POLITECNICO
MILANO 1863

SCUOLA DELL'INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE

COMPUTER SCIENCE AND ENGINEERING

Internet Of Things

2nd Challenge

Simone Pio Bottaro - 10774229

Gabriele Lorenzetti - 10730455

A.Y 2024/2025

Introduction

The purpose of this challenge is to analyse a file .pcap in order to respond to some questions about the packet traffic. In order to solve this exercise, we decided to use only WireShark, which is a program that allows the user to capture and display details of network traffic.

For each question, are been reported the filters used in WireShark with related explanations.

Challenge Questions and Responses

CQ1) How many different Confirmable PUT requests obtained an unsuccessful response from the local CoAP server?

((coap.code == 3 && coap.type == 0) or (coap.code >= 128 && coap.type == 2)) && ip.src == 127.0.0.1

- coap.code == 3 && coap.type == 0: this part of the filter catches PUT requests that use Confirmable messages
- coap.code >= 128 && coap.type == 2: this part of the filter captures error responses in an ACK message
- ip.src == 127.0.0.1: this part ensures that the packet is sent from the local server

Verify that requests and responses have the same MID and token, and that tokens are different for different requests, **the requests are 22**.

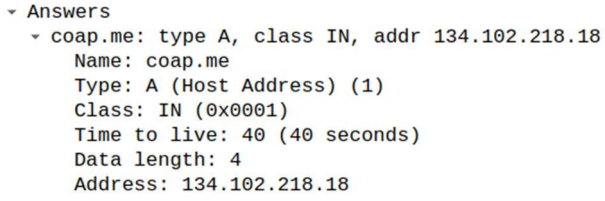
CQ2) How many CoAP resources in the coap.me public server received the same number of unique Confirmable and Non-Confirmable GET requests?

coap.code ==1 && (coap.type == 0 or coap.type == 1) && ip.dst == 134.102.218.18

- coap.code ==1: this part selects only GET requests
- coap.type == 0 or coap.type == 1: this part selects types CON and NOT
- ip.dst == 134.102.218.18: restrict analysis only to packets sent towards coap.me server

DNS traffic and packet specifications were analysed to find the IP address of coap.me and it is 134.102.218.18.

- dns.qry.name contains "coap.me"



Manually counting the number of CONs and NONs of each resource, **the result is 3.**

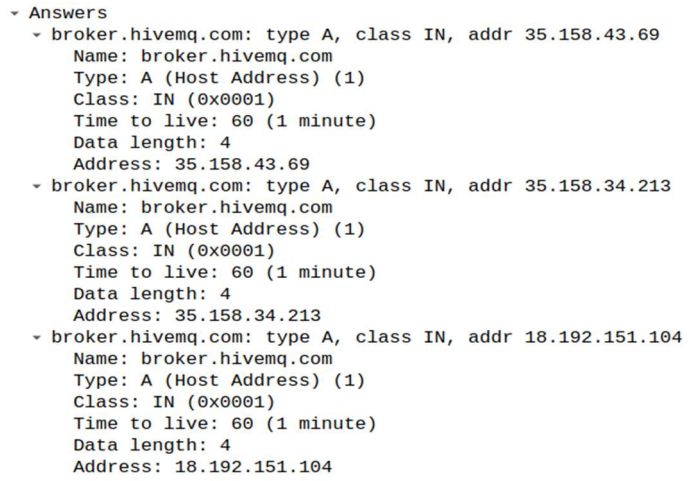
CQ3) How many different MQTT clients subscribe to the public broker HiveMQ using multi-level wildcards?

mqtt.msgtype == 8 && mqtt.topic contains "#" && (ip.dst == 18.192.151.104 or ip.dst == 35.158.34.213 or ip.dst == 35.158.43.69)

- mqtt.msgtype == 8: this part only captures SUBSCRIBE messages
- mqtt.topic contains "#": this part only filters out subscriptions that use #
- ip.dst == 18.192.151.104 or ip.dst == 35.158.34.213 or ip.dst == 35.158.43.69: restrict analysis only to packets sent towards HiveMQ server

DNS traffic and packet specifications were analysed to find the IP address of HiveMQ and they are: 35.158.43.69, 35.158.34.213 and 18.192.151.104

- dns.qry.name contains "hivemq"



Only counting those with different source ports for different clients, **the result is 4.**

CQ4) How many different MQTT clients specify a last Will Message to be directed to a topic having as first level "university"?

mqtt.conflag.willflag == 1 && mqtt.willtopic contains "university"

- mqtt.conflag.willflag == 1: this part only captures connections (CONNECT) with a Last Will set
- mqtt.willtopic contains "university": this part only selects packets with an last Will Message assigned to a topic containing ‘university’.

The result is 1 by checking as first level "university”:

CQ5) How many MQTT subscribers receive a last will message derived from a subscription without a wildcard?

First of all, CONNECT packages with a last will message set were identified.
mqtt.conflag.willflag == 1

- selects only MQTT packets of type CONNECT where the Will Flag bit is set to 1, i.e. active

Then subscriptions without wildcards were found for the various topics previously found; only 'university/department12/room1/temperature' appears to have them.
mqtt.msgtype == 8 && mqtt.topic == "university/department12/room1/temperature"

- mqtt.msgtype == 8: filters only MQTT packets of type SUBSCRIBE
- mqtt.topic == "university/department12/room1/temperature": check that the topic of the posted message is exactly 'university/department12/room1/temperature'

Then the PUBLISH packets are checked to see if any last will messages were actually sent on that particular topic and subscriber, controlling the Ports, and that the Will Message is the same as the one set.
mqtt.msgtype == 3 && mqtt.topic == "university/department12/room1/temperature"

- mqtt.msgtype == 3 : filters only MQTT packets of type PUBLISH
- mqtt.topic == "university/department12/room1/temperature": check that the topic of the posted message is exactly 'university/department12/room1/temperature'

The final result is 3.

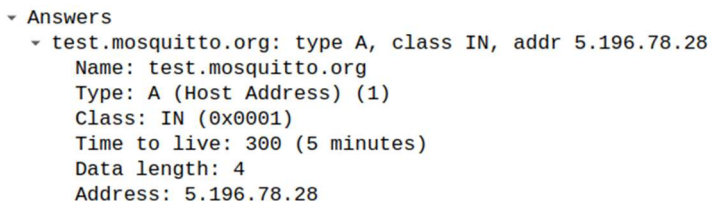
CQ6) How many MQTT publish messages directed to the public broker mosquitto are sent with the retain option and use QoS "At most once"?

mqtt.msgtype == 3 && mqtt.retain == 1 && mqtt.qos == 0 && ip.dst == 5.196.78.28

- mqtt.msgtype == 3: this part only captures PUBLISH packages
- mqtt.retain == 1: this part only captures messages with the retain flag active
- mqtt.qos == 0: this part only captures messages with QoS 0
- ip.dst == 5.196.78.28: restrict analysis only to packets sent towards public broker mosquitto

DNS traffic and packet specifications were analysed to find the IP address of mosquitto and it is 5.196.78.28.

- dns.a == 5.196.78.28



The result is 208.

CQ7) How many MQTT-SN messages on port 1885 are sent by the clients to a broker in the local machine?

mqttsn && udp.port == 1885

- mqttsn: this part ensures that only MQTT-SN packets are selected
- udp.port == 1885: this part ensures that only packets on UDP port 1885 are selected

There are no messages matching this filter.