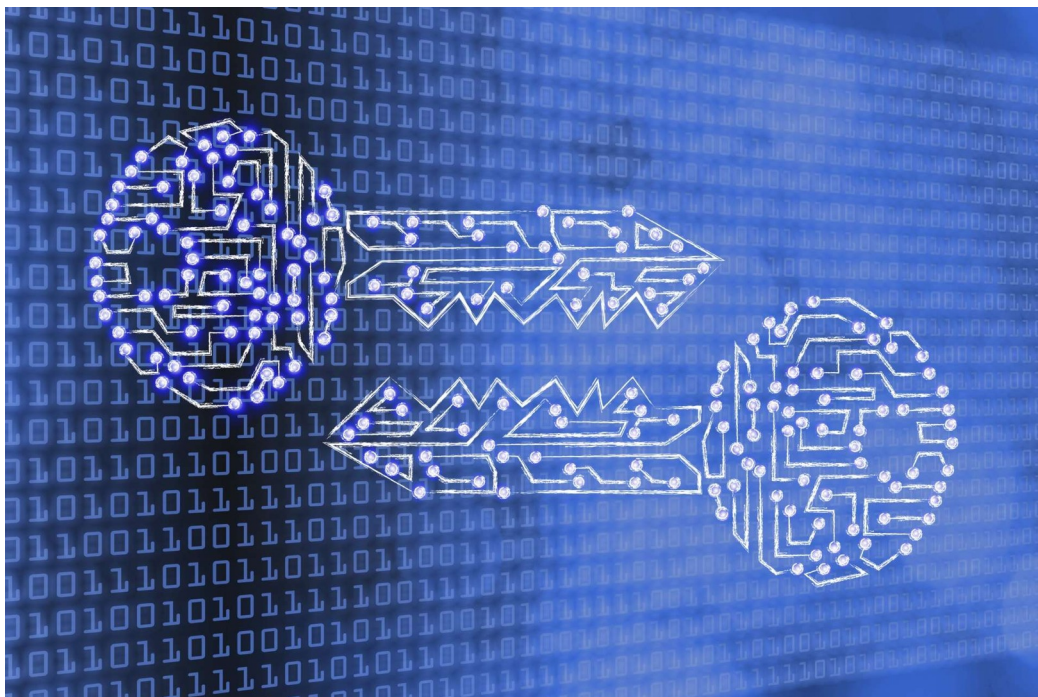


CRÉATION DE CERTIFICAT AVEC UN SERVEUR WEB



Objectif :

l'objectif de ce TP est de créer tout d'abord un certificat auto signé root à l'aide d'une paire de clé (public et privé), ensuite de la même manière un certificat intermédiaire qui sera signé par la root, et enfin un certificat serveur également, qui sera signé par le certificat intermédiaire.

Ensuite il faudra configurer le certificat serveur avec un virtualhost apache pour que le serveur web puisse être accessible en https grâce au certificat du serveur qui a été signé.

Tout d'abord je prépare mes dossiers pour les CA avec ces commandes :

```
mkdir -p pki/rootCA/{certs,crl,newcerts,private,csr}
```

```
mkdir -p pki/intermediateCA/{certs,crl,newcerts,private,csr}
```

```
touch pki/rootCA/index.txt
```

```
touch pki/intermediateCA/index.txt
```

```
(root@kratos)-[~]
# tree
.
├── pki
│   ├── intermediateCA
│   │   ├── certs
│   │   ├── crl
│   │   ├── crlnumber
│   │   ├── csr
│   │   ├── index.txt
│   │   ├── newcerts
│   │   ├── private
│   │   └── serial
│   └── rootCA
│       ├── certs
│       ├── crl
│       ├── crlnumber
│       ├── csr
│       ├── index.txt
│       ├── newcerts
│       ├── private
│       └── serial
└── .

14 directories, 6 files

(root@kratos)-[~]
#
```

Voici mon arborescence :

Détails des fichiers et dossiers :

Dossiers:

- **certs** : contient tous les certificats signés par cette CA intermédiaire (par exemple intermediate.cert.pem, certificats des serveurs signés, etc.)
- **crl** : contient les listes de révocation de certificats (CRL, fichiers .crl.pem). C'est ici que je stocke les fichiers qui indiquent quels certificats ont été révoqués.
- **newcerts** : dossier utilisé par OpenSSL pour stocker temporairement chaque nouveau certificat signé (il leur donne un numéro unique)
- **private** : contient les clés privées de la CA intermédiaire et des serveurs (ca.key.pem, intermediate.key.pem, server.key.pem)
- **csr** : contient les Certificate Signing Requests (demandes de signature de certificat et fichiers .csr.pem) générées avant la signature par la CA.

Fichiers:

- **index.txt** : base de données listant tous les certificats émis, valides ou révoqués, par la CA.
- **serial** : fichier contenant le numéro de série à attribuer au prochain certificat émis.
- **crlnumber** : fichier contenant le numéro de version de la prochaine liste de révocation (CRL) générée.

Configuration des fichiers .cnf

Un fichier .cnf (comme openssl.cnf) sert de fichier de configuration pour OpenSSL : il définit tous les paramètres et chemins nécessaires pour générer, signer, et gérer les certificats et clés dans une infrastructure PKI, notamment les dossiers utilisés, les politiques de certification, les extensions à appliquer, et les valeurs par défaut pour les certificats.

Je crée mon fichier openssl_root.cnf, voici juste le haut fichier : (le fichier entier sera fourni sur la totalité du rapport)

```
└─# cat ~/pki/rootCA/openssl_root.cnf
# ~/pki/rootCA/openssl_root.cnf

[ ca ]
default_ca = CA_default

[ CA_default ]
dir                = /home/$USER/pki/rootCA
certs              = $dir/certs
crl_dir            = $dir/crl
new_certs_dir      = $dir/newcerts
database           = $dir/index.txt
serial             = $dir/serial
RANDFILE           = $dir/private/.rand

private_key        = $dir/private/ca.key.pem
certificate         = $dir/certs/ca.cert.pem

crlnumber          = $dir/crlnumber
crl                = $dir/crl/ca.crl.pem
crl_extensions     = crl_ext
default_crl_days   = 30

default_md         = sha256
name_opt           = ca_default
cert_opt           = ca_default
default_days       = 3650
preserve           = no
policy             = policy_strict

[ policy_strict ]
countryName        = France
stateOrProvinceName = Toulouse
organizationName    = ESGI
organizationalUnitName = Biram-ESGI
commonName          = CA-root
emailAddress        = biramdandare@gmail.com

[ req ]
default_bits       = 4096
```

Je crée également mon fichier openssl_intermediate.cnf, voici à quoi ressemble le haut du fichier (le fichier entier sera fournie dans la totalité du rapport)

```

└─# cat ~/pki/intermediateCA/openssl_intermediate.cnf
# ~/pki/intermediateCA/openssl_intermediate.cnf

[ ca ]
default_ca = CA_default

[ CA_default ]
dir                = /home/$USER/pki/intermediateCA
certs              = $dir/certs
crl_dir            = $dir/crl
new_certs_dir      = $dir/newcerts
database           = $dir/index.txt
serial             = $dir/serial
RANDFILE           = $dir/private/.rand

private_key        = $dir/private/intermediate.key.pem
certificate         = $dir/certs/intermediate.cert.pem

crlnumber          = $dir/crlnumber
crl                = $dir/crl/intermediate.crl.pem
crl_extensions     = crl_ext
default_crl_days   = 30

```

Création de la CA root

Je crée ma CA root :

- `openssl genrsa -aes256 -out private/ca.key.pem 4096`
- `chmod 400 private/ca.key.pem`

Je crée d'abord ma clé privé `ca.key.pem` avec `openssl`, et je lui assigne des droit 400 qui veulent dire des droits de lecteur seule pour l'utilisateur.

```

└─(root@kratos)-[~/pki/rootCA]
└─# openssl genrsa -aes256 -out private/ca.key.pem 4096
chmod 400 private/ca.key.pem
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

└─(root@kratos)-[~/pki/rootCA]
└─#

```

Je crée maintenant son certificat avec les commandes

- `openssl req -config openssl_root.cnf \`
`-key private/ca.key.pem \`
`-new -x509 -days 3650 -sha256 -extensions v3_ca \`
`-out certs/ca.cert.pem`
- `chmod 444 certs/ca.cert.pem`

J'utilise la clé privée pour générer le certificat et en out j'aurai mon certificat `ca.cert.pem`.

La seconde commande assigne au certificat des droits 444 qui veulent dire lectures uniquement pour l'utilisateur le groupe et les autres.

```
(root@kratos)~[/pki/rootCA]
# openssl req -config openssl_root.cnf \
-key private/ca.key.pem \
-new -x509 -days 3650 -sha256 -extensions v3_ca \
-out certs/ca.cert.pem
chmod 444 certs/ca.cert.pem
Enter pass phrase for private/ca.key.pem:
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.
-----
FR [FR]:
Haute-Garonne [Haute-Garonne]:
Toulouse [Toulouse]:
ESGI [ESGI]:
Biram-ESGI []:
Common Name []:
biramdandare@gmail.com []:
(root@kratos)~[/pki/rootCA]
#
```

Ayant modifié mes valeurs directement dans le fichier `.cnf`, je peux laisser celle par défaut, elle sont déjà correctes.

Création de la CA intermédiaire

Je fais exactement pareil avec ma CA intermédiaire, je me place dans le bon répertoire (`/root/pki/intermediateCA`) et je lance les même commandes du dessus, mais adaptés pour la CA intermédiaire

J'ai d'abord générer la clé privée `intermediate.key.pem`, ensuite je crée la CSR, c'est un fichier de configuration contenant les paramètres nécessaires pour générer et signer les certificats.

Le certificat intermédiaire peut avoir des données différentes (être localisé à Dubaï etc) mais lors de la création de la csr je choisis de donner les mêmes informations que ma CA root pour une cohérence.

```
(root@kratos)~/pki/intermediateCA
# openssl genrsa -aes256 -out private/intermediate.key.pem 4096
chmod 400 private/intermediate.key.pem
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:

(root@kratos)~/pki/intermediateCA
# openssl req -config openssl_intermediate.cnf \
  -new -sha256 \
  -key private/intermediate.key.pem \
  -out csr/intermediate.csr.pem
Enter pass phrase for private/intermediate.key.pem:
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.
-----
FR [FR]:
Haute-Garonne [Haute-Garonne]:
Toulouse [Toulouse]:
ESGI [ESGI]:
Biram-ESGI []:
Common Name []:
biramdandare@gmail.com []:

(root@kratos)~/pki/intermediateCA
#
```

Nous allons maintenant signer la CSR intermédiaire (`intermediate.csr.pem`) avec la root et lui attribuer les droits adéquats aussi avec les commandes :

- `openssl ca -config openssl_root.cnf \`
 `-extensions v3_intermediate_ca \`
 `-days 1825 -notext -md sha256 \`
 `-in ../intermediateCA/csr/intermediate.csr.pem \`
 `-out ../intermediateCA/certs/intermediate.cert.pem`
- `chmod 444 ../intermediateCA/certs/intermediate.cert.pem`

J'utilise mon fichier `openssl_root.cnf` pour signer la csr intermédiaire et avoir en sortie le certificat auto signé `intermediate.cert.pem`, avec les droits 444 encore une fois.

```

(root@kratos)-[~/pki/rootCA]
# openssl ca -config openssl_root.cnf \
  -extensions v3_intermediate_ca \
  -days 1825 -notext -md sha256 \
  -in /root/pki/intermediateCA/csr/intermediate.csr.pem \
  -out /root/pki/intermediateCA/certs/intermediate.cert.pem
chmod 444 /root/pki/intermediateCA/certs/intermediate.cert.pem
Using configuration from openssl_root.cnf
Enter pass phrase for /root/pki/rootCA/private/ca.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 4096 (0x1000)
  Validity
    Not Before: May 20 19:07:14 2025 GMT
    Not After : May 19 19:07:14 2030 GMT
  Subject:
    countryName           = FR
    stateOrProvinceName   = Haute-Garonne
    organizationName      = ESGI
    commonName            = ESGI
  X509v3 extensions:
    X509v3 Subject Key Identifier:
      B1:B3:AB:49:01:B8:92:FF:09:C5:04:3B:C6:22:B4:C7:26:14:13:43
    X509v3 Authority Key Identifier:
      A1:29:ED:76:CD:43:AD:C9:14:76:7B:A8:A0:3D:55:61:88:A4:7A:80
    X509v3 Basic Constraints: critical
      CA:TRUE, pathlen:0
    X509v3 Key Usage: critical
      Digital Signature, Certificate Sign, CRL Sign
Certificate is to be certified until May 19 19:07:14 2030 GMT (1825 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
Database updated

(root@kratos)-[~/pki/rootCA]
#

```

Création de la CA serveur

Maintenant on va générer la CA serveur de la même manière que les deux autres, avec un couple de clé privé et publique

On crée une clé privée `server.key.pem`, ensuite une CSR.

```

(root@kratos)-[~/pki/intermediateCA]
# openssl genrsa -out private/server.key.pem 4096
chmod 400 private/server.key.pem

(root@kratos)-[~/pki/intermediateCA]
# openssl req -new -sha256 \
  -key private/server.key.pem \
  -out csr/server.csr.pem \
  -subj "/C=FR/ST=Haute-Garonne/L=Toulouse/O=ESGI/OU=FR/CN=www.monsite.local"

(root@kratos)-[~/pki/intermediateCA]
#

```


Nous allons signer la CA serveur avec la CA intermédiaire avec la commande :

- `openssl ca -config openssl_intermediate.cnf \`
`-extensions server_cert \`
`-days 825 -notext -md sha256 \`
`-in csr/server.csr.pem \`
`-out certs/server.cert.pem`
- `chmod 444 certs/server.cert.pem`

On utilise le fichier `openssl_intermediate.cnf` pour signer la `csr` `server.csr.pem` et générer le certificat `server.cert.pem`.

```
(root@kratos)~[~/pki/intermediateCA]
# openssl ca -config openssl_intermediate.cnf \
  -extensions server_cert \
  -days 825 -notext -md sha256 \
  -in csr/server.csr.pem \
  -out certs/server.cert.pem
chmod 444 certs/server.cert.pem

Using configuration from openssl_intermediate.cnf
Enter pass phrase for /root/pki/intermediateCA/private/intermediate.key.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
  Serial Number: 4096 (0x1000)
  Validity
    Not Before: May 20 19:13:20 2025 GMT
    Not After : Aug 23 19:13:20 2027 GMT
  Subject:
    countryName           = FR
    stateOrProvinceName   = Haute-Garonne
    localityName          = Toulouse
    organizationName      = ESGI
    organizationalUnitName = FR
    commonName            = www.monsite.local
  X509v3 extensions:
    X509v3 Basic Constraints:
      CA:FALSE
    Netscape Cert Type:
      SSL Server
    Netscape Comment:
      OpenSSL Generated Server Certificate
    X509v3 Subject Key Identifier:
      B4:5C:DD:31:97:4D:48:09:3B:7A:34:10:9B:98:40:29:A2:03:B2:AB
    X509v3 Authority Key Identifier:
      keyid:B1:B3:AB:49:01:B8:92:FF:09:C5:04:3B:C6:22:B4:C7:26:14:13:43
      DirName:/C=FR/ST=Haute-Garonne/L=Toulouse/O=ESGI/CN=ESGI
      serial:10:00
    X509v3 Key Usage: critical
      Digital Signature, Key Encipherment
    X509v3 Extended Key Usage:
      TLS Web Server Authentication
    X509v3 Subject Alternative Name:
      DNS:www.monsite.local
```

À l'aide de commande je crée ma chaîne de certificat :

Cette chaîne de certificat est obligatoire, elle est à fournir dans la configuration `ssl` de `apache`.
Ça permet à `apache` de s'assurer de l'intégrité de la chaîne `rootCA` → `intermediateCA` → `serverCA`.

Je crée la chaîne avec la commande suivante :

Donc `server-chain.pem` contient juste les trois certificats (root, intermédiaire et serveur).

- `cat certs/server.cert.pem certs/intermediate.cert.pem ../rootCA/certs/ca.cert.pem > certs/server-chain.pem`

J'active ensuite le module ssl d'apache :

- `sudo a2enmod ssl`
- `sudo systemctl reload apache2`

Ensuite je reload le serveur apache.

```
(root@kratos)~[/pki/intermediateCA]
# sudo a2enmod ssl
sudo systemctl reload apache2

perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = (unset)
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
    systemctl restart apache2

(root@kratos)~[/pki/intermediateCA]
#
```

Je configure mon virtual sur le port HTTPS (443)

```
(root@kratos)~[/pki/intermediateCA]
# cat /etc/apache2/sites-available/monserveur-ssl.conf
<VirtualHost *:443>
    ServerName www.monsite.local
    DocumentRoot /var/www/html

    SSLEngine on

    SSLCertificateFile      /etc/ssl/certs/server.cert.pem
    SSLCertificateKeyFile   /etc/ssl/private/server.key.pem
    SSLCertificateChainFile /etc/ssl/certs/intermediate.cert.pem

    # OU si tu utilises la chaîne complète :
    # SSLCertificateFile /etc/ssl/certs/server-chain.pem

    ErrorLog ${APACHE_LOG_DIR}/error.log
    CustomLog ${APACHE_LOG_DIR}/access.log combined
</VirtualHost>

(root@kratos)~[/pki/intermediateCA]
#
```

Le virtual host contient la configuration du site https.

Je vais copier chaque certificat dans les fichiers de configuration ssl grâce à ses commandes :

```
sudo cp ~/pki/intermediateCA/private/server.key.pem
/etc/ssl/private/

sudo cp ~/pki/intermediateCA/certs/server.cert.pem
/etc/ssl/certs/

sudo cp ~/pki/intermediateCA/certs/intermediate.cert.pem
/etc/ssl/certs/

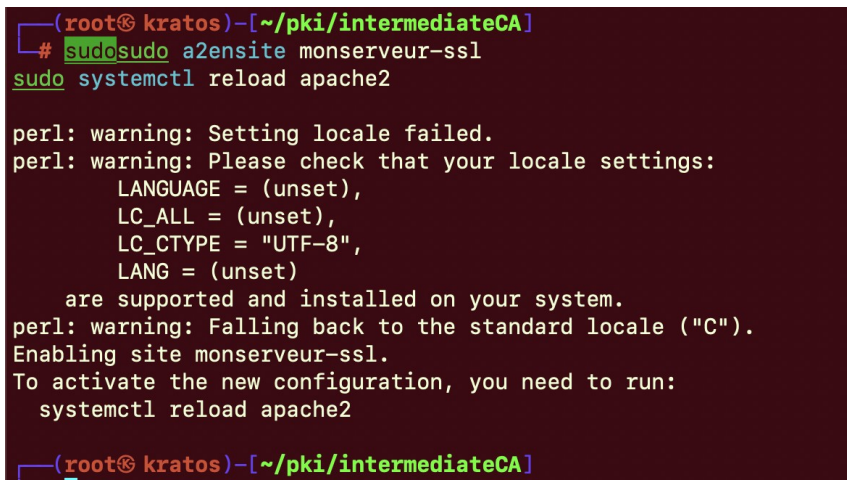
sudo cp ~/pki/rootCA/certs/ca.cert.pem /etc/ssl/certs/

sudo cp ~/pki/intermediateCA/certs/server-chain.pem
/etc/ssl/certs/

sudo chmod 400 /etc/ssl/private/server.key.pem
```

Ensuite comme l'on peut le voir sur la configuration ssl du virtual host j'ai mis le chemin vers le certificat du serveur (`server.cert.pem`), le chemin de la clé privé du serveur (`server.key.pem`), et le chemin du certificat de la CA intermédiaire (`intermediate.cert.pem`).

J'active la configuration ssl de mon virtual host et je recharge apache :



```
(root@kratos)~[~/pki/intermediateCA]
# sudo a2ensite monserveur-ssl
sudo systemctl reload apache2

perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
    LANGUAGE = (unset),
    LC_ALL = (unset),
    LC_CTYPE = "UTF-8",
    LANG = (unset)
are supported and installed on your system.
perl: warning: Falling back to the standard locale ("C").
Enabling site monserveur-ssl.
To activate the new configuration, you need to run:
    systemctl reload apache2

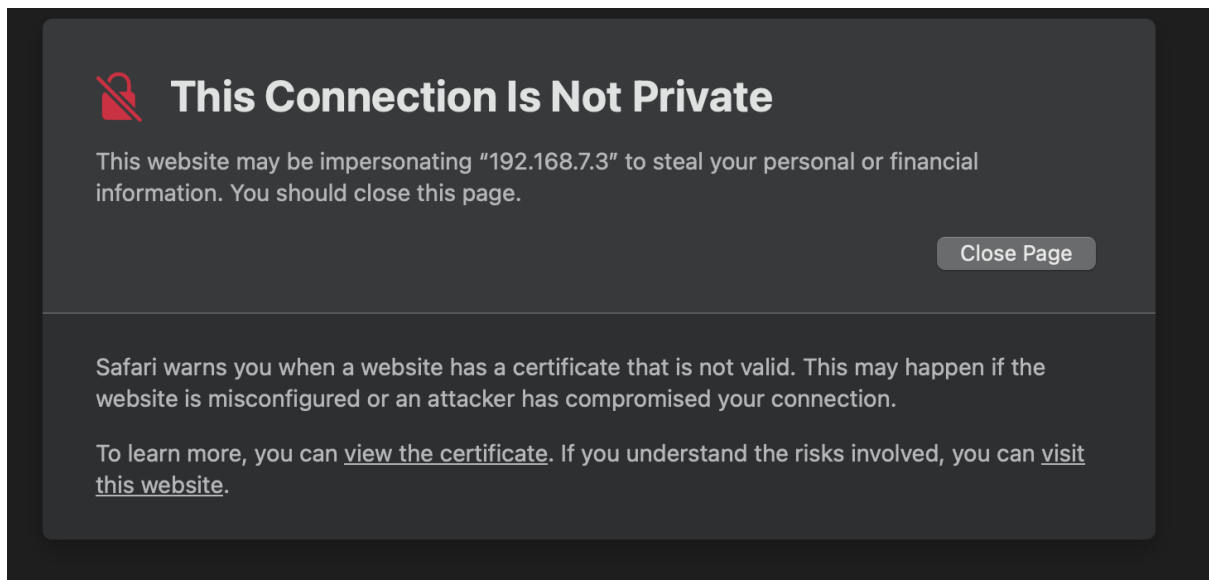
(root@kratos)~[~/pki/intermediateCA]
```

Enfin je redémarre le service apache :

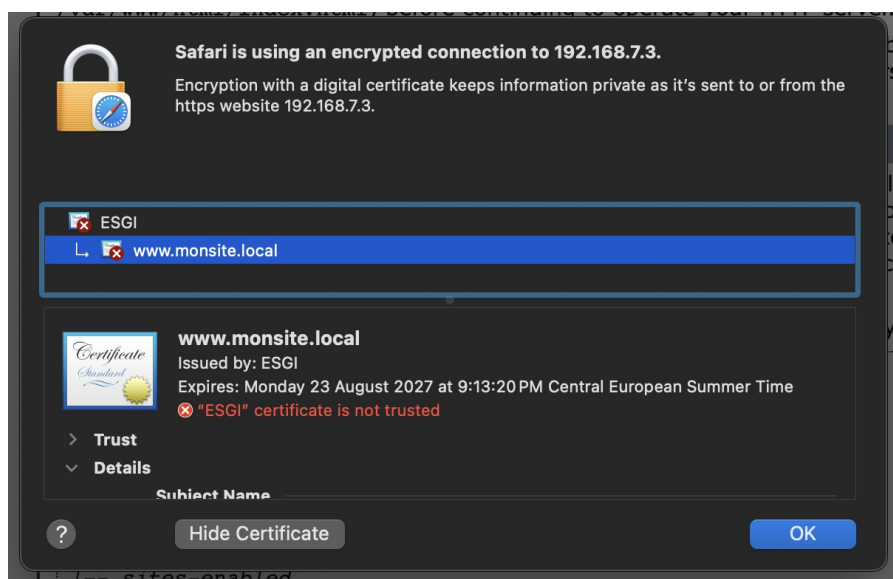
```
sudo systemctl restart apache2
```

Vérification sur le navigateur

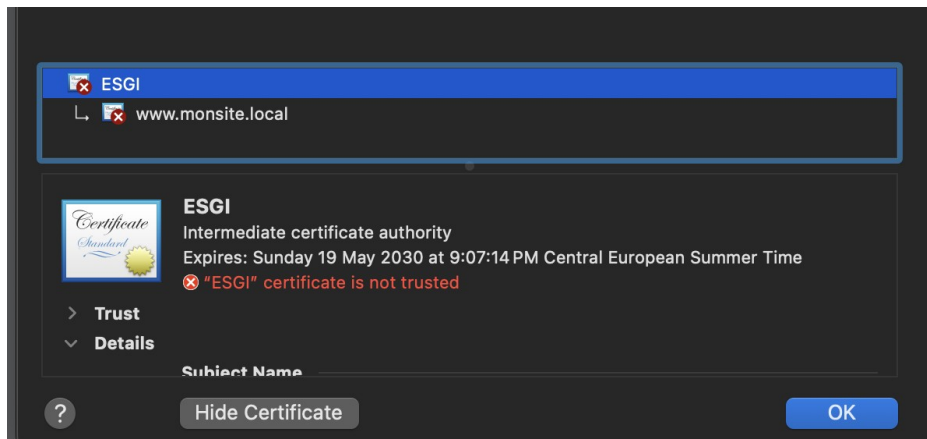
Je tente d'accéder à mon serveur web sur ma machine (192.168.7.3) en https



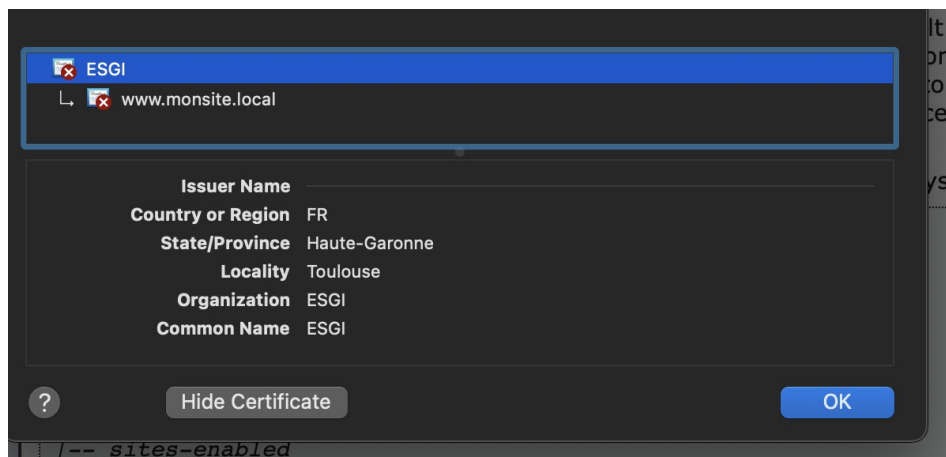
En affichant le certificat j'ai bien mes informations.



L'on remarque que le certificat `monsite.local` est issue de ESGI.



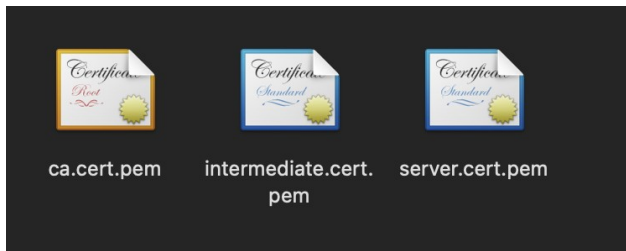
ESGI est marqué comme Intermediate certificate authority.



Et ce certificat intermédiaire est issue de ESGI qui est également le nom de ma CA root.

Liste des fichiers joints au TP.

Certificats :

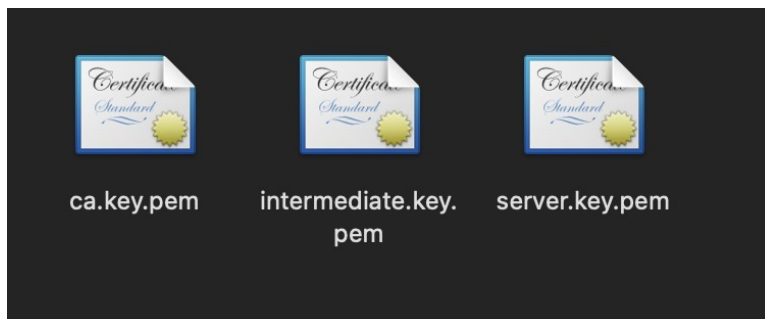


ca.cert.pem = certificat root

intermediate.cert.pem = certificat intermédiaire

server.cert.pem = certificat serveur

Clé privés :



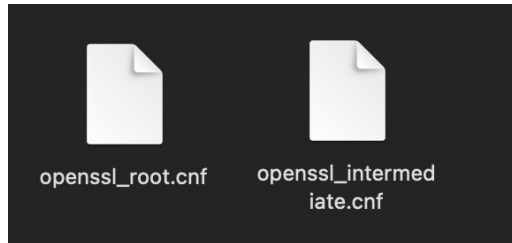
ca.key.pem = clé privé root

intermediate.key.pem = clé privé certificat intermédiaire

server.key.pem = clé privé certificat serveur.

Configuration cnf :

Le .cnf est un fichier de configuration qui définit tous les paramètres nécessaires à OpenSSL pour générer, signer et gérer les certificats et clés dans une infrastructure PKI.



`openssl_root.cnf` = configuration pour root

`openssl_intermediate.cnf` = configuration pour la CA intermédiaire