MQTT in the Enterprise
How to successfully run an MQTT Message Broker

Arnold Bechtoldt
(Senior Systems Engineer)

5 learnings from two years of suffering challenges!
Thousands of cars (MQTT clients)
Dozens of backend services (MQTT clients)
~ 40,000 MQTT sessions
> 100,000 incoming MQTT msgs per minute
> 1,000,000 customers
Today’s Learning Goal:
How not to kill customers with IoT!
Many (many) developers, sysadmins, supporters, QA
inovex to the rescue!
Message Broker in the Middle (MITM)
Lesson #1

MQTT: Forget what you've learned before!
MQTT is different

- Rules from HTTP world won’t apply
- “MQTT is simple.” vs. “MQTT is hard.”
- MQTT tooling support (was) missing
- HiveMQ is a rather stateful application
Lesson #2

The(re is no) “I” in IoT!
Most MQTT clients connected via GSM/UMTS
Miserable internet connection
Almost all support tickets caused by network issues
Embedded hardware lacks network fault tolerance
Comparing (MQTT) monitoring metrics helps
NOT SURE IF IT'S A NETWORK ISSUE

OR LAYER 8 ISSUE
Lesson #3
Monitor your (MQTT) clients to understand their behaviour!
### HiveMQ Prometheus Metrics (> 1800)

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>metrics_com_hivemq.messages_dropped_before_publish_send_count_count</td>
<td>0.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_count_count</td>
<td>6092213.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_in_flight_window_count_count</td>
<td>835421.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_internal_error_count_count</td>
<td>0.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_not_connected_count_count</td>
<td>1238.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_not_writable_count_count</td>
<td>4775796.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_qos_0_queue_not_empty_count_count</td>
<td>479758.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_queue_full_count_count</td>
<td>0.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_rate_count</td>
<td>6092213.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_rate_fifteenminuterate</td>
<td>0.0026798488064779242</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_rate_fiveminuterate</td>
<td>0.003673811334623905</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_rate_meanrate</td>
<td>1.1690383860629943</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_dropped_rate_oneminuterate</td>
<td>0.0024693484286362477</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_connect_count_count</td>
<td>1.0997633E7</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_connect_rate_count</td>
<td>1.0997633E7</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_connect_rate_fifteenminuterate</td>
<td>2.6954264466661912</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_connect_rate_fiveminuterate</td>
<td>2.84684980596273</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_connect_rate_meanrate</td>
<td>2.110342385291968</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_connect_rate_oneminuterate</td>
<td>2.890580707427736</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_disconnect_count_count</td>
<td>3926444.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_disconnect_rate_count</td>
<td>3926444.0</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_disconnect_rate_fifteenminuterate</td>
<td>0.8603151573544764</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_disconnect_rate_fiveminuterate</td>
<td>0.8937906297328476</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_disconnect_rate_meanrate</td>
<td>0.7534476952336496</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_disconnect_rate_oneminuterate</td>
<td>0.848894289974284</td>
</tr>
<tr>
<td>metrics_com_hivemq.messages_incoming_pingreq_count_count</td>
<td>1.6680187E8</td>
</tr>
</tbody>
</table>
Visualizing Metrics
MQTT Broker Dashboard (Grafana)
(Bad) Dev/Prod Parity

Testing

Integration

Production
Lesson #4

Log all MQTT events and people will start loving you!
Notice: Don’t break data privacy law!
## Quick Links

The following links help you to find some query examples. Open an example and modify to your own needs:

<table>
<thead>
<tr>
<th>Use Case</th>
<th>Index</th>
<th>TEST</th>
<th>INT</th>
<th>PROD</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Login details (after connect)</td>
<td><em>-mqttbroker-authplugin-</em></td>
<td>Kibana</td>
<td>Kibana</td>
<td>Kibana</td>
<td>(user abc)</td>
</tr>
<tr>
<td>Login details by VIN (after connect)</td>
<td><em>-mqttbroker-authplugin-</em></td>
<td>Kibana</td>
<td>Kibana</td>
<td>Kibana</td>
<td>(vehicle XYZ)</td>
</tr>
<tr>
<td>Connects/Disconnects by VIN</td>
<td><em>-mqttbroker-events-</em></td>
<td>Kibana</td>
<td>Kibana</td>
<td>Kibana</td>
<td>(vehicle XYZ)</td>
</tr>
<tr>
<td>Non-graceful disconnects</td>
<td><em>-mqttbroker-events-</em></td>
<td>Kibana</td>
<td>Kibana</td>
<td>Kibana</td>
<td></td>
</tr>
<tr>
<td>MQTT Message Drops</td>
<td><em>-mqttbroker-events-</em></td>
<td>Kibana</td>
<td>Kibana</td>
<td>Kibana</td>
<td></td>
</tr>
<tr>
<td>HTTP IP Triggers</td>
<td><em>-mqttbroker-restplugin-</em></td>
<td>Kibana</td>
<td>Kibana</td>
<td>Kibana</td>
<td></td>
</tr>
<tr>
<td>MQTT actions by VIN</td>
<td><em>-mqttbroker-logs-</em></td>
<td>Kibana</td>
<td>Kibana</td>
<td>Kibana</td>
<td>(vehicle XYZ)</td>
</tr>
</tbody>
</table>
Unstructured logs as base ...

2018-06-01T18:25:37,347Z [single-writer-14] DEBUG event.client-disconnected - Client ID: mqttlinkstatusaggregator, IP: 192.168.0.188, was disconnected. re
... manipulated with Logstash

```ruby
# fix a few inconsistencies in the original log format before offering the log line to kv for parsing
mutate {
  gsub => [
    "LOGMESSAGE", "connected\.$", "",
    "LOGMESSAGE", "was disconnected\.", ",",
    "LOGMESSAGE", "disconnected gracefully\.$", ",", reason: graceful_disconnect",
    "LOGMESSAGE", "disconnected ungracefully\.$", ",", reason: ungraceful_disconnect",
    "LOGMESSAGE", "An other client connected with the same client id.$", "clientID_in_new_connection",
    "LOGMESSAGE", "Another client connected with the same clientId.$", "clientID_in_new_connection",
    "LOGMESSAGE", "Client was idle for too long.$", "idle_timeout",
    "LOGMESSAGE", "Outgoing publish message was dropped\.", ",",
    "LOGMESSAGE", "reason: Internal error\.$", "reason: internal_error",
    "LOGMESSAGE", "reason: No CONNECT sent in time.$", "reason: no_connect_sent_in_time",
    "LOGMESSAGE", "reason: ", "event_reason:",
    "LOGMESSAGE", "\.$", ","
  ]
}

kv {
  source => "LOGMESSAGE"
  prefix => "mqtt_
  field_split => ",",
  value_split => ":"
  remove_char_value => " ",
  remove_char_key => " ",
  transform_key => "lowercase"
}
```
... to show MQTT client login

```
<table>
<thead>
<tr>
<th>Time</th>
<th>mqtt.ip</th>
<th>mqtt_username</th>
<th>mqtt_clientid</th>
<th>mqtt_auth_result</th>
</tr>
</thead>
<tbody>
<tr>
<td>June 1st 2018, 14:10:41.312</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:40.975</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:40.657</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:40.485</td>
<td></td>
<td></td>
<td>CLEAN</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:40.457</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:40.083</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.992</td>
<td></td>
<td></td>
<td>CLEAN</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.812</td>
<td></td>
<td></td>
<td>CLEAN</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.779</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.745</td>
<td>10.14.72.32</td>
<td>internal.healthcheck.user</td>
<td>prod-m</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.728</td>
<td></td>
<td></td>
<td>CLEAN</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.606</td>
<td></td>
<td></td>
<td>CLEAN</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.597</td>
<td>10.14.72.14</td>
<td>internal.healthcheck.user</td>
<td>prod-m</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.584</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
<tr>
<td>June 1st 2018, 14:10:39.584</td>
<td></td>
<td></td>
<td>PERSIST</td>
<td>successful</td>
</tr>
</tbody>
</table>
```
... or dropped messages!
Lesson #5

“S” in IoT stands for security!
MQTT security mechanisms are okay
IoT security is possible
*Problem Exists Between Chair And Keyboard* (PEBCAK)
Fight for security!
Ultimative list of *recommendations*

Don’t …

- force passwords with length of 8 chars.
- log passwords in plaintext.
- forget authorization after implementing authentication.
- use credentials across ten thousands of clients.
- publish passwords on Github!!!
Mission accomplished!

[...] die pragmatische, unkomplizierte und produktive Zusammenarbeit mit inovex positiv hervorzuheben, da [...] komplexe Problemstellungen [...] gemeinsam gelöst wurden [...].

Generell finde ich die Zusammenarbeit [...] schon ziemlich gut und v. a. auch pragmatisch. Auch dass sich alle [...] verantwortlich fühlen, dass die Dinge laufen/funktionieren.

Thank you to all [...] and giving us the chance to provide such a solution, from monitoring point of view it seems that we have the best MQTT connection ever.
Open Source now ...

- [github.com/inovex/mqtt-stresser](https://github.com/inovex/mqtt-stresser)
- [github.com/inovex/mqtt_blackbox_exporter](https://github.com/inovex/mqtt_blackbox_exporter)
- [inovex.de/blog (HAProxy MQTT Health Check)](https://inovex.de/blog)
- ...