HOME ASSIGNMENT #3

IoT 19-20

Name Surname	Person code	Id Number
Andrea Crivellin	10491856	928320
Gabriele Guelfi	10491169	916207

QUESTION #1

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

The message with MID 3978 is Confirmable (CON) and the other one is Non-Confirmable (NON-CON). That means that the first one waits an ACK message in response, the other one doesn't. If the CON message does not receive the ACK response within a timeout it repeats the message.

QUESTION #2

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

Yes, it does. The response is in message No. 6953. It is an error message with code 4.05 (method not allowed).

QUESTION #3

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

Zero. Each message with code "Content" is an ACK or a NON-CON message. This means that Content is sent directly in the ACK message, when the request is confirmable.

QUESTION #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. (btw, * is NOT an MQTT wildcard)*

Jane connects four times. We looked for those messages coming from the same IP/port that Jane used in this way:

'ip.src == 127.0.0.1 && tcp.srcport == XXXXX && mqtt.msgtype == 3'

We found that there were seven publish messages, but none of them matches the requested topic. Some of them were of the type *factory/department*/+/+* and one was in a totally different format.

QUESTION #5

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

First of all we looked at the DNS messages to figure out which IP address hivemq broker has. We found two different IPs: 3.120.68.56 and 18.185.199.22. Then we filtered as follows:

'mqtt.msgtype == 1 && mqtt.willmsg && (ip.dst == 3.120.68.56 || ip.dst == 18.185.199.22)'

We found 10 messages:

- 6 with ClientID field empty, this results in a connection without any state
- 2 with the same ClientID value
- 2 with unique ClientIDs

So the clients connected are at least 3 and at most 9.

QUESTION #6

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

We figured out that the publish messages with QoS 1 are 124 ('mqtt.msgtype == 3 && mqtt.qos == 1') and that the pubACK are only 74 ('mqtt.msgtype == 4'), so the publishes with QoS 1 that don't receive the ACK are 50. We counted also the possible duplicates using the following filter:

'mqtt.dupflag == 1 && mqtt.msgtype == 3'

there are only 2 messages with a DUP flag, so we can say that the solution is 48.

QUESTION #7

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

We find out that each last will message is in the type 'error : xxxxx', so we applied the filter:

'mqtt.msgtype == 3 && mqtt.msg contains "error" && mqtt.qos == 0'

it results in just one message (No. 4164).

QUESTION #8

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

First of all we looked for the Connect Command message of the specified client finding its IP/port and the IP of the broker it has connected to. We applied this filter:

'ip.src == 10.0.2.15 && tcp.srcport == 58313 && ip.dst == 5.196.95.208 && mqtt.msgtype == 3 && mqtt.qos > 0'

and we discovered that just one message with QoS > 0 was published by that client. Finally we used the following filter:

'ip.src == 5.196.95.208 && mqtt.msgtype == 3 && mqtt.msg == "{"id": "Actuator 2", "value": 812, "lat": 100, "lng": 140, "unit": "C", "type": "temperature"}"'

that results in zero messages, so the broker has not forwarded the message to any subscriber.

QUESTION #9

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

Most of the messages have length of 71 bytes, that is also the minimum, the maximum reached is 103 bytes. The main differences are caused by username, password and ClientID fields.

QUESTION #10

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

There aren't any REQ/RESP pings in the pcap because it means that some interaction between client and broker occurred before the keep alive timer expired.