Intrusion Detection System in IoT for MQTT Protocol

Presented by :

Abin Patric Melvin - TRV20IT004
Akshara K. Anilkumar - TRV20IT013
Karthik S.S. - TRV20IT041
Rithika Raj - TRV20IT053

Guided by : Dr. Haripriya A. P.

Dept. of Information Technology
Government Engineering College, Barton Hill,Trivandrum

May 9, 2024

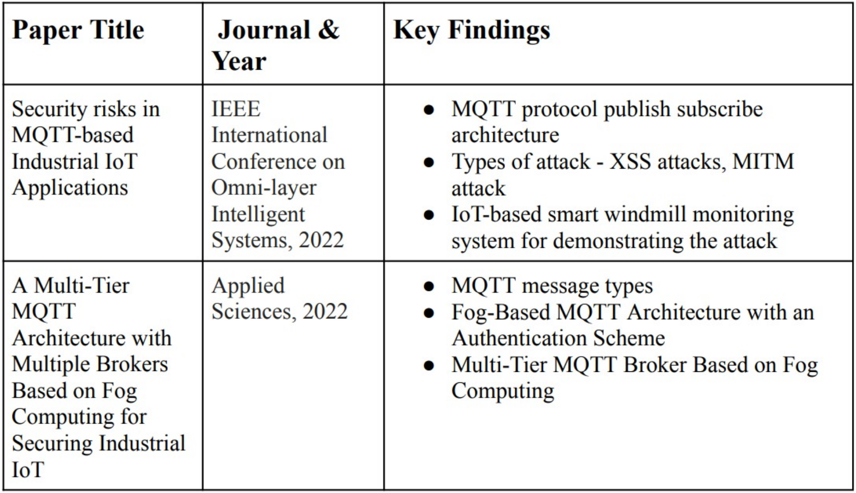
Contents

1. Introduction
2. Literature Survey
3. Problem Statement and Objectives
4. Proposed Solution
5. High Level Design
6. Block Diagram
7. Module Description
8. Tools/ Methodologies Used
9. Result and Analysis
10. Conclusion
11. References

Introduction

- Internet of Things(IoT) is a network of connected devices exchanging data for intelligent decision-making and enhanced efficiency
- Message Queueing Telemetry Transport or MQTT is a protocol that enables efficient real-time communication among IoT devices
- The diverse nature of IoT environments and lightweight nature of MQTT in IoT can lead to various kinds of intrusions and threats among the network
- This amplifies the need for an Intrusion Detection System(IDS) in the IoT-MQTT environment
- IDS detects anomalies and protect sensitive data by travesing the MQTT communication channels

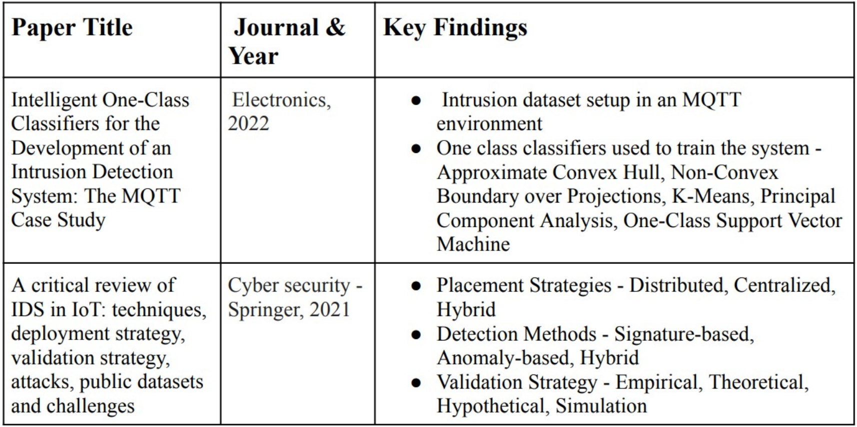
Literature Survey



Paper Title	Journal & Year	Key Findings
Security risks in MQTT-based Industrial IoT Applications	IEEE International Conference on Omni-layer Intelligent Systems, 2022	• MQTT protocol publish subscribe architecture • Types of attack - XSS attacks, MITM attack • IoT-based smart windmill monitoring system for demonstrating the attack
A Multi-Tier MQTT Architecture with Multiple Brokers Based on Fog Computing for Securing Industrial IoT	Applied Sciences, 2022	• MQTT message types • Fog-Based MQTT Architecture with an Authentication Scheme • Multi-Tier MQTT Broker Based on Fog Computing

Table 1 : Literature Survey

Literature Survey



Paper Title	Journal & Year	Key Findings
Intelligent One-Class Classifiers for the Development of an Intrusion Detection System: The MQTT Case Study	Electronics, 2022	• Intrusion dataset setup in an MQTT environment • One class classifiers used to train the system - Approximate Convex Hull, Non-Convex Boundary over Projections, K-Means, Principal Component Analysis, One-Class Support Vector Machine
A critical review of IDS in IoT: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges	Cyber security - Springer, 2021	• Placement Strategies - Distributed, Centralized, Hybrid • Detection Methods - Signature-based, Anomaly-based, Hybrid • Validation Strategy - Empirical, Theoretical, Hypothetical, Simulation

Table 2 : Literature Survey

Problem Statement and Objectives

- **Problem Statement**
  - The diverse nature of IoT environments and lightweight nature of MQTT in IoT can lead to various kinds of intrusions and threats among the network and systems that monitor environmental parameters such as temperature

- **Specific Objectives**
  1. Design and implement a temperature monitoring system using IoT technology
  2. Implement Enhanced Security Alerts within the IoT Ecosystem
  3. Achieve Real-time Threat Detection in Connected Devices
  4. Recognise unauthorised actions and promptly notifying users upon detection

Proposed Solution

- To develop a system for accurately detecting intrusions in an IoT environment utilizing a DHT11 Sensor, and promptly notifying users upon detection
- Create an IoT environment by connecting DHT11 sensor and ESP32 Microcontroller unit.
- ESP32 is then configured to read temperature values, transmit them to brokers and detect intrusions by porogramming them in Arduino IDE
- ESP32 and MQTT Explorer acts as the MQTT Clients where as Mosquitto is the broker acting as intermediate between the clients to transmit information
- Different attacks such as missing key in subscribed topics, subscription to unauthorised topics and external attacks such temperature manipulation will be detected based on the training given to Node MCU.
- Alerts will be genrated based on each of the intrusions and notify users in both Node MCU's output serial monitor as well MQTT Explorer

High Level Design

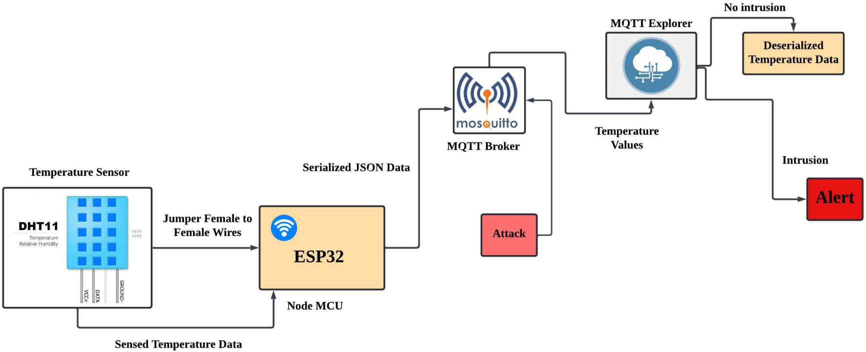


Fig 1: High Level Design of the Proposed Model

Block Diagram

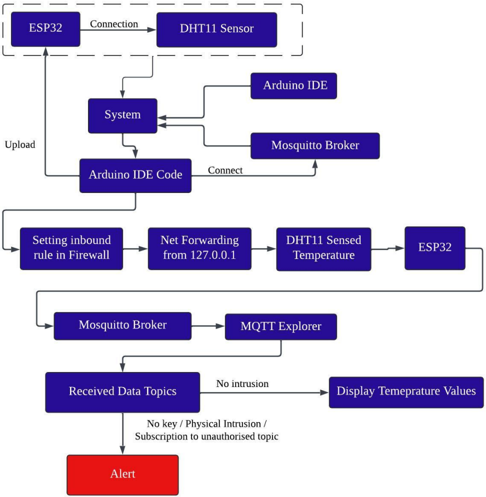


Fig 2: Block Diagram of Working of Proposed Model

Module Description

1) Configuration of IoT Environment

- DHT11 Sensor and ESP32, the Node MCU are connected using female-to-female jumper wires
- The WiFi module of ESP32 is connected to local WiFi network to form an IoT network
- ESP32 is the microcontroller that interfaces the digital output produced by the sensor.
- ESP32 includes a micro USB port for power supply and programming, enabling easy interfacing with computers for code uploading
- This USB Port is connected to port COM4 of the system/laptop to upload code to ESP32
- The software platform used to upload code to ESP32 is Arduino IDE
- The ESP32 Dev Module in Arduino IDE can be used to program the Nodede MCU by choosing the appropriate port on the system

Module Description (Contd.)

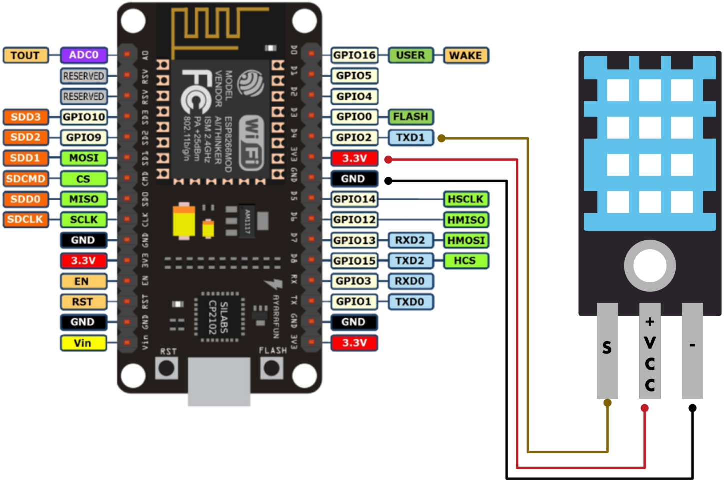


Fig 3: Node MCU - Temperature Sensor Connection

Module Description (Contd.)

2) Setting up MQTT Client-Broker communication

- MQTT is a lightweight messaging protocol designed for IoT devices and applications.
- It is based on the publish-subscribe messaging pattern, making it efficient for communication between distributed systems
- Clients publishing messages as MQTT topics are called publishers and those subscribing them are called subscribers
- MQTT Brokers act as intermediate and receives messages published to topics and delivers them to subscribers based on their subscriptions.
- Mosquitto is an open-source message MQTT broker that is used here to implement the MQTT protocol.
- The MQTT Explorer is the graphical MQTT client tool used for monitoring and interacting with MQTT brokers.

Module Description (Contd.)

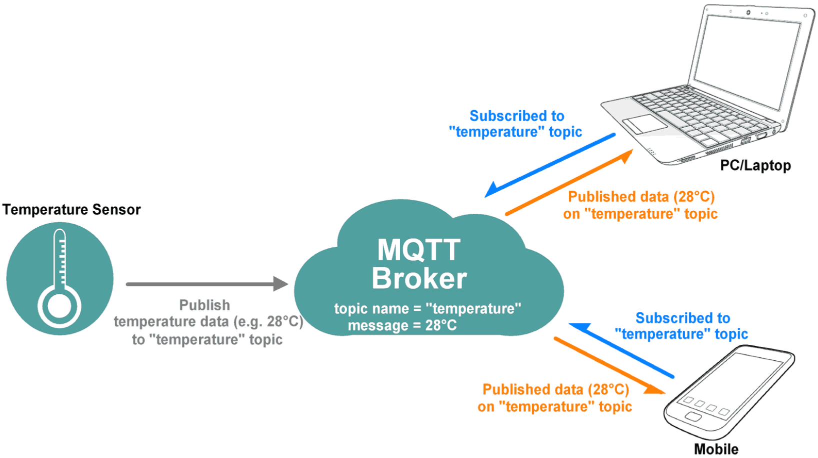


Fig 4: MQTT Client-Broker Communication

Module Description (Contd.)

3) MQTT Topic Handling and Publishing of Sensed Data

- The DHT11 sensor-node collects temperature data, likely in a raw format.
- The MCU would then fetch the data using dht.readTemperature() and serialize this data into a JSON document before publishing it to the MQTT broker.
- This sensed data will be published as a message from MQTT client,i.e, ESP32 by specifying the topic to which the message belongs
- To publish this message, both the payload (the actual data you're sending, e.g., the temperature value) and the topic under which the message should be published have to be provided.
- Once the data is formatted as a JSON document, the MCU publishes it to the MQTT broker.

Module Description (Contd.)

3) MQTT Topic Handling and Publishing of Sensed Data

- MQTT treats the data as a raw stream of bytes converting the JSON document into a byte stream before publishing the data over MQTT.
- The MQTT broker receives the raw byte stream from the MCU and facilitate the transmission of the raw data between publishers (ESP32) and subscribers (MQTT Explorer)
- To subscribe to the published messages using MQTT client,i.e, MQTT Explorer, we have to specify the MQTT topics that we are interested in receiving messages from.
- When the MQTT client,i.e, MQTT Explorer, subscribes to the temperature topic on the MQTT broker, the broker sends the raw data as it received it from the publisher.
- Deserialization occurs on the client side, where the raw byte stream is parsed and converted back into a JSON document for interpretation and display.

Module Description (Contd.)

4) Intrusion Detection and Alerting

- **Case 1 :** Data manipulation by an external physical intruder that causes sudden unexpected value changes in the sensor reading is detected by the system and shown as alert in both serial monitor as well MQTT Explorer's message tab
- **Case 2 :** Messages are exchanged between MQTT clients via Mosquitto broker along with a secret key. Whenever a message subscribed from a topic arrives without a key, it will be notified to users via alert in both serial monitor as well MQTT Explorer's message tab
- **Case 3 :** Subscription to messages from or to unauthorized topics will be detected and generated as an alert in both serial monitor as well MQTT Explorer's message tab

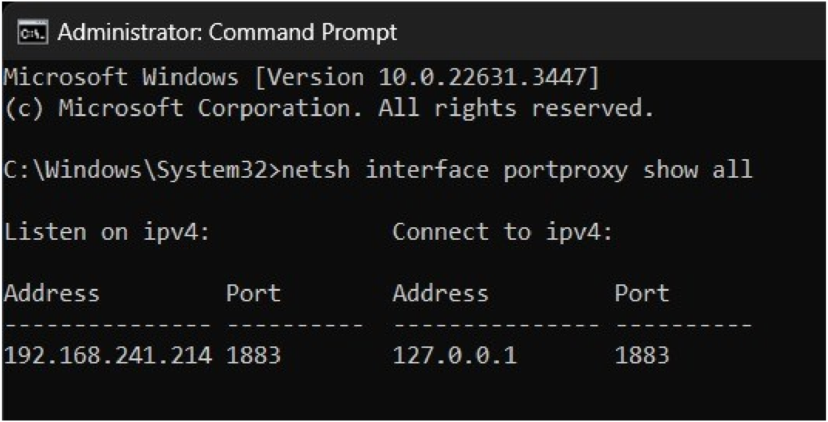
Tools/ Methodologies Used

1. **Hardware :**
   - **ESP32 (Node MCU)** : The microcontroller used for implementing IoT nodes and connecting sensors.
   - **DHT11 Sensor** : Collects temperature and humidity data.
   - **Jumper Wires** : Establish electrical connections between components.
   - **Laptop** : Utilized for development and interfacing with the ESP32 via the COM4 port.
2. **Software :**
   - **Arduino IDE** : Integrated Development Environment for programming the ESP32.
   - **Mosquitto Broker** : Open-source MQTT broker facilitating communication between MQTT clients. .
   - **MQTT Explorer** : MQTT client application for monitoring and interacting with MQTT messages.
   - **Firewall Configuration** : Software-based rules set up to allow incoming messages on port 1883.
   - **Command Prompt** : Software used for Net Forwarding, i.e, to forward messages from localhost to the network.

Tools/ Methodologies Used

3. **Libraries :**
   - **WiFi.h** : Library for connecting the ESP32 to a Wi-Fi network, enabling internet connectivity.
   - **PubSubClient.h** : MQTT client library for the Arduino platform, facilitating communication with the MQTT broker.
   - **ArduinoJson.h**: Library for parsing and generating JSON data, commonly used for handling data exchange in IoT applications.
   - **DHT.h** : Library for interfacing with DHT series sensors like DHT11 to read temperature and humidity data.
4. **Techniques/Methodologies :**
   - **Serialization** : Process of converting sensor data into a suitable format for transmission over the network.
   - **Deserialization** : The process of reconstructing data from its serialized form back into its original format. .
   - **Alert Generation** : Process triggered by the MQTT Explorer client upon detecting anomalies or intrusions in the IoT network.
5. **Programming Languages :**
   - **C** : The primary programming language used for coding the ESP32 microcontroller in the Arduino IDE.

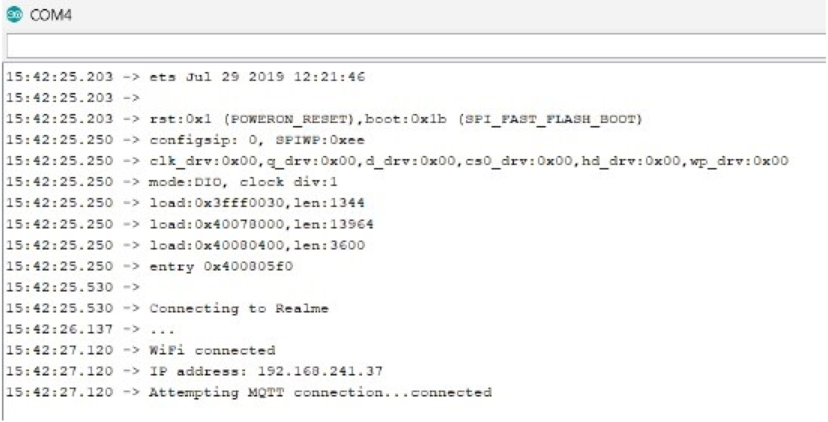
Result and Analysis



```
Microsoft Windows [Version 10.0.22631.3447]
(c) Microsoft Corporation. All rights reserved.

C:\Windows\System32>netsh interface portproxy show all

Listen on ipv4:             Connect to ipv4:

Address         Port        Address         Port
--------------- ----------  --------------- ----------
192.168.241.214 1883        127.0.0.1       1883
```

Fig 5: Net Forwarding

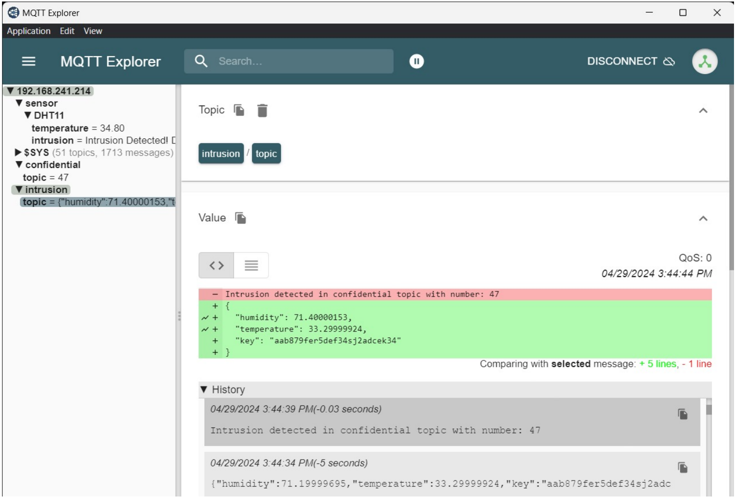
Result and Analysis (Contd.)



```
15:42:25.203 -> ets Jul 29 2019 12:21:46
15:42:25.203 ->
15:42:25.203 -> rst:0x1 (POWERON_RESET),boot:0x1b (SPI_FAST_FLASH_BOOT)
15:42:25.250 -> configsip: 0, SPIWP:0xee
15:42:25.250 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
15:42:25.250 -> mode:DIO, clock div:1
15:42:25.250 -> load:0x3fff0030,len:1344
15:42:25.250 -> load:0x40078000,len:13964
15:42:25.250 -> load:0x40080400,len:3600
15:42:25.250 -> entry 0x400805f0
15:42:25.530 ->
15:42:25.530 -> Connecting to Realme
15:42:26.137 -> ...
15:42:27.120 -> WiFi connected
15:42:27.120 -> IP address: 192.168.241.37
15:42:27.120 -> Attempting MQTT connection...connected
```

Fig 6: WiFi Module Connection

Result and Analysis (Contd.)



```
▼ 192.168.241.214
  ▼ sensor
    ▼ DHT11
      temperature = 34.80
      intrusion = Intrusion Detected! D
  ▶ $SYS (51 topics, 1713 messages)
  ▼ confidential
    topic = 47
  ▼ intrusion
    topic = {"humidity":71.40000153,"t
```

QoS: 0

04/29/2024 3:44:44 PM

```
 −  Intrusion detected in confidential topic with number: 47
 +  {
 +    "humidity": 71.40000153,
 +    "temperature": 33.29999924,
 +    "key": "aab879fer5def34sj2adcek34"
 +  }
```

Comparing with **selected** message: + 5 lines, - 1 line

04/29/2024 3:44:39 PM(-0.03 seconds)

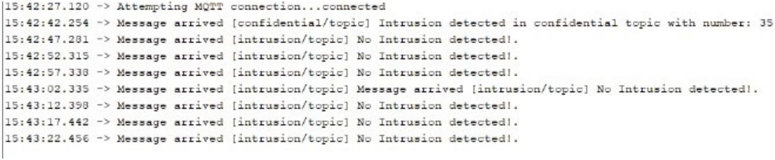
```
Intrusion detected in confidential topic with number: 47
```

04/29/2024 3:44:34 PM(-5 seconds)

```
{"humidity":71.19999695,"temperature":33.29999924,"key":"aab879fer5def34sj2adc
```

Fig 7: Temperature Reading in MQTT Explorer

Result and Analysis (Contd.)



```
15:42:27.120 -> Attempting MQTT connection...connected
15:42:42.254 -> Message arrived [confidential/topic] Intrusion detected in confidential topic with number: 35
15:42:47.281 -> Message arrived [intrusion/topic] No Intrusion detected!.
15:42:52.315 -> Message arrived [intrusion/topic] No Intrusion detected!.
15:42:57.338 -> Message arrived [intrusion/topic] No Intrusion detected!.
15:43:02.335 -> Message arrived [intrusion/topic] Message arrived [intrusion/topic] No Intrusion detected!.
15:43:12.398 -> Message arrived [intrusion/topic] No Intrusion detected!.
15:43:17.442 -> Message arrived [intrusion/topic] No Intrusion detected!.
15:43:22.456 -> Message arrived [intrusion/topic] No Intrusion detected!.
```

Fig 8: Temperature Reading in Serial Monitor

Result and Analysis (Contd.)

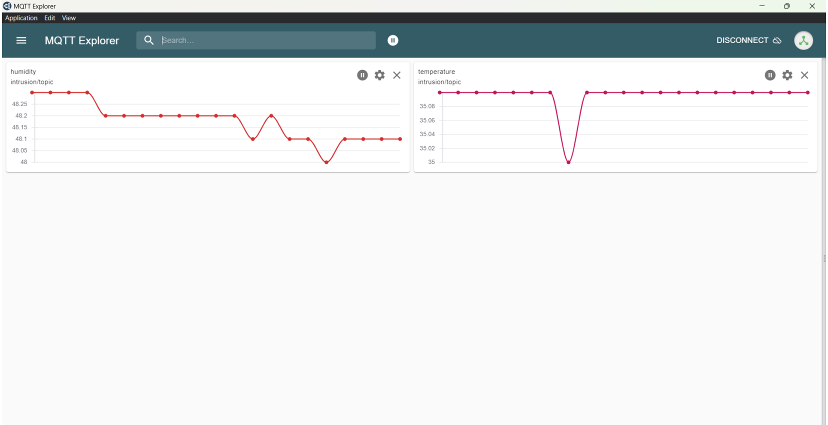
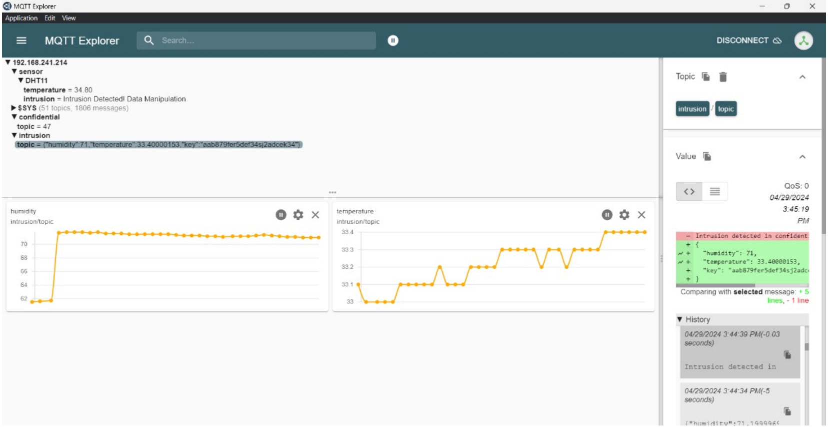


Fig 9: Pattern of Sensed Data in MQTT Explorer

Result and Analysis (Contd.)



```
▼ 192.168.241.214
  ▼ sensor
    ▼ DHT11
      temperature = 34.80
      intrusion = Intrusion Detected! Data Manipulation
  ▶ $SYS (51 topics, 1806 messages)
  ▼ confidential
    topic = 47
  ▼ intrusion
    topic = {"humidity":71,"temperature":33.40000153,"key":"aab879fer5def34sj2adcek34"}
```

QoS: 0
04/29/2024
3:45:19
PM

```
 - Intrusion detected in confident
 + {
 +   "humidity": 71,
 +   "temperature": 33.40000153,
 +   "key": "aab879fer5def34sj2adc
 + }
```

Comparing with **selected** message: + 5 lines, - 1 line

▼ History

04/29/2024 3:44:39 PM(-0.03 seconds)

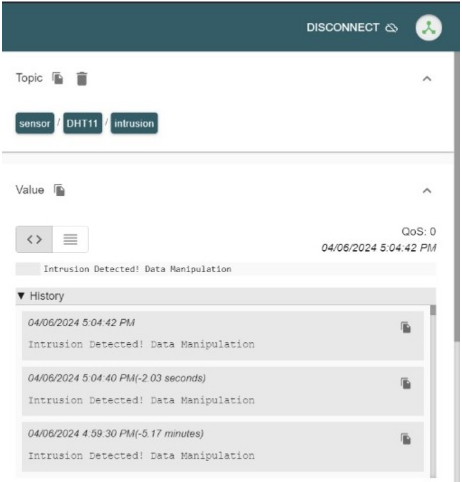
```
Intrusion detected in
```

04/29/2024 3:44:34 PM(-5 seconds)

```
{"humidity":71.199996
```

Fig 10: Pattern of Sensed Data in MQTT Explorer

Result and Analysis (Contd.)



QoS: 0

04/06/2024 5:04:42 PM

```
Intrusion Detected! Data Manipulation
```

04/06/2024 5:04:42 PM

```
Intrusion Detected! Data Manipulation
```

04/06/2024 5:04:40 PM(-2.03 seconds)

```
Intrusion Detected! Data Manipulation
```

04/06/2024 4:59:30 PM(-5.17 minutes)

```
Intrusion Detected! Data Manipulation
```

Fig 11: Alert in case of physical intrusion

Conclusion

- The implementation of an IDS in IoT for MQTT represents a crucial step towards fortifying the security infrastructure of IoT devices
- By actively monitoring MQTT communication, the IDS can guard against threats, ensuring data integrity and confidentiality
- This research project has focused on the development of an IDS tailored specifically for IoT environments utilizing the MQTT protocol, with a primary emphasis on temperature sensor data.
- We designed a solution that addresses some of the challenges of securing IoT networks while considering the resource constraints inherent in such environments.
- Our findings underscore the significance of leveraging lightweight protocols like MQTT for efficient communication in IoT systems, highlighting the critical role of IDS in safeguarding against potential threats.

References

[1] Jove E, Aveleira-Mata J, Alaiz-Moretón H, Casteleiro-Roca JL, Marcos del Blanco DY, Zayas-Gato F, Quintián H, Calvo-Rolle JL. Intelligent one-class classifiers for the development of an intrusion detection system: the MQTT case study. Electronics. 2022 Jan 30;11(3):422.

[2] Alsoufi MA, Razak S, Siraj MM, Nafea I, Ghaleb FA, Saeed F, Nasser M. Anomaly-based intrusion detection systems in iot using deep learning: A systematic literature review. Applied sciences. 2021 Sep 9;11(18):8383.

[3] Boppana TK, Bagade P. Security Risks in MQTT-Based Industrial IoT Applications. In2022 IEEE International Conference on Omni-layer Intelligent Systems (COINS) 2022 Aug 1 (pp. 1-5). IEEE.

[4] Kurdi H, Thayananthan V. A Multi-Tier MQTT architecture with multiple brokers based on fog computing for securing industrial IoT. Applied Sciences. 2022 Jul 16;12(14):7173.

References

[5] Ciklabakkal E, Donmez A, Erdemir M, Suren E, Yilmaz MK, Angin P. ARTEMIS: An intrusion detection system for MQTT attacks in Internet of Things. In2019 38th Symposium on Reliable Distributed Systems (SRDS) 2019 Oct 1 (pp. 369-3692). IEEE.

[6] Khraisat A, Alazab A. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges. Cybersecurity. 2021 Dec;4:1-27.

[7] Husnain M, Hayat K, Cambiaso E, Fayyaz UU, Mongelli M, Akram H, Ghazanfar Abbas S, Shah GA. Preventing mqtt vulnerabilities using iot-enabled intrusion detection system. Sensors. 2022 Jan 12;22(2):567.

Thank you.