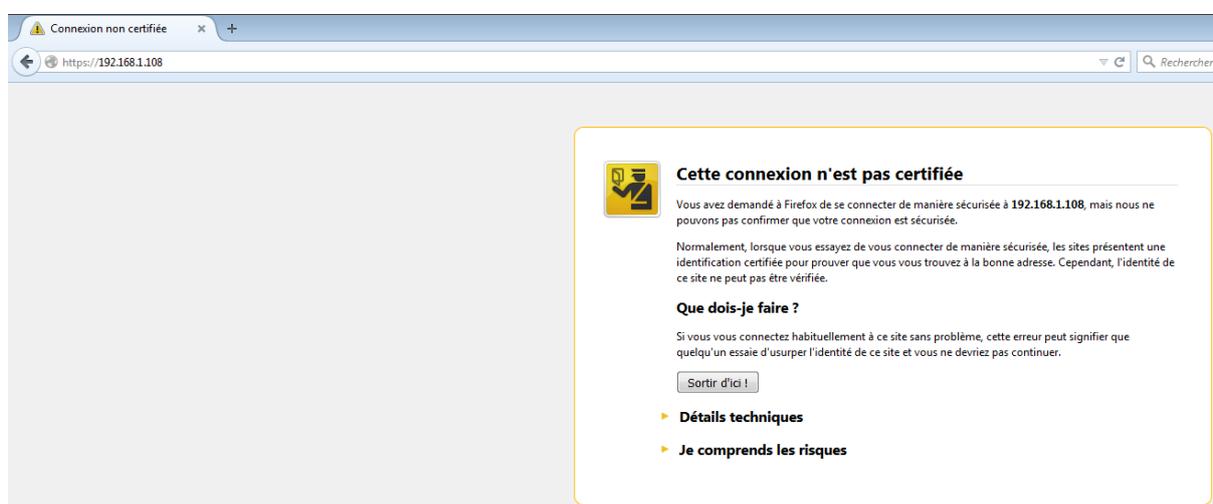
```
root@debian:/etc/apache2/ssl# a2ensite default-ssl
Enabling site default-ssl.
To activate the new configuration, you need to run
service apache2 reload
root@debian:/etc/apache2/ssl# service apache2 rest
[...] Restarting web server: apache2apache2: Coul
server's fully qualified domain name, using 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 :

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



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



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

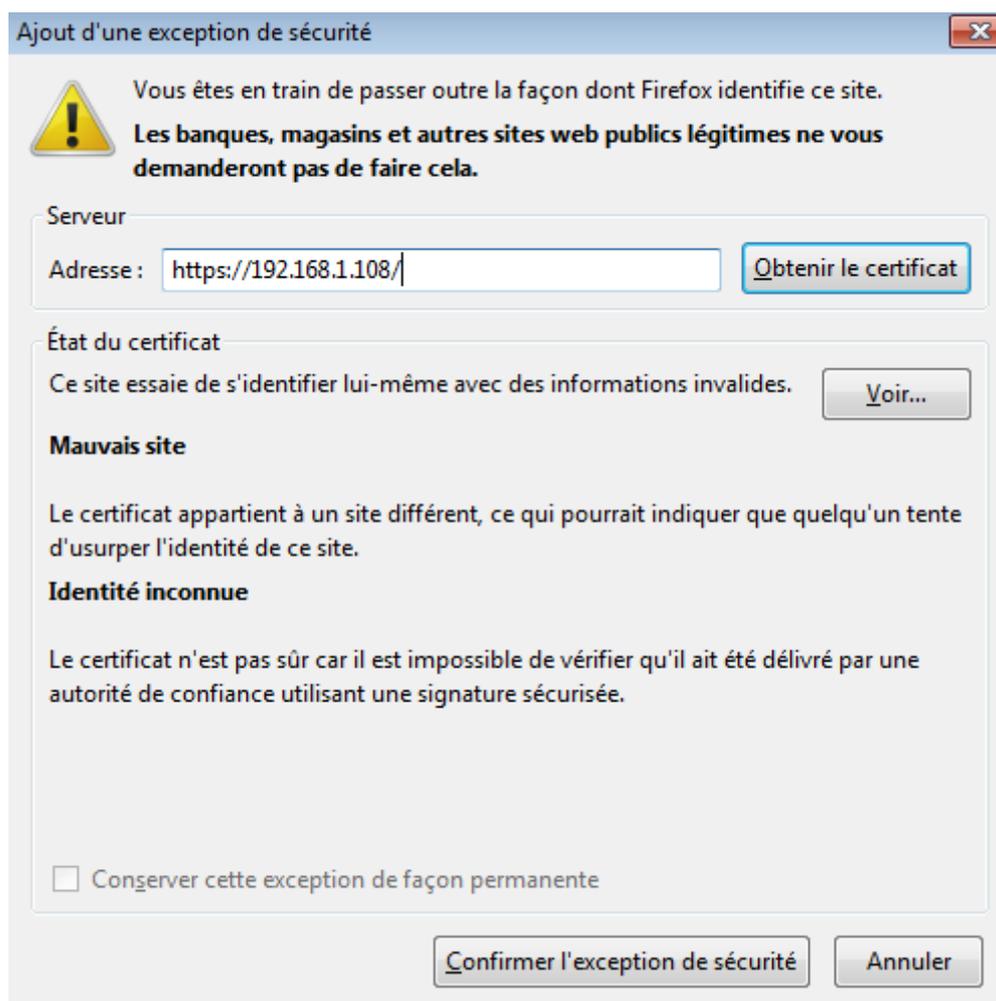
▼ **Je comprends les risques**

Si vous comprenez ce qui se passe, vous pouvez indiquer à Firefox de commencer à faire confiance à l'identification de ce site. **Même si vous avez confiance en ce site, cette erreur pourrait signifier que quelqu'un est en train de pirater votre connexion.**

N'ajoutez pas d'exception à moins que vous ne connaissiez une bonne raison pour laquelle ce site n'utilise pas d'identification certifiée.

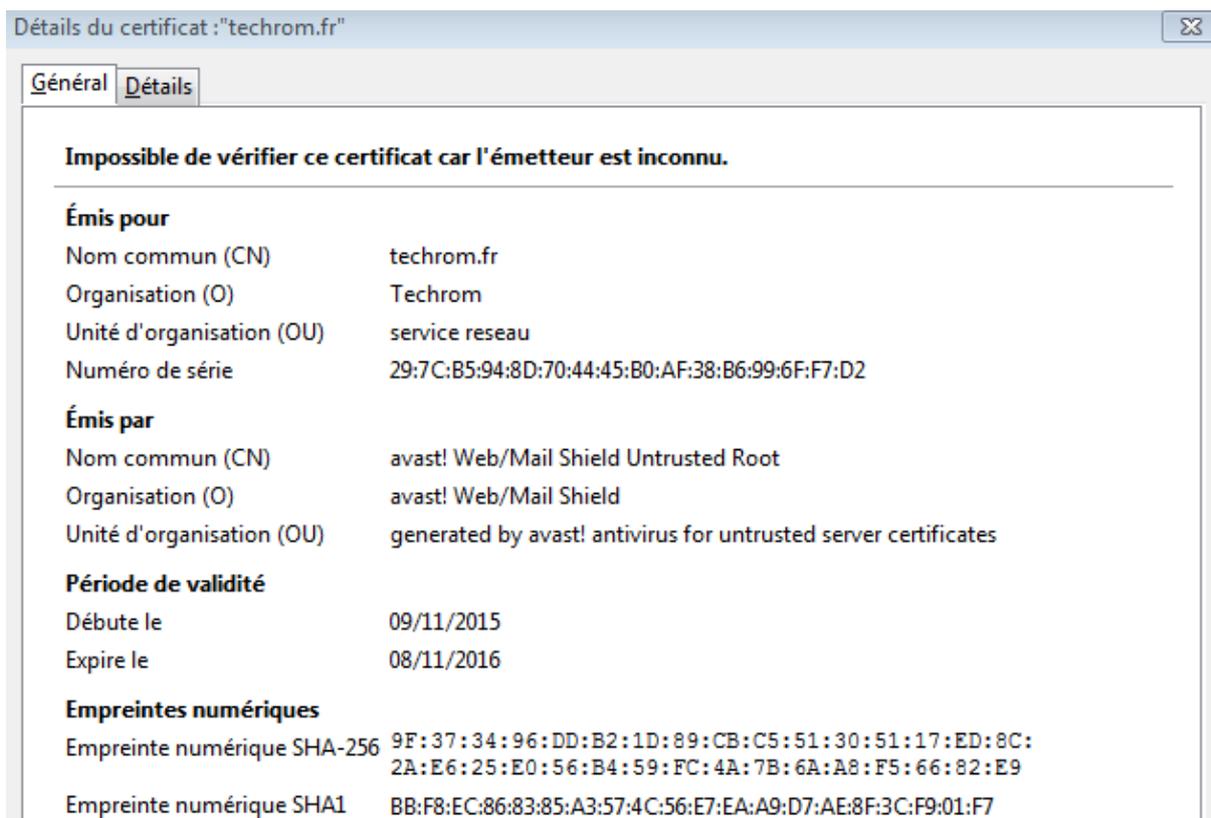
- Dans la zone « **Adresse** », nous renseignons l'adresse IP du serveur **SSL** et nous cliquons sur « **Obtenir le certificat** » :

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

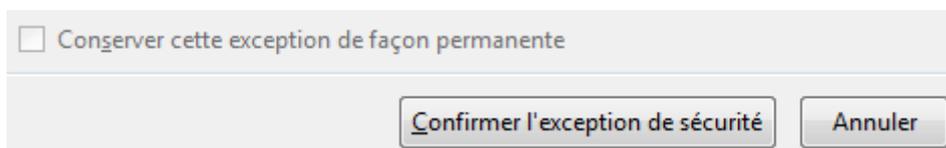


- Nous pouvons visualiser les détails du certificat « **techrom.fr** » :

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



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



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



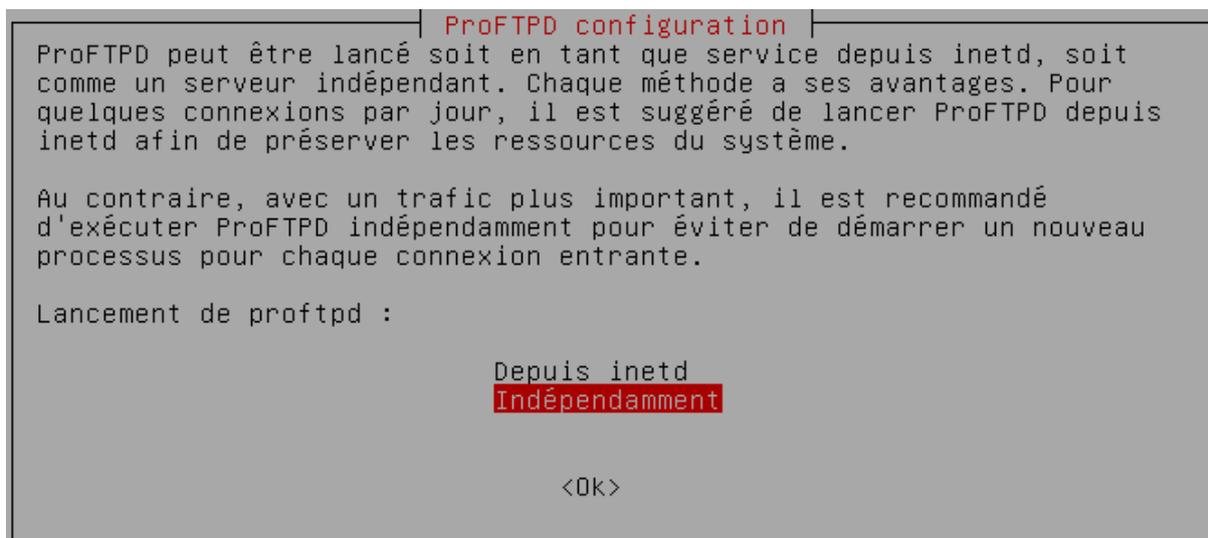
VIII) Installation du service FTP

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

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

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

- 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 les lignes « **</anonymous ~ftp>** » et « **MaxClients** » :

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

```

GNU nano 2.2.6          Fichier : proftpd.conf
# A basic anonymous configuration, no upload direc

<Anonymous ~ftp>
  User                ftp
  Group               nogroup
  # We want clients to be able to login with "ano
  UserAlias           anonymous ftp
  # Cosmetic changes, all files belongs to ftp us
  DirFakeUser  on ftp
  DirFakeGroup on ftp
#
  RequireValidShell  off
#
  # Limit the maximum number of anonymous logins
  MaxClients         10

```

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

```

# We want 'welcome.msg' displayed
# in each newly chdir'd directory
DisplayLogin           welc
DisplayChdir           .message

```

- Nous décommentons la dernière ligne « **<Anonymous>** » :

```

_</Anonymous>

```

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

```

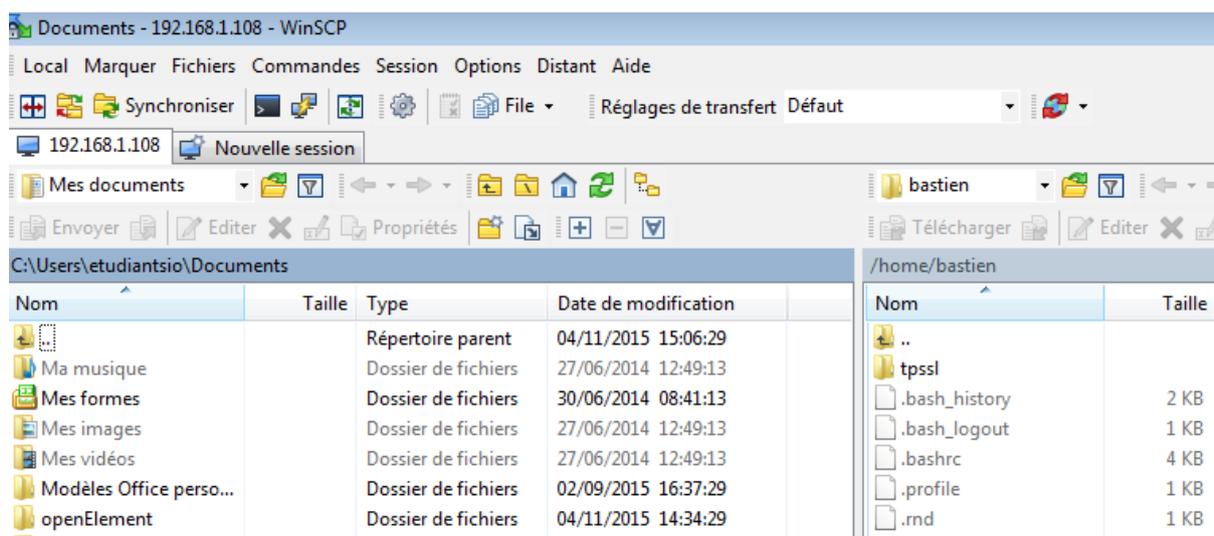
root@debian:/etc/proftpd# service proftpd stop
[ ok ] Stopping ftp server: proftpd.
root@debian:/etc/proftpd# service proftpd start
[...] Starting ftp server: proftpddebian proft
notice: unable to register 'memcache' SSL sessio
bled
. ok
root@debian:/etc/proftpd# _

```

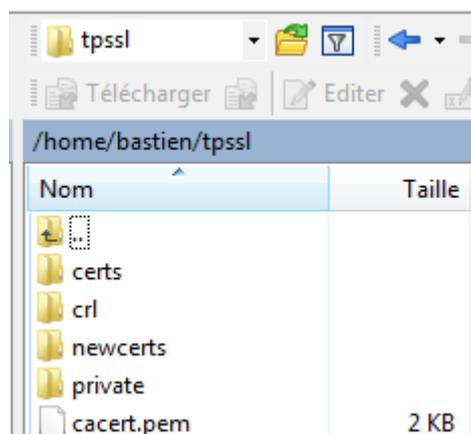
X) Vérification de connexion avec un client FTP

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

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



- 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** » :

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

Avancé

Général Données collectées Réseau Mises à jour **Certificats**

Requêtes

Lorsqu'un serveur demande mon certificat personnel :

en sélectionner un automatiquement

me demander à chaque fois

Interroger le répondeur OCSP pour confirmer la validité de vos certificats

[Afficher les certificats](#) [Périphériques de sécurité](#)

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

Gestionnaire de certificats

Vos certificats Personnes Serveurs **Autorités** Autres

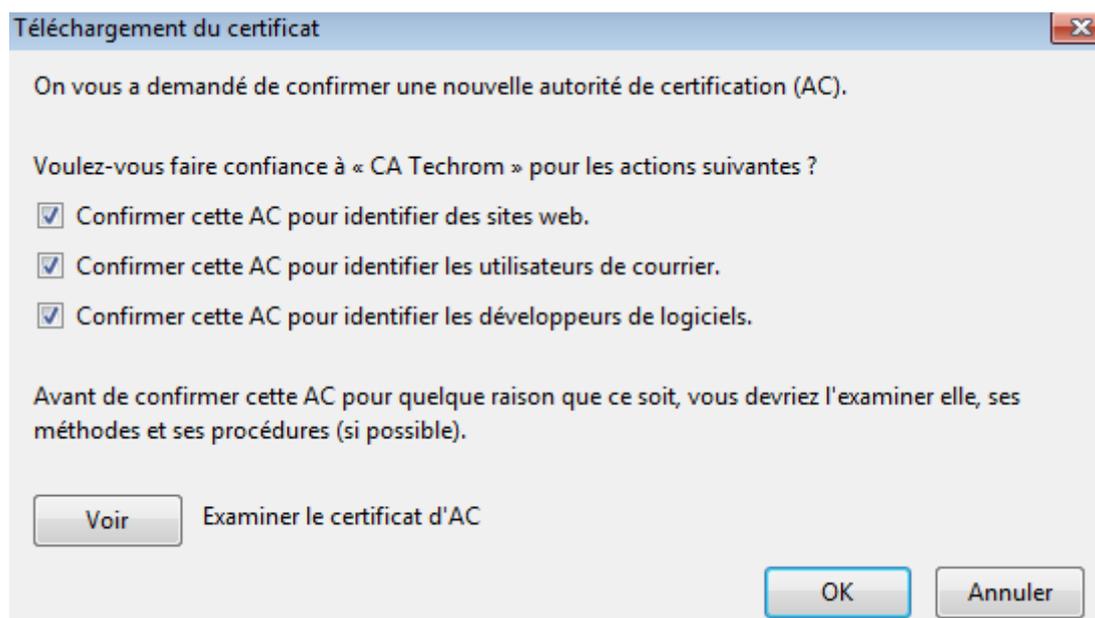
Vous possédez des certificats enregistrés identifiant ces autorités de certification :

Nom du certificat	Périphérique de sécurité
▲(c) 2005 TÜRKTRUST Bilgi İletişim ve Bilişim Güve...	
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayı...	Builtin Object Token
▲A-Trust Ges. f. Sicherheitssysteme im elektr. Date...	
A-Trust-nQual-03	Builtin Object Token
▲AC Camerfirma S.A.	
Chambers of Commerce Root - 2008	Builtin Object Token
Global Chambersign Root - 2008	Builtin Object Token
▲AC Camerfirma SA CIF A82743287	

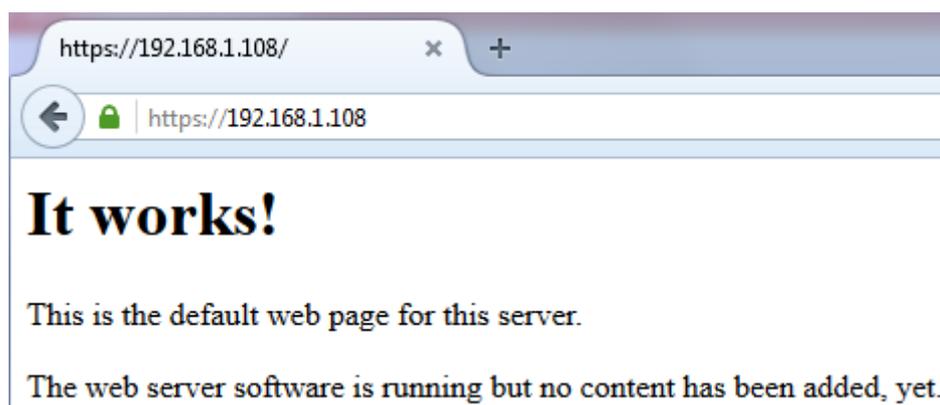
[Voir...](#) [Modifier la confiance...](#) **[Importer...](#)** [Exporter...](#) [Supprimer](#)

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

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



- 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_
```

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

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** :

No.	Time	Source	Destination	Protocol	Length	Info
33	4.862211000	192.168.1.108	192.168.1.73	TCP	66	443-52811 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=16
34	4.871102000	192.168.1.108	192.168.1.73	TCP	54	443-52811 [ACK] Seq=1 Ack=198 win=15680 Len=0
35	4.873332000	192.168.1.108	192.168.1.73	TLSv1.2	1506	Server Hello, Certificate
36	4.873432000	192.168.1.108	192.168.1.73	TLSv1.2	91	Server Key Exchange
38	4.923818000	192.168.1.108	192.168.1.73	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
39	4.939521000	192.168.1.108	192.168.1.73	TLSv1.2	85	Encrypted Alert
126	12.046200000	192.168.1.108	192.168.1.73	TCP	66	443-52812 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=16
127	12.056445000	192.168.1.108	192.168.1.73	TCP	54	443-52812 [ACK] Seq=1 Ack=198 win=15680 Len=0
128	12.058987000	192.168.1.108	192.168.1.73	TLSv1.2	1506	Server Hello, Certificate
129	12.059090000	192.168.1.108	192.168.1.73	TLSv1.2	91	Server Key Exchange
130	12.107133000	192.168.1.108	192.168.1.73	TLSv1.2	312	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
131	12.107658000	192.168.1.108	192.168.1.73	TLSv1.2	651	Application Data, Application Data, Application Data, Application Data
132	12.249720000	192.168.1.108	192.168.1.73	TLSv1.2	673	Application Data, Application Data, Application Data, Application Data
133	12.252045000	192.168.1.108	192.168.1.73	TCP	66	443-52813 [SYN, ACK] Seq=0 Ack=1 win=14600 Len=0 MSS=1460 SACK_PERM=1 WS=16
134	12.254897000	192.168.1.108	192.168.1.73	TCP	54	443-52813 [ACK] Seq=1 Ack=518 win=15680 Len=0
135	12.255222000	192.168.1.108	192.168.1.73	TLSv1.2	196	Server Hello, Change Cipher Spec, Encrypted Handshake Message
136	12.259212000	192.168.1.108	192.168.1.73	TCP	54	443-52813 [ACK] Seq=143 Ack=925 win=16752 Len=0
137	12.259779000	192.168.1.108	192.168.1.73	TLSv1.2	674	Application Data, Application Data, Application Data, Application Data
270	17.251939000	192.168.1.108	192.168.1.73	TLSv1.2	85	Encrypted Alert
271	17.252106000	192.168.1.108	192.168.1.73	TCP	54	443-52812 [FIN, ACK] Seq=2995 Ack=1025 win=17824 Len=0
272	17.252974000	192.168.1.108	192.168.1.73	TCP	54	443-52812 [ACK] Seq=2996 Ack=1026 win=17824 Len=0
273	17.263268000	192.168.1.108	192.168.1.73	TLSv1.2	85	Encrypted Alert
274	17.263411000	192.168.1.108	192.168.1.73	TCP	54	443-52813 [FIN, ACK] Seq=794 Ack=925 win=16752 Len=0
275	17.264250000	192.168.1.108	192.168.1.73	TCP	54	443-52813 [ACK] Seq=795 Ack=926 win=16752 Len=0

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.