МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Кіберзахист об’єктів критичної інфраструктури

Лабораторний практикум №5

Впровадження MQTT з протоколом безпеки транспортного рівня використовуючи бібліотеку OpenSSL

Перевірив:
Войцеховський А. В.

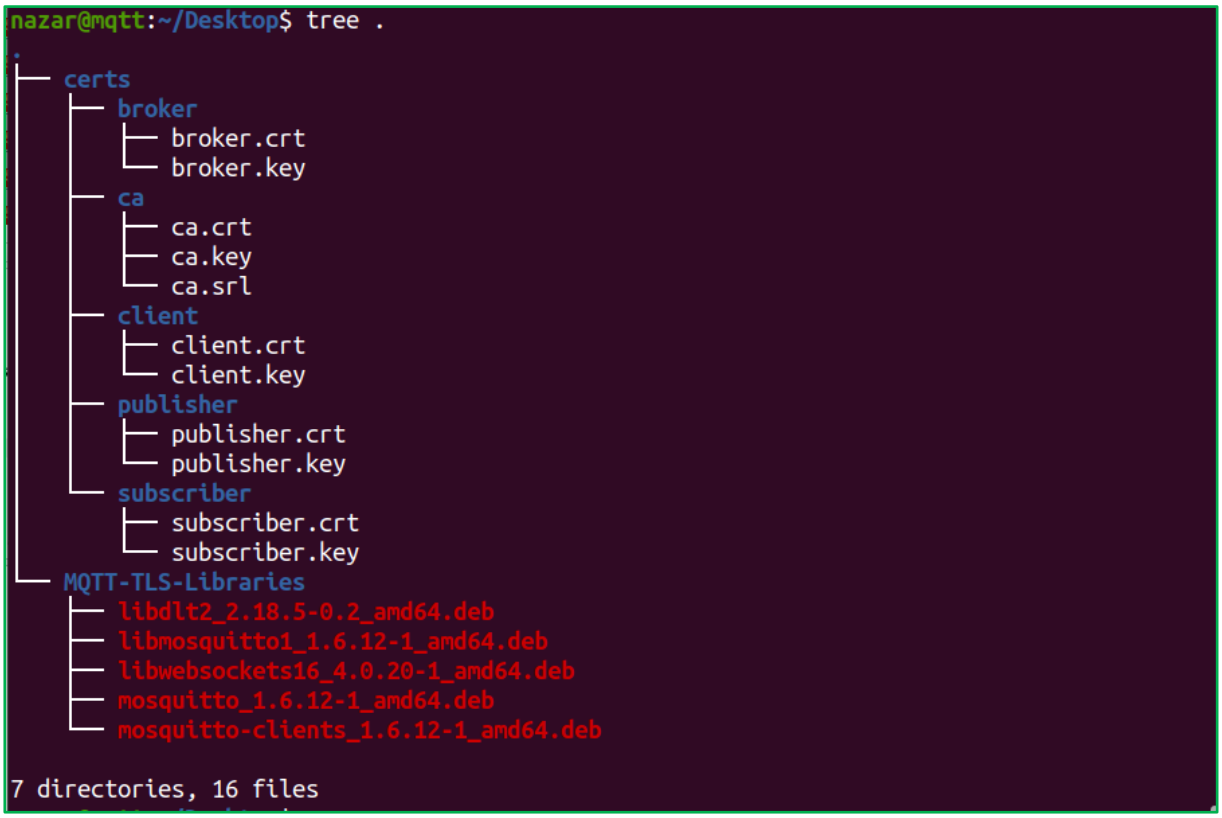
Виконав:
студент І курсу
групи ФБ-41мп
Сахній Н. Р.

Київ 2024

Мета роботи: Навчитися встановлювати безпечне/зашифроване з'єднання MQTT між клієнтами MQTT і Mosquitto Broker, що працюють на комп'ютері з використанням бібліотеки OpenSSL, яка є однією з найбільш широко використовуваних криптографічних інструментів з відкритим кодом у системах на базі UNIX.

Завдання до виконання:

1. Дерево сертифікатів та інсталяційних бібліотек для Mosquitto MQTT.



Дерево сертифікатів використовується для встановлення MQTT(S)-з'єднань (M2M-комунікацій на основі SSL/TLS) із метою забезпечення автентифікації та шифрування між різними учасниками (брокерами, клієнтами, публікаторами (**publishers**) та підписниками (**subscribers**)).

2. Надсилання вимірювальних показників із датчиків температури OVEN.

```
nazar@mqtt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert pub
lisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 125°C
nazar@mqtt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert pub
lisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 140°C
nazar@mqtt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert pub
lisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 160°C
nazar@mqtt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert pub
lisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m 200°C
nazar@mqtt:~/Desktop/certs/publisher$ mosquitto_pub -p 8883 --cafile ../ca/ca.crt --cert pub
lisher.crt --key publisher.key -h localhost -t /home/kitchen/oven_temperature -m ***°C
```

Отже, для прикладу, в даному випадку датчик температури виступає в ролі “Publisher”, тому його завдання зводиться лише до публікації даних у бік брокера. Стандартний порт MQTT-брокера для вхідних TCP-з'єднань – 1883. При використанні захищеного підключення SSL/TLS – порт 8883.

3. Отримання даних брокером від **publisher** та надсилання їх до **subcriber**.

```
nazar@mqtt:~/Desktop/certs$ sudo mosquitto -v -c /etc/mosquitto/mosquittotest.conf
1732008102: mosquitto version 1.6.12 starting
1732008102: Config loaded from /etc/mosquitto/mosquittotest.conf.
1732008102: Opening ipv4 listen socket on port 8883.
1732008102: Opening ipv6 listen socket on port 8883.
1732008102: mosquitto version 1.6.12 running
1732008106: New connection from 127.0.0.1 on port 8883.
1732008106: New client connected from 127.0.0.1 as mosq-BDtPH0chR1gs8Yf7fR (p2, c1, k60).
1732008106: No will message specified.
1732008106: Sending CONNACK to mosq-BDtPH0chR1gs8Yf7fR (0, 0)
1732008106: Received SUBSCRIBE from mosq-BDtPH0chR1gs8Yf7fR
1732008106:     /home/kitchen/oven_temperature (QoS 0)
1732008106: mosq-BDtPH0chR1gs8Yf7fR 0 /home/kitchen/oven_temperature
1732008106: Sending SUBACK to mosq-BDtPH0chR1gs8Yf7fR
1732008110: New connection from 127.0.0.1 on port 8883.
1732008110: New client connected from 127.0.0.1 as mosq-KLxuZD2w9H45W4o1gl (p2, c1, k60).
1732008110: No will message specified.
1732008110: Sending CONNACK to mosq-KLxuZD2w9H45W4o1gl (0, 0)
1732008110: Received PUBLISH from mosq-KLxuZD2w9H45W4o1gl (d0, q0, r0, m0, '/home/kitchen/oven
_temperature', ... (6 bytes))
1732008110: Sending PUBLISH to mosq-BDtPH0chR1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_te
mperature', ... (6 bytes))
1732008110: Received DISCONNECT from mosq-KLxuZD2w9H45W4o1gl
1732008110: Client mosq-KLxuZD2w9H45W4o1gl disconnected.
1732008112: New connection from 127.0.0.1 on port 8883.
1732008112: New client connected from 127.0.0.1 as mosq-jtRO8fGaabTJzlnIYb (p2, c1, k60).
1732008112: No will message specified.
1732008112: Sending CONNACK to mosq-jtRO8fGaabTJzlnIYb (0, 0)
1732008112: Received PUBLISH from mosq-jtRO8fGaabTJzlnIYb (d0, q0, r0, m0, '/home/kitchen/oven
_temperature', ... (6 bytes))
```

```
1732008112: Received DISCONNECT from mosq-jtRO8fGaabTJzlnIYb
1732008112: Client mosq-jtRO8fGaabTJzlnIYb disconnected.
1732008114: New connection from 127.0.0.1 on port 8883.
1732008114: New client connected from 127.0.0.1 as mosq-PKBUSAZc6nr2eK5PbM (p2, c1, k60).
1732008114: No will message specified.
1732008114: Sending CONNACK to mosq-PKBUSAZc6nr2eK5PbM (0, 0)
1732008114: Received PUBLISH from mosq-PKBUSAZc6nr2eK5PbM (d0, q0, r0, m0, '/home/kitchen/oven
_temperature', ... (6 bytes))
1732008114: Sending PUBLISH to mosq-BDtPH0chR1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_te
mperature', ... (6 bytes))
1732008114: Received DISCONNECT from mosq-PKBUSAZc6nr2eK5PbM
1732008114: Client mosq-PKBUSAZc6nr2eK5PbM disconnected.
1732008116: New connection from 127.0.0.1 on port 8883.
1732008116: New client connected from 127.0.0.1 as mosq-RkNEpd1FW13L3DAZPa (p2, c1, k60).
1732008116: No will message specified.
1732008116: Sending CONNACK to mosq-RkNEpd1FW13L3DAZPa (0, 0)
1732008116: Received PUBLISH from mosq-RkNEpd1FW13L3DAZPa (d0, q0, r0, m0, '/home/kitchen/oven
_temperature', ... (6 bytes))
1732008116: Sending PUBLISH to mosq-BDtPH0chR1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_te
mperature', ... (6 bytes))
1732008116: Received DISCONNECT from mosq-RkNEpd1FW13L3DAZPa
1732008116: Client mosq-RkNEpd1FW13L3DAZPa disconnected.
1732008118: New connection from 127.0.0.1 on port 8883.
1732008118: New client connected from 127.0.0.1 as mosq-pNGrydNdMz2p8PhI57 (p2, c1, k60).
1732008118: No will message specified.
1732008118: Sending CONNACK to mosq-pNGrydNdMz2p8PhI57 (0, 0)
1732008118: Received PUBLISH from mosq-pNGrydNdMz2p8PhI57 (d0, q0, r0, m0, '/home/kitchen/oven
_temperature', ... (6 bytes))
1732008118: Sending PUBLISH to mosq-BDtPH0chR1gs8Yf7fR (d0, q0, r0, m0, '/home/kitchen/oven_te
mperature', ... (6 bytes))
1732008118: Received DISCONNECT from mosq-pNGrydNdMz2p8PhI57
1732008118: Client mosq-pNGrydNdMz2p8PhI57 disconnected.
```

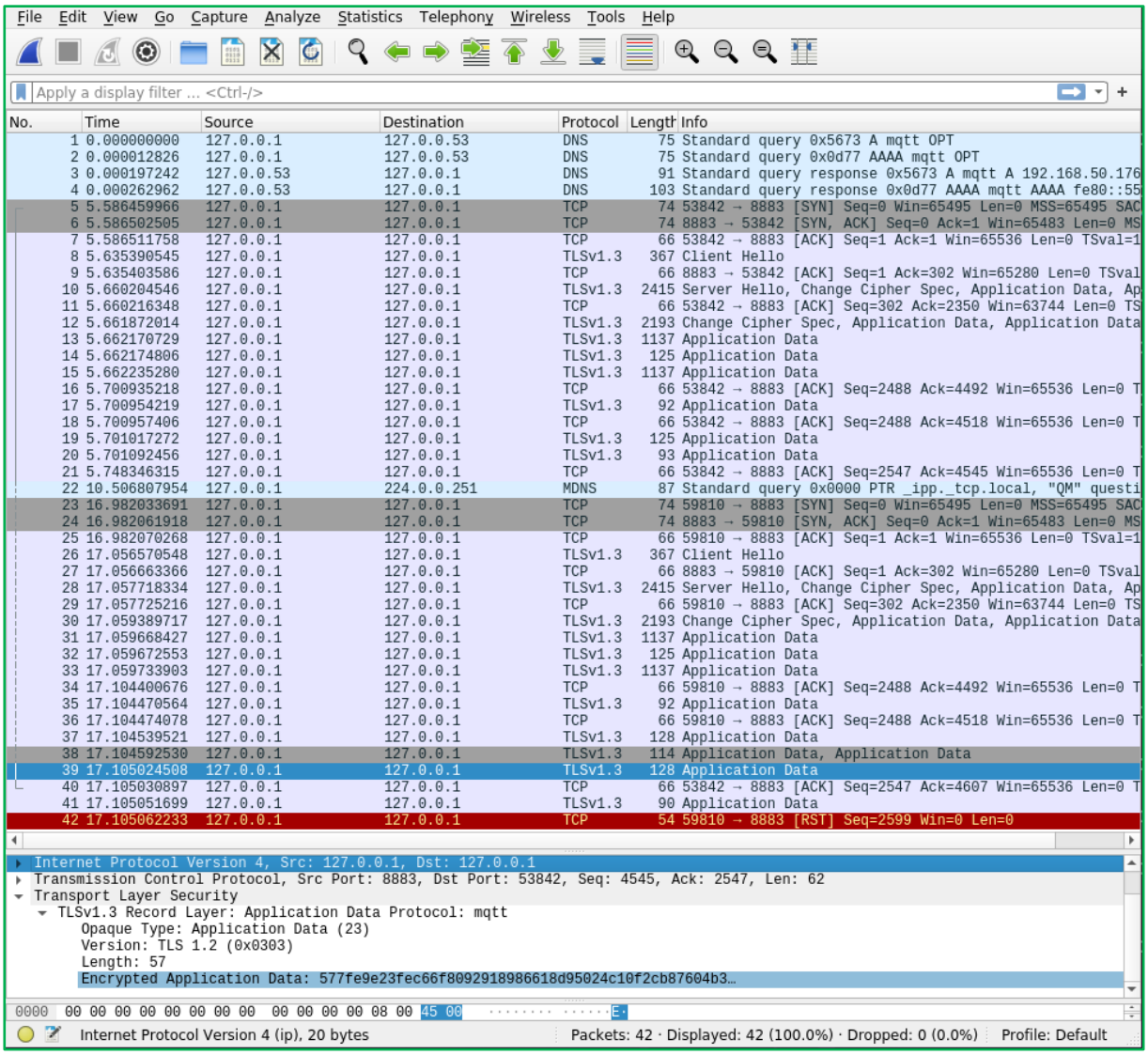
Брокер – це центральний вузол MQTT, що забезпечує взаємодію клієнтів. Обмін даними між клієнтами відбувається лише через брокера. У ролі брокера може виступати як серверне ПЗ, так і контролер. До його завдань входить отримання даних від клієнтів, обробка та збереження даних, доставка даних клієнтам та контроль за доставкою їх повідомлень.

4. Отримання даних **subscriber**, надісланих від імені **publisher** ч/з брокера:

```
nazar@mqtt:~/Desktop/certs/subscriber$ mosquitto_sub -p 8883 --cafile ../ca/ca.crt --cert sub
scriber.crt --key subscriber.key -h localhost -t /home/kitchen/oven_temperature
125°C
140°C
160°C
200°C
***°C
```

Роль “Subscriber” означає, що клієнт підписується на оновлення показників.

5. Перехоплення даних між брокером та іншими учасниками спілкування за допомогою “Wireshark” для підтвердження, що з’єднання шифроване.



З’являються записи протоколу **TLSv1.3**, що вказує на те, що з’єднання зашифроване. Поле “Encrypted Application Data” підтверджує, що вміст переданих даних є зашифрованим. Неможливо побачити текст повідомлення без доступу до приватного ключа. Також у записах помітно пакет RST (Reset), який свідчить про завершення сеансу цього з’єднання.

Додатково перевірено, що без шифрування йде обмін пакетами MQTT.

6. Відповісти на контрольні запитання.

- **Як працює взаємодія “Клієнт-Брокер-Публікатор-Підписник»? У чому полягають переваги використання такої схеми?**

Отже, в цій схемі MQTT-клієнти, тобто публікатор і підписник, не знають про існування один одного і не взаємодіють безпосередньо. Брокер може отримувати дані з різних джерел, проводити над ними маніпуляції, наприклад, розраховувати середнє значення від кількох публікаторів і вже оброблені дані повертати підписнику.

При цьому, асинхронність протоколу MQTT передбачає, що публікатор та підписник можуть бути онлайн у різний час, втрачати пакети, і бути недоступними. Брокер подбає про те, щоб зберегти в пам'яті останні дані, отримані від датчика, та забезпечити їх доставку.

- **До якого рівня моделі OSI належить MQTT?**

MQTT знаходиться на прикладному рівні моделі OSI, а водночас для структури пакету він також використовує Ethernet, TCP/IP та SSL/TLS.

- **Які альтернативи MQTT ви можете вказати?**

Modbus: Організація зв’язку між цифровими пристроями в області релейного захисту і автоматики. Може використовуватись для передачі даних через послідовні лінії зв’язку RS-485, RS-422, RS-232, а також мережі TCP/IP (Modbus TCP). Розроблений для використання в програмно-логічних контролерах.

DNP3 (Distributed Network Protocol): Підтримка роботи з подіями типу: зміна стану і подія з міткою часу. Обмін даними по мережах Ethernet і через інтерфейси RS232/RS485. Використання при передачі повідомлення великого розміру. Широкомовна розсилка повідомлень. Віддалене конфігурування програмно-логічних контролерів.

- **Які основні кроки слід зробити, щоби налаштувати криптографічно захищену передачу інформації?**
  1) *Вибір криптографічного алгоритму та методу шифрування* – визначення типу шифрування й алгоритму, які будуть використані.
  2) *Генерація криптографічних ключів та сертифікатів* – після вибору алгоритмів потрібно згенерувати відповідні ключі та сертифікати для підтвердження автентичності кінцевого пристрою.
  3) *Налаштування криптографічного захисту на пристроях* – після цього налаштовуються пристрої для використання обраних алгоритмів, що включає конфігурацію криптографічних модулів, налаштування для захищеного каналу для шифрованого зв'язку.

- **Для чого в роботі використовуються TCPdump та Wireshark?**

TCPdump і **Wireshark** є інструментами для перехоплення і аналізу мережевого трафіку. Вони дозволяють спостерігати, як дані передаються через мережу, що важливо для діагностики та моніторингу з’єднань.

Для з’єднань, що використовують шифрування (SSL/TLS), можна було побачити, що комунікації здійснюються через порт **8883**, який є стандартним для **MQTT(S)**. Проаналізувавши більш детально пакети, було підтверджено наявність TLSv1.3-пакетів, тому всі дані були зашифровані.