Faculté des Sciences & Techniques — Université de Limoges

Master 2nde année

TMC

TP n°3

Serveur MQTT sécurisé

Chiffrement ECC : clés et certificats

Pour la génération de la clé et du certificat pour l'AC :

xterm
```
$ openssl ecparam -out ecc.ca.key.pem -name prime256v1 -genkey

$ openssl req -config <(printf
"[req]\ndistinguished_name=dn\n[dn]\n[ext]\nbasicConstraints=CA:TRUE") -new -nodes
-subj "/C=FR/L=Limoges/O=TMC/OU=IOT/CN=ACTMC" -x509 -days 3650 -extensions ext
-sha256 -key ecc.ca.key.pem -text -out ecc.ca.cert.pem
```

Pour le serveur MQTT on peut uiliser son adresse IP ou son FQDN (il faut un serveur DNS) :

xterm
```
$ openssl ecparam -out ecc.server.key.pem -name prime256v1 -genkey

$ openssl req -config <(printf
"[req]\ndistinguished_name=dn\n[dn]\n[ext]\nbasicConstraints=CA:FALSE") -new -subj
"/C=FR/L=Limoges/O=TMC/OU=IOT/CN=serveur.iot.com" -reqexts ext -sha256 -key
ecc.server.key.pem -text -out ecc.server.csr.pem

$  openssl x509 -req -days 3650 -CA ecc.ca.cert.pem -CAkey ecc.ca.key.pem
-CAcreateserial -extfile <(printf
"basicConstraints=critical,CA:FALSE\n\nsubjectAltName=DNS:localhost") -in
ecc.server.csr.pem -text -out ecc.server.pem
```

On a utilisé un « nom alternatif *» permettant d'accéder au serveur par «* localhost *».*

Si on veut aussi authentifier le client auprès du serveur :

xterm
```
$ openssl ecparam -out ecc.client.key.pem -name prime256v1 -genkey

$ openssl req -config <(printf
"[req]\ndistinguished_name=dn\n[dn]\n[ext]\nbasicConstraints=CA:FALSE") -new -subj
"/C=FR/L=Limoges/O=TMC/OU=IOT/CN=capteur" -reqexts ext -sha256 -key
ecc.client.key.pem -text -out ecc.client.csr.pem

$ openssl x509 -req -days 3650 -CA ecc.ca.cert.pem -CAkey ecc.ca.key.pem
-CAcreateserial -extfile <(printf "basicConstraints=critical,CA:FALSE") -in
ecc.client.csr.pem -text  -out ecc.client.pem
```

L'options `-days 3650` définit des certificats valables 10 ans !.

Firewall/configuration dnsmasq

Vous ajouterez les règles de firewall et la configuration de dnsmasq pour effectuer l'association DNS/adresse IP pour le bon fonctionnement du certificat.

MQTT : installation et connexion sécurisée avec authentification du client

Pour l'installation du serveur MQTT :

xterm
```
$ sudo apt-get install mosquitto
$ sudo apt-get install mosquitto-clients
```

Pour supprimer l'enregistrement de fichiers de « *log* », vous mettrez en commentaire la ligne suivante dans le fichier « `/etc/mosquitto/mosquitto.conf` » :

```
#log_dest file /var/log/mosquitto/mosquitto.log
```

Pour activer la protection d'accès au serveur MQTT par mot de passe, vous ajouterez dans le fichier « `/etc/mosquitto/mosquitto.conf` » :

```
allow_anonymous false
password_file /etc/mosquitto/mosquitto_passwd
```

Vous utiliserez la commande « `mosquitto_passwd` » pour créer le contenu du fichier password.

`https://mosquitto.org/man/mosquitto_passwd-1.html`

xterm
```
$ sudo mosquitto_passwd -c /etc/mosquitto/mosquitto_passwd <user_name>
```

Pour s'abonner à un topic MQTT :

xterm
```
$ mosquitto_sub -h localhost -p 1883 -t mon_topic -u <user_name> -P <password>
```

Pour la connexion par TLS, vous créerez les fichiers « `tcp.conf` » et « `tls.conf` » dans le répertoire `/etc/mosquitto/conf.d` :

« `tcp.conf` »
```
listener 1883
```

« `tls.conf` »
```
listener 8883
cafile /home/pef/ECC_CERTIFICATES/ecc.ca.cert.pem
certfile /home/pef/ECC_CERTIFICATES/ecc.server.pem
keyfile /home/pef/ECC_CERTIFICATES/ecc.server.key.pem
#require_certificate true
#use_identity_as_username false
```

Explications :

- ▷ *la ligne « `require_certificate true` » oblige le client à fournir un certificat, vous pouvez la supprimer si vous ne l'avez pas encore mis en place ;*
- ▷ *la ligne « `use_identity_as_username false` » indique si le « CN », « common name », du certificat doit correspondre ou non à l'utilisateur enregistré sur le serveur MQTT (accès login/mdp).*

Pour activer le service mosquitto et le lancer :

xterm
```
$ sudo systemctl enable mosquitto.service
$ sudo systemctl start mosquitto.service
```

Suite à une modification des fichiers de configuration, vous devez redémarrer le service :

xterm
```
$ sudo systemctl restart mosquitto.service
```

Vous pouvez tester la connexion TLS avec authentification du client par certificat de la manière suivante :

xterm
```
$ openssl s_client -connect localhost:8883 -CAfile ecc.ca.cert.pem -cert
ecc.client.pem -key ecc.client.key.pem
```

Si la connexion TLS réussit alors la commande ne retourne pas tout de suite mais attend des données à transmettre (connexion TLS de base équivalente à une connexion TCP mais chiffrée).

ATTENTION

Le CN, « *Common Name* », du certificat du serveur doit correspondre au nom DNS utilisé pour permettre l'authentification du serveur.

Pour accéder au serveur mosquitto une fois l'authentification serveur et client activée :

xterm
```
$ mosquitto_pub -h localhost -p 8883 -t capteur -m 'bonjour' --cafile
ecc.ca.cert.pem  --cert ecc.client.pem --key ecc.client.key.pem
```