

Installation d'une PKI

L'installation se passe pour une machine debian stretch

N'ayant pas réussi à faire valider ma demande de certificat en tant qu'autorité de certification secondaire sur Active Directory (trouvable sur le bonus), j'ai donc fait signé ma demande par une autorité racine autosigné.

1. Autorité de certification racine

Nous allons commencer par créer notre autorité de certification racine. Pour ce faire, on va déplacer le dossier « ca » se trouvant dans notre repository dans le chemin « /root/ » (Être en sudo ou en su).

Après cela, se déplacer dans le dossier « /root/ca ».

On va créer notre clé privée en faisant :

```
openssl ecparam -genkey -name secp384r1 | openssl ec -aes256 -out private/ca.cheese.key.pem
```

Maintenant que l'on a notre clé privée dans le dossier « /root/ca/private », on va générer notre certificat auto-signé en faisant :

```
openssl req -config openssl_root.cnf -new -x509 -sha384 -extensions v3_ca -key private/ca.cheese.key.pem -out certs/ca.cheese.crt.pem
```

```
root@OpenSSLDebian:~/ca# openssl req -config openssl_root.cnf -new -x509 -sha384 -days 3650 -extensions v3_ca -key private/ca.cheese.key.pem -out certs/ca.cheese.crt.pem
Enter pass phrase for private/ca.cheese.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.

Country Name (2 letter code) [FR]:
State or Province Name [Gironde]:
Locality Name [Bordeaux]:
Organization Name [Epsi]:
Organizational Unit Name [I4]:
Common Name []:Epsi Root CA
Email Address [peter.xu@epsi.fr]:
root@OpenSSLDebian:~/ca#
```

Maintenant que l'on a notre certificat racine dans « /root/ca/certs/ca.cheese.crt.pem » on va pouvoir créer notre certificat secondaire.

Si on veut vérifier notre certificat, on peut faire la commande :

```
openssl x509 -noout -text -in certs/ca.cheese.crt.pem
```

```
ss = peter.xu@epsi.fr
Validity
  Not Before: May  1 18:45:55 2019 GMT
  Not After : Apr 28 18:45:55 2029 GMT
  Subject: C = FR, ST = Gironde, L = Bordeaux, O = Epsi, OU = I4, CN = Epsi Root CA, emailAddr
ss = peter.xu@epsi.fr
Subject Public key info:
  Public Key Algorithm: id-ecPublicKey
  Public-Key: (384 bit)
  pub:
    04:69:59:ff:d6:5a:e2:9e:bd:8a:2f:13:5f:3b:8f:
    97:f2:ac:53:aa:76:7a:bf:a9:81:a3:c0:bd:ca:43:
    71:1b:04:0d:66:44:99:0c:56:a8:75:77:56:d7:20:
    2e:ef:a1:a5:a6:52:1c:e3:84:d7:a6:dd:29:6d:dd:
    ec:40:77:3b:01:68:6b:0b:67:70:6f:ec:9b:ee:d7:
    be:f3:07:93:ff:c8:10:9d:c4:d1:00:47:db:b9:35:
    40:81:fd:3b:3c:8f:cd
  ASN1 OID: secp384r1
  NIST CURVE: P-384
X509v3 extensions:
  X509v3 Subject Key Identifier:
    EE:6B:39:FD:18:99:F7:61:3C:FB:F0:4B:88:BE:2E:4B:45:52:F3:DC
  X509v3 Authority Key Identifier:
    keyid:EE:6B:39:FD:18:99:F7:61:3C:FB:F0:4B:88:BE:2E:4B:45:52:F3:DC

  X509v3 Basic Constraints: critical
    CA:TRUE
  X509v3 Key Usage: critical
    Digital Signature, Certificate Sign, CRL Sign
  Signature Algorithm: ecdsa-with-SHA384
    30:64:02:30:61:a6:78:c4:b1:c9:aa:5a:4d:cd:fa:4c:17:ca:
    82:e9:ca:7c:f1:5b:ad:1c:65:6f:0a:52:67:05:2b:75:39:fb:
    83:fc:c7:7d:d3:cd:6c:21:9d:e8:76:86:66:e6:48:b7:02:30:
    19:39:cd:09:00:8f:85:dc:f7:bc:5e:16:db:db:67:9f:07:40:
    a3:00:c5:6c:1e:ac:b9:60:de:a9:41:aa:58:dd:4c:3e:04:f1:
    9a:17:20:85:e2:1c:79:3d:47:87:fb:1f
root@openSSLDebian:~/ca#
```

Comme montré sur l'encadré, le certificat contient les informations renseignées au-dessus et sa durée de validité.

2. Autorité de certificat secondaire

On va donc générer notre clé privée et notre demande de certificat en faisant :

```
openssl req -config intermediate/openssl_intermediate.cnf -new -newkey ec:<(openssl ecparam
-name secp384r1) -keyout intermediate/private/int.cheese.key.pem -out intermediate/csr/int.
cheese.csr
```

```

root@OpenSSLDebian:~/ca# openssl req -config intermediate/openssl_intermediate.cnf -new -newkey ec:<
(openssl ecparam -name secp384r1) -keyout intermediate/private/int.cheese.key.pem -out intermediate/
csr/int.cheese.csr
Generating an EC private key
writing new private key to 'intermediate/private/int.cheese.key.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [US]:FR
State or Province Name [WA]:Gironde
Locality Name [Seattle]:Bordeaux
Organization Name [Grilled Cheese Inc.]:Epsi
Organizational Unit Name [Grilled Cheese Intermediary CA]:I4
Common Name []:Epsi Intermediate CA
Email Address [grilledcheese@yummyinmytummy.us]:peter.xu@epsi.fr
root@OpenSSLDebian:~/ca#

```

Après cela, on fait signer notre demande par notre autorité racine en faisant :

```

openssl ca -config openssl_root.cnf -extensions v3_intermediate_ca -days 3600 -md sha384 -i
n intermediate/csr/int.cheese.csr -out intermediate/certs/int.cheese.crt.pem

```

```

.c:74:fopen('/root/ca/index.txt.attr','r')
140350325370944:error:2006D080:BI0 routines:BI0_new_file:no such file:../crypto/bio/bss_file.c:81:
Check that the request matches the signature
Signature ok
Certificate Details:
    Serial Number: 4096 (0x1000)
    Validity
        Not Before: May  1 18:50:19 2019 GMT
        Not After : Mar  9 18:50:19 2029 GMT
    Subject:
        countryName           = FR
        stateOrProvinceName   = Gironde
        organizationName      = Epsi
        organizationalUnitName = I4
        commonName            = Epsi Intermediate CA
        emailAddress          = peter.xu@epsi.fr
    X509v3 extensions:
        X509v3 Subject Key Identifier:
            3E:95:69:5D:17:D1:2B:55:A2:7F:27:3B:C5:BC:C0:8F:A2:5D:6B:91
        X509v3 Authority Key Identifier:
            keyid:EE:6B:39:FD:18:99:F7:61:3C:FB:F0:4B:88:BE:2E:4B:45:52:F3:DC

        X509v3 Basic Constraints: critical
            CA:TRUE, pathlen:0
        X509v3 Key Usage: critical
            Digital Signature, Certificate Sign, CRL Sign
        X509v3 CRL Distribution Points:

        Full Name:
            URI:http://crl.grilledcheese.us/whomovedmycheese.crl

        Authority Information Access:
            CA Issuers - URI:http://ocsp.grilledcheese.us/cheddarcheeseroot.crt
            OCSP - URI:http://ocsp.grilledcheese.us/

Certificate is to be certified until Mar  9 18:50:19 2029 GMT (3600 days)
Sign the certificate? [y/n]:_

```

Notre certificat sera signée une fois que l'on valide (en faisant y).

On peut donc vérifier notre certificat en faisant :

```
openssl x509 -noout -text -in intermediate/certs/int.cheese.crt.pem
```

3. Le site internet

Le site tourne sous Python3 avec le framework Flask.

Il y a un fichier « requirements.txt » pour installer les dépendances, il faut faire :

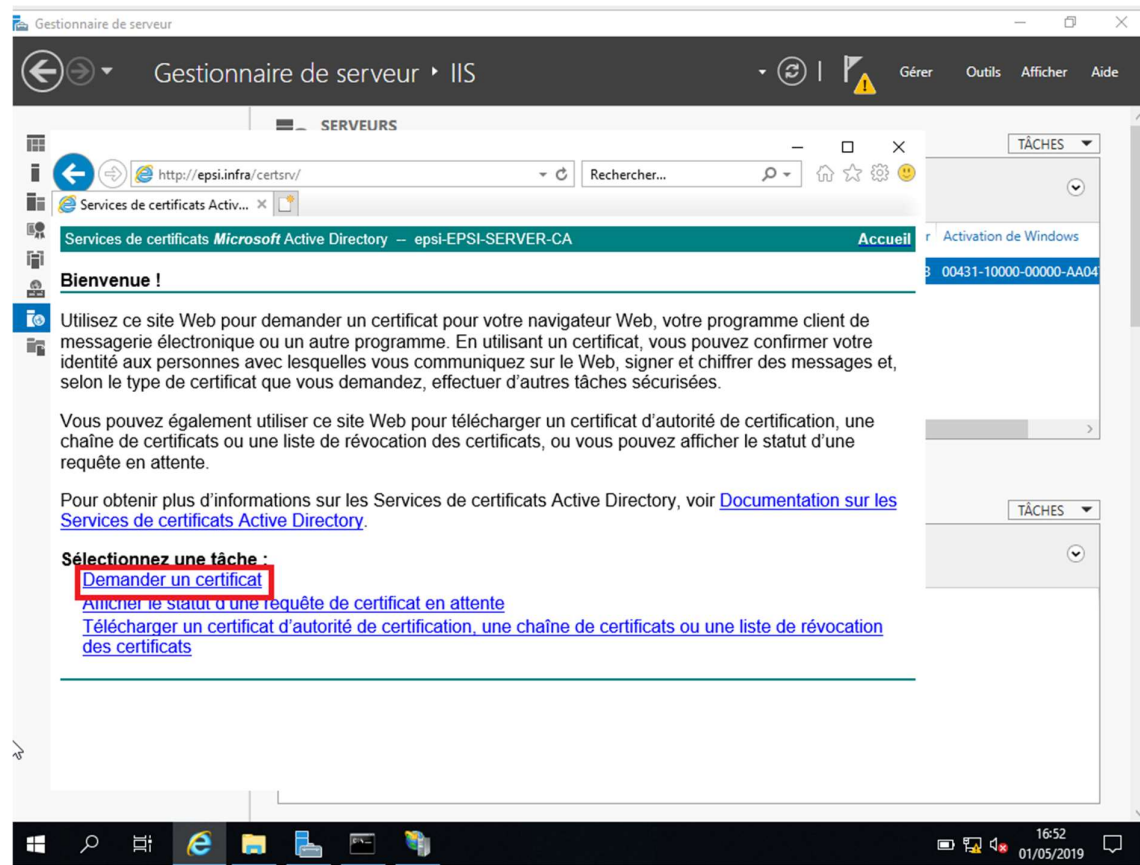
```
pip3 install -r requirements.txt
```

Puis pour lancer le serveur web, il faut faire :

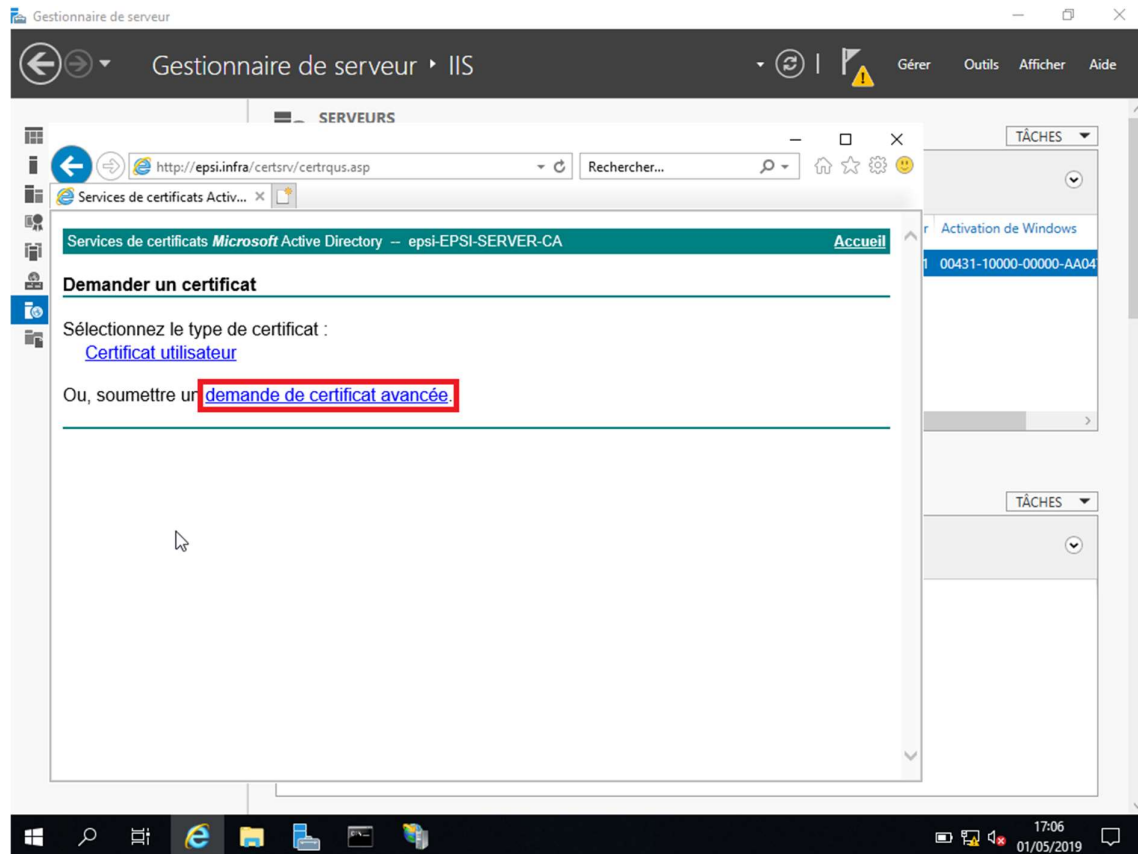
```
python index.py
```

Bonus

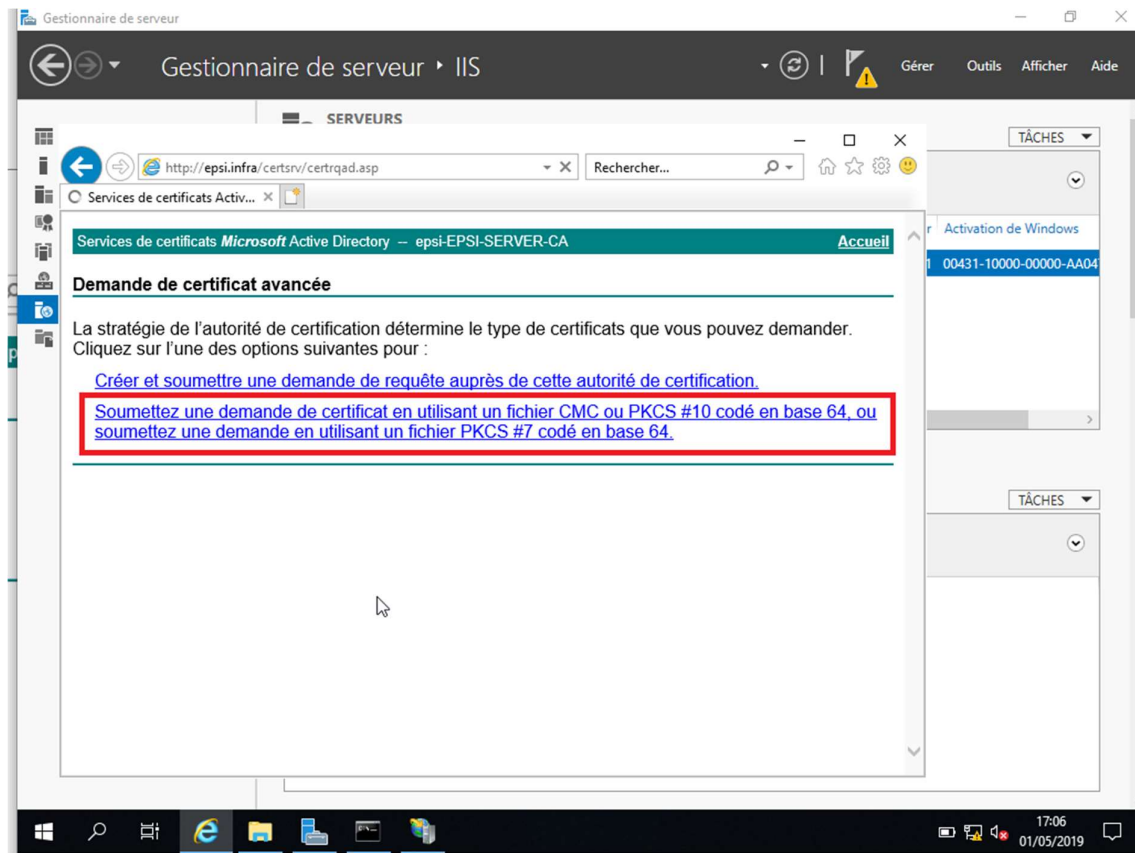
Accédez à la page http://<mon_hote>/certsrv et cliquez sur « Demander un certificat »



Après cela, vous arriverez sur la page suivante et cliquez sur « Demande de certificat avancée »



Puis sur « Soumettez une demande de certificat en utilisant un fichier [...] en base 64 »



Collez votre certificat en Base64 dans le champs « Base-64 encoded » comme sur le 1, sélectionnez « Autorité de certificat secondaire » et validez.

Fichier Machine Écran Entrée Périphériques Aide

Corbeille

Administrateur

http://epsi.infra/certsrv/certrqxt.asp

Rechercher...

Services de certificats Activ...

Soumettre une demande de certificat ou de renouvellement

Afin de soumettre une demande enregistrée à l'autorité de certification, collez une demande de certificat CMC ou PKCS #10 codé en base 64 ou une demande de renouvellement PKCS #7 générée par une source externe (telle qu'un serveur Web) dans la zone Demande enregistrée.

Demande enregistrée :

Base-64-encoded
Requête de certificat (CMC ou PKCS #10 ou PKCS #7):

```
tfmTAE/y4aZyP39ncx5wFqAS8nUHUG1fP18jJ1G  
1OvhN7RfZNzhwzL10tc9vipdydFCs81z81TXb7Vo/  
PgCveBwdDYxGF60Awubt13GirG1i1FZot35W0ZsC  
mL3BVQR/hhkT12+DctTDeUrA0NyrFW/YPDuKabx2  
JbIqUmPRn125MBRh1+VNXvOE  
-----END CERTIFICATE REQUEST-----
```

Modèle de certificat :

Autorité de certification secondaire

Attributs supplémentaires :

Attributs :

Envoyer >

Windows Server 2019 Standard Evaluation
Licence Windows valide pour 163 jours
Build 17763.rs5_release.180914-1434
17:07
01/05/2019

Malheureusement, je n'ai pas réussi à télécharger mon certificat, donc je me suis arrêté là.