

MQTT comment faire ?

Description

1- Bugs

1-1 Bug publication impossible en QOS 2

Si il y a un bug de publication d'un message en QOS 2 cela peut venir du fait que le client MQTT n'est pas dans une boucle. Il faut forcément que le client soit dans une boucle loop, loop_start ou loop_forever.

```
self.client.connect(host, port)self.client.loop_start()  
ret = self.client.publish(self.topic, self.message, qos=2)  
print('Publication result = ', ret)self.client.loop_stop()
```

Information

Il est à noter que pour envoyer des messages en QOS 0 (QOS 1 pas certain que c

```
self.client.connect(host, port)  
ret = self.client.publish(self.topic, self.message, qos=2)  
print('Publication result = ', ret)
```

2- MQTTS

2-1 Création des certificats

Pour créer des certificats (certificats CA auto-signé et certificats pour le broker), un script a été codé pour réaliser cela automatiquement. Ce script se trouve sur le share dans BE_Sirea/Alicia/MQTT/TLS/certificat_final.py. Si vous ouvrez ce script vous y trouverez les chemins des fichiers stockés ainsi que le contenu des champs des certificats. Ce script gère

Avertissement

Un erreur qui arrive fréquemment lors de la création des certificats est que l'information de l'entité commune n'est pas renseignée. Par contre, il est important de ne pas renseigner les mêmes informations lors

2.2 Accès au broker TLS depuis l'extérieur

Accéder au broker TLS depuis l'intérieur de Sirea est facile mais tout se complique si on souhaite se connecter au broker depuis l'extérieur. La description suivante est un exemple pour y accéder:

Serveur de production = dalkia.sirea.local (192.168.1.49)

Serveur de recette = securecharge.sirea.local (192.168.1.45)

Autres noms de domaine pour accessibilité depuis l'extérieur de Sirea :

Sont déclarés sur OVH securecharge.microserver.fr et securecharge-dev.microserver.fr tous les deux

pointant vers 213.223.121.194.

Ces noms de domaine sont aussi déclarés sur le DNS interne à Sirea.

Règles de NAT dans notre Firewall :

- IP source 213.223.121.194 port source 58884 -> IP cible 192.168.1.49 port cible 8884
- IP source 213.223.121.194 port source 48884 -> IP cible 192.168.1.45 port cible 8884

Création d'un certificat autosigné avec OpenSSL sur le serveur cible (prod ou préprod).

A partir de ce certificat, créer le certificat racine. Puis une clef et un certificat client signés par l'autorité de certification. Dans tous les cas, le CN à renseigner est le nom de domaine cible.

Ainsi le certificat et la clef que devront présenter le client (les bornes Securecharge) doivent être ceux créés sur la cible.

- Si ce sont des tests, il faut que la borne interroge securecharge-dev.microserver.fr via le port 48884 avec le certificat généré par ce serveur et dont le CN est securecharge-dev.microserver.fr. Par le NAT, les paquets seront transférés au serveur 192.168.1.45 et le port 8444. mosquito serveur va comparer les clefs de certifications présentées par mosquito client et notamment le CN. Il peut alors répondre par un OK.
- Si c'est la production, la borne doit interroger securecharge.microserver.fr via le port 58884 avec le certificat généré par ce serveur et dont le CN est securecharge.microserver.fr. Même fonctionnement que précédemment (NAT vers 192.168.1.49 port 8884).

2.3 Erreurs à ne pas commettre

Si les appareils se connectent avec un identifiant et un mot de passe le fichier mosquito.conf sera:

```
# Place your local configuration in /etc/mosquitto/conf.d/  
#  
# A full description of the configuration file is at  
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example  
persistence true  
persistence_location /var/lib/mosquitto/  
log_dest file /var/log/mosquitto/mosquitto.log  
include_dir /etc/mosquitto/conf.d  
listener 8883  
per_listener_settings false  
cafile /etc/mosquitto/ca_certificates/ca.crt  
keyfile /etc/mosquitto/certs/server.key  
certfile /etc/mosquitto/certs/server.crt  
tls_version tlsv1.3
```

Par contre si les appareils se connectent sans utiliser d'identifiants alors le fichier mosquito.conf contiendra:

```
# Place your local configuration in /etc/mosquitto/conf.d/  
#  
# A full description of the configuration file is at  
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example  
persistence true  
persistence_location /var/lib/mosquitto/  
log_dest file /var/log/mosquitto/mosquitto.log
```

```
include_dir /etc/mosquitto/conf.d
listener 8883
per_listener_settings false
cafile /etc/mosquitto/ca_certificates/ca.crt
keyfile /etc/mosquitto/certs/server.key
certfile /etc/mosquitto/certs/server.crt
require_certificate true
use_identity_as_username true
tls_version tlsv1.3
```

La ligne en gras n'est pas à oublier sinon l'erreur qui va être soulevée est Error: Protocol error.

De plus si `require_certificate` est vrai, le client doit fournir un certificat valide afin de se connecter avec succès.