Autenticación en MQTT

Requisitos previos:

- Mosquitto instalado en un servidor (en nuestro caso una instancia EC2 en AWS).
- OpenSSL instalado para generar certificados.
- Cliente MQTT en Python (paho-mqtt).
- ESP32 con la biblioteca **pubsubclient**.

Caso 1: Autenticación con usuario y contraseña

1. **Configuración del servidor Mosquitto**:
   - Edita el archivo de configuración de Mosquitto (`/etc/mosquitto/mosquitto.conf`):

     ```
     allow_anonymous false
     password_file /etc/mosquitto/passwd
     ```

   - Crea el archivo de contraseñas con `mosquitto_passwd` :

     ```
     sudo mosquitto_passwd -c /etc/mosquitto/passwd username
     ```

     Ingresa la contraseña cuando se solicite.
2. **Habilitar TLS (si no está configurado ya)**:
   - Si aún no has configurado TLS, sigue estos pasos:
     - Genera los certificados usando OpenSSL:

       ```
       openssl genpkey -algorithm RSA -out mosquitto-server.key
       openssl req -new -key mosquitto-server.key -out server.csr
       openssl x509 -req -in server.csr -signkey mosquitto-server.key -out
       mosquitto-server.crt
       ```

     - Configura Mosquitto para utilizar estos certificados:

       ```
       listener 8883
       cafile /path/to/ca.crt
       certfile /path/to/mosquitto-server.crt
       keyfile /path/to/mosquitto-server.key
       ```

3. **Cliente MQTT con Python (Paho)**:
   - Crea un script en Python que se conecte al broker Mosquitto usando usuario y contraseña:

```
import paho.mqtt.client as mqtt

def on_connect(client, userdata, flags, rc):
    print("Connected with result code " + str(rc))
    client.subscribe("test/topic")

def on_message(client, userdata, msg):
    print(msg.topic + " " + str(msg.payload))

client = mqtt.Client()
client.username_pw_set("username", "password")
client.tls_set("/path/to/ca.crt")
client.connect("broker_address", 8883, 60)

client.on_connect = on_connect
client.on_message = on_message

client.loop_forever()
```

4. **Prueba con el ESP32**:
    - Modifica el código del ESP32 utilizando la librería PubSubClient:

```
#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "tu_SSID";
const char* password = "tu_clave_WIFI";
const char* mqtt_server = "broker_address";
const int mqtt_port = 8883;
const char* mqtt_user = "username";
const char* mqtt_pass = "password";

WiFiClientSecure espClient;
PubSubClient client(espClient);

void setup() {
  Serial.begin(115200);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }
  espClient.setCACert(ca_cert);
  client.setServer(mqtt_server, mqtt_port);
}

void loop() {
  if (!client.connected()) {
    client.connect("ESP32Client", mqtt_user, mqtt_pass);
  }
  client.loop();
}
```

Caso 2: Autenticación con certificados del lado del cliente

1. **Generación de certificados para el cliente**:
   - Genera una clave y certificado para el cliente:

```
openssl genpkey -algorithm RSA -out client.key
openssl req -new -key client.key -out client.csr
openssl x509 -req -in client.csr -CA mosquitto-server.crt -CAkey mosquitto-
server.key -set_serial 01 -out client.crt
```

2. **Configuración del servidor Mosquitto**:
   - Edita la configuración para habilitar la autenticación con certificados:

```
listener 8883
require_certificate true
use_identity_as_username true
cafile /path/to/ca.crt
certfile /path/to/mosquitto-server.crt
keyfile /path/to/mosquitto-server.key
```

3. **Cliente MQTT en Python (Paho)**:
   - Modifica el script de Python para usar el certificado del cliente:

```python
import paho.mqtt.client as mqtt

client = mqtt.Client()
client.tls_set(ca_certs="/path/to/ca.crt",
               certfile="/path/to/client.crt",
               keyfile="/path/to/client.key")
client.connect("broker_address", 8883, 60)

client.loop_forever()
```

4. **ESP32 con certificados**:
   - Carga el certificado en el ESP32 y usa el siguiente código:

```
espClient.setCertificate(client_cert);
espClient.setPrivateKey(client_key);
```

Pruebas:

1. Conecta el cliente Python y el ESP32 al broker y verifica que ambos puedan autenticar correctamente.
2. Publica y suscríbete a temas para verificar la comunicación.

Este laboratorio te permitirá probar la autenticación con Mosquitto utilizando dos métodos: con usuario/contraseña y con certificados.