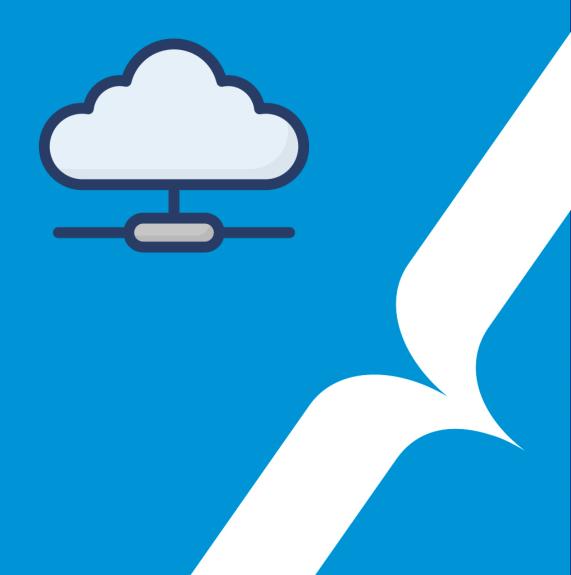


WEATHER

SESSION 4 - SECURING COMMUNICATIONS



WEATHER

Session 4: Securing Communications with SSL/TLS

Objectives

By the end of this session, you will:

- Add a security layer to the communications between weather stations and the backend.
- Encrypt MQTT and API requests using SSL/TLS.
- Understand the importance of secure communications by simulating basic intrusion attempts.

Project Steps

Part 1: Securing MQTT Communications

1. Introduction to MQTT Security

- What are the risks of unsecured MQTT communications?
- How does TLS protect MQTT messages?

2. Configuring the MQTT Broker for TLS

- Modify the Mosquitto configuration file to enable TLS.
- Generate or use existing certificates (self-signed or provided by Let's Encrypt).
- Configure Mosquitto to use these certificates for encrypted connections.
- Update MQTT clients (e.g., your weather station script) to use the encrypted connection.



3. Testing Encrypted MQTT

- Publish and subscribe to MQTT topics using TLS-enabled clients.
- Verify that the connection is encrypted by monitoring the network traffic with tools like wireshark or Wireshark.

Part 2: Encrypting API Requests with HTTPS

1. Introduction to HTTPS

- What is HTTPS, and how does it differ from HTTP?
- Why is HTTPS important for securing API communications?

2. Setting Up SSL/TLS for the API

- Generate SSL/TLS certificates (self-signed or Let's Encrypt).
- Configure your API framework (e.g., Flask, Express) to use HTTPS.
- Update any client applications interacting with the API to use HTTPS endpoints.

3. Testing Encrypted API Requests

- Use tools like Postman or cURL to ensure API requests are encrypted.
- Confirm that HTTPS is working correctly by checking the certificate details in the client response.

Part 3: Managing Certificates

1. Generating Certificates

- Generate self-signed certificates using openss1.
- (Optional) Use Let's Encrypt for obtaining free SSL/TLS certificates.

2. Storing and Renewing Certificates

Discuss best practices for managing certificates securely.



	— If using Let's Encrypt, automate renewal with сетtbot.
	Part 4: Basic Intrusion Testing
1.	Simulating Attacks — Attempt to publish or subscribe to the MQTT broker without using TLS.
	— Try accessing the API over HTTP instead of HTTPS.
2.	Observing the Results — Use Wireshark to demonstrate how unencrypted data can be intercepted.
	— Compare this with the encrypted setup to highlight the benefits of SSL/TLS.

Expected Outcomes

By the end of this session, you should have:

- 1. An MQTT broker with TLS encryption enabled.
- 2. An HTTPS-enabled API for secure communications.
- 3. A basic understanding of certificate management and renewal.
- 4. Practical experience with intrusion testing to validate the security measures.

Tips

— If using Let's Encrypt, ensure the system has access to the internet for certificate validation.

Good luck!



