КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (27), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.27.727
УДК 004.056.5:004.75+004.738.5

Онищук Оксана Олександрівна
кандидат технічних наук, доцент
Волинський національний університет, Луцьк, Україна
ORCID ID: 0000-0002-8342-3011
oksanaoo2024@gmail.com

КІБЕРБЕЗПЕКА MQTT-З'ЄДНАНЬ НА ПРИКЛАДІ СИСТЕМИ АВТОМАТИЗАЦІЇ ВОРІТ

Анотація. З розвитком Інтернету речей (IoT) постає питання забезпечення захисту даних та безпечного функціонування IoT-систем. Однією з основних загроз є незахищені MQTT-з'єднання, які можуть бути вразливими до перехоплення трафіку (MitM-атаки), підробки команд (spoofing), несанкціонованого доступу та DDoS-атак. У цій статті досліджуються методи захисту MQTT-з'єднань на прикладі автоматизованої системи керування воротами. Представлено аналіз останніх досліджень у сфері кібербезпеки IoT, визначено основні вразливості MQTT-брокерів та клієнтів, а також запропоновано заходи для захисту IoT-інфраструктури. Особлива увага приділяється застосуванню TLS/SSL для шифрування трафіку, використанню автентифікації клієнтів MQTT, обмеженню доступу за допомогою ACL (Access Control List) та ізоляції IoT-пристроїв у окремих мережах (VPN/VLAN). Результати дослідження підтверджують, що комплексне впровадження заходів безпеки дозволяє значно знизити ризики атак та забезпечити надійну і безпечну роботу IoT-проєктів.

Ключові слова: Інтернет речей (IoT); захист даних; MQTT-з'єднання; TLS/SSL; контролер; датчик; сканер.

ВСТУП

Інтернет речей (IoT) активно впроваджується у різні сфери життя, автоматизуючи процеси та підвищуючи комфорт користувачів. Проте разом із перевагами з'являються і серйозні виклики, пов'язані з кібербезпекою. Багато IoT-пристроїв не мають вбудованих механізмів захисту, що робить їх вразливими до атак. Однією з ключових проблем є MQTT-з'єднання — протокол обміну повідомленнями, що використовується для комунікації між пристроями. Без належного рівня захисту зловмисники можуть перехоплювати трафік, підробляти команди або отримувати несанкціонований доступ до системи. Це особливо критично для автоматизованих систем, таких як автоматичні ворота, де атака може призвести до порушення безпеки об'єкта або фізичного доступу сторонніх осіб. У цій статті проаналізовано вразливості IoT-проєктів, пов'язані з MQTT-з'єднанням, а також методи його захисту на прикладі системи автоматизації воріт.

Постановка проблеми. Розглянемо проєкт, який використовує MQTT для обміну повідомленнями між пристроями, а також кібербезпеку передачі даних. MQTT забезпечує гнучкий і не залежний від даних підхід до публікації повідомлень між клієнтами та брокерами. Використовуючи теми для фільтрації повідомлень, клієнти можуть швидко та легко підписатися на вміст, який їх цікавить. Корисне навантаження кожного повідомлення можна налаштувати відповідно до конкретних потреб кожного клієнта, а підтримка MQTT різних типів даних робить його універсальним рішенням для багатьох випадків використання. Крім того, розуміння атрибутів повідомлення PUBLISH, таких як рівень QoS і прапор збереження, може допомогти клієнтам і брокерам забезпечити ефективну та надійну доставку повідомлень. MQTT не залежить

від даних, що означає, що корисне навантаження можна структурувати відповідно до конкретного випадку використання клієнта. Корисне навантаження — це і єосновний вміст повідомлення, на що клієнти підписуються, отримують і обробляють.

Аналіз останніх досліджень і публікацій. З розвитком Інтернету речей (IoT) значно зросла увага до питань безпеки, зокрема захисту MQTT-з'єднань. У сучасних наукових дослідженнях та інженерних публікаціях розглядаються основні загрози для IoT-систем, а також ефективні методи їхнього усунення та безпека MQTT-протоколу [1], [3], [4]. Розглянемо автоматизовану систему керування воротами, яка використовує MQTT для комунікації між сенсорами, контролерами та виконавчими пристроями.

Метою статті є аналіз і порівняння рівня кібербезпеки IoT-проєктів, зокрема захисту MQTT-з'єднань на прикладі системи автоматизації воріт. Для досягнення цієї мети необхідно визначити основні вразливості MQTT-з'єднання, дослідити методи шифрування та аутентифікації в MQTT, провести порівняльний аналіз існуючих підходів до безпеки IoT та розробити практичні рекомендації щодо безпечного використання MQTT у проєктах автоматизації.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

MQTT є одним із найпопулярніших протоколів, проте його базова конфігурація не передбачає шифрування або аутентифікації. У статті [2] проаналізовано основні атаки на MQTT-брокер, серед яких:Man-in-the-Middle (MitM), тобто перехоплення MQTT-трафіку, Spoofing-атаки, тобто підробка MQTT-повідомлень, Denial of Service (DoS), а саме перевантаження брокера запитами.

Дослідження [2], [3] демонструє, що TLS/SSL-шифрування та використання ACL (Access Control List) значно знижують ризик атак на MQTT-брокер. Крім того, впровадження сертифікатів безпеки [4], [5] та ідентифікації клієнтів на рівні брокера може повністю усунути несанкціонований доступ до пристроїв. Згідно з [4], [6], рекомендується поєднання методів безпеки, зокрема: MQTTS (MQTT + TLS/SSL) — шифрування даних, аутентифікація та авторизація користувачів MQTT, обмеження доступу до критичних тем MQTT за ACL [6] – [8]. Таким чином, аналіз останніх досліджень підтверджує, що MQTT-з'єднання є вразливим без належного рівня захисту, проте застосування комплексних заходів може усунути більшість загроз [9] – [11].

МЕТОДИКА ДОСЛІДЖЕННЯ

MQTT (Message Queuing Telemetry Transport) — це легкий протокол обміну повідомленнями, який працює за принципом підписки та публікації. Його переваги: мінімальне навантаження на мережу та підходить для IoT-пристроїв, швидка передача даних, оскільки працює у реальному часі, гнучкість через підтримку великої кількості підключених клієнтів [2]. Проте, MQTT не містить вбудованих механізмів безпеки, що робить його вразливим до атак [3].

Проаналізувавши теоретичним методом безпекові аспекти, виявлено що основні загрози в MQTT-з'єднаннях можуть бути несанкціонований доступ: якщо MQTT-брокер відкритий для всіх, зловмисники можуть підключитися та перехопити дані. Наступна загроза — це підробка повідомлень, адже без перевірки автентичності пристроїв можна отримати фальшиві дані. Перехоплення трафіку можливе без шифрування передані

даних, бо можуть бути прочитані третіми особами та DDoS-атаки, коли зловмисники можуть перевантажити брокер фейковими запитами [5].

До методів захисту MQTT-брокера відноситься використання авторизації та аутентифікації, щоб обмежити доступ до MQTT-брокера. Для цього потрібно впровадити логін та пароль для кожного клієнта та використовувати ACL (Access Control List). Access Control List- це так звані списки доступу, які визначають, хто може публікувати або читати певні теми. Щодо методів аутентифікації, то тут використовуємо парольний захист, де кожен пристрій отримує унікальний логін і пароль або цифрові сертифікати, де кожен клієнт використовує сертифікати безпеки для перевірки автентичності. Дана ACL (Access Control List) виступає як система розмежування прав доступу до тем MQTT. Це означає, що тільки адміністратор може відкривати і закривати ворота, а звичайні сенсори можуть лише надсилати дані.

Приклад конфігурації для Mosquitto [3], [5] (файл mosquitto.conf): allow_anonymous false
password_file /etc/mosquitto/passwd

Створення пароля [3], [5]:
mosquitto_passwd -c /etc/mosquitto/passwd username

Наступним методом є шифрування з’єднання (SSL/TLS). Звичайне MQTT-з’єднання передає дані у відкритому вигляді, що небезпечно, а використання TLS (Transport Layer Security) гарантує, що всі повідомлення будуть зашифровані. Це дозволяє використовувати MQTT через TLS (MQTTS), що значно покращує безпеку.

Налаштування Mosquitto для використання SSL [3], [5]:
listener 8883
cafile /etc/mosquitto/ca.crt
certfile /etc/mosquitto/server.crt
keyfile /etc/mosquitto/server.key
require_certificate true

Ще один з методів захисту є обмеження доступу до критичних тем. Якщо брокер містить важливі теми (наприклад, керування пристроями), потрібно обмежити доступ. Оскільки адміністратори можуть публікувати в темах керування (control/relay); а звичайні пристрої можуть тільки читати повідомлення з sensors/#.

Приклад ACL для Mosquitto (acl_file) [3], [5]:
user admin
topic write control/relay
topic read sensors/#
user device1
topic read sensors/temperature
topic read sensors/humidity

Останнім методом захисту від DDoS-атак є необхідність уникати перевантаження сервера. Для цього встановлюємои ліміти підключень (наприклад, не більше 10 підключень від одного IP) та включаємо автоматичне відключення неактивних клієнтів та відстежуємо аномальну активність за допомогою логів MQTT.

Приклад обмежень у mosquitto.conf [3], [5]:
max_connections 100
max_inflight_messages 20
autosave_interval 180

Як видно з табл. 1 найефективніший підхід — це поєднання кількох методів: TLS/SSL + ACL + автентифікація клієнтів. Аналітично визначено ключові переваги та недоліки кожної платформи з точки зору захисту інформації та можливостей виявлення доказових даних.

Таблиця 1

Порівняльний аналіз підходів до безпеки IoT

Метод захисту	Ефективність	Переваги	Недоліки
Паролі (логін/пароль)	Середня	Простота реалізації	Легко підібрати (Brute Force)
ACL (обмеження доступу до тем)	Висока	Гнучке налаштування доступу	Не запобігає MitM-атакам
TLS/SSL (MQTTS)	Висока	Захищає від MitM-атак	Потребує сертифікатів
VPN + MQTT	Дуже висока	Максимальний рівень захисту	Складна конфігурація

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розглянемо IoT-проєкт на прикладі система автоматизації воріт. Даний проект демонструє основи збору, аналізу та передачі даних з датчиків через MQTT. Він дозволяє зрозуміти принципи взаємодії сенсорів, контролерів та виконавчих пристроїв у реальних системах. Врахуємо додатковий акцент на кібербезпеку, тобто використання авторизації MQTT-брокера, шифрування з'єднань та обмеження доступу до критичних тем.

Опис проєкту. Наша система автоматизованих воріт працює, аналізуючи дискретні сигнали: виявлення автомобіля та сканування картки доступу. У режимі очікування система реагує на появу автомобіля біля воріт, а після аутентифікації картки відкриває ворота. При відкритих воротах система перевіряє чи автомобіль залишив зону детекції. Якщо автомобіль все ще знаходиться поблизу, то ворота залишаються відкритими. Після від'їзду автомобіля система автоматично закриває ворота. Приклад роботи системи наведено на рис. 1.

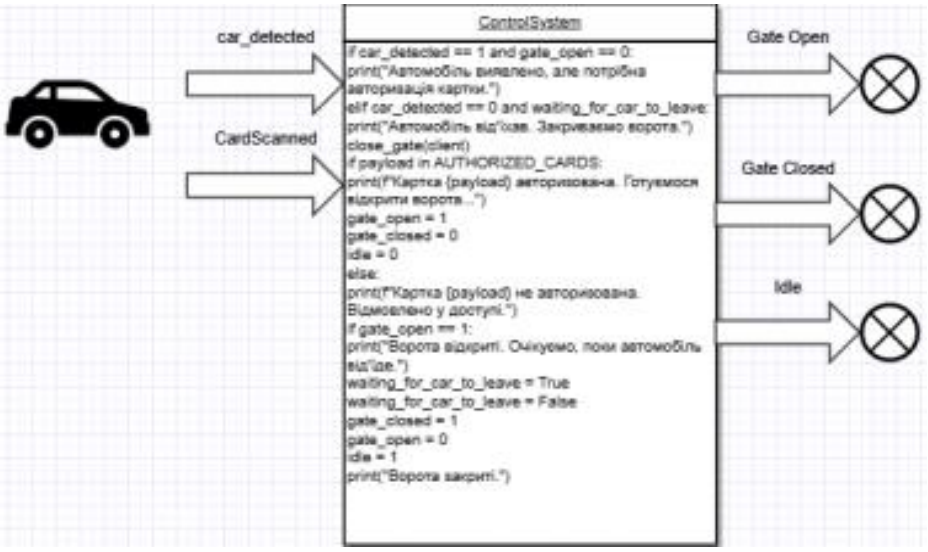

Рис. 1. Блок схема проєкту

Для імітації та моделювання роботи автоматизованої системи воріт використовуємо MQTT. Через протокол MQTT передаються сигнали виявлення автомобіля, авторизації картки доступу та стану воріт. Система обробляє ці сигнали для виконання дій: відкриття, очікування, перевірки та закриття воріт. Для тестування створено блок-схему роботи системи автоматизації воріт на основі MQTT та теми, як показано на рис. 2 та рис. 3.

Система використовує MQTT для комунікації між компонентами, подані в табл. 2. Компоненти системи: датчик наближення автомобіля (ультразвуковий / інфрачервоний); система доступу (RFID або NFC-сканер); контролер (ESP32/ESP8266/Raspberry Pi); привід воріт (електродвигун + реле); MQTT-брокер (Mosquitto/HiveMQ)

Таблиця 2

Комунікація між компонентами

Тема	**Опис**	**Відправник**	**Отримувач**
gate/sensor/detect	Виявлення автомобіля (1 — є авто, 0 — немає)	Датчик автомобіля	Контролер
gate/access/card	Дані зчитаної картки доступу	Сканер карток	Контролер
gate/control/open	Команда на відкриття воріт	Контролер	Привід воріт
gate/control/close	Команда на закриття воріт	Контролер	Привід воріт
gate/status	Статус воріт (open, closed, moving)	Контролер	Всі підписані пристрої

Логіка роботи системи: режим очікування — датчик автомобіля передає 0 (немає автомобіля), тому ворота закриті; виявлення автомобіля- датчик автомобіля передає 1 у gate/sensor/detect, при цьому контролер очікує дані з картки доступу; автентифікація: користувач прикладає картку до зчитувача, при цьому якщо картка є в базі даних → відправляється gate/control/open, то ворота відкриваються; очікування виїзду: поки автомобіль у зоні детекції → ворота залишаються відкритим, коли автомобіль залишає зону → gate/control/close відбувається закриття воріт: ворота закриваються; статус оновлюється (gate/status = closed).

Блок-схема роботи системи: Авто наближається → Датчик виявляє авто → Сканування картки доступу → Перевірка картки → Відкриття воріт → Авто виїжджає → Закриття воріт

1. 1.Очікування автомобіля — система перебуває в режимі очікування.
2. Датчик виявляє автомобіль — надсилає сигнал у MQTT (gate/sensor/detect).
3. Сканування картки доступу — перевірка ідентифікації через gate/access/card.
4. Перевірка аутентифікації — якщо картка правильна, ворота відкриваються (gate/control/open).
5. Очікування виїзду — система перевіряє, чи автомобіль залишив зону детекції.
6. Закриття воріт — якщо автомобіль виїхав, система надсилає команду gate/control/close.
7. Повернення в режим очікування — система готова до нового автомобіля.

Згідно [5], [7], [8] подано відповідно лістинг 1.

```
import paho.mqtt.client as mqtt
import time
BROKER = "mqtt.example.com"  # Адреса вашого MQTT-брокера
PORT = 1883
def on_message(client, userdata, msg):
    topic = msg.topic
    payload = msg.payload.decode()

    if topic == "gate/sensor/detect":
        if payload == "1":  # Виявлено авто
            print("Автомобіль виявлено, очікуємо картку...")

    elif topic == "gate/access/card":
        if payload == "123456":  # Припустимо, що це коректна картка
            print("Картка підтверджена, відкриваємо ворота...")
            client.publish("gate/control/open", "1")

    elif topic == "gate/status" and payload == "open":
        print("Ворота відкриті, чекаємо виїзду авто...")

    elif topic == "gate/sensor/detect" and payload == "0":
        print("Автомобіль поїхав, закриваємо ворота...")
        client.publish("gate/control/close", "1")

client = mqtt.Client()
client.on_message = on_message
client.connect(BROKER, PORT, 60)

client.subscribe("gate/sensor/detect")
client.subscribe("gate/access/card")
client.subscribe("gate/status")
client.loop_forever()
```

Лістинг 1. Код для ESP32 (контролер)

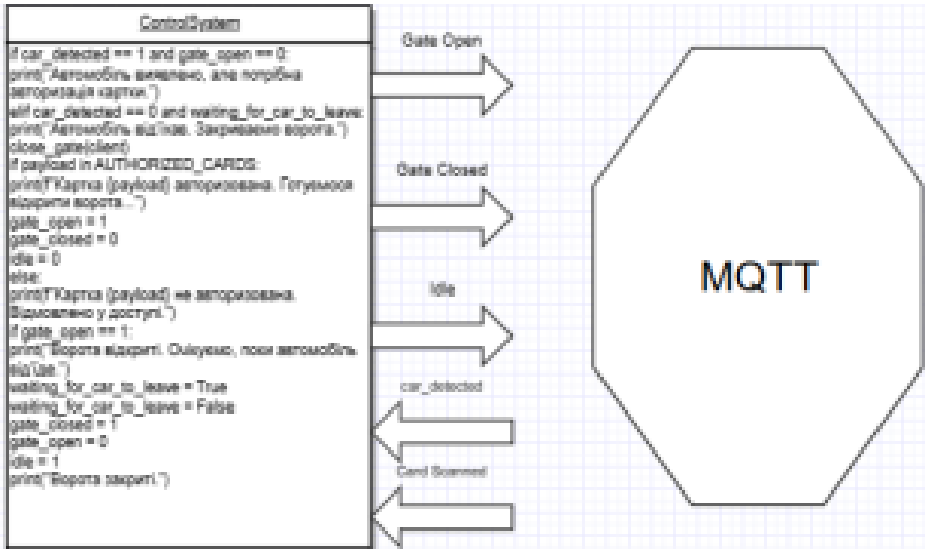

Рис. 2. Блок схема тестового проєкту

ISSN 2663 - 4023

Рис. 3. Теми в MQTT — Explorer

Розглянемо захист системи автоматизації воріт, проблеми безпеки та їх рішення. Врахуємо можливість підробки повідомлень (spoofing), коли без захисту зловмисник може підробити команду gate/control/open, тоді як запобіжний захід можна використати паролі MQTT та перевірити цифрові підписи.

Наступна проблема полягає у можливості перехоплення трафіку (sniffing). Якщо MQTT працює без шифрування, то можна отримати картку доступу, при цьому варто використовувати TLS/SSL (порт 8883). Перехоплення трафіку працює, якщо з'єднання між пристроями не зашифроване. Зловмисник може перехопити MQTT-трафік і бачити всі передані дані, а також він знаходиться в тій самій мережі Wi-Fi, використовує інструменти для перехоплення трафіку (наприклад, Wireshark або tcpdump) та отримує всі MQTT-повідомлення (наприклад, значення температури, стан реле тощо). Для початку варто встановити Wireshark, відкрити інтерфейс мережі, через який передаються MQTT-пакети (наприклад, wlan0) та відфільтрувати MQTT-трафік (фільтр: mqtt). При цьому відображається весь MQTT-трафік, включаючи паролі (якщо вони передаються у відкритому вигляді).

Для захисту треба використовувати TLS (порт 8883), щоб дані передавалися у зашифрованому вигляді, використовувати VPN або окрему підмережу для IoT та включити авторизацію на рівні MQTT-брокера.

Щоб уникнути несанкціонованого доступу, коли будь-який клієнт може підключитися до MQTT, варто використовувати ACL та логіни для MQTT-клієнтів. А також обмежити дозволи: датчики можуть тільки публікувати, а виконавчі пристрої — тільки підписуватися (лістинг 2) [5], [7], [8].

```
allow_anonymous false
password_file /etc/mosquitto/passwd
listener 8883
cafile /etc/mosquitto/ca.crt
certfile /etc/mosquitto/server.crt
keyfile /etc/mosquitto/server.key
Пароль для користувачів:
mosquitto_passwd -c /etc/mosquitto/passwd admin
```

Лістинг 2. Захист MQTT-брокера (Mosquitto mosquitto.conf)

Отже, підсумуємо практичні рекомендації щодо безпечного використання MQTT у проєктах автоматизації. Для захисту варто використовувати шифрування (MQTTS, TLS/SSL), налаштувати TLS (порт 8883) для зашифрованої комунікації, запровадити аутентифікація клієнтів та використовувати логін і пароль або цифрові сертифікати для кожного пристрою. Для захисту MQTT-брокера потрібно встановити Mosquitto ACL для обмеження доступу до критичних тем та вимкнути анонімний доступ (allow_anonymous false). Для використання VPN або окремої VLAN потрібно відокремити IoT-пристрої від загальної мережі. Якщо ж MQTT не захищений, атакувальники можуть змінювати дані сенсорів, викликати помилкові дії системи або навіть перехоплювати важливі повідомлення. Впровадження цих методів значно зменшить ризики атак і забезпечить безпечну роботу розумних пристроїв у нашій IoT-системі.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

У роботі було виявлено, що даний проект демонструє, як можна автоматизувати ворота на основі MQTT. Головні переваги даного з’єднання: швидка реакція, відкриття та закриття в реальному часі, гнучкість, адже можна додати камери; розпізнавання номерів та безпека через шифрування MQTT-з’єднань та контроль доступу.

Дослідження показало, що автоматизовані системи на основі MQTT, такі як система воріт, вразливі до атак без належного захисту. Найбільші загрози — це перехоплення трафіку (MitM), підробка команд (spoofing) та DoS-атаки. Однак, використовуючи такі заходи, як TLS-шифрування, автентифікація клієнтів, контроль доступу та VPN-з’єднання, можна значно підвищити рівень безпеки. Впровадження цих методів гарантує безпеку IoT-проєктів, включаючи автоматизовані системи воріт, та забезпечує захищену передачу даних у реальному часі.

Перспективи подальших досліджень можуть бути у дослідженнях автоматичного виявлення аномалій у MQTT-трафіку, використанні штучного інтелекту для аналізу загроз у IoT, інтеграції з мобільним застосунком, віддалене керування через смартфон та використання RFID/NFC + PIN-коду для підвищеної безпеки.

Таким чином, розробники IoT-рішень повинні інтегрувати механізми безпеки на етапі проектування системи, щоб уникнути потенційних загроз та забезпечити надійний захист IoT-пристроїв у майбутньому. Кібербезпека IoT-систем є критично важливим аспектом їхньої безпечної роботи. Реалізація цих заходів дозволяє значно підвищити рівень безпеки IoT-проєкту та забезпечити захищене передачу даних між пристроями. Інтернет речей має величезний потенціал, але лише за умови належного рівня кібербезпеки ми можемо використовувати його переваги без ризику загроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Information technology – Security techniques – Guidelines for identification, collection, acquisition and preservation of digital evidence (ISO/IEC 27037:2012). (2012).
2. Zhang, X. (2023). Security Challenges in MQTT-Based IoT Systems. *Journal of Cybersecurity and IoT, 15(3),* 45–62.
3. Patel, Y. (2022). Enhancing MQTT Security Using TLS and Access Control Mechanisms. *International Journal of IoT Security Studies, 8(2),* 78–94.
4. IoT Security Research Group. (2023). *Best Practices for Securing MQTT Communications in Industrial IoT. Proceedings of the IoT Security Conference, 2023,* 112–130.
5. *MQTT Specification v5.0.* (2023). MQTT.org. https://mqtt.org

6. *MQTT Security Fundamentals. Technical Report on IoT Communication Security, OASIS Open.* (2022). OASIS.
7. *Eclipse Mosquitto MQTT Broker Security Guide.* (2023). Mosquitto Project. https://mosquitto.org
8. *Detecting and Analyzing MQTT Traffic Using Wireshark.* (2023). Wireshark Foundation. https://wireshark.org
9. Singh, R., & Kumar, P. (2023). Mitigating IoT-Based Attacks through Secure MQTT Implementations. *Cybersecurity & IoT Journal, 10(1),* 23–40.
10. *Implementing VPN and VLAN Isolation for IoT Networks.* (2023). Cisco Systems. https://cisco.com
11. Schneider Electric. (2023). Secure Gate Automation Using MQTT and IoT Protocols. *Industry 4.0 Security Handbook, 14(4),* 98–115.

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (27), 2025
ISSN 2663 - 4023

Oksana Onyshchuk
Candidate of Technical Sciences, Associate Professor
Volyn National University, Lutsk, Ukraine
ORCID ID: 0000-0002-8342-3011
oksanaoo2024@gmail.com

CYBERSECURITY OF MQTT CONNECTIONS IN AN AUTOMATED GATE CONTROL SYSTEM

Abstract. With the development of the Internet of Things (IoT), the issue of data protection and the secure operation of IoT systems has become increasingly important. One of the major threats is unprotected MQTT connections, which are vulnerable to traffic interception (MitM attacks), command spoofing, unauthorized access, and DDoS attacks.This paper explores MQTT security methods using the example of an automated gate control system. It presents an analysis of recent research in IoT cybersecurity, identifies the main vulnerabilities of MQTT brokers and clients, and proposes measures to secure IoT infrastructure.Special attention is given to TLS/SSL encryption for traffic protection, MQTT client authentication, access restrictions using ACL (Access Control List), and the isolation of IoT devices in separate networks (VPN/VLAN).The research findings confirm that a comprehensive implementation of security measures significantly reduces attack risks and ensures the reliable and secure operation of IoT projects.

Keywords: Internet of Things (IoT); data protection; MQTT connection; forensic analysis; TLS/SSL; controller; sensor; scanner.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Information technology – Security techniques – Guidelines for identification, collection, acquisition and preservation of digital evidence (ISO/IEC 27037:2012). (2012).
2. Zhang, X. (2023). Security Challenges in MQTT-Based IoT Systems. *Journal of Cybersecurity and IoT, 15(3),* 45–62.
3. Patel, Y. (2022). Enhancing MQTT Security Using TLS and Access Control Mechanisms. *International Journal of IoT Security Studies, 8(2),* 78–94.
4. IoT Security Research Group. (2023). *Best Practices for Securing MQTT Communications in Industrial IoT. Proceedings of the IoT Security Conference, 2023,* 112–130.
5. *MQTT Specification v5.0.* (2023). MQTT.org. https://mqtt.org
6. *MQTT Security Fundamentals. Technical Report on IoT Communication Security, OASIS Open.* (2022). OASIS.
7. *Eclipse Mosquitto MQTT Broker Security Guide.* (2023). Mosquitto Project. https://mosquitto.org
8. *Detecting and Analyzing MQTT Traffic Using Wireshark.* (2023). Wireshark Foundation. https://wireshark.org
9. Singh, R., & Kumar, P. (2023). Mitigating IoT-Based Attacks through Secure MQTT Implementations. *Cybersecurity & IoT Journal, 10(1),* 23–40.
10. *Implementing VPN and VLAN Isolation for IoT Networks.* (2023). Cisco Systems. https://cisco.com
11. Schneider Electric. (2023). Secure Gate Automation Using MQTT and IoT Protocols. *Industry 4.0 Security Handbook, 14(4),* 98–115.