

# Securing MQTT-Based SCADA Systems: Threats, Risks, and Defensive Best Practices

This document provides a comprehensive defensive security overview for systems that use MQTT in SCADA environments. It focuses on understanding risks, recognizing common categories of threats at a high level, and implementing practical protections. This guide deliberately avoids procedural instructions for carrying out cyberattacks, and instead emphasizes how to safeguard and harden systems.

## 1. Overview of MQTT and SCADA Security Context

SCADA systems control and monitor industrial processes. MQTT is a lightweight publish/subscribe protocol frequently used for telemetry and control messaging. Because SCADA systems often interact with the physical world, system compromise can result in safety, financial, operational, or environmental damage. Security must therefore prioritize confidentiality, integrity, availability, and safety.

## 2. Threat Landscape (High-Level Categories)

- Unauthorized access to brokers or dashboards due to weak authentication or misconfiguration.
- Eavesdropping on unencrypted MQTT traffic leading to data exposure.
- Message tampering or manipulation that changes values or control messages during transit.
- Impersonation of legitimate devices or clients, including spoofed client identifiers.
- Replay of previously valid messages in the absence of integrity validation or timestamps.
- Excessive traffic causing resource exhaustion and denial of service.
- Exploitation of insecure web dashboards or APIs associated with SCADA HMIs.
- Misuse of default credentials or overly permissive access roles.
- Accidental operator actions due to lack of safeguards or confirmation mechanisms.
- Physical security weak points such as unsecured edge devices and field equipment.

## 3. Risks of Using MQTT Without Security Controls

When MQTT is deployed without encryption or authentication, traffic may be readable or modifiable in transit. Sensitive process data, credentials, and system states can be exposed. In extreme scenarios, control messages could be altered, delayed, or replayed, affecting industrial processes. Even where safety systems exist, unnecessary risk is introduced into operational environments.

## 4. Warning Signs and Indicators of Compromise

- Unexpected device disconnects, reconnect storms, or unfamiliar client identifiers.
- Unusual traffic volume increases or unexpected retained messages.
- Dashboard or HMI sessions appearing from unknown locations or times.
- Unexpected changes in set-points, states, or operator logs.
- Alarms or logs showing repeated authentication failures.

- Devices behaving erratically or reporting inconsistent values.

## **5. Defensive Best Practices and Hardening Measures**

- Use TLS encryption for MQTT communication to protect confidentiality and integrity in transit.
- Disable anonymous access and require strong authentication for all clients and users.
- Implement role-based authorization with least-privilege principles for topics and commands.
- Apply per-topic Access Control Lists to restrict which entities may publish or subscribe.
- Rotate credentials and certificates regularly, and revoke compromised identities promptly.
- Segment networks so that SCADA and critical systems are not directly exposed to the public internet.
- Use VPNs, private networks, or dedicated industrial network segments for field device traffic.
- Keep MQTT brokers, dashboards, and dependencies regularly patched and updated.
- Limit use of wildcard topics and large retained messages to minimize blast radius.
- Log and monitor connection attempts, topic access, and configuration changes and enable alerts.

## **6. Secure Development and Configuration Practices**

When extending or customizing SCADA software such as FUXA, apply secure development life-cycle practices. Avoid hard-coded secrets, validate all external inputs, implement server-side authorization checks, and perform dependency vulnerability scanning. Provide strong session handling and protect API endpoints that interact with control or configuration functions.

## **7. Incident Response and Recovery Preparedness**

- Maintain regular offline backups of configuration and critical data.
- Create an incident response plan defining roles, responsibilities, and contact paths.
- Test backup restoration and failover procedures periodically.
- Document asset inventory and network topology to support rapid containment actions.
- Ensure safety interlocks and manual overrides exist outside the software layer.

## **8. Relevant Security Frameworks and Standards**

Organizations may reference industrial cybersecurity standards such as IEC 62443, NIST guidance for operational technology, or sector-specific regulations. These frameworks reinforce defense-in-depth strategies and structured risk management across people, process, and technology.