IoT 2023 CHALLENGE 1

(1) Name: Giovanni De Lucia Person Code: 10700658

(2) Name: Lorenzo Battiston Person Code: 10618906

Referring to the file challenge2023_1.pcap, answer the following questions:

1. How many CoAP GET requests are directed to non-existing resources in the local CoAP server? How many of these are of type Non confirmable? **(0.2 pts)**

   Result: 11 and 6 of those are NON confirmable

   Filter: “coap.code==1 && ip.dst==127.0.0.1”, to find GET requests to local CoAP server

   This has a result of 306 packets, then we extracted those packets as JSON and filtered them with a Python script by unique message ID and token. Then using the filters: “coap.token in **TOKEN** && coap.code==132”, and “coap.mid in **MID** and ip.src==127.0.0.1 and coap.code==132”; where **TOKEN** and **MID** are the output of the script. The results are 5 packets ‘following’ the mid and 9 with the token.

   As a proof also the other way around has been done (first filters not found response then find the relative request and filter by GET)

   NOTE: There is an overlap of 3 req/res that has the same token and mid

2. How many CoAP DELETE requests directed to the "coap.me" server did not produce a successful result? How many of these are directed to the "/hello" resource? **(0.2 pts)**

   Result: 105 and 5 of those are directed to the “/hello” resource

   DNS solves coap.me as 134.102.218.18 then the filter “coap.code==4 && ip.dst==134.102.218.18” gets all the delete directed to “coap.me”, they are 115.

   With “coap.code==66 && ip.src==134.102.218.18” we get all the DELETE OK reply messages, these indicate the successful result. They are 10.

   Then we have 115-10 = 105 unsuccessful result.

   We note that none of the DELETE OK messages refer to the /hello resource, then all the delete requests on that resource are unsuccessful. Using coap.opt.uri_path=="hello" in AND with the first filter we find out that we have 5 unsuccessful delete to the “hello” resource

3. How many different MQTT clients subscribe to the public broker mosquitto using single-level wildcards? How many of these *clients* **WOULD** receive a publish message issued to the topic "hospital/room2/area0“ **(0.2 pts)**

   Result: 3 different clients and 2 of them would receive a publish message on that topic

   Filter: (“mqtt.msgtype==8 && ip.dst ==91.121.93.94”) to find all the subscribe request, then filter the 13 that specifies a single-level wildcards. Now we have 3 different client since there are 3 different srcport ({'51531', '35239', '43133'}) and we check that at least one of the subscription for each client received a SUBACK (in fact, all the subscription was acked).

For the second part we used the filter: “mqtt.msgtype==8 && ip.dst==91.121.93.94 && tcp.srcport==**SRC”**, src is 51531 or 35239, 43133 to get ALL the subscription of those clients (even the one WITHOUT the + wildcards). And among these just 2 clients (35239, 43133) have a subscription that matches that topic (hospital/room2/area0)

Assumption: We think that response to the second part is NOT 1 since we should consider ALL the subscription by each client (the ones that have at least 1 subscription with a + wildcard) and see if they match, even the ones that doesn’t contain the wildcards

4. How many MQTT clients specify a last Will Message directed to a topic having as first level "university"? How many of these Will Messages are sent from the broker to the subscribers? **(0.2 pts)**

   Result: 2 client has a Last Will directed to a topic having as first level “university” and 0 of these messages are sent to the subscribers

   Filter: “mqtt.willtopic_len>0”, gives the Connect Commands with a LW, among these 7 only 2 have /university/ as first level (no. 1697, 35836). None of the Will Messages are sent to the subscribers, (filter “mqtt.topic == **TOPIC** ip.src == **BROKER”**, where topic is the one of the Last Will and none of the results display the error message and broker is the IP address of the broker to which the clients are connected)

   Note that the first part of this question changes using a newer version of Wireshark since one of the Connection Packet (no. 35726) is not marked as malformed while on the VM it is.

5. How many Publish messages with QoS = 1 are received by the MQTT clients connected to the HiveMQ broker with MQTT version 5? **(0.1 pts)**

   Result: 60
   Filter: “mqtt.ver==5 && (ip.dst==52.29.173.150 || ip.dst==3.65.137.17)”, find all the clients connected to the HiveMQbroker with MQTT version 5 (they are 9)
   “mqtt.msgtype==4 && (ip.dst==3.65.137.17 || ip.dst==52.29.173.150) && tcp.srcport in **PORTS”**, where ports are the one of the connected clients with MQTT version 5 found before. It gives 60 ACKs then there are 60 Pub messages with QoS=1 that are received by the clients above
   Note1: with “(ip.src==3.65.137.17 || ip.src==52.29.173.150) && tcp.dstport in **PORTS** && mqtt.qos==1” we can found ALL the Publish messages that has been sent (they are 73). We have done a script that count how many of them have mqtt.qos==1 and they are 60.
   Note2: in the VM ALL the 60 PUBACK above are malformed, this is due to the older version of Wireshark MQTT dissector that refers to an older protocol version. We think that even if they are malformed, they are still a proof that the messages were received by the clients (even if not correctly acknowledged)

6. How many MQTT-SN (on port 1885) publish messages sent after the hour 3.16PM (Milan Time) are directed to topic 9? Are these messages handled by the server? **(0.1 pts)**

   Result: 15. Not handled by the server

   Filters: “udp.dstport==1885 && mqttsn.topic.id == 9 && mqttsn.msg.type == 0x0c”, there are 28 packets and 15 are sent after the 3.16PM (frame > Arrival Time)

And the second part is no because we can see that all the publish messages are accompanied by a “ICMP 82 Destination unreachable (Port unreachable)” packet.