Assignment 2

Report

isep

instituto
superior de
engenharia do
porto

Carlos Rijo - 1101626

Mar 15, 2023

Contents

List of Acronyms and Definitions

MQTT	Message Queuing Telemetry Transport
CoAP	Constrained Application Protocol
CVE	Common Vulnerabilities and Exposures
DoS	Denial of Service
UDP	User Datagram Protocol
DTLS	Datagram Transport Layer Security
CBOR	Certificate-Based Authentication
PSK	Pre-Shared Key
RFC	Request for Comments

1 CoAP Security

Constrained Application Protocol (CoAP) is a protocol designed for resource-constrained devices and networks which uses User Datagram Protocol (UDP) as its transport layer with RESTful protocol. This protocol provides methods for resource discovery, retrieval, and manipulation. It also provides features for reliable message exchange, resource observation, and group communication. This allows clients to receive updates from servers when the resources change and can be very useful for monitoring data from sensors or other devices in real-time.

CoAP is designed to be a very efficient protocol that minimizes the use of network resources, making it ideal for devices that have limited processing power, memory, and battery life. In terms of security, CoAP provides several security mechanisms (with different security levels) to ensure secure communication between devices. Some examples are:

- Datagram Transport Layer Security (DTLS): Can be use to provide end-to-end security for data transmission between devices since it provides encryption, authentication, and integrity protection of messages.
- Resource access control: The protocol allows for access control of resources based on user identity and permissions.
- Secure group communication: Secure multicast and group communication.
- Security profiles: CoAP defines several security profiles that can be used to specify security requirements for different application scenarios including Pre-Shared Key (PSK) and Certificate-Based Authentication (CBOR).

There are documentations developed through a collaborative process and maintained by the Internet Engineering Task Force (IETF) called Request for Comments (RFC) that describe standards, protocols, and best practices for the Internet and other computer networks.

RFCs are essentially technical documents that describe how different Internet protocols and technologies work, as well as provide recommendations and guidelines for their implementation and usage. They can cover a wide range of topics, including network architecture, routing, security, email protocols, web protocols, and more.

RFCs are assigned unique identification numbers, which are used to reference and cite them in other technical documents, academic papers, and other publications.

CoAP also has RFC documentation to describe the features, behavior, guidelines and best practices of the protocol Here are some of the relevant RFCs related to CoAP:

- RFC 7228:
- RFC 7252: This RFC defines the CoAP basic features including its message format, request/response methods, Uniform Resource Identifier mapping, and transport bindings
- RFC 7641: This RFC provides guidelines for the use of CoAP in the Internet of Things (IoT) and machine-to-machine (M2M) applications.
- RFC 7662:
- RFC 7959: Defines the Block-wise Transfers in CoAP, which is a mechanism to allow large payloads to be split into multiple smaller messages and transferred over networks with limited resources.
- RFC 8075: This RFC defines the usage of CoAP over TCP, which is an alternative transport protocol for CoAP messages.

- RFC 8323: This RFC defines the CoAP Management Interface (CoMI), which provides a standardized interface for managing CoAP resources.
- RFC 8613: This RFC defines the CoAPs protocol, which is a secure version of CoAP that uses Datagram Transport Layer Security (DTLS) to provide end-to-end security.
- RFC 8890: This RFC specifies the CoAP Content-Format Indicators, which provide a standardized way of indicating the format of CoAP payloads.

These RFCs are important references for developers who want to implement CoAP-based applications or protocols. They provide a comprehensive understanding of the protocol and its use cases.

Like MQTT, there is also several CVEs related to CoAP implementations. Here are some examples:

- CVE-2018-18913: A vulnerability in Contiki OS that allows a remote attacker to execute arbitrary code or cause a Denial of Service (DoS).
- CVE-2019-12415: A vulnerability in Eclipse Californium that could allow a remote attacker to cause a DoS condition by sending a specially crafted CoAP packet with a large block option value.
- CVE-2020-15523: A vulnerability in TinyOS which allows a remote attacker to cause a buffer overflow and execute arbitrary code.

Even through CoAP and MQTT are both protocols used for communication, they have different security mechanisms due to their design and requirements.

Overall, both CoAP and MQTT provide security features that can be used to protect device communications and data. However, the choice between the two protocols may depend on the specific security requirements of the application and the level of support for security mechanisms.