Formal verification of the implementation of the MQTT protocol in IoT devices

Kristiyan Mladenov

University of Amsterdam
Faculty of Physics, Mathematics and Informatics
MSc System and Network Engineering
Research Project 2

July 3, 2017
Introduction

- Mirai botnet producing one of the largest DDoS attacks ever.
- We can also talk about botnet "wars".
- Compromise due to human error.
IoT testing

- Rapid7 IoT Security Testing Methodology
- OWASP IoT Top 10
- IoT Inspector (SEC Technologies)
IoT testing

- Rapid7 IoT Security Testing Methodology
- OWASP IoT Top 10
- IoT Inspector (SEC Technologies)

What would happen if we dig deeper?

- One of the main goals of the IoT devices is to exchange data using some message exchange mechanism.
- How can we assure a proper protocol implementation?
- Could we make sure that it is correct in a more formal way?
Protocol of choice

MQTT
Message Queue Telemetry Transport

• Designed for message transfer with small code footprint and limited bandwidth in mind.

• First version was available in 1999. Version 3.1.1 is standardised by OASIS (2014) and ISO (2016).
Protocol of choice

**MQTT**

Message Queue Telemetry Transport

- Designed for message transfer with small code footprint and limited bandwidth in mind.
- First version was available in 1999. Version 3.1.1 is standardised by OASIS (2014) and ISO (2016).
- Publish/Subscribe communication mechanism similar to IRC.
- Adds the concept of Last Will and QoS.
MQTT use cases

MQTT is implemented in:

- The backend of The Things Network (LoRa)
- AWS IoT, Google Cloud IoT
MQTT use cases

MQTT is implemented in:

- The backend of The Things Network (LoRa)
- AWS IoT, Google Cloud IoT

Applications that use MQTT

- Fitness trackers, Medical equipment, ATM machines
- Implemented by Deutsche Bahn (DB)
- Facebook Messenger (Unconfirmed)
Research Question

Can the MQTT protocol implementation in IoT devices be verified formally?

Subquestions

- What methods can be used to formally assess the implementation of a communication protocol?
- Using the chosen formal testing methods, does the MQTT implementation in certain selected IoT devices adhere to the standard?
Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework.
Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open

- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179
Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open

- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

...
Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open

- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.
Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.
- Finite State Machines and TTCN successfully used to verify IIoT protocol implementations.
Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.
- Finite State Machines and TTCN successfully used to verify IIoT protocol implementations.

There is a tool for every approach
Related Work

From some of the major standardisation organisations:

- ISO/IEC 9646 - Conformance testing methodology and framework. Not open
- Testing and Test Control Notation version 3 (TTCN-3) included in part 3 of the above. Formal Description Technique as of ITU-T Z.160 - Z.179

Relevant scientific research:

- Mapping TTCN to Labelled Transition Systems.
- Finite State Machines and TTCN successfully used to verify IIoT protocol implementations.

There is a tool for every approach

- The testing to follow is focused on Eclipse Titan.
### MQTT Packet Structure

**Figure**: MQTT Packet structure

![MQTT Packet Structure Diagram](image-url)
Example test

**[MQTT-2.3.1-1]**

SUBSCRIBE, UNSUBSCRIBE, and PUBLISH (in cases where QoS > 0) Control Packets MUST contain a non-zero 16-bit Packet Identifier.

**Figure:** Publish with Packet ID 0

**Figure:** Test execution flowgraph
Room for improvement

Writing is nature’s way of letting you know how sloppy your thinking is.\textsuperscript{1}

\textsuperscript{1}Dick Guidon
Room for improvement

*Writing is nature’s way of letting you know how sloppy your thinking is.*

- Translating a specification from natural to formal language is prone to errors.
- How can we safely come up with new values for the tests?
- If the specification is defined in a formal language, testing might be easier.

---

1Dick Guidon
Intermezzo

The Die Hard challenge

- You have two buckets
  - 3 litres
  - 5 litres
- You have an infinite amount of water.
- You can waste as much water as you want.
- How do you fill the large bucket with exactly 4 litres?

Intermezzo approach (enter TLA+)
EXTENDS Integers

VARIABLES small, big

TypeOK $\triangleq \land \text{small} \in 0 \ldots 3$
$\land \text{big} \in 0 \ldots 5$

Init $\triangleq \land \text{big} = 0$
$\land \text{small} = 0$

FillSmall $\triangleq \land \text{small}' = 3$
$\land \text{big}' = \text{big}$

FillBig $\triangleq \land \text{big}' = 5$
$\land \text{small}' = \text{small}$

EmptySmall $\triangleq \land \text{small}' = 0$
$\land \text{big}' = \text{big}$

EmptyBig $\triangleq \land \text{big}' = 0$
$\land \text{small}' = \text{small}$

SmallToBig $\triangleq$ IF $\text{big} + \text{small} \leq 5$
THEN $\land \text{big}' = \text{big} + \text{small}$
$\land \text{small}' = 0$
ELSE $\land \text{big}' = 5$
$\land \text{small}' = \text{small} - (5 - \text{big})$

BigToSmall $\triangleq$ IF $\text{big} + \text{small} \leq 3$
THEN $\land \text{big}' = 0$
$\land \text{small}' = \text{big} + \text{small}$
ELSE $\land \text{big}' = \text{small} - (3 - \text{big})$
$\land \text{small}' = 3$

Next $\triangleq \lor \text{FillSmall}$
$\lor \text{FillBig}$
$\lor \text{EmptySmall}$
$\lor \text{EmptyBig}$
$\lor \text{SmallToBig}$
$\lor \text{BigToSmall}$
• Define different invariant in the TLA+ model checker.

• Observe the behaviour of the model; relax constraints if necessary.

• Map the observed behaviour in terms of TTCN-3 tests.

• The problem of translating natural to formal language is still not solved.

Figure: TLA+ simplified keepalive
What follows is a list of the normative requirements and how do the tested implementations conform to them.

<table>
<thead>
<tr>
<th>Normative Requirements</th>
<th>2.2.2</th>
<th>2.3.1-1</th>
<th>3.1.0-1a</th>
<th>3.1.0-1b</th>
<th>3.1.0-2</th>
<th>3.1.2-2</th>
<th>3.1.2-24</th>
<th>3.1.3-8</th>
<th>3.3.1-4</th>
<th>3.6.1-1</th>
<th>3.8.1-1</th>
<th>3.8.3-4</th>
<th>3.12.4-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mosquitto</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Emqtt</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RabbitMQ</td>
<td>x</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>x</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
What follows is a list of the normative requirements and how do the tested implementations conform to them.

<table>
<thead>
<tr>
<th></th>
<th>2.2.2</th>
<th>2.3.1-1</th>
<th>3.1.0-1a</th>
<th>3.1.0-1b</th>
<th>3.1.0-2</th>
<th>3.1.2-2</th>
<th>3.1.2-24</th>
<th>3.1.3-8</th>
<th>3.3.1-4</th>
<th>3.6.1-1</th>
<th>3.8.1-1</th>
<th>3.8.3-4</th>
<th>3.12.4-1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mosquitto</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Emqt</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>RabbitMQ</td>
<td>X</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Conclusion

- There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.

- Using the TTCN-3 language, three different MQTT implementations were tested and inconsistencies with the specification were found.

As a side note, adhering to the standard does not mean that a device is secure, especially in the cases of bad protocol design.
Conclusion

- There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.

- Using the TTCN-3 language, three different MQTT implementations were tested and inconsistencies with the specification were found.

As a side note, adhering to the standard does not mean that a device is secure, especially in the cases of bad protocol design.
Conclusion

• There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.

• Using the TTCN-3 language, three different MQTT implementations were tested and inconsistencies with the specification were found.

• Those inconsistencies can be used to fingerprint and identify implementations.
Conclusion

- There are plenty of ways to model the implementation of a communication protocol, using Finite State Machines, Labelled Transition Systems, even Set Theory and First Order Logic.
- Using the TTCN-3 language, three different MQTT implementations were tested and inconsistencies with the specification were found.
- Those inconsistencies can be used to fingerprint and identify implementations.

As a side note, adhering to the standard does not mean that a device is secure, especially in the cases of bad protocol design.
Future work

- Building a complete TLA+ model could be able to identify additional behavioural differences between different implementations.
- The output derived from the TLA+ model might be used for fuzzing.
- It could also help in identifying deficiencies in the protocol design itself, rendering all implementations vulnerable.
Questions?

Share your thoughts?
References

- Image depicting the interaction between the MQTT Client and Server taken from: http://www.hivemq.com/blog/mqtt-essentials-part2-publish-subscribe