IoT Security and Privacy

MQTT

Instructions:

1. Note: Blue text points to a web link. Ctrl + Click to follow link.
2. Answers to all questions must be put into **ONE** document. That is, every time, each student can only submit one report document, answering all questions of this assignment, if not explicitly stated otherwise.
3. Students must put answers following each question in this assignment. The instructor will not grade a report with only answers in it and the student gets zero for such an assignment. An assignment report must include original questions.
4. Students MUST submit the finished assignment in either Microsoft Word or pdf format to Blackboard. The doc must be submitted as ONE standalone file and cannot be tarred or zipped into a container.
5. All required files or docs must be submitted in one submission (last submission). Note: Blackboard allows unlimited number of submission of one assignment by students.

Review questions

1. A messaging broker system uses a publish/subscribe protocol based on a "hub and spoke" model. (Yes/No)

2. In MQTT, clients can only publish (Pub) messages. (Yes/No)

3. Representational state transfer (REST) is HTTP. (Yes/No)

4. Open source MQTT Mosquitto is a broker server. (Yes/No)

5. Mosquitto supports username/password authentication. (Yes/No)

6. Mosquitto supports certificate based encryption. (Yes/No)

7. Mosquitto supports pre-shared-key based encryption. (Yes/No)

Lab 1 (Needs instructional support)

Each team is required to set up a mosquitto MQTT system [1][3]. The system has three players: Client 1, Client 2, and MQTT broker (server). Client 1 must be on a Raspberry Pi. Client 2 and MQTT broker can be on the same Pi, or on different computers. Python is the recommended programming language although students are free to use C/C++/Java.

In the mosquitto MQTT system, Client 1 subscribes to the broker and Client 2 publishes to the broker. Client 1 should be able to receive messages published by Client 2. The Raspberry Pi can be installed with the Python client package paho-mqtt [2]. Please refer to [4][5] for how to set up the system.

Students will also set up the TLS/SSL transport security for the MQTT system and use certificate based authentication for authenticating the clients by the broker.

NOTE 1: Instructions in the provided citations are only for reference. They may not work. It is the students' responsibility to correctly set up the system and meet the requirements below.

NOTE 2: Students can run the following command and get an example of bash script creating private keys, certificates and others.

wget https://github.com/owntracks/tools/raw/master/TLS/generate-CA.sh

Students **CANNOT** use private keys, certificates originally generated by generate-CA.sh. Students must use individual openssl commands to create those keys and certificates. Please provide the *openssl* commands in the report when asked. Students can read generate-CA.sh, dig out the openssl commands and use them. Students just cannot use generate-CA.sh directly although students can try this command and see what correct keys and certificate look like.

NOTE 3: openssl can view the content of a certificate. For example, the following command will display the content of the certificate file ca.crt.

```
openssl x509 -noout -text -in CA.crt
```

Requirements:

1. Set up the mosquitto MQTT system. Test the system works with either programs or *mosquitto_sub* and *mosquitto_pub* from *mosquitto*. Document the setup procedure and test results, including all the commands. (4 points)

2. Set up the mosquitto broker with SSL/TLS transport security. Please refer to [6][7][8]. Test the setup. Document the setup procedure and test results, including all the commands. (3 points)

3. Set up the certificate based authentication between each client and the broker while using the mosquitto broker with SSL/TLS transport security. Test the setup. Document the setup procedure and test results, including all the commands. (3 points)

References

[1] Mosquitto, An Open Source MQTT v3.1/v3.1.1 Broker, Documentation, 2016
[2] Python Software Foundation, paho-mqtt 1.2, 2016
[3] mosquitto.conf — the configuration file for mosquitto, 2016
[4] James Lewis, MQTT Introduction and Tutorial Part One - Message Brokers and why

your IoT device should use them, February 17, 2016.
[5] James Lewis, MQTT Tutorial for Raspberry Pi, Arduino, and ESP8266 - Send MQTT messages between 3 different platforms, February 24, 2016
[6] Primal Cortex, MQTT Mosquitto broker with SSL/TLS transport security, March 31, 2016
[7] J. Dunmire, SSL/TLS Client Certs to Secure MQTT, 2016
[8] HuyITF, Configure SSL/TLS for MQTT broker mosquitto, Jun 2, 2016