WEATHER

SESSION 3 - SECURING ACCESS

WEATHER

Session 3 : Securing Access to Weather Stations with RFID Authentication

Objectives

By the end of this session, you will have :
- Integrated an RFID module to enable authentication using RFID badges.
- Implemented a user management system with secure access controls.
- Enhanced the backend to enforce role-based access control.

Project Steps

Part 1 : Integrating the RFID Module

---

1. **Introduction to RFID**
   - What is RFID, and how does it work?
   - Describe the components : RFID reader, badges, and their communication protocol.
2. **Hardware Setup**
   - Connect the RFID module to the Raspberry Pi (e.g., using GPIO pins).
   - Verify the wiring and connections.
3. **Reading RFID Tags**
   - Write a Python script using a library like `MFRC522` to detect and read RFID badge IDs.
   - Test the script to ensure it outputs badge IDs when scanned.

Part 2 : Implementing an Authentication System

1. **Introduction to Authentication**
   - What is authentication, and why is it important for securing the weather stations ?
2. **User Management**
   - Design a simple user system that stores :
     - Usernames
     - RFID badge IDs
     - Roles (e.g., Read-Only, Admin)
   - Use a database (e.g., SQLite or your existing InfluxDB) to store user details securely.
3. **Authentication Logic**
   - Write a script that verifies if a scanned RFID badge ID exists in the user database.
   - If the ID is valid, allow access. Otherwise, deny it.

Part 3 : Integrating Authentication with the Backend

1. **API Enhancement**
   - Extend the backend API to include user authentication :
     - Endpoint for authenticating users via their RFID badge ID.
     - Return appropriate responses based on the user's role.
   - Example flow :
     - RFID scans badge → Backend verifies badge ID → Backend returns user role and permissions.
2. **Securing Backend Access**
   - Add middleware to API endpoints to enforce access control based on user roles.
   - Example : Only Admins can modify settings; Read-Only users can view data but not change it.

Part 4 : Implementing Role-Based Access Control

1. **Understanding Roles**
   — Discuss the purpose of roles (e.g., separating Admin and Read-Only users).

2. **Role Management in the API**
   — Extend the user database schema to include roles.

   — Adjust API routes to respect role-based permissions :
      — Admin : Full access (read/write).

      — Read-Only : Limited access (read-only).

3. **Testing Role-Based Access**
   — Create test users with different roles.

   — Simulate API requests and ensure access is granted or denied based on the user's role.

Expected Outcomes

By the end of this session, you should have :
1. An RFID-based authentication system for weather stations.
2. A user management system with secure access control.
3. A backend that enforces role-based permissions for API interactions.

Tips

— Test the RFID module thoroughly to ensure reliable reads.

— Handle errors gracefully, such as invalid badge scans or unauthorized access attempts.

— Use secure practices for storing and transmitting user credentials.

v1