M2M Straw-Man Proposition

# MQTT Security SC

The purpose of this document is to propose a straw man for the upcoming security section of the MQTT specification. It was agreed during the kick-off meetings that the security section would comprise 1) a sub-section with high level M2M security recommendations based on the upcoming NIST Cyber Security Framework and 2) a sub-section describing hands down legacy cryptographic support. This straw-man will attempt to formulate the former sub-section.

## Proposed Table of Contents

1. Introduction
2. Definition of Terms
3. M2M Threat Overview
   1. Zero Day Threat
   2. Persistent Threat
4. Cyber Security Function Matrix
   1. Know
   2. Prevent
   3. Detect
   4. Respond
   5. Recover
5. Security Level Profiles Definitions
   1. Unsecured
   2. Base Secured
   3. Industry Secured
   4. Cyber Critical Secured
6. Security Profile Implementation Use Case
7. Historical MQTT Security Implementations

## 1. Introduction

As the high level M2M security recommendations will map directly to the NIST Cyber Security Framework guidelines and references contained, we propose that the introduction explains the purpose of the NIST Framework, its structure, and most importantly, the reasons for its applicability towards the MQTT specification and M2M in general. More precisely, the NIST Framework introduces the concept of “*functions*”: Know, Prevent, Detect, Respond, and Recover. In the context of MQTT, these functions should offer a way to take a high-level, overarching view of an organization’s management of M2M cyber security risk. We suggest that the introduction briefly introduce these 5 functions and explain how each function will point towards informative references such as existing standards, guidelines, and practices.

## 2. Definition of terms

The definition of terms should define the 5 functions and any other sub-category that may be associated with a given function. Any other term that we will not consider common knowledge should be defined as well. These terms will come up as the section evolves.

## 3. M2M Threat Overview

This section should summarize general M2M threats applicable in most use cases. We propose to divide these threats by introducing them in two categories: Zero Day Threats (unknown threats) and Persistent Threats. For each threat we could introduce systematic descriptive categories such as: threat objectives, timeliness, risk tolerance, skills and methods used for this threat, actions used throughout the threat, attack origination points, numbers involved in the attack, knowledge source, etc.

### Zero Day Threats

### Persistent Threats

## 4. Cyber Security Function Matrix

This is the core of the M2M security section. It should contain the 5 functions each with their dedicated introduction and appropriate guide lines and references. The main effort to be accomplished in writing the M2M security section should lie in crafting the guide lines and selecting the correct references.

### Know

This function describes the knowledge required to safe keep an M2M environment. For example this can be asset management (physical M2M devices or software applications) network mapping or identifying specific risks. An example of a potentially relevant specification is the “*Cyber Vulnerability Assessment – Electronic Access Points*” (CIP-005-3a R4).

### Prevent

This function should help the reader prevent security threats described in section 3 (threat overview). The suggested guidelines and references should cover a wide spectrum of threats ranging from physical security of M2M devices and infrastructure to Credential and Access Management.

### Detect

This function should help the reader identify potential security threats. We should point towards references that explain how to monitor hardware devices, software applications, the underlying network, users and staff.

### Response

This function should help the reader respond to an M2M security breach. More particularly we should make recommendations on how to implement risk based decision trees and how to manage incidents. The NIST Framework contains many standards and specifications for Incident management and response planning that we should analyze.

### Recover

This last function should describe system and disaster recovery. Crisis communication and continuous improvement are also subjects we propose to address. Again there are many standards and specifications that cover these subjects that we could recommend.

## 5. Security Level Profiles Definition

Based on the level of compliance with the functions and their content, we propose to create a metric that would help the reader determine her/his organization’s level of security. This qualitative metric could, e.g., be measured using a matrix where the reader would assess, for each function, the number of guidelines and references with which the organization is already compliant with. We propose to establish 4 security levels: Unsecured, Base Secured, Industry Secured (Base + Industry customizations), Cyber Critical Secured. Corresponding scores could be: <25%, <50%, <75%, >75%.

## Security Profile Implementation Use Case

The final section would give one or many (diverse) example use cases. Each use case would have its own illustrative list of threats. The 5 functions would then be applied to the use case and general discussion would argue whether the current use case is compliant with the guide lines and references contained in each function. Subsequently we could apply the qualitative metric to the use case and return a security level score as described in section 5. Finally recommendations could be given as to how the level could be raised.

### Example Use Case: Aircraft Turnaround M2M Ecosystem

An airline company establishes an M2M infrastructure that gathers information in order to optimize aircraft turnaround at its home base airport. The information gathered originates from the company’s and partners remote sensors. They include passenger buses and refueling trucks geo-location and real time fuel consumption sensors. The objective is to optimize routes, locate key assets, forecast unavailability periods, and ultimately reduce turnaround time. However as the information is potentially shared between several organizations, the ability to secure and accurately apportion data to the authorized members is important. The airline has identified the following list of M2M cyber security threats:

1. Malfunctioning or misconfigured partner hardware.
2. Key management.
3. Security perimeters with partners.
4. Airport Networking Infrastructure.
5. …

The Airline Company knows it is compliant with standards X, Y, and Z (to be defined in the example) and has set up internal policies that enforce A, B, C, and D (to be defined as well). Applying the 5 functions framework to the Use Case leads to the following discussion and comments:

Know

The company follows a strict asset management policy based on ISO 27001 also specified by MQTT security specification. It also imposes this policy on its partners. However it does not consider more specific hardware recommendations such as the SANS Top 20 controls for unauthorized and misconfigured devices (also specified in the MQTT security specification).

#### Prevent

The Company has established extensive cyber security prevention measures. The Company has invested in staff awareness training, information protection (data at rest and in transit) and infrastructure protection (system and network protection). Although the Company is not using the same standards as those defined in the security section, we conclude that they are similar or equivalent. The Company is using sufficient and correct prevention measures and is considered compliant.

#### Detect

The company is monitoring network activity and access control by using firewalls and white listing policies as specified in NIST 800-12 and NIST SP 800-83. It does not however monitor physical devices. The company exposes itself to security breaches on its end devices (and those of its partners).

#### Respond

The company has incident management guidelines that staff should follow in case of security breach detection. These guidelines were crafted internally and overlook several recommendations specified in the MQTT security section especially those contained in a NIST publication.

#### Recover

To recover from a potential cyber-attack the company has established a contingency and disaster recovery plan based on guidelines specified in the same specifications that are found in the security section.

#### Security Level Profile Score

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Functions** | **Guidelines** |  | **Publication** |  | **Score** |
| Know | A  B  C | x  x | X  Y  Z | x | 10/20 |
| Prevent | A  B  C  D | x | X  Y | x  x | 12/20 |
| Detect | A |  | X  Y  Z | x  x  x | 15/20 |
| Respond | A  B  C | x  x  x | X | x | 20/20 |
| Recover | A  B | x | X  Y  Z | x | 14/20 |
| TOTAL SCORE |  |  |  |  | 71/100 |

#### Conclusion

The qualitative metric ranks the Company at 71% secured corresponding to Level 2 “Industry Secured”. To increase its security level, the Company should make efforts to comply with the overlooked guidelines and publications.

## Historical MQTT Security Implementation

The primary market that MQTT was first developed for was for real-time SCADA systems in the Oil/Gas market sector. In the late 1990's early 2000's TCP/IP was just emerging as an available transport for the various communications providers in this market. So regardless of the physical connection (VSAT, PSTN, SS Radio, etc) the only thing we could count on is that it would be slow, expensive, and unreliable. Case and point is that there are still many MQTT monitoring solutions deployed that run over 300 baud dial up lines that use SLIP (Serial Line IP) to establish the MQTT session (I guess you could consider that security by obscurity). We knew in the beginning that security would eventually become a requirement for MQTT in many applications and that was actually one of the key decision factors as we made the decision to use TCP in lieu of UDP as the underlying transport.

So early deployments of MQTT tended to run on private VSAT/Microwave/SS (Spread Spectrum) Radio/PSTN and at the time security was not really considered as a primary concern. But over time we did have to start addressing the security aspects. One of the first was to ensure that we could authenticate an MQTT client over PSTN as customers didn't want intruder access to the SCADA host over dial up modems. This is where we implemented one of the only changes to the original MQTT specification and added the option for Username/Password in addition to the ClientID. Some customers implemented an additional layer of authentication by requiring authentication to the modem router before setting up SLIP connections to the MQTT broker.

As the networks evolved, became faster, and somewhat more reliable, they also tended to move towards public network offerings instead of the customers running their own private TCP/IP networks. Now these networks did touch the Internet and security started to become more of a concern. In some of the VSAT installations this was accomplished by setting up VPN's from the remote VSAT terminal equipment into the customers facilities. Although not part of the MQTT transport, the TCP/IP session to the broker was done over a secure VPN tunnel that was setup by the VSAT interface. But this did require some complex configuration and management of the network by the customer.

By the mid 2000's MQTT was starting to branch out into other industries and embedded systems were getting more mature OS's and TCP/IP stacks. For several medical company projects security and HIPPA compliance required a higher level of security than we had seen previously. For these systems a double SSH tunnel was used to VPN the MQTT session to the Broker. An initial SSH tunnel was created from the MQTT client to the medical facilities DMZ, and then a second SSH tunnel was created to connect to the broker. Just to note here, most of the early MQTT implementations that used a TCP/IP layer of security used SSH in lieu of SSL. The Unix/Linux community seemed quite comfortable with this technology and methodology for secure MQTT sessions.

But now TLS/SSL security schemes tend to be preferred over SSH (albeit I feel SSH tunnels are still a very viable security scheme ). Most of the deployed MQTT solutions that I am aware of use one of these methods to secure the MQTT session. In some cases it is still the VSAT terminal equipment that establishes the secure connection and in other cases it's the MQTT client implementation on the embedded device that establishes the session.

Although the MQTT message transport does not specify session security (nor should it), In all of these cases where required, MQTT security has tended to be one off, project specific, implementations. In some cases it is done by the network equipment (i.e. router, terminal server, VSAT IDU, etc) and in some cases it's done by the platform running the MQTT client. Regardless of "where" the security it implemented, I am seeing more and more customers wanting to use MQTT and just wanting to know what "best practices" are to do this. I think that having OASIS add a security "Appendix" or “Best Practices” to the MQTT specification that would outline these best practices is an excellent idea.