

Travail pratique 2

Thème

Infrastructures à clés publiques

Réalisé par

- ADLA ilyes chiheb eddine

Sécuriser l'échange entre un client et un serveur Web Apache avec SSL :

Créer un espace de Publication Web Apache :

Tout d'abord on doit créer un espace de publication Web Apache on suivant les étapes suivantes :

1-Création d'un répertoire pour test « delta » dans le répertoire « /opt/lampp/htdocs » avec la commande : **sudo mkdir delta**

```
ipls@ipls-VirtualBox:/opt/lampp/htdocs$ sudo mkdir delta
[sudo] password for ipls:
ipls@ipls-VirtualBox:/opt/lampp/htdocs$ ls
applications.html  dvwa          img           submit.php    welcome.html
bitnami.css       favicon.ico   index.html    submit.php~   xampp
delta             guestbook.html  index.php     tmp.php
display.php       guestbook.html~ myadmin       webalizer
ipls@ipls-VirtualBox:/opt/lampp/htdocs$
```

2-Modification du fichier de configuration httpd.conf dans le répertoire « /opt/lampp/etc » avec la commande : **sudo nano httpd.conf**

```
ipls@ipls-VirtualBox: /opt/lampp/etc
GNU nano 2.2.6      Fichier : httpd.conf      Modifié

# below.
#
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
#DocumentRoot "/opt/lampp/htdocs"
DocumentRoot "/opt/lampp/htdocs/delta"
#Directory "/opt/lampp/htdocs"
<Directory "/opt/lampp/htdocs/delta">
#
# Possible values for the Options directive are "None", "All",
# or any combination of;
```

3- Création d'une page web **index.html** avec la commande : **sudo nano index.html**

```
ip1s@ip1s-VirtualBox: /opt/lampp/htdocs/delta
GNU nano 2.2.6      Fichier : index.html      Modifié

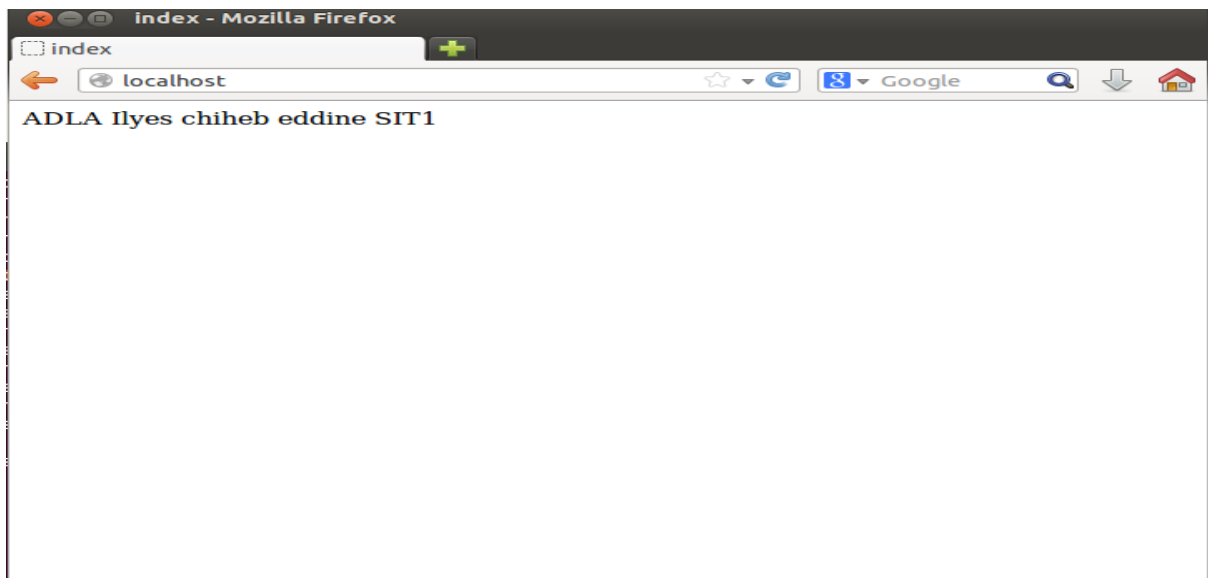
<html>
<head>
<meta charset="utf-8"/>

<title> index </title>
</head>
<body>
<p>ADLA Ilyes chiheb eddine SIT1</p>
</body>
</html>
```

4-Redémarrage du serveur Apache avec la commande :**sudo ./lampp restart**

```
ip1s@ip1s-VirtualBox:~$ cd /opt/lampp
ip1s@ip1s-VirtualBox:/opt/lampp$ sudo ./lampp restart
Restarting XAMPP for Linux 1.8.3-3...
XAMPP: Stopping Apache...ok.
XAMPP: Stopping MySQL...ok.
XAMPP: Starting Apache...ok.
XAMPP: Starting MySQL...ok.
ip1s@ip1s-VirtualBox:/opt/lampp$
```

5-test du fonctionnement du serveur avec le navigateur web



Créer un répertoire pour la zone sécurisé :

La 2 eme phase consiste à créer un répertoire pour la zone sécurisée avec les étapes suivantes :

1- création d'un répertoire « Secure » dans le chemin « **/opt/lampp/htdocs/delta** » avec la commande : **sudo mkdir secure**

2- modification du fichier de configuration « **httpd-ssl.conf** » qui se trouve dans le chemin « **opt/lampp/etc/extra** » avec la commande : **sudo nano httpd-ssl.conf**

```
ipls@ipls-VirtualBox: /opt/lampp/etc/extra
GNU nano 2.2.6      Fichier : httpd-ssl.conf      Modifié

#   Inter-Process Session Cache:
#   Configure the SSL Session Cache: First the mechanism
#   to use and second the expiring timeout (in seconds).
#SSLSessionCache      "dbm:/opt/lampp/logs/ssl_scache"
SSLSessionCache      "shmcb:/opt/lampp/logs/ssl_scache(512000)"
SSLSessionCacheTimeout 300

##
## SSL Virtual Host Context
##

<VirtualHost _default_:443>

#   General setup for the virtual host
#DocumentRoot "/opt/lampp/htdocs"
DocumentRoot "/opt/lampp/htdocs/delta/secure"
ServerName localhost:443
#ServerName localhost:443

^G Aide      ^O Écrire    ^R Lire fich.^Y Page préc.^K Couper      ^C Pos. cur.
^X Quitter   ^J Justifier ^W Chercher  ^V Page suiv.^U Coller     ^T Orthograp.
```

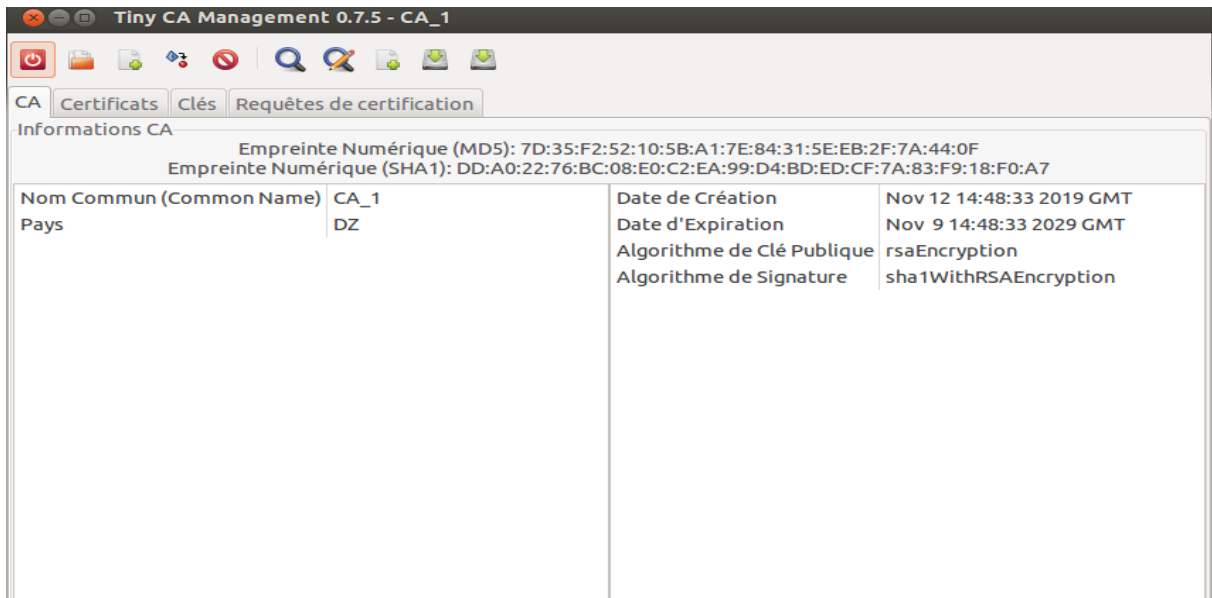
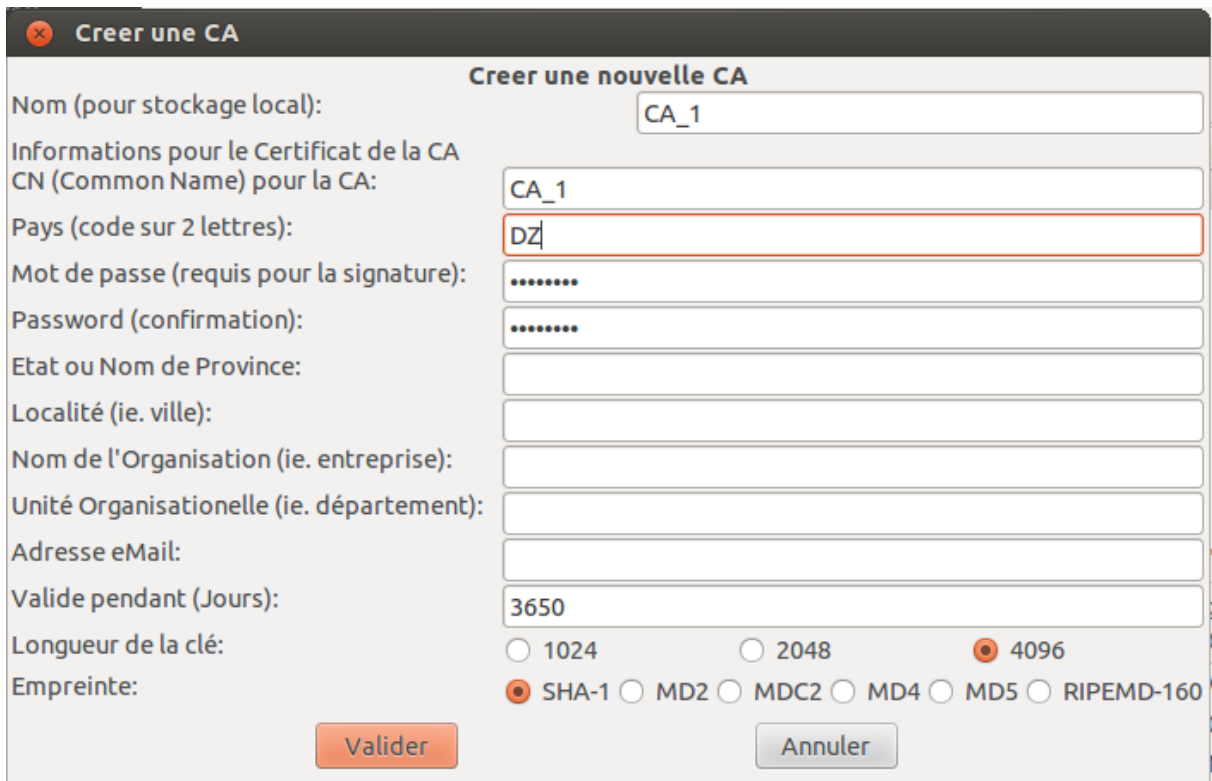
3-Création d'une page « index.html » dans le répertoire
« **/opt/lampp/htdocs/delta/secure** »

```
ipls@ipls-VirtualBox:/opt/lampp/etc/extra$ sudo nano httpd-ssl.conf
ipls@ipls-VirtualBox:/opt/lampp/etc/extra$ cd /opt/lampp/htdocs/delta/secure
ipls@ipls-VirtualBox:/opt/lampp/htdocs/delta/secure$ sudo nano index.html
ipls@ipls-VirtualBox:/opt/lampp/htdocs/delta/secure$ more index.html
<html>
<head>
<meta charset="utf-8" />
<title> index secure </title>
</head>
<body>
<p> page secure </p>
</body>
</html>
ipls@ipls-VirtualBox:/opt/lampp/htdocs/delta/secure$
```

Créer les certificats et les clés pour la CA et le Serveur Web :

La 3 eme phase est basée sur la création des certificats et les clés pour la CA et le serveur WEB on utilisant tinyCA . tinyCA est un utilitaire graphique permettant de créer des certificats et qui se base sur le système de cryptographie OpenSSL , le TinyCa est déjà installé, pour l'ouvrir on doit juste taper la commande tinyca2. Pour assurer cette phase de réalisation on doit suivre les étapes suivantes :

1-Création de l'autorité racine (la clé privée et certificat du CA)



Nom Commun (Common Name)	CA_1	Date de Création	Nov 12 14:48:33 2019 GMT
Pays	DZ	Date d'Expiration	Nov 9 14:48:33 2029 GMT
		Algorithme de Clé Publique	rsaEncryption
		Algorithme de Signature	sha1WithRSAEncryption

2- Création des sous répertoires de delta pour le stockage des certificats et les clés de la CA et du serveur avec les commendes suivantes :

```
ipls@ipls-VirtualBox:~$ cd /opt/lampp/etc
ipls@ipls-VirtualBox:/opt/lampp/etc$ ls
extra          magic          pear.conf      ssl.crt
freetds.conf   mime.types    php.ini        ssl.key
httpd.conf     my.cnf        php.ini-pre1.7.2  webalizer.conf
httpd.conf~    openssl.cnf   php.ini-pre1.7.2~ webalizer.conf.sample
httpd.conf.bak original       pool.conf      xampp
locales.conf   proftpd.conf

ipls@ipls-VirtualBox:/opt/lampp/etc$ mkdir delta
mkdir: impossible de créer le répertoire «delta»: Permission non accordée
ipls@ipls-VirtualBox:/opt/lampp/etc$ sudo mkdir delta
[sudo] password for ipls:
ipls@ipls-VirtualBox:/opt/lampp/etc$ cd delta
ipls@ipls-VirtualBox:/opt/lampp/etc/delta$ sudo mkdir certifs
ipls@ipls-VirtualBox:/opt/lampp/etc/delta$ sudo mkdir cles
ipls@ipls-VirtualBox:/opt/lampp/etc/delta$ ls
certifs  cles
ipls@ipls-VirtualBox:/opt/lampp/etc/delta$
```

3- Création du certificat du serveur on utilisant TinyCA pour générer une clé privée du serveur plus un certificat signé par l'autorité de confiance, la procédure de création d'un certificat se fait en deux étapes :

- Création de requête de signature (certificat)
- Signature de la requête
- Création de la requête de signature : on sélectionnant « **Requêtes de certifications** » dans l'onglet principal on peut créer une requête on spécifiant « **localhost** » comme un nom de cette requête.

Créer une nouvelle requête de certificat

Nom commun (ie. votre Nom, votre adresse eMail ou le Nom du serveur): localhost

Adresse eMail:

Mot de passe (protrège votre Clé privée):

Password (confirmation):

Pays (code sur 2 lettres): DZ

Etat ou Nom de Province:

Localité (ie. ville):

Nom de l'Organisation (ie. entreprise):

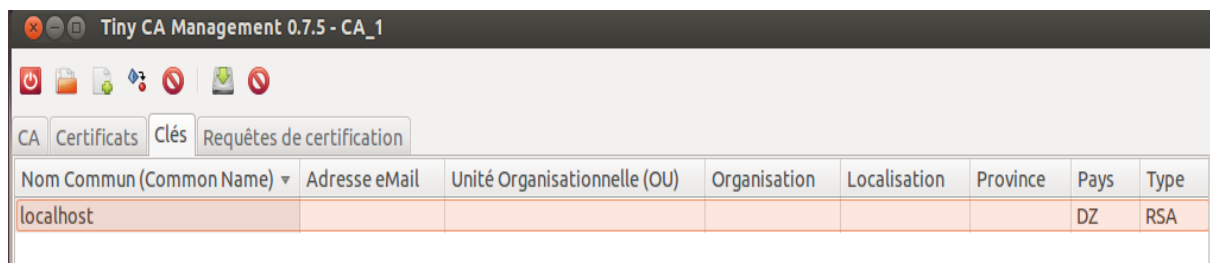
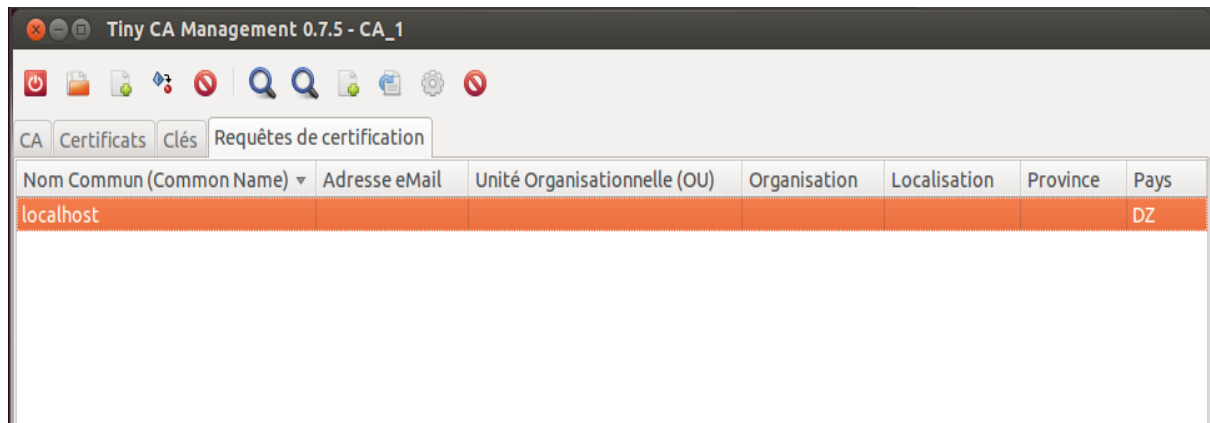
Unité Organisationelle (ie. département):

Longueur de la clé: ☒ 4096 ☐ 1024 ☐ 2048

Empreinte: ☒ SHA-1 ☐ MD2 ☐ MDC2 ☐ MD4 ☐ MD5 ☐ RIPEMD-160

Algorithme: ☒ RSA ☐ DSA

Valider Annuler

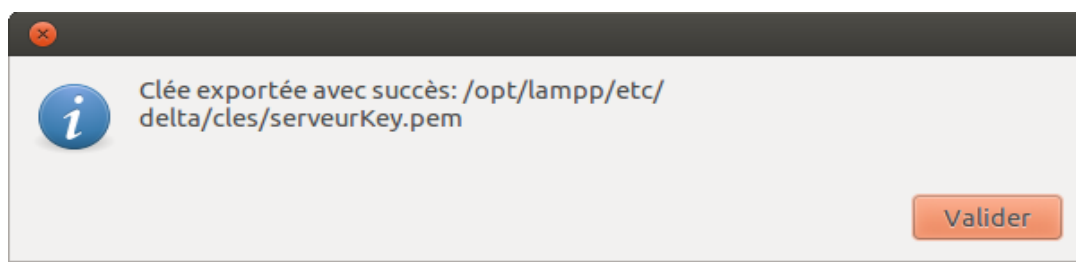
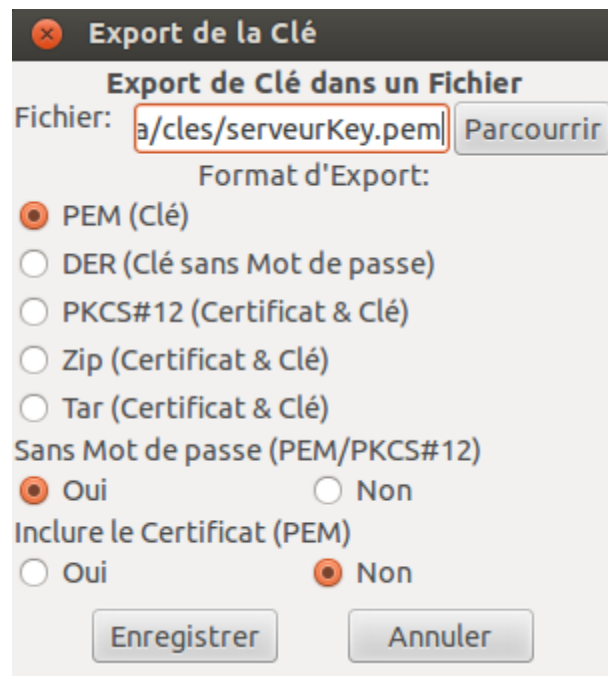


- Signature de la requête : c'est grâce à l'autorité qu'on va signer la requête précédente : clique droit sur la requête => signer la requête



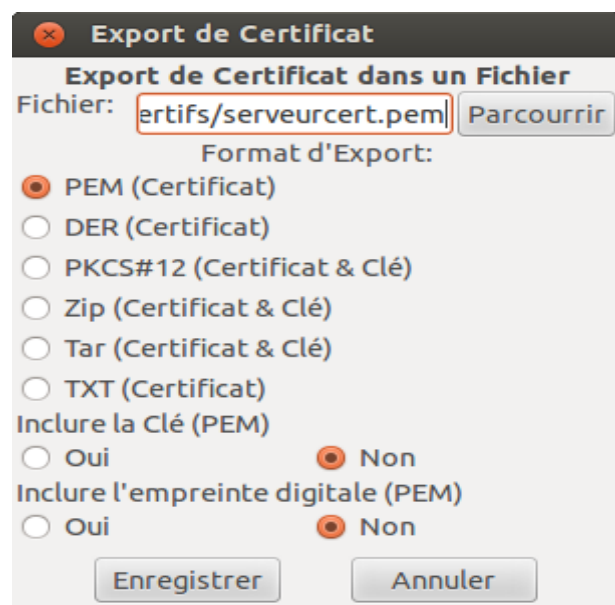
Exportation de la clé privée du serveur : on exporte donc la clé privée du serveur Web dans le fichier **/opt/lampp/etc/delta/cles/serveurkey.pem** .

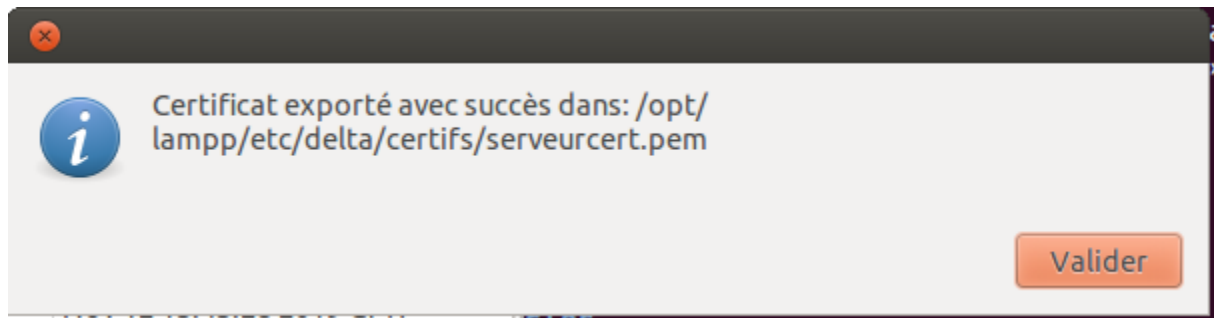
On sélectionnant « clés » dans l'onglet principal => clic droit => export de la clé



Exportation du certificat du serveur : on exporte donc le certificat du serveur Web dans le fichier **/opt/lampp/etc/delta/certs/serveurcert.pem**

On sélectionnant « certificats » dans l'onglet principal => clic droit => export de certificat





4-Modification du fichier de configuration « **httpd-ssl.config** » avec la commande :

sudo nano httpd-ssl.config

```
# Some ECC cipher suites (http://www.ietf.org/rfc/rfc4492.txt)
# require an ECC certificate which can also be configured in
# parallel.
#SSLCertificateFile "/opt/lampp/etc/ssl.crt/server.crt"
SSLCertificateFile "opt/lampp/etc/delta/certifs/serveurcert.pem"
#SSLCertificateFile "/opt/lampp/etc/server-dsa.crt"
#SSLCertificateFile "/opt/lampp/etc/server-ecc.crt"
```

```
ipls@ipls-VirtualBox: /opt/lampp/etc/extra
# directive to point at the key file.  Keep in mind that if
# you've both a RSA and a DSA private key you can configure
# both in parallel (to also allow the use of DSA ciphers, etc.)
# ECC keys, when in use, can also be configured in parallel
#SSLCertificateKeyFile "/opt/lampp/etc/ssl.key/server.key"
SSLCertificateKeyFile "/opt/lampp/etc/delta/cles/serveurKey.pem"
#SSLCertificateKeyFile "/opt/lampp/etc/server-dsa.key"
#SSLCertificateKeyFile "/opt/lampp/etc/server-ecc.key"
```

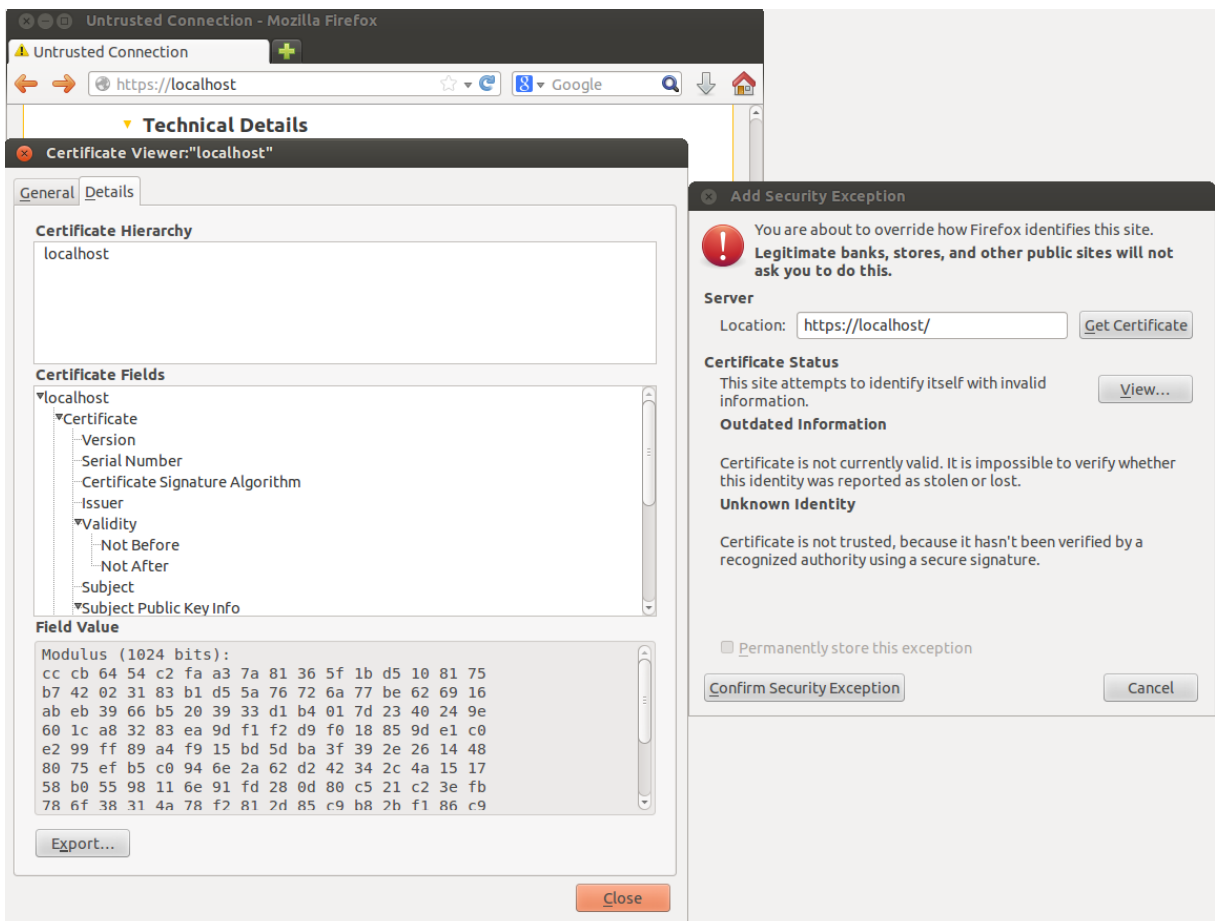
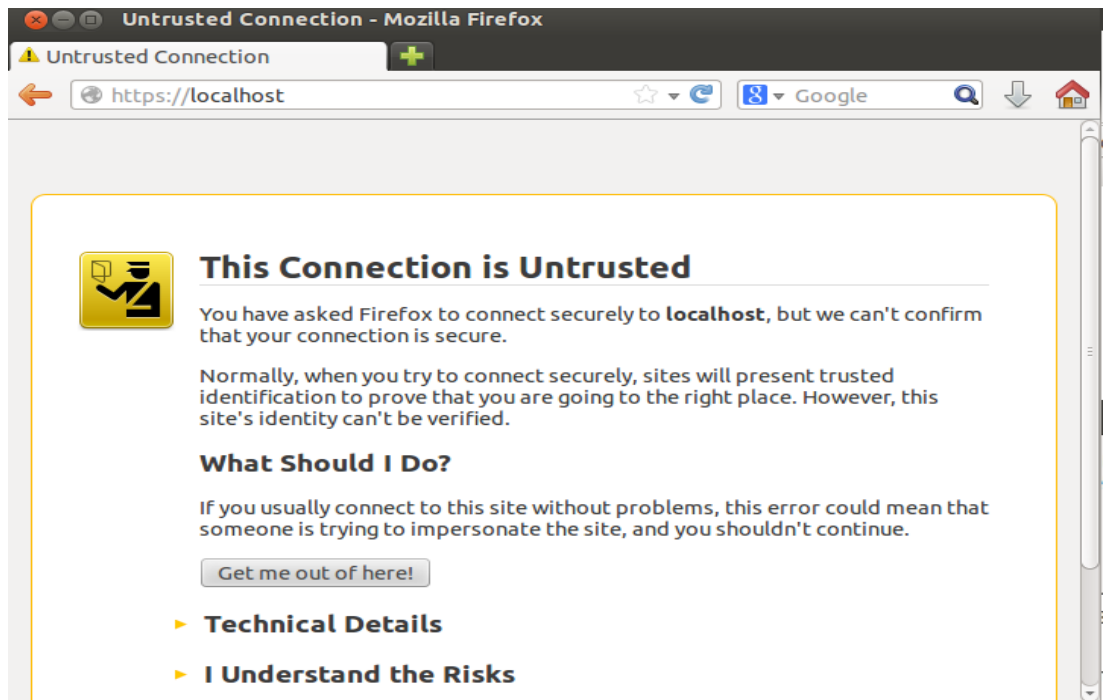
Les tests :

la phase des tests est caractérisée avec les étapes suivantes :

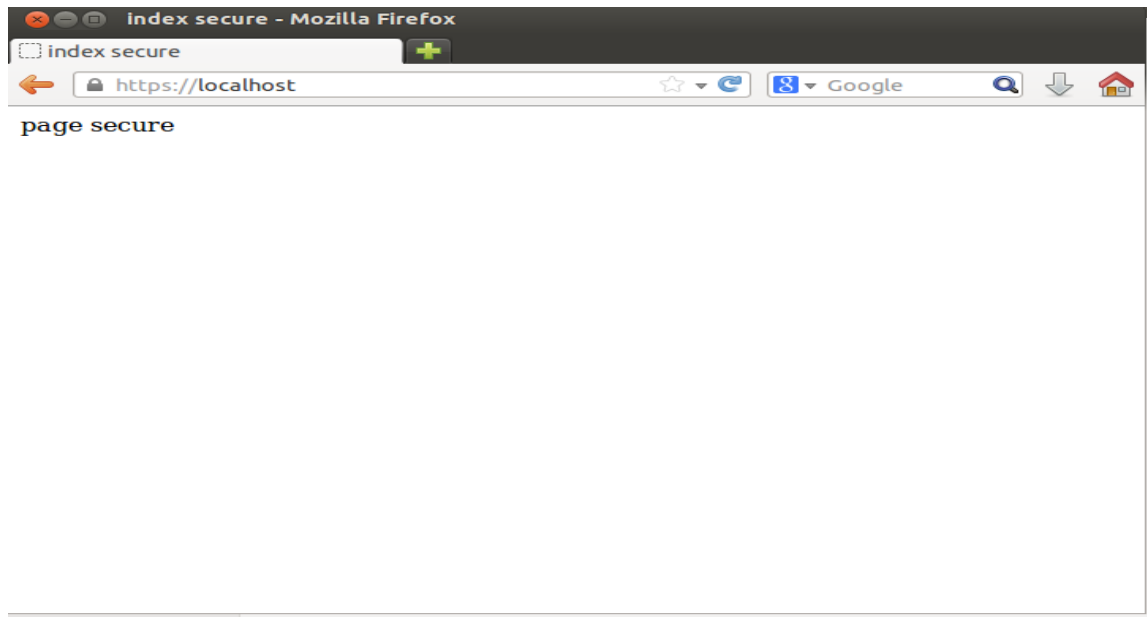
1. Relancement du Apache avec la commande : **sudo ./lampp restart**

```
ipls@ipls-VirtualBox:~$ cd /opt/lampp
ipls@ipls-VirtualBox:/opt/lampp$ sudo ./lampp restart
Restarting XAMPP for Linux 1.8.3-3...
XAMPP: Stopping Apache...fail.
apachectl returned 1.
XAMPP: Stopping MySQL...ok.
XAMPP: Starting Apache...already running.
XAMPP: Starting MySQL...ok.
ipls@ipls-VirtualBox:/opt/lampp$
```

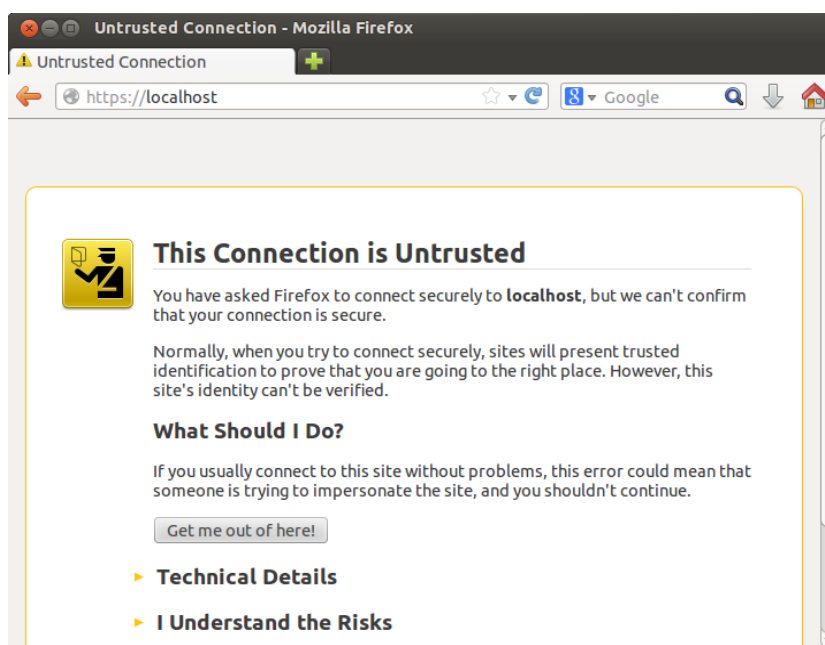
2. Test du serveur avec le navigateur : la page nous informe que la connection n'est pas securisée , pour afficher la page on doit confirmer une exception de securité (temporairement)



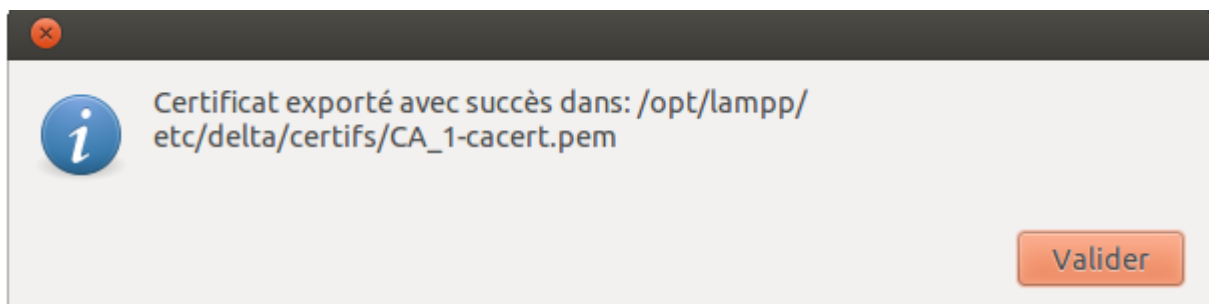
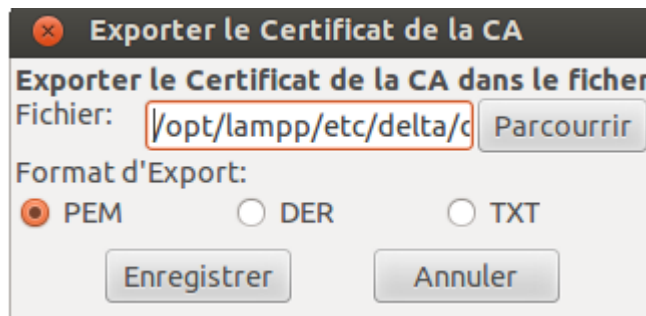
Après l'acceptation de l'exception, le navigateur affiche la page suivante :



Mais si on relance le navigateur on peut voir que le certificat n'est pas accepté automatiquement et l'alerte s'affiche de nouveau. L'explication de ce problème est que la partie tierce (CA) n'est pas une autorité de confiance pour le navigateur (client).

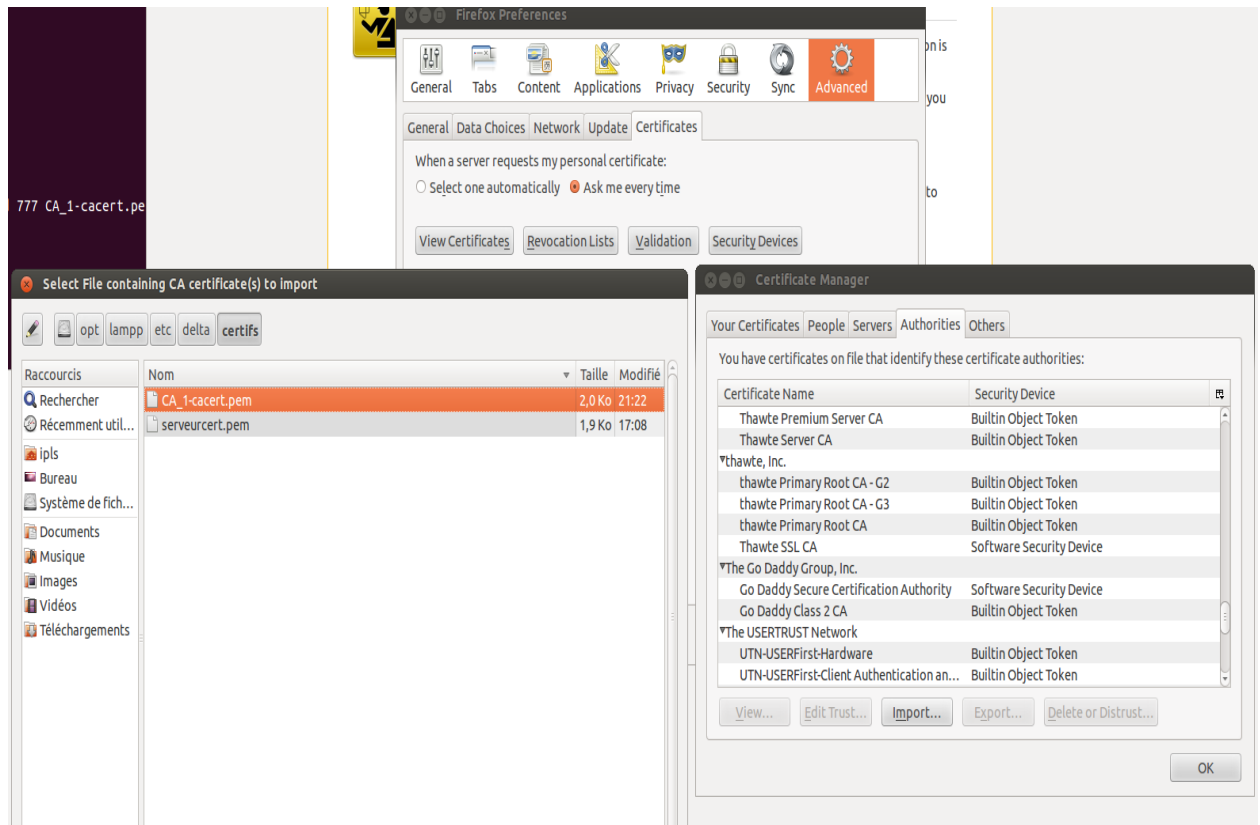


Pour régler ce problème on doit exporter la certificat de l'autorité (CA) dans le serveur « **/opt/lampp/etc/delta/certifs** » et l'inclure dans le navigateur.



```
ipls@ipls-VirtualBox:/opt/lampp/etc/delta/certifs$ ls
CA_1-cacert.pem  serveur-cert.pem
ipls@ipls-VirtualBox:/opt/lampp/etc/delta/certifs$
```

Pour inclure le certificat de l'autorité , on doit y'accéder au preferences de FireFox ,puis cliquer sur « View certificates » pour importer la certificat de CA

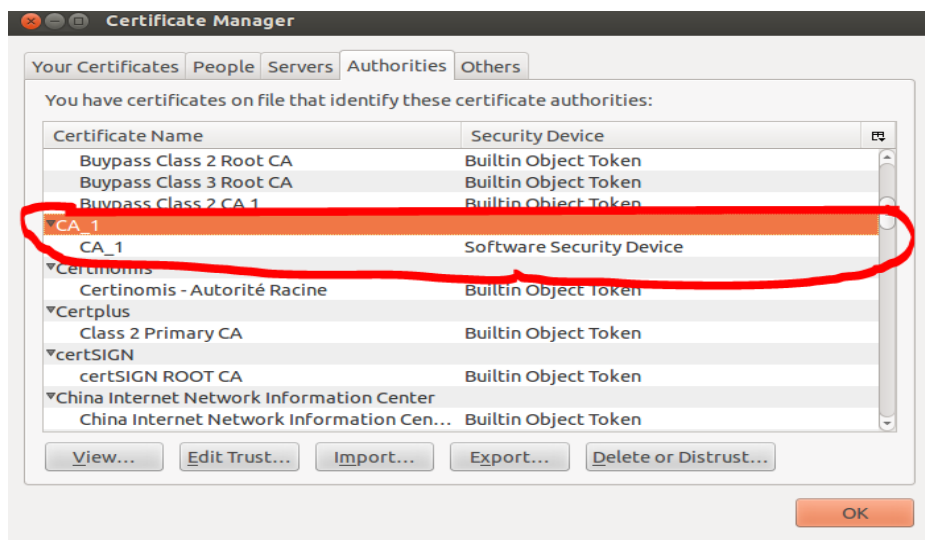
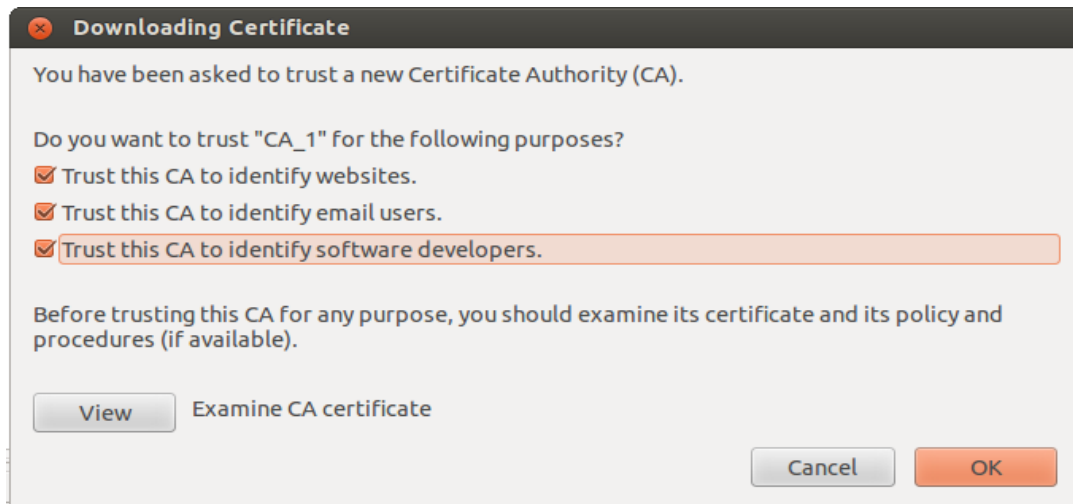


Mais malheureusement toutes les fichiers créés dans le type de répertoire comme « /opt » ou autres « /etc , /bin » ne sont disponibles que en mode lecture seul, ce qui explique l'utilisation du mode super utilisateur/admin « sudo » dans la partie précédente. donc la seule solution c'est de changer le mode d'accès à ce fichier « CA_1-cacert.pem » pour être disponible en lecture/écriture

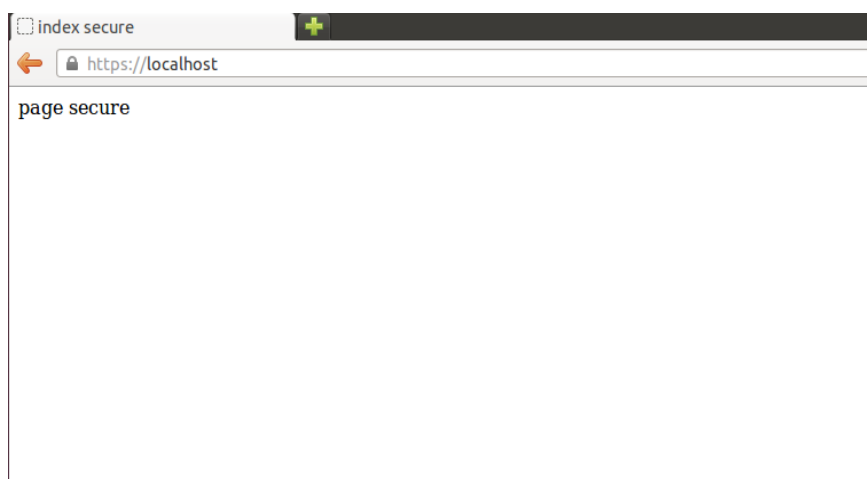
```

ipls@ipls-VirtualBox:/opt/lampp/etc/delta/certifs$ ls
CA_1-cacert.pem  serveurcert.pem
ipls@ipls-VirtualBox:/opt/lampp/etc/delta/certifs$ ls -la
total 16
drwxr-xr-x 2 root root 4096 nov. 12 21:24 .
drwxr-xr-x 4 root root 4096 nov. 12 15:55 ..
-rw----- 1 root root 2017 nov. 12 21:22 CA_1-cacert.pem
-rw----- 1 root root 1980 nov. 12 17:08 serveurcert.pem
ipls@ipls-VirtualBox:/opt/lampp/etc/delta/certifs$ sudo chmod 777 CA_1-cacert.pem
ipls@ipls-VirtualBox:/opt/lampp/etc/delta/certifs$ ls -la
total 16
drwxr-xr-x 2 root root 4096 nov. 12 21:24 .
drwxr-xr-x 4 root root 4096 nov. 12 15:55 ..
-rwxrwxrwx 1 root root 2017 nov. 12 21:22 CA_1-cacert.pem
-rw----- 1 root root 1980 nov. 12 17:08 serveurcert.pem
ipls@ipls-VirtualBox:/opt/lampp/etc/delta/certifs$

```



Et maintenant que le navigateur dispose du certificat de l'autorité (CA_1), il peut s'assurer que le certificat du serveur est signé par une autorité de confiance (CA_1), donc le navigateur peut afficher la page « index.html » de **secure**.



Analyse et comparaison des échanges :

5 eme phase concerne l'analyse des échanges.

Echange HTTP:

No.	Time	Source	Destination	Protocol	Length	Info
1146	23909.02563	127.0.0.1	127.0.0.1	TCP	74	http > 47979 [SYN, ACK] Seq=0 Ack=1 Win=43690 Len=0 MSS=65495 SACK_PERM=1 TSval=11678889 TSecr=11678889
1147	23909.02564	127.0.0.1	127.0.0.1	TCP	66	47979 > http [ACK] Seq=1 Ack=1 Win=43776 Len=0 TSval=11678889 TSecr=11678889
1148	23909.02643	127.0.0.1	127.0.0.1	HTTP	351	GET / HTTP/1.1
1149	23909.02646	127.0.0.1	127.0.0.1	TCP	66	http > 47979 [ACK] Seq=1 Ack=286 Win=44800 Len=0 TSval=11678890 TSecr=11678890
1150	23909.03446	127.0.0.1	127.0.0.1	HTTP	534	HTTP/1.1 200 OK (text/html)
1151	23909.03453	127.0.0.1	127.0.0.1	TCP	66	47979 > http [ACK] Seq=286 Ack=469 Win=44800 Len=0 TSval=11678892 TSecr=11678892
1152	23909.10715	127.0.0.1	127.0.0.1	HTTP	332	GET /favicon.ico HTTP/1.1
1153	23909.10850	127.0.0.1	127.0.0.1	TCP	668	[TCP segment of a reassembled PDU]
1154	23909.10850	127.0.0.1	127.0.0.1	TCP	66	47979 > http [ACK] Seq=552 Ack=1071 Win=45000 Len=0 TSval=11679010 TSecr=11679010

Options: (12 bytes)

- No-Operation (NOP)
- No-Operation (NOP)
- Timestamps: TSval 11678892, TSecr 11678890

[SEQ/ACK analysis]

[Bytes in flight: 468]

Hypertext Transfer Protocol

HTTP/1.1 200 OK\r\n

[Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]

Request Version: HTTP/1.1

Status Code: 200

Response Phrase: OK

On peut remarquer que l'échange n'est pas sécurisé entre les deux parties de la communication ce qui explique le fait qu'aucun protocole de sécurité n'est utilisé dans le HTTP.

On consultant la partie **Hypertext Transfer Protocol** on remarque que le fichier text « **text/html** » n'est pas crypté et son code source est disponible en clair.

```
▼ Hypertext Transfer Protocol
▼ HTTP/1.1 200 OK\r\n
  [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
    Request Version: HTTP/1.1
    Status Code: 200
    Response Phrase: OK
    Date: Wed, 13 Nov 2019 19:03:30 GMT\r\n
    Server: Apache/2.4.7 (Unix) OpenSSL/1.0.1f PHP/5.5.9 mod_perl/2.0.8-dev Perl/v5.16.3\r\n
    Last-Modified: Tue, 12 Nov 2019 13:25:53 GMT\r\n
    ETag: "82-5972630068bc1"\r\n
    Accept-Ranges: bytes\r\n
    ► Content-Length: 130\r\n
    Keep-Alive: timeout=5, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html\r\n
    \r\n
▼ Line-based text data: text/html
  <html>\n
  <head>\n
  <meta charset="utf-8"/>\n
  \n
  <title> index </title>\n
  </head>\n
  <body>\n
  <p>ADLA Ilyes chiheb eddine SIT1</p>\n
  </body>\n
  </html>\n
```

1- Creation du certificat du client : on utilisant TinyCA on peut generer la clé et le certificat du client dans les deux repertoire « **/opt/lampp/etc/delta/cles/** » et « **/opt/lampp/etc/delta/certifs/** »

Créer Requête

Créer une nouvelle requête de certificat

Nom commun (ie. votre Nom, votre adresse eMail ou le Nom du serveur):

Adresse eMail:

Mot de passe (protège votre Clé privée):

Password (confirmation):

Pays (code sur 2 lettres):

Etat ou Nom de Province:

Localité (ie. ville):

Nom de l'Organisation (ie. entreprise):

Unité Organisationnelle (ie. département):

Longueur de la clé: ☒ 4096 ☐ 1024 ☐ 2048

Empreinte: ☒ SHA-1 ☐ MD2 ☐ MDC2 ☐ MD4 ☐ MD5 ☐ RIPEMD-160

Algorithme: ☒ RSA ☐ DSA

Tiny CA Management 0.7.5 - CA_1

CA Preferences Aide

CA Certificats Clés Requetes de certification

Nom Commun (Common Name)	Adresse eMail	Unité Organisationnelle (OU)	Organisation	Localisation	Provi
client					
localhost					

Exportation des Cles dans les repertoires mentioné precedament :

Export de la Clé

Export de Clé dans un Fichier

Fichier:

Format d'Export:

☒ PEM (Clé)

☐ DER (Clé sans Mot de passe)

☐ PKCS#12 (Certificat & Clé)

☐ Zip (Certificat & Clé)

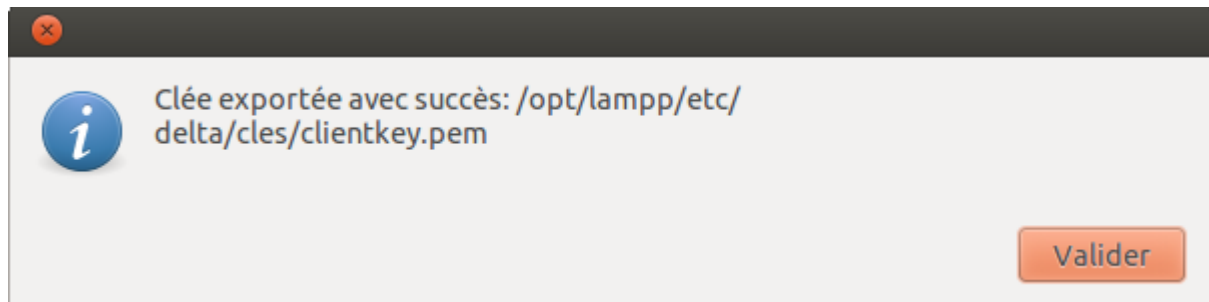
☐ Tar (Certificat & Clé)

Sans Mot de passe (PEM/PKCS#12)

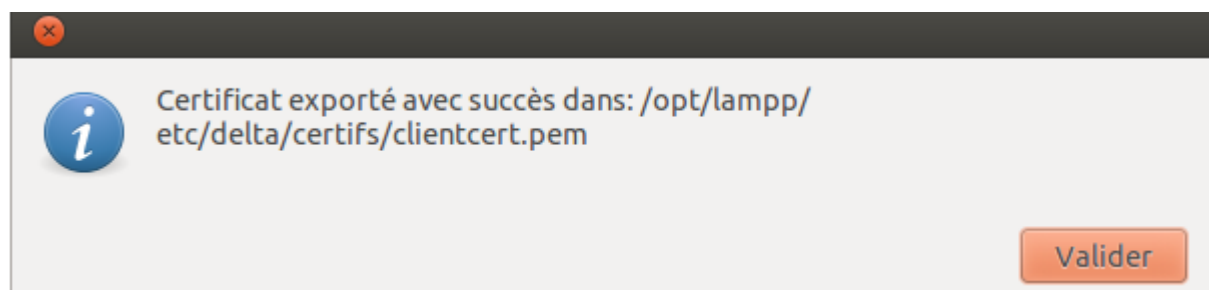
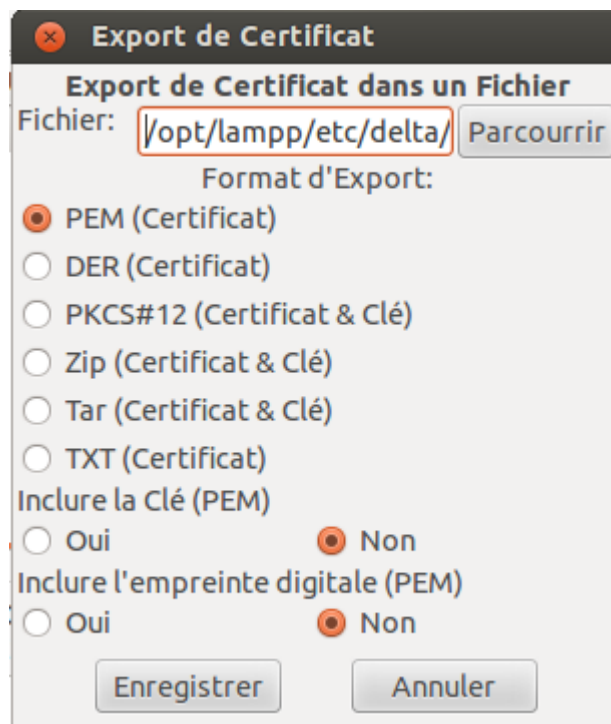
☒ Oui ☐ Non

Inclure le Certificat (PEM)

☐ Oui ☒ Non



Exportation du certificat dans les répertoires « **/opt/lampp/etc/delta/certifs/** » :



2- Modification du fichier « httpd-ssl.conf » avec la commande :

sudo nano httpd-ssl.conf

```
# huge file containing all of them (file must be PEM encoded)
# Note: Inside SSLCertificatePath you need hash symlinks
# to point to the certificate files. Use the provided
# Makefile to update the hash symlinks after changes.
#SSLCertificatePath "/opt/lampp/etc/ssl.crt"
#SSLCertificateFile "/opt/lampp/etc/ssl.crt/ca-bundle.crt"
SSLCertificatePath "/opt/lampp/etc/delta/certifs/"
SSLCertificatePath "/opt/lampp/etc/delta/certifs/cacert.pem"
# Certificate Revocation Lists (CRL):
# Set the CA revocation path where to find CA CRLs for client
# authentication or alternatively one huge file containing all
# of them (file must be PEM encoded).
```

```
# number which specifies how deeply to verify the certificate
# issuer chain before deciding the certificate is not valid.
#SSLVerifyClient require
#SSLVerifyDepth 10
SSLVerifyClient require
SSLVerifyDepth 2
```

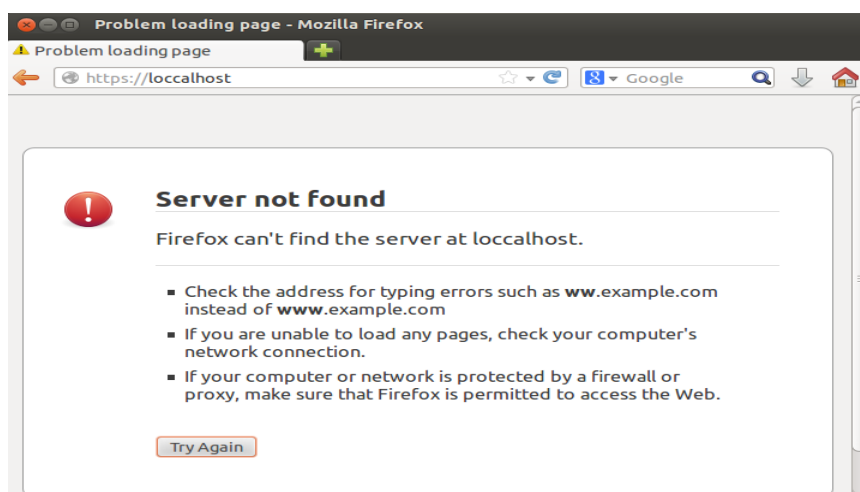
```
# Access Control:
```

3- On relance le serveur Apache

```
ipls@ipls-VirtualBox:/opt/lampp/etc/extra$ cd /opt/lampp
ipls@ipls-VirtualBox:/opt/lampp$ sudo ./lampp restart
Restarting XAMPP for Linux 1.8.3-3...
XAMPP: Stopping Apache...fail.
apachectl returned 1.
XAMPP: Stopping MySQL...ok.
XAMPP: Starting Apache...already running.
XAMPP: Starting MySQL...ok.
ipls@ipls-VirtualBox:/opt/lampp$
```

Activer Windows
Accédez aux pages

Et on teste : le resultat explique le besoin de certification du client



4- On ajoute le certificat client dans le navigateur

Export de Certificat

Export de Certificat dans un Fichier

Fichier:

Format d'Export:

☐ PEM (Certificat)

☐ DER (Certificat)

☒ PKCS#12 (Certificat & Clé)

☐ Zip (Certificat & Clé)

☐ Tar (Certificat & Clé)

☐ TXT (Certificat)

Inclure la Clé (PEM)

☐ Oui ☒ Non

Inclure l'empreinte digitale (PEM)

☐ Oui ☒ Non

Exporter au format PKCS#12

Exporter au format PKCS#12

Mot de passe de la Clé:

Mot de passe d'Export:

Friendly Name:

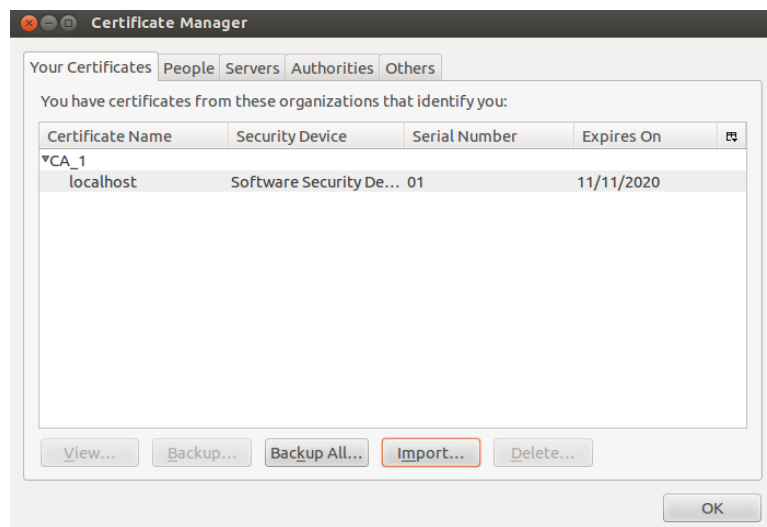
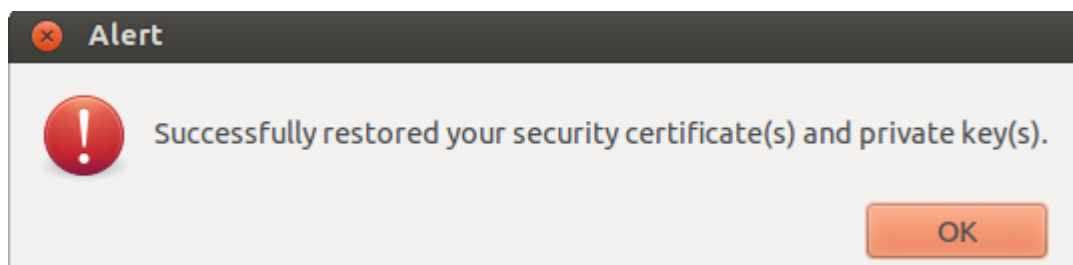
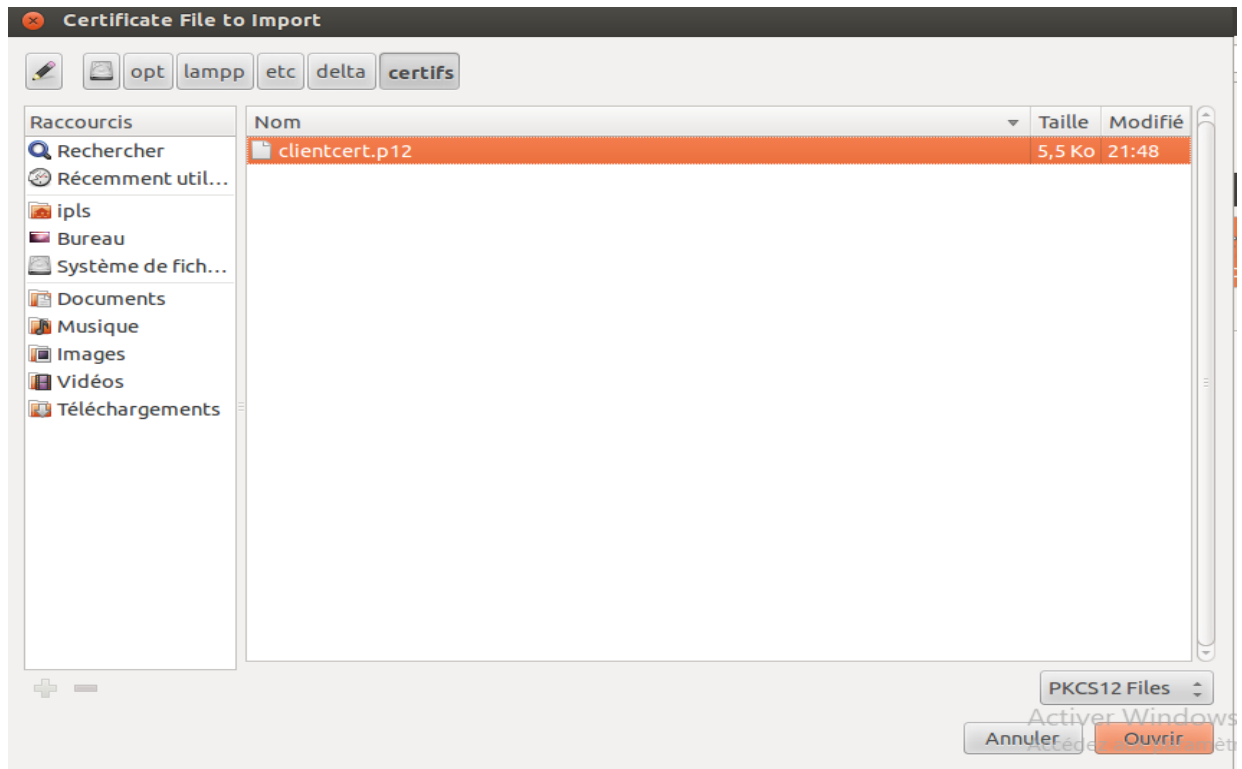
Sans Mot de passe

☒ Oui ☐ Non

Ajouter le Certificat de la Ca à la structure PKCS#12

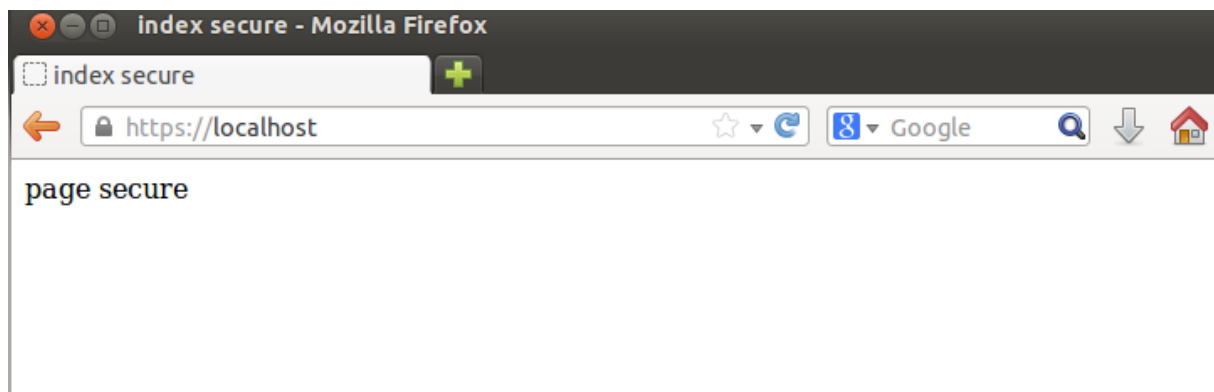
☒ Oui ☐ Non

Le certificat et la clé privée ont été correctement exportés dans /opt/lampp/etc/delta/certifs/clientcert.p12



5-On relance le serveur Apache et on teste de nouveau :

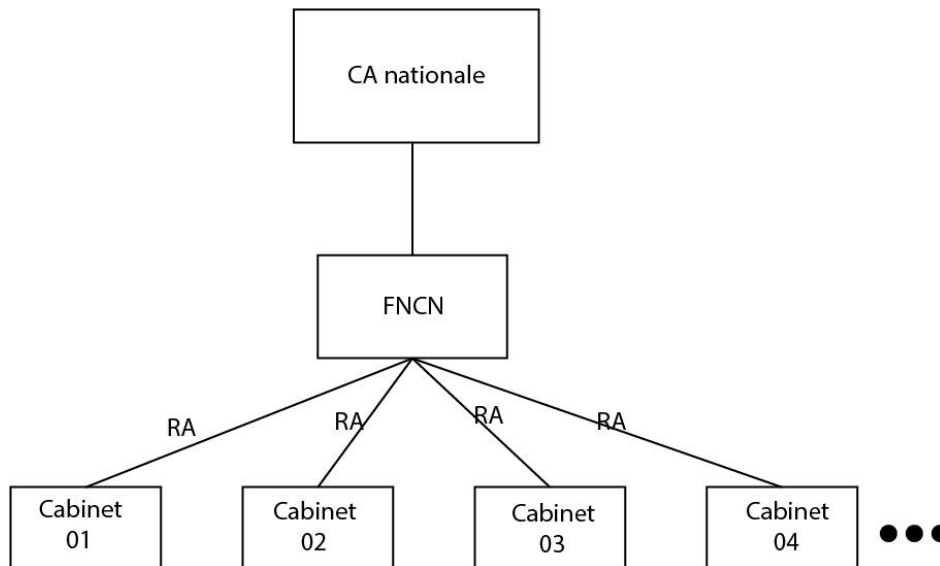
```
ippls@ippls-VirtualBox:/opt/lampp/etc/delta/certifs$ ls
CA_1-cacert.pem  clientcert.pem      localhost-cert.p12
clientcert.p12   localhost-cert1.p12  serveururcert.pem
ippls@ippls-VirtualBox:/opt/lampp/etc/delta/certifs$ cd /opt/lampp
ippls@ippls-VirtualBox:/opt/lampp$ sudo ./lampp restart
Restarting XAMPP for Linux 1.8.3-3...
XAMPP: Stopping Apache...fail.
apachectl returned 1.
XAMPP: Stopping MySQL...ok.
XAMPP: Starting Apache...already running.
XAMPP: Starting MySQL...ok.
ippls@ippls-VirtualBox:/opt/lampp$
```



Exercice B :

Question 01 : réaliser la conception du système

- La FNCN doit être une autorité de certification pour délivrer des certificats numériques
- La CA nationale délègue le pouvoir de certification à la FNCN
- La FNCN certifie les cabinets
- La FNCN certifie les clients
- Avec le modèle monopoliste (RA) la délégation de l'enregistrement au niveau des cabinets pour qu'ils puissent communiquer avec les clients en délivrant les actes à signer



Question 02 : la feuille de route

1. On aura besoin d'un serveur pour la gestion des certificats des clients et des cabinets au niveau de la FNCN
2. Les cabinets se chargent de l'enregistrement des clients et la gestion des actes, Le système applicatif se trouve au niveau des cabinets
3. L'information remonte jusqu'à la FNCN qui fait la validation finale
4. Grâce au parapheur électronique qui va transférer les actes aux clients qui pourront les signer, le parapheur assure la signature électronique
5. La confidentialité : elle est assurée grâce au processus de chiffrement et déchiffrement asymétrique par la paire de clé (privé, publique).
6. La confidentialité : elle est assurée grâce au processus de chiffrement et déchiffrement asymétrique par la paire de clé (privé, publique).
7. L'intégrité : est assurée grâce à la signature du document le haché du document sera chiffré avec la clé privée du client.
8. L'authentification : également assurée avec l'hachage à base de clé privé.
9. La non répudiation sera assurée avec la signature électronique

Question 03 : Spécifications techniques du parapheur

Un **parapheur électronique** est un logiciel permettant la validation d'un document électronique suivant un circuit avant sa signature électronique il permet :

- Création d'un objet DOCUMENT
- Paramétrage d'un circuit de validation et de signature.
- Signatures électroniques grâce aux certificats électroniques

Spécifications :

- Assurer le fonctionnement sur réseau local et externe
- Utilisation PKI et certificats électronique dans les fonctionnalités du parapheur
- Le parapheur électronique doit s'interfacer avec différents logiciel grâce aux protocoles :
 - CMIS : pour échanger avec les GED (gestion électronique des documents).
 - SEDA : standard d'échange de données pour l'archivage pour échanger avec un système d'archivage électronique.
 - WS : web services pour connecter le parapheur avec un logiciel tiers.