Raffaele Cicellini 10626081
Francesca Forbicini 10628756

Challenge 01 – Internet of Things

1. Filter: “coap.mid==53533 || coap.mid==42804”
   The message with MID 53533 is a GET request and is of type “Confirmable” (in fact, there is an ACK response related to this message), whereas the message with MID 42804 is a DELETE request and is of type “Non-Confirmable” (in fact, no response is present related to this message).
2. The response to message no. 2428 is message no. 2429: it is a Non-Confirmable message with code 2.02, which means that the resource was deleted successfully as requested by message no. 2428. No filter was used in this case.
3. Filter: “coap && ip.dst==127.0.0.1 && coap.code==69”. Applying this filter, we see that 8 replies of this kind are received by client localhost. The replies’ numbers are: 90, 1047, 1337, 2124, 2537, 2673, 2921, 3055.
4. Filter: “coap.code==1 || coap.code==132”. Applying this filter, we obtain all GET requests and all responses with code “Not found”. We search for matching request-response without the optional field “Observe” and obtain 6 requests that answer the question: Message 2430 (with response 2431), message 2712 (2713), 2806 (2807), 2836 (2837), 3052(3053), 4911(4912).
5. We first find tcp ports for users connecting with password “admin” (filter “mqtt.passwd==”admin””), then we search for publish messages coming from one of these tcp ports (filter “mqtt.msgtype==3 && (tcp.srcport== 51565 || 41869 ||60395||47135||44429||40989||60419||55953)”). Then we select only messages with the requested topic. Doing this, we find 13 messages. We consider as wildcard “factory/department*/#”, if we use the “+” we find 0 messages.
6. We ping test.mosquitto.org to retrieve the destination ip address for the public broker, then we apply filter “mqtt.willmsg && ip.dst==5.196.95.208”, thus obtaining 9 clients connected.
7. If we apply filter “mqtt.msgtype==6” to find PUBREL we don’t find anything. So it means that all 94 publish messages didn’t receive PUBREL.
8. We consider only messages that specify the Will Message applying filter “mqtt.willmsg && mqtt.clientid_len==0”, thus obtaining an average Will Topic Length of 37.77 bytes.
9. We search messages from the requested client with filter “mqtt.clientid==6M5H8y3HJD5h4EEscWknTD”, thus obtaining the correspondent tcp port. Next, we search for messages with this port as tcp destination port: we see that it received 1 Connect Ack (message no. 4764), 3 Subscribe Acks in the same message (no. 4766), 1 Publish Ack (message no. 4768). These are Acks at MQTT level (application level), it also received 6 Acks (1 of which Syn Ack) at TCP level (transmission level).
10. We select only Connect messages with MQTT version 3.1 with filter “mqtt.msgtype==1 && mqtt.ver=3”, then we compute the average of MQTT message length obtaining 63.596 bytes. The different sizes are due to the presence of optional fields in the message, such as the Will Message and Topic, the Username and Password, each of which can have different lengths depending on how long the message, topic, username, or password specified by the user is.