MQTT – SSL Certification

Opis stosowania certyfikacji SSL przy użyciu protokołu MQTT

|  |  |
| --- | --- |
| Autor | Bartłomiej Krasoń |
| Wersja | 1 |
| Data modyfikacji | 05-01-2019 |

Spis treści

[Wprowadzenie 3](#_Toc529998823)

[Dane techniczne stanowiska 3](#_Toc529998824)

[Przygotowanie stanowiska 4](#_Toc529998825)

[Podłączenie płytki 4](#_Toc529998826)

[Instalowanie sterowników 4](#_Toc529998827)

[Konfiguracja Edisona 4](#_Toc529998828)

[Flashowanie 4](#_Toc529998829)

[Połączenie szeregowe UART 5](#_Toc529998830)

[Połączenie Wi-Fi\* 5](#_Toc529998831)

[Instalacja oprogramowania 6](#_Toc529998832)

[Przygotowanie Yocto Toolchaina 6](#_Toc529998833)

[Zakładanie projektu C++ 7](#_Toc529998834)

[Wrzucanie aplikacji na Edisona 11](#_Toc529998835)

[Uruchamianie aplikacji na Edisonie 13](#_Toc529998836)

# Wprowadzenie

Niniejszy dokument prezentuje w jaki sposób zagwarantować zabezpieczony – kodowany przesył danych pomiędzy serwerem a klientem przy wykorzystaniu protokołu MQTT. W tym celu wykorzystamy certyfikację SSL. System ten opiera się na wyprodukowaniu plików, które będę wymagane przez serwer/klienta aby możliwa była komunikacja miedzy nimi. Pliki takie nazywa się kluczami oraz certyfikatami.

# Wymagania

## Wymagane pliki

Zestaw potrzebnych plików:

* Brokera:
  + server.key – klucz certyfikatu brokera, potrzeby do utworzenia pliku server.csr
  + server.csr – plik żądania certyfikatu, potrzeby do utworzenia pliku server.crt
  + server.crt – właściwy plik certyfikujący brokera
* Klienta:
  + ca.key – klucz certyfikatu klienta, potrzebny do utworzenia pliku ca.crt
  + ca.crt – właściwy plik certyfikujący klientów

## Wymagane oprogramowanie

Korzystamy z pakietu **openssl**, który można pozyskać w następujący sposób:

$ wget <https://www.openssl.org/source/openssl-1.0.2q.tar.gz>

$ tar –xzvf openssl-1.0.2q.tar.gz

$ cd openssl-1.0.2q

$ ./config

$ make

$ sudo make install

$ /usr/local/ssl/bin/openssl version <- sprawdzenie czy instalacja zadziałała

# Utworzenie certyfikatów

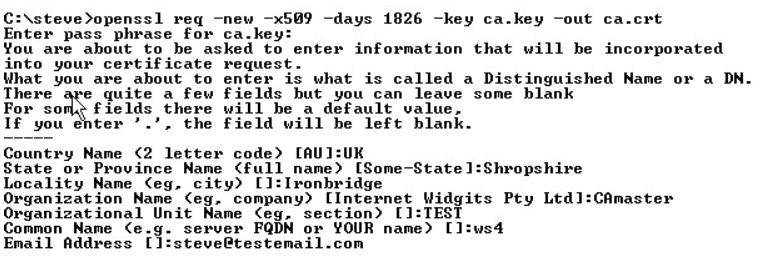
Wszystkie kroki muszą być wykonane w danej kolejności:

1. Utworzenie klucza CA klienta:

$ openssl genrsa –des3 –out ca.key 2048

1. Utworzenie certyfikatu klienta przy wykorzystaniu klucza:

$ **openssl req -new -x509 -days 1826 -key ca.key -out ca.crt**



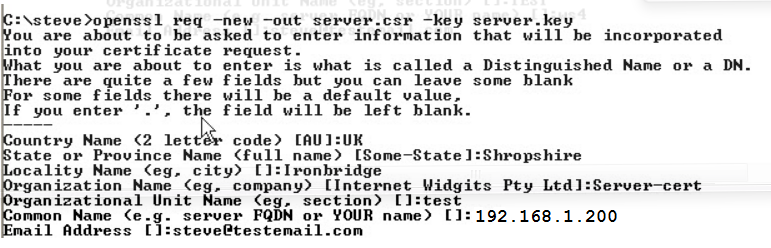
1. Utworzenie klucza CA brokera

$ **openssl genrsa -out server.key 2048**

1. Utworzenie żądanie certyfikatu brokera:

$ **openssl req -new -out server.csr -key server.key**

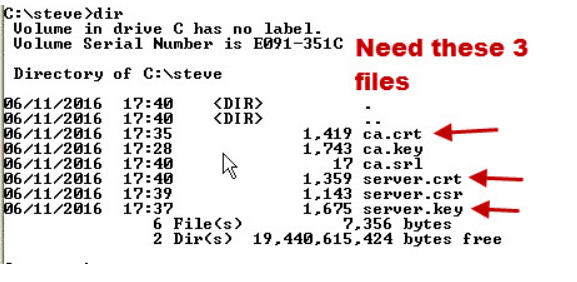
**WAŻNE** Common Name ma być ustawione na adres IP brokera (192.168.1.200) oraz dane muszą się choć trochę różnić od ca.crt (punkt 2)



1. Utworzenie certyfikatu brokera przy użyciu klucza klientów:

$ **openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 360**

1. Upewnienie się że mamy wszystkie potrzebne pliki



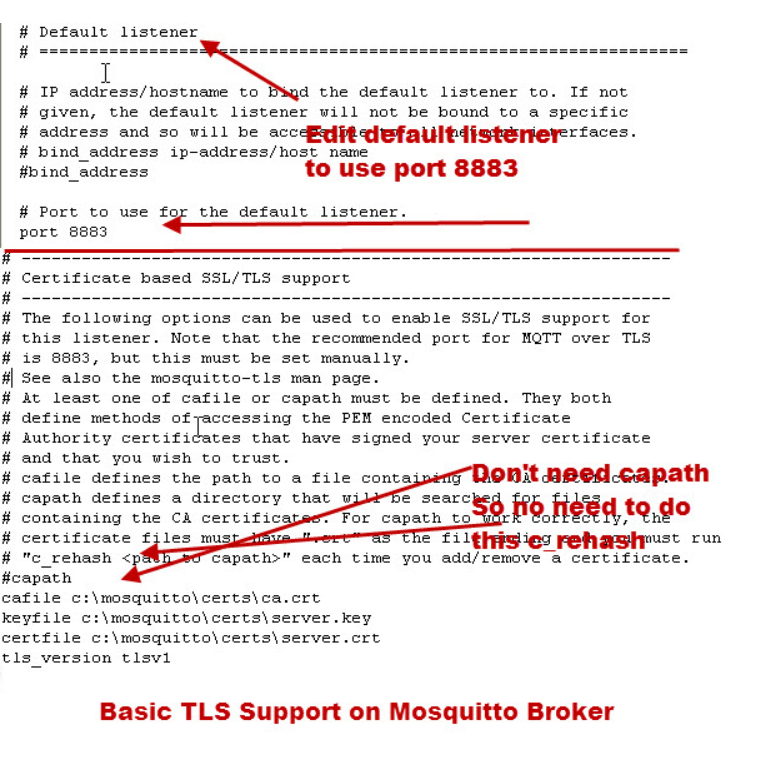
1. Przeniesienie potrzebnych plików:

$ mkdir /etc/mosquito/certs

$ cp ca.crt /etc/mosquito/certs/ca.crt

$ cp ca.crt /etc/mosquito/certs/server.crt

$ cp ca.crt /etc/mosquito/certs/server.key

1. Edytowanie pliku /etc/mosquito/mosquitto.conf
2. Od teraz broker po uruchomieniu będzie działał na porcie **8883** który jest zalecany do certyfikacji SSL.

# Modyfikacja programu klienta