Activity 3

COAP and MQTT

Diego Costanzelli
diego.costanzelli@mail.polimi.it
ID number: 10527966 - 928941

Francesco Maffezzoli
francesco.maffezzoli@gmail.com
ID number: 10576556 - 944914

Marco Passera
passera.marco@alice.it
ID number: 10531470 - 944947

Answers

COAP

1) The difference between the two messages is that the one with MID = 3978 is of type CON (confirmable – an ACK is required) and has block size of 64 byte while the message MID = 22636 is of type NON (non-confirmable) and block size is of 1024 byte
   Filter: *coap.mid == 3978 || coap.mid == 22636*
2) Yes, the client receives an ACK at message No. 6953
   Filter: *frame.number == 6949 || coap.mid == 28357*
3) There are 8 replies of type confirmable and result code “Content”. They are replies to messages of type CON (confirmable).
   Filter: *coap.type == 2 && ip.dst== 127.0.0.1 && coap.code == 69*

MQTT

4) First of all, we have looked to the messages with username “jane”, which are Connect Commands (frame number 1554, 1623, 3708, 5816). Then we have taken note of source IPs and ports of the clients and then filtered the packets with that info, MQTT topic “factory/department” and MQTT publish type. We hadn’t found messages with just one level under the department, as required from the question, because all the messages have two levels under the factory/department*/. We found 6 messages published by “jane” to the topics “factory/department*/section*/device”
   Filter: *(tcp.srcport == 42821 || tcp.srcport == 50985 || tcp.srcport == 40005 || tcp.srcport == 40989) && (ip.src == 127.0.0.1) && (mqtt.topic contains "factory/department") && (mqtt.msgtype == 3)*
5) We have looked to the destination IPs of the broker hivemq with the first filter and then we have filtered the connect MQTT messages with Will flag set and destination IP taken from first filter. The number of messages with second filter is 16.
   Filter 1: *(dns.qry.name == "broker.hivemq.com")*

Filter 2: *(mqtt.msgtype == 1) && (mqtt.conflag.willflag == 1) && ((ip.dst == 18.185.199.22) || (ip.dst == 3.120.68.56))*

6) The client keeps retransmitting the packet while an ACK is received, so the retransmitted packet has the DUP flag (duplication flag) set to 1. The number of messages with the characteristics required is two.
   Filter: *((mqtt.msgtype == 3) && mqtt.qos == 1) && (mqtt.dupflag == 1)*
7) The number of Last Will Message delivered to broker with QoS = 0 is 19.
   The number of Last Will Message delivered to clients because of hard disconnection is 1. We have found that every will message contains the word “error”, so looking at Publish messages with message containing the word “error” we have found only frame number 4164 (the corresponding connect frame is 3619)
   Filter 1: *mqtt.conflag.qos == 0 && (mqtt.conflag.willflag == 1)*
   Filter 2: *(mqtt.msg contains "error:") && (mqtt.qos == 0)*
8) The client “4m3DWYzWr40pce6OaBQAfk” has this IP and port: 10.0.2.15:58313
   It sends only one message with QoS = 2 (frame number 2423) found with filter 1. By looking for the ACKs with filter 1 and 2, we found that the broker answers with the PUBREC (frame number 2425) but the client will never answer with the PUBREL, so the broker will never send the message to subscribers.
   Filter 1: *ip.src == 10.0.2.15 && tcp.srcport == 58313 && mqtt*
   Filter 2: *ip.dst == 10.0.2.15 && tcp.dstport == 58313 && mqtt*
9) The average message length of connect messages using MQTT v5 is 91.08
   Messages have different sizes because of the fields of the packet that not always are present (will topic, will message, username, password, client ID, etc)
   Statistics > Packet Lenghts with filter *(mqtt.ver == 5) && (mqtt.msgtype == 1)*
10) By looking to different packets, it appears that the Keep Alive timer is frequently longer than the window of observation of this pcap file, while the packets with shorter Keep Alive have activity before the timer expires.