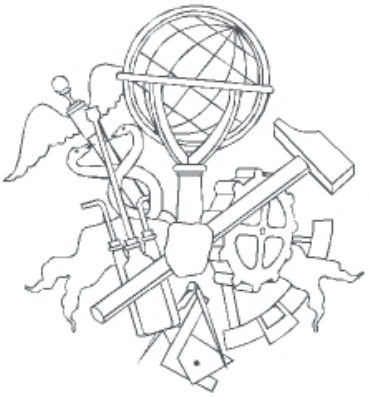
Assingment 1

Report



instituto
superior de
engenharia do
porto

Mar 08, 2023

Contents

List of Acronyms and Definitions **3**

1 MQTT Security **4**

List of Acronyms and Definitions

1 MQTT Security

MQTT stands for Message Queuing Telemetry Transport. It is a lightweight messaging protocol designed for low-bandwidth, high-latency, and unreliable networks. It is commonly used in the Internet of Things (IoT) to enable communication between devices and applications.

MQTT uses a publish/subscribe messaging model, where devices publish messages to topics, and other devices subscribe to topics to receive those messages. Topics are hierarchical, and clients can subscribe to entire hierarchies of topics using wildcards.

MQTT messages consist of a header and a payload. The header contains information such as the topic, message quality of service (QoS), and other metadata. The payload contains the actual message data.

MQTT supports three levels of QoS:

QoS 0: At most once delivery. The message is sent once and is not guaranteed to be delivered. This level of QoS is suitable for non-critical data such as sensor readings. QoS 1: At least once delivery. The message is sent at least once and is guaranteed to be delivered, but may be delivered multiple times. This level of QoS is suitable for critical data such as alarms or alerts. QoS 2: Exactly once delivery. The message is sent exactly once and is guaranteed to be delivered only once. This level of QoS is suitable for mission-critical data such as financial transactions.

security is indeed a critical aspect of any communication protocol, and MQTT provides several security mechanisms to ensure secure communication between devices and applications. Here are some of the security features of MQTT:

Transport Layer Security (TLS): MQTT supports TLS encryption for secure transport of messages over the network. TLS provides encryption, integrity, and authentication, protecting against eavesdropping, tampering, and unauthorized access.

User authentication: MQTT allows devices and applications to authenticate each other using username and password authentication. This helps to prevent unauthorized access to the MQTT broker or topics.

Access control: MQTT brokers can restrict access to topics and messages based on user roles and permissions. This allows fine-grained control over who can publish or subscribe to specific topics.

Message encryption: MQTT messages can be encrypted end-to-end using application-level encryption mechanisms. This provides an additional layer of security, ensuring that the message content is protected even if it is intercepted by an attacker.

Retained messages: MQTT supports retained messages, which are messages that are stored by the broker and are delivered to any new subscribers to the topic. This can be a security risk if the retained message contains sensitive information. MQTT brokers should be configured to limit the use of retained messages.

Security considerations are critical when designing and implementing any communication protocol, especially when it comes to IoT and industrial control systems, where security breaches can have severe consequences.

The MQTT protocol has been designed with security in mind, and several security features have been implemented, as I discussed in my previous answer. However, like any software, there may be vulnerabilities that could be exploited by attackers. Therefore, it is important to keep

an eye on the Common Vulnerabilities and Exposures (CVE) database to understand any known security issues.

After checking the CVE database, I found several CVEs related to MQTT, some of which could be used to exploit the protocol's security. One example is CVE-2017-7652, which is a vulnerability in the Mosquitto MQTT broker that allows an attacker to cause a denial of service (DoS) by sending a specially crafted message to the broker. This vulnerability can be exploited remotely, allowing an attacker to disrupt the MQTT broker's operation, leading to communication disruptions between IoT devices.

Another example is CVE-2017-7653, which is another vulnerability in the Mosquitto MQTT broker that allows an attacker to cause a DoS by sending a specially crafted message that triggers an infinite loop in the broker. This vulnerability can also be exploited remotely, allowing an attacker to disrupt the MQTT broker's operation.

These CVEs show that even though MQTT has been designed with security in mind, there are still vulnerabilities that can be exploited by attackers. It is important to keep MQTT implementations up to date with the latest security patches and to properly configure and use the security features to minimize the risk of attacks.