УДК 004.738

Н. О. Черневський

ВРАЗЛИВОСТІ ПРОТОКОЛУ MQTT ТА МЕТОДИ ЇХ УСУНЕННЯ В IOT-СИСТЕМАХ

Вінницький національний технічний університет

Анотація

У роботі розглянуто основні вразливості протоколу MQTT, що широко застосовується в системах Інтернету речей. Проаналізовано загрози на рівні транспорту, автентифікації та управління доступом до топіків. Запропоновано практичні методи захисту: використання TLS шифрування, сертифікатної автентифікації, налаштування ACL на брокері та моніторингу трафіку.

Ключові слова: IoT, MQTT, кібербезпека, TLS, брокер повідомлень, вразливості протоколу.

Abstract

The paper examines the main vulnerabilities of the MQTT protocol widely used in Internet of Things systems. Threats at the transport, authentication, and topic access control levels are analyzed. Practical security methods are proposed: TLS encryption, certificate-based authentication, broker-level ACL configuration, and traffic monitoring.

Keywords: IoT, MQTT, cybersecurity, TLS, message broker, protocol vulnerabilities.

Вступ

MQTT (Message Queuing Telemetry Transport) — це протокол обміну повідомленнями на базі моделі "видавець–підписник", розроблений IBM у 1999 році для зв'язку з пристроями в умовах нестабільного мережевого з'єднання. Сьогодні він є одним із ключових транспортних протоколів в IoT системах, його підтримують платформи AWS IoT, Azure IoT Hub, Google Cloud IoT, а також тисячі комерційних брокерів на кшталт Mosquitto та HiveMQ.

Популярність MQTT зумовлена мінімальним накладним трафіком, заголовок пакета займає лише 2 байти, що критично для мікроконтролерів з обмеженою пам'яттю та каналів з вузькою смугою пропускання. Однак за простотою ховається серйозна проблема, протокол у базовій специфікації не передбачає ні шифрування, ні обов'язкової автентифікації. Ці особливості перетворюють недбало розгорнуту MQTT інфраструктуру на зручну точку входу для зловмисника.

У відкритому доступі в мережі Інтернет постійно знаходяться десятки тисяч незахищених MQTT брокерів. Частина з них керує реальним промисловим обладнанням, системами опалення будівель або медичними сенсорами. Мета цієї роботи систематизувати типові вразливості протоколу та описати практичні методи їх усунення.

Аналіз вразливостей протоколу

Перша і найбільш критична проблема це відсутність шифрування за замовчуванням. Стандартний MQTT трафік передається відкритим текстом через TCP порт 1883. Будь-який учасник мережевого сегмента здатний перехопити трафік. Наприклад, через ARP-спуфінг у локальній мережі або моніторинг публічного Wi-Fi зловмисник отримає повний доступ до змісту повідомлень.

Друга суттєва вразливість це механізм автентифікації. Специфікація MQTT дозволяє підключатися до брокера з порожніми полями username та password, і більшість брокерів у конфігурації за замовчуванням таке підключення приймають. Навіть коли паролі використовуються, вони передаються у відкритому вигляді в полі CONNECT пакета, що робить їх легко перехоплюваними без TLS. Крім того, специфікація не регламентує формат або мінімальну складність пароля.

Третя проблема це відсутність контролю доступу до топіків. У MQTT топік є простим рядком виду home/floor2/sensor/temperature, а символи # та + є wildcards для підписки. Клієнт без обмежень може підписатися на # і отримати всі повідомлення брокера, або публікувати довільні дані в будь-який топік, включаючи системні. Це відкриває можливість для атак типу data injection (підміни показань датчиків або надсилання хибних команд виконавчим пристроям).

Небезпека Retained Messages полягає в тому, що зловмисник може зчитати актуальні критичні дані (наприклад, останній переданий пароль, стан замка чи конфігурацію системи), або ж опублікувати

шкідливе Retained-повідомлення, яке буде застосовуватися до всіх нових легітимних клієнтів відразу після їх підключення.

Також варто розуміти, що MQTT не обмежує частоту CONNECT запитів, масове підключення дешевих ботів здатне вичерпати ресурси брокера і вивести з ладу всю мережу пристроїв.

Методи усунення вразливостей

Базовим і обов'язковим кроком є увімкнення TLS шифрування на порту 8883. TLS 1.2 або 1.3 усуває загрозу перехоплення трафіку та підміни брокера. На практиці сертифікати можна отримати безкоштовно через Let's Encrypt для хмарних брокерів або згенерувати власний CA для ізольованих промислових мереж. Важливо налаштувати взаємну автентифікацію (mTLS), при якій брокер перевіряє не лише себе перед клієнтом, а й клієнт перед брокером за допомогою X.509-сертифіката. Це повністю виключає підключення неавторизованих пристроїв.

Для управління доступом до топіків слід налаштувати Access Control List на рівні брокера. У Mosquitto це реалізується через директиву acl_file, де кожному користувачу явно перераховуються дозволені топіки і тип доступу read або write. Доступ до wildcard підписок типу # слід заборонити для всіх клієнтів, крім внутрішніх моніторингових агентів. Принцип мінімальних привілеїв: датчик температури повинен мати право публікувати лише у свій топік і не може нічого більше.

Для захисту від DoS атак рекомендується налаштувати rate limiting на рівні мережевого екрана та самого брокера. HiveMQ та Mosquitto підтримують обмеження кількості CONNECT запитів за одиницю часу і максимальну кількість одночасних підключень з однієї IP адреси. Додатково слід вимкнути anonymous access і явно задати max_keepalive, щоб уникнути накопичення неактивних з'єднань.

Для зниження ризиків, пов'язаних із Retained Messages, необхідний регулярний аудит топіків. Через підписку на # (з адміністративного акаунта) є можливість виявити застарілі або підозрілі повідомлення. У MQTT 5.0 з'явилася можливість вказати Message Expiry Interval безпосередньо в пакеті, що автоматично очищає застарілі дані.

На рівні архітектури ефективним є ізоляція брокера у DMZ та використання VPN тунелів для підключення пристроїв. Розгортання MQTT брокера у внутрішній мережі без прямого доступу з Інтернету суттєво звужує поверхню атаки. Для виявлення аномалій рекомендується інтеграція з SIEM системами (незвичні патерни публікацій, масові підписки або нові клієнтські ID є надійними індикаторами компрометації).

Висновки

Отже, більшість вразливостей MQTT є наслідком не архітектурних дефектів протоколу, а неправильного налаштування. Базова специфікація свідомо залишає реалізацію безпеки на розсуд розробника. За умов швидкого прототипування IoT проектів це часто призводить до пропуску критичних кроків налаштування. Мінімально необхідний захищений стек включає: TLS 1.3 з mTLS, заборону анонімного доступу, ACL з мінімальними привілеями та rate limiting на брокері. Ці чотири заходи перекривають переважну більшість реальних векторів атак і можуть бути впроваджені без зміни клієнтського коду пристроїв.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. MQTT Specification v5.0. URL: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html (дата звернення: 27.05.2026).

2. Mosquitto MQTT Broker Documentation. URL: https://mosquitto.org/documentation/ (дата звернення: 27.05.2026).

3. Shodan Search Engine — Open MQTT Brokers. URL: https://www.shodan.io (дата звернення: 27.05.2026).

4. HiveMQ Security Guide for MQTT. URL: https://www.hivemq.com/mqtt-security-fundamentals/ (дата звернення: 27.05.2026).

Черневський Назар Олександрович — студент групи 2КІ-25м, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький Національний Технічний Університет, Вінниця, e-mail: chernevskijnazar@gmail.com

Chernevskyi Nazar Oleksandrovich — student of group 2KI-25m, faculty of information technologies and computer engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: chernevskijnazar@gmail.com