

# Home Challenge #1: Sniffing

Prof. Matteo Cesana - Year 2020/2021

*Gabba Rohit [codice persona: 10706944]*

*Tortorelli Giuseppe [codice persona: 10582962]*

## 1 What's the difference between the message with MID: 3978 and the one with MID: 22636?

Using Wireshark filter:

```
coap.mid == 3978 && coap.mid == 22636
```

we have isolated the two packets. After that we noticed that they have a different token and also that the first is of type *CON* and the second of type *NON*. Therefore the first one requires an *ACK* message and the second one does not.

## 2 Does the client receive the response of message No. 6949?

First of all using Wireshark *No.* column we found the message number 6949 and we read its *MID*. After this using filter:

```
coap.mid == 28357
```

we discovered that there was an *ACK* containing the error message *4.05*, which means the method is not allowed therefore the client did not receive any other response.

## 3 How many replies of type confirmable and result code "Content" are received by the server "localhost"?

Using filter:

```
coap.type == 0 && ip.dst == 127.0.0.1 && coap.code == 69
```

we found out that there were 8 replies.

## 4 How many messages containing the topic "factory/department\*/+" are published by a client with user name: "jane"? Where \* replaces the dep. number, e.g. factory/department1/+, factory/department2/+ and so on?

Using filter:

```
mqtt.username == jane
```

we got 4 connection messages. We noticed that these 4 connect messages have 4 different tcp ports (50985, 40989, 42821, 40004).

Using the filter:

```
mqtt.topic contains "factory" && (tcp.port == 50985 || tcp.port == 40989 || tcp.port == 42821 || tcp.port == 40004) && mqtt.msgtype == 3
```

we found 10 messages but none of them were of the topic "factory/department\*/+".

## 5 How many clients connected to the broker "hivemq" have specified a will message?

Using filter:

```
ip.dst_host == broker.hivemq.com && mqtt.msgtype == 1 && mqtt.willmsg_len > 0
```

we have selected all connection messages with *will message* specified sent to *hivemq*. Counting also the connection messages with the same *ClientId* and those in which the *CleintId* was not specified we found 16 messages: 3 of them have distinct *ClientID*, 1 is a duplicate and the rest of them have the *ClientID* not specified.

## 6 How many publishes with QoS 1 don't receive the ACK?

Using filter:

$$mqtt.msgtype == 3 \&\& mqtt.qos == 1$$

we got all published messages with QoS equals to 1 (124).

Using filter:

$$mqtt.msgtype == 4$$

we got all *ACK* messages for the published one (74).

By computing the difference

$$124 - 74 = 50$$

we obtained the number of messages that did not receive an *ACK*.

During the count we have considered also the duplicate messages because those are also the ones which do not receive the *ACK* messages.

## 7 How many last will messages with QoS set to 0 are actually delivered?

We tried to retrieve all the last will messages delivered to us because those were the the only last will messages that can be seen by this pcap. We noticed that all the will messages sent during the connection contained the keyword "error".

Using the filter

$$mqtt.msgtype == 3 \&\& mqtt.qos == 0 \&\& mqtt.msgcontains "error"$$

to see all the publish messages containing the word "error", we found only one message.

## 8 Are all the messages with QoS>0 published by the client "4m3DWYzWr40pce6OaBQAfk" correctly delivered to the subscribers?

Using filter:

$$mqtt.clientid == "4m3DWYzWr40pce6OaBQAfk"$$

we found sender ip address (10.0.2.15), source port (58313), destination ip (5.196.95.208).

Using filter:

$$ip.src == 10.0.2.15 \&\& tcp.srcport == 58313 \&\& mqtt.qos > 0 \&\& mqtt.msgtype == 3$$

we found that the client sent one pub message with *QoS* set to 2, to the same server as previous connection message.

Using filter:

$$ip.src == 5.196.95.208 \&\& tcp.dstport == 58313 \&\& ip.dst == 10.0.2.15 \&\& mqtt.msgtype == 5$$

we got the *PUBREC* message (same *Message Identifier* equal to 3) send from the broker to the client.

It means that the broker successfully received the message and has sent it to the "factory/department1/section1/deposit" subscribers correctly. So it can be said that all publish messages sent by the client have been correctly delivered even if there is no a *PUBREL* message sent to the server

## 9 What is the average message length of a connect msg using mqttv5 protocol? Why messages have different size?

Using filter:

$$mqtt.ver == 5$$

we got all connection messages that use mqtt version 5.

Analyzing the "Msg Len" field of each message we obtained:

- 35 messages of length 13
- 1 message of length 29
- 4 messages of length 69
- 4 messages of length 65
- 1 message of length 20
- 4 messages of length 77
- 2 messages of length 32
- 5 messages of length 25
- 1 message of length 27
- 1 message of length 86
- 1 message of length 33
- 2 messages of length 30
- 1 message of length 78
- 1 message of length 83

Calculating the average, we obtained an average length of 30.22 bytes.

The length changes from message to message because the connection message may or may not contain additional information such as the last will message.

(The wireshark version used to analyze the pcap didn't detect any malformed messages)

## 10 Why there aren't any REQ/RESP pings in the pcap?

There aren't any REQ/RESP pings in the pcap because the clients and the broker keep interacting without ever letting the Keep Alive timers of the clients expire.