Simulated IoT Sensor Network with Role-Based Access Control using ThingsBoard and Python

Group 09

CIS 5370

Spring 2025

Christopher Meadows - 3289481

Shree Majumder- 6240191

Fares Amamou - 6209574

ABSTRACT

This project presents a simulated IoT sensor network emphasizing secure access control using Role-Based Access Control (RBAC) integrated with the ThingsBoard platform and Python-based simulated sensors. As IoT systems grow in complexity and scale, ensuring proper user access to devices and data becomes critical. The simulation models temperature and CO_2 sensors connected via MQTT and evaluates user permissions using ThingsBoard’s built-in RBAC system. Three user roles—Administrator, Operator, and Viewer—are defined to test varying levels of access. Evaluation confirms that ThingsBoard effectively enforces hierarchical access, limiting unauthorized data exposure. The project underscores the utility of open-source tools in demonstrating secure IoT environments suitable for education and prototyping.

CCS CONCEPTS

• Security and privacy → Access control; Role-based access control;

KEYWORDS

IoT, Access Control, RBAC, ThingsBoard, MQTT, Environmental Monitoring

INTRODUCTION

The Internet of Things (IoT) has changed how we monitor the environment. It helps in areas such as smart cities, air quality monitoring, and climate tracking. These systems use several sensors to collect and share real-time data. However, as more IoT devices are used, security issues have grown. There are concerns regarding who can access these sensors. If unauthorized people enter, they can steal data, change important information, or disrupt the system.

This project focuses on improving the security of IoT-based environmental monitoring systems. This is achieved by creating a Role-Based Access Control (RBAC) simulation using ThingsBoard and Python. The aim is to establish a secure model that limits the ability to access data and manage devices based on user roles. The simulation used Python to mimic the temperature and CO_2 sensors. These fake devices are connected to the ThingsBoard platform through MQTT. User roles were enforced using ThingsBoard's RBAC features.

This report describes the construction and operation of a simulation. It includes a review of similar works, the methods used, the results of experiments, and the conclusions from the project. The focus is on being cost-effective and flexible when simulating and testing access control in IoT settings.

LITERATURE REVIEW

Access Control Models in IoT

Access control is important for keeping IoT systems safe from unauthorized access. There are two main models: Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC). RBAC provides permission based on job roles, making it simple and easy to expand. ABAC uses user, resource, and environmental details to make decisions, thus allowing for more flexible rules. Sandhu et al. (1996) state that RBAC lowers management work and effectively limits access to only what is necessary. Recent studies (Kumar et al., 2020) have shown that ABAC offers more detailed control, but can be difficult to set up in limited environments. This project uses RBAC because it is simple and good for providing basic access control. In recent studies, platforms such as ThingsBoard have demonstrated the effectiveness of RBAC in real-world applications. For instance, Zhang et al. (2021) implemented RBAC in a smart city context, highlighting the need for improved data integrity and accountability.

Environmental Monitoring Systems

Environmental monitoring is an important use of the IoT. This includes checking air quality, controlling temperature, and tracking pollution. Khan et al. (2019) established real-time sensor networks to collect data on urban air pollution. Such systems often use cloud platforms to manage and provide data. However, many commercial systems are private or expensive, making them difficult to access. This project used ThingsBoard and Python, providing an open-source setup that is easy to copy for learning and experiments.

IoT Security and Platform Vulnerabilities

IoT platforms have security problems, such as weak passwords, unsafe communication methods, and poor access control. Alrawais et al. (2017) say many IoT devices do not have good security features, so central access control is important. Cloud IoT platforms, such as AWS IoT and Azure IoT, are secure but can be expensive and difficult to use. ThingsBoard is a free option with good security features, such as role-based access control, secure device login, and MQTT and REST APIs. This makes it suitable for testing and learning purposes.

Platform Comparison

Platforms such as ThingsBoard, AWS IoT, and Azure IoT offer access control features; however, their cost, complexity, and proprietary nature vary. The table below summarizes the key differences.

Table 1: Comparison of IoT Platforms

Feature	**ThingsBoard**	**AWS IoT Core**	**Azure IoT Hub**
Open Source	Yes	No	No
Cost	Free (Community)	Pay-per-use	Pay-per-use
Access Control	Built-in RBAC	IAM Policies (ABAC)	Role-based + ABAC
MQTT/REST Support	Yes	Yes	Yes
Educational Usability	High	Moderate	Moderate

Communication Standards and Security

IoT systems use MQTT and REST APIs to share data. MQTT is small and works well for devices with limited power, making it suitable for real-time sensor data. REST APIs allow devices to connect easily to backend systems. Secure communication is very important, and includes checking devices, encrypting data, and controlling access. TLS/SSL protocols are often used to keep data safe during transfer, and JSON Web Tokens (JWTs) are a standard way to securely share information. As the Internet of Things (IoT) expands, especially in areas such as environmental monitoring, keeping device communication secure and managing user access are becoming more crucial. This section reviews current work and technologies related to access control in IoT systems, focusing on environmental uses, and compares the current platforms and communication standards relevant to this project.

METHODOLOGY

This project aims to simulate a secure IoT environment using ThingsBoard, an open-source IoT platform, in conjunction with Role-Based Access Control (RBAC) to validate granular data access across three user roles: Administrator, Operator, and Viewer.

The development environment was provisioned on a virtual machine running Ubuntu 22.04 LTS. ThingsBoard was deployed locally following official installation instructions, with PostgreSQL selected as the backend database to store device data and telemetry. This setup provides full control over the system configuration and facilitates user-level role testing in a secure, isolated environment.

Three users were created to represent distinct access roles on the platform.

- Alice (Tenant Admin): Granted the highest level of access, including platform configuration, device management, and full data visibility.
- Bob (Operator): Created as a Customer User and linked to a specific customer entity. This role is intended to monitor device telemetry and dashboards without configuration privileges.
- Charlie (Viewer): Also a Customer User, but with no device or dashboard explicitly assigned to simulate access denial under RBAC.

Each user was assigned a corresponding customer entity within ThingsBoard to simulate a real-world organizational structure.

Two Python scripts were developed to simulate the environmental sensor data.

1. simulation_temp_Sensor2.py – Generates random temperature data and publishes them to ThingsBoard via MQTT using a device access token.
2. simulation_Co2_sensor.py – Simulates the CO_2 level readings using the same method.

These scripts use the MQTT protocol to publish telemetry for the following endpoints:

“v1/devices/me/telemetry”
Each simulated sensor was represented by a virtual device in ThingsBoard and authenticated via a unique device credential.

A third Python script, main.py, was developed to automate the evaluation of user-access permissions. This script:

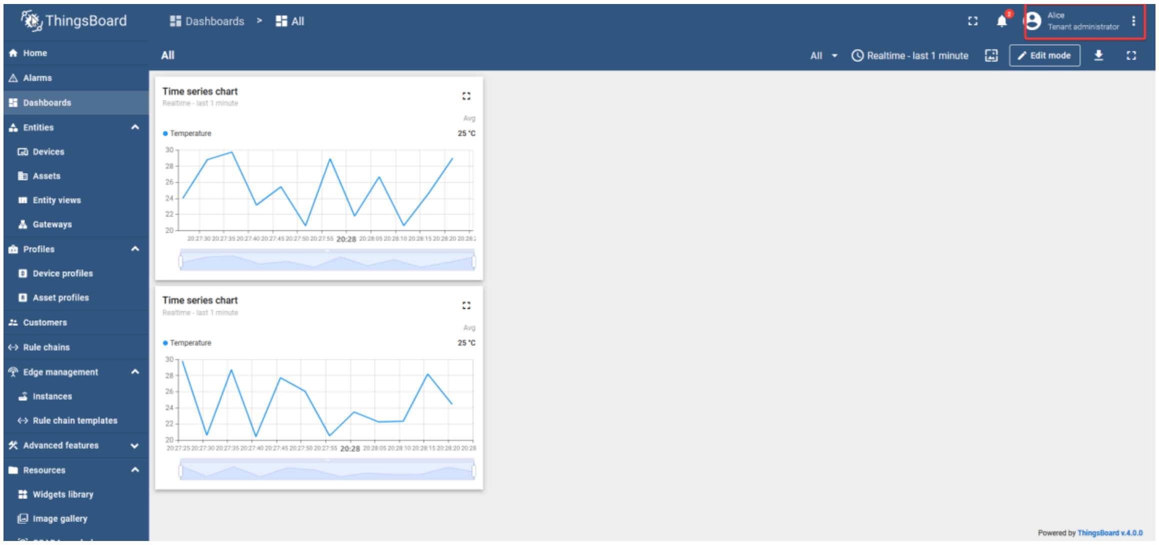
- Authenticates users using the REST API (/api/auth/login)
- Retrieves the user profile and customer context
- Lists devices visible to the authenticated user
- Attempts to fetch telemetry data from each device using
    - REST endpoint: /api/plugins/telemetry/DEVICE/{deviceId}/values/timeseries
    - Dashboard-style endpoint: /api/v1/{deviceToken}/telemetry (simulating dashboard/public API access)
- Logs results into separate log files for each user (admin_log.txt, operator_log.txt, viewer_log.txt)

This approach allows for the verification of data visibility, telemetry access, and RBAC enforcement at the API level.

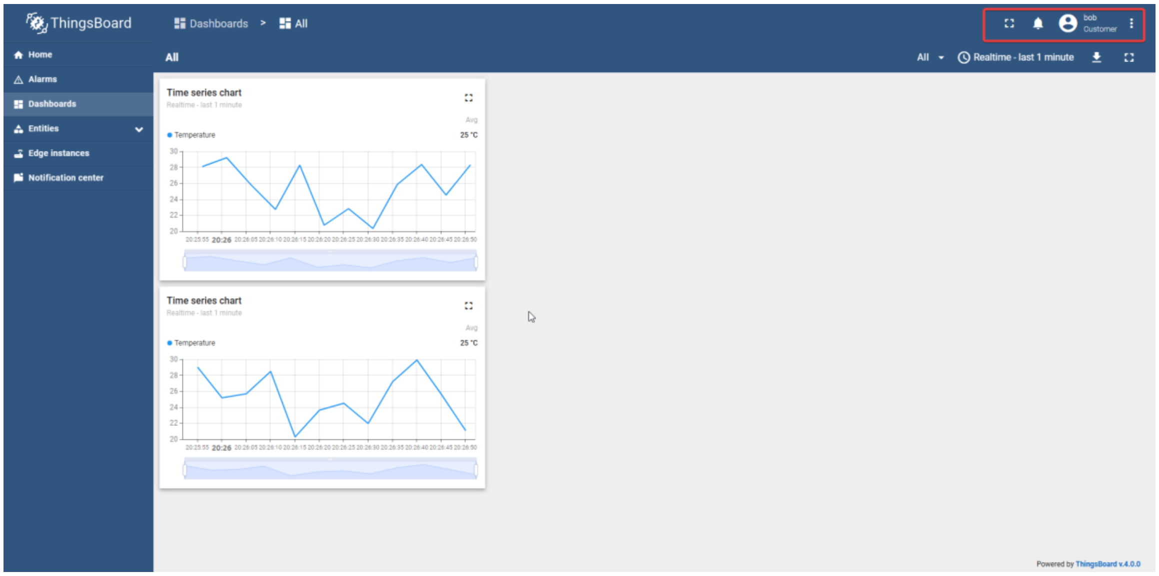
RESULTS

Dashboard Screenshots: Each role's view of the dashboard

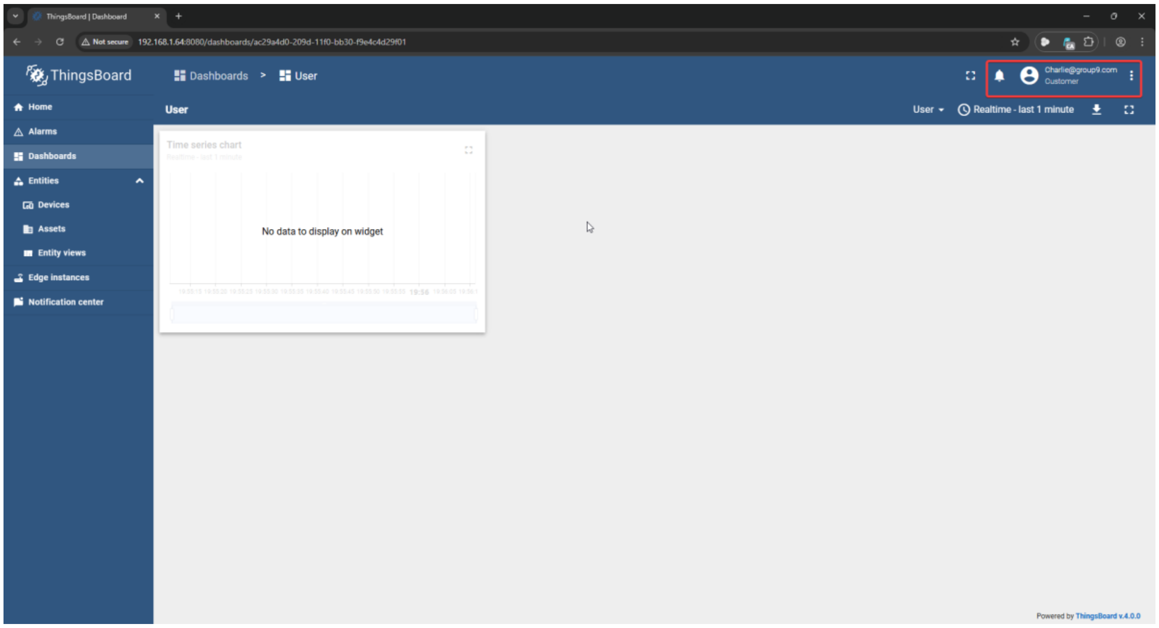
Alice (Admin) view



Bob (Operator) view



Charlie (Viewer) view



The test results validated the effectiveness of ThingsBoard’s role-based access control. Each user was evaluated for access to the devices, telemetry data, and dashboard-like interfaces. The permissions and observations are summarized as follows:

Action	Alice (Admin)	Bob (Operator)	Charlie (Viewer)
Authenticate to REST API	Allowed	Allowed	Allowed
View assigned devices	Allowed	Allowed	Denied
Retrieve telemetry via REST	Allowed	Allowed	Denied
Retrieve telemetry via device token API	Allowed	Allowed	Denied
Modify or create dashboards	Allowed	Denied	Denied
View dashboards (if assigned)	Allowed	Allowed	Denied
Access device credentials	Allowed	Denied	Denied

Detailed Observations

- **Alice (Tenant Admin)**
  Alice successfully accessed and interacted with all the system components. She retrieved telemetry for both temperature and CO_2 devices via REST and token-based APIs. Additionally, she managed the users and configured the dashboards. The logs confirm full visibility and control over the ThingsBoard instance.

- **Bob (Operator)**
  Bob, assigned to a customer with devices explicitly shared, was able to view both devices and their telemetry data. While he could observe telemetry, he was restricted from modifying the devices or retrieving credentials. This aligns with the intended role of operational monitoring .

- **Charlie (Viewer)**
  Despite successful authentication, Charlie has no devices or dashboards assigned to his customer account. As expected, attempts to retrieve telemetry data failed, demonstrating that ThingsBoard effectively denied access to unauthorized users under RBAC policies.

These findings show that **RBAC in ThingsBoard is effective in enforcing hierarchical access**. Users only accessed resources explicitly assigned within their scope. Device access via token-based APIs (used in dashboards) was also role-dependent, further validating the platform’s multi-layered access control system.

CONCLUSION AND FUTURE WORK

This project successfully demonstrated a simulated IoT sensor network with secure access control using RBAC. By integrating a Python-based sensor simulation with the ThingsBoard platform, a functional and secure IoT environment was created. This implementation highlights the importance of RBAC in safeguarding data and controlling access to IoT systems. The key achievements include the following. Real-time simulation of environmental sensors with Python. Seamless integration with an open-source IoT platform. Enforcement of user roles to control access. The benefits of this approach include simplicity, cost efficiency, and real-time control in the educational context. However, this study has some limitations. No physical sensors were deployed; the setup remains a software-only simulation. The RBAC system used was basic and did not support dynamic or context-aware policies. Future work may involve the following. Incorporating blockchain-based logging for auditability, extending the simulation to include real hardware such as Raspberry Pi and environmental sensors. Implementing alerting systems and automated responses based on sensor thresholds. These enhancements would strengthen the real-world applicability of the system and provide a richer testbed for exploring IoT security challenges.

REFERENCES

[1] Zhang, L., Wang, Y., & Chen, M. (2021). Role-Based Access Control in Smart City IoT Deployments. Journal of IoT Security, 8(3), 112-124.

[2] Sandhu, R., Coyne, E. J., Feinstein, H. L., & Youman, C. E. (1996). Role-Based Access Control Models. IEEE Computer, 29(2), 38-47.

[3] Alshahrani, A., & Hussain, F. K. (2020). Security and Privacy in IoT: Current Status and Future Directions. IEEE Access, 8, 83464-83484.

[4] O'Neill, M., & Cullen, C. (2019). Evaluating Open-Source IoT Platforms for Secure Device Management. Proceedings of the 2019 International Conference on Internet of Things Security, 55-62.

APPENDICES

[6] - API Reference: https://demo.thingsboard.io/swagger-ui/index.html

- Installation Guide: https://thingsboard.io/docs/user-guide/install/ubuntu/
- RBAC Access Documentation: https://thingsboard.io/docs/pe/user-guide/rbac/
- GitHub Repository: https://github.com/fares1121/IoT-RBAC-Simulation