POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

IoT Challenge #2, Packet Sniffing

Internet of Things

Authors: **Kevin Ziroldi - 10764177**
Matteo Volpari - 10773593

Professors: Alessandro Redondi, Fabio Palmese
Academic Year: 2024-2025
Version: 1.0
Release date: 6-4-2025

Contents

1 | Packet sniffing PCAP file

1.1. CQ1

Question

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

Answer

TODO

Explanation

Domanda 1 Confirmable put request

```
coap && coap.type == 0 && coap.code == 3
```

45 frames Response

```
coap && (coap.code >= 128)
```

228 frames Dovrei matcharli per token o message id (quale????), troppi. In realtà posso filtrare anche ip src = ip dst. Quindi pyshark.

1.2. CQ2

Question

How many CoAP resources in the coap.me public server received the same number of unique Confirmable and Non Confirmable GET requests?
Assuming a resource receives X different CONFIRMABLE requests and Y different NON-CONFIRMABLE GET requests, how many resources have X=Y, with X>0?

Answer

TODO

Explanation

Domanda 2 Get request confirmable a coap.me

```
coap.type == 0 && coap.code == 1 && ip.dst==134.102.218.18
```

39 frames Get non confirmable a coap.me

```
coap.type == 1 && coap.code == 1 && ip.dst==134.102.218.18
```

31 frames Dovrei vedere a quale risorsa fanno riferimento e poi confrontare. Troppo, quindi pyshark.

1.3. CQ3

Question

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

Answer

The number of clients who subscribe to the public broker HiveMQ using multi-level wildcards is 4.

Explanation

In order to find the IP address of the HiveMQ broker, we filter the response of the DNS server using the following Wireshark filter:

```
dns.qry.name == "broker.hivemq.com"
```

All DNS responses return 3 addresses: 18.192.151.104, 35.158.34.213 and 35.158.43.69.

We use a second filter to find SUBSCRIBE messages, with message type 8, sent to HiveMQ broker, to one of the IP addresses found above, with a multi-level wildcard, ending with "#":

```
mqtt && mqtt.msgtype == 8 &&
(ip.dst == 18.192.151.104 || ip.dst == 35.158.34.213
```

```
|| ip.dst == 35.158.43.69) && mqtt.topic contains "#"
```

We find out that HiveMQ broker receives 6 messages of this type, all at the IP address 18.192.151.104.

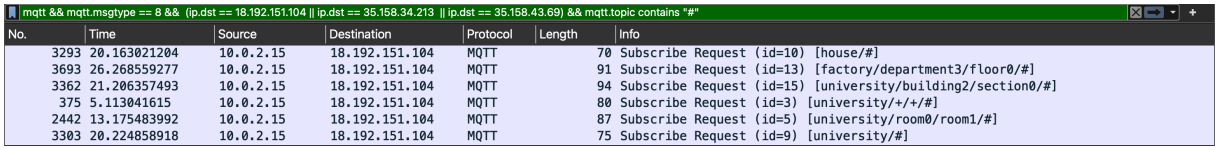

Figure 1.1: SUBSCRIBE messages to HiveMQ broker with "#"

Since the question asks for the number of MQTT clients who subscribe, we need to identify the clients who sent these messages. For each message, we select the TCP stream, which identifies the client.

Message number	TCP stream
375	8
2442	15
3293	20
3303	15
3362	3
3693	15

Table 1.1: TCP streams

Since there are 4 TCP streams, the 6 messages have been sent by 4 different client.
We can also find the Client ID of these clients by finding the CONNECT message, of type 1, they sent to the broker. For the TCP stream 8, we can use the following filter:

```
mqtt && mqtt.msgtype == 1 && tcp.stream == 8
```

The same filter with different TCP stream can be used for other clients.

TCP stream	Client ID
3	cpoepjzkhibxgjiu
8	dzcxnwdqef
15	tukvxesuhe
20	fcthvjikxjul

Table 1.2: Client IDs table

1.4. CQ4

Question

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

Answer

The number of clients who specify a Last Will Message to be directed to a topic having as first level "university" is 1.

Explanation

MQTT clients can specify a Last Will Message in the CONNECT message. In order to find the described messages, we filter CONNECT messages, of type 1, with a Last Will Topic:

```
mqtt && mqtt.msgtype == 1 && mqtt.willtopic
```

We find four messages, but only one of them has a Last Will Topic having as first level "university".

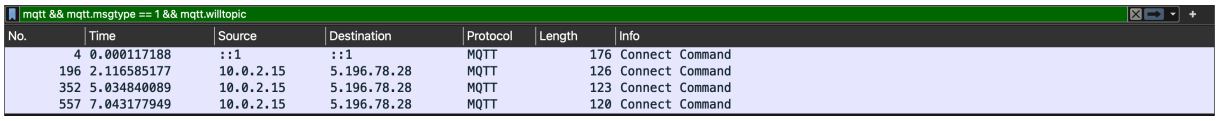

Figure 1.2: CONNECT messages specifying a Last Will Topic

We can find the result by enriching the filter and avoiding manually checking the topics, using the following filter:

```
mqtt && mqtt.msgtype == 1 && mqtt.willtopic matches "^university"
```

Using this filter, we directly get the only message asked by CQ4.

Figure 1.3: CONNECT messages specifying a Last Will Topic starting with "university"

1.5. CQ5

Question

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

Answer

The number of subscribers who receive a Last Will Message derived from a subscription without a wildcard is 3.

Explanation

We start by identifying the possible Last Will Messages (LWM). To do so, we find all the CONNECT messages, with message type 1, that specify a LWM.

```
mqtt && mqtt.msgtype== 1 && mqtt.willmsg
```

We find four messages.

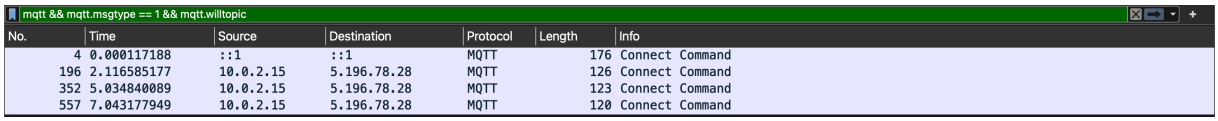

Figure 1.4: CONNECT messages specifying a LWM

These messages specify the Last Will Messages, which we don't report in the table, and Last Will Topics.

Message number	Destination	Last Will Topic
4	::1	university/department12/room1/temperature
196	5.196.78.28	metaverse/room2/floor4
352	5.196.78.28	hospital/facility3/area3
557	5.196.78.28	metaverse/room2/room2

Table 1.3: Last Will Topics

Starting from the first Last Will Topic (LWT), we filter all PUBLISH messages with that topic and with same message as the LWM, found in the CONNECT message.

```
mqtt && mqtt.msgtype==3 &&
mqtt.topic == "university/department12/room1/temperature" &&
mqtt.msg contains 6572726f723a20612056495020436c69656e74206a7573742064696564
```

We find four messages:

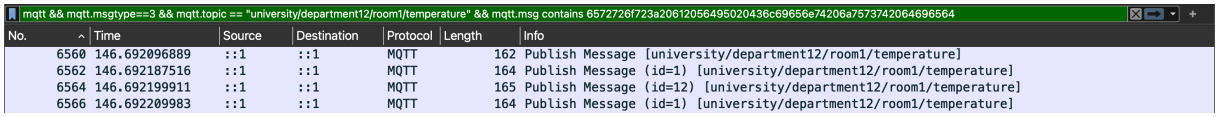

Figure 1.5: LWM with topic "university/department12/room1/temperature"

By looking at the messages, we can see a TCP Reset message, representing an hard disconnection, followed by four Last Will Messages sent by the broker, on port 1883, to different client, on different ports and using different TCP streams.

Figure 1.6: TCP Reset and Last Will Messages

The clients who receive the LWM are grouped in the following table.

Message number	Subscriber port	TCP stream
6560	39551	2
6562	53557	6
6564	51743	10
6566	41789	14

Table 1.4: Last Will Topics

In order to answer to QC5, we need to find which of these clients subscribed to the Last Will Topic without a wildcard. We can do so by filtering the SUBSCRIBE messages, with message type 8, with the correct TCP stream. For each of them we also find the CONNECT message and Client ID.

```
mqtt && mqtt.msgtype == 8 && mqtt.topic matches "^university" &&
(tcp.stream == 2 || tcp.stream == 6 || tcp.stream == 10 || tcp.stream == 14)
```

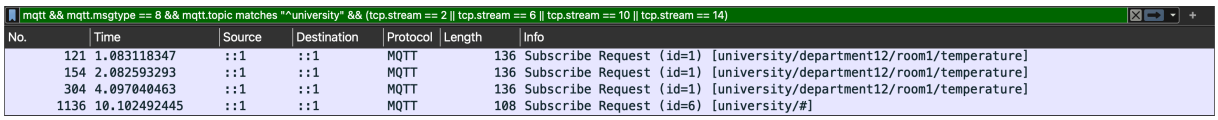

Figure 1.7: TCP Reset and Last Will Messages

We can see that only three out of four clients subscribed to the Last Will Topic without a wildcard.

TCP stream	Client ID	Specified topic
2	auyvhrhdudnm	university/department12/room1/temperature
6	ntpiopsqc	university/department12/room1/temperature
10	zmjnxudohrkaegmh	university/#
14	mjdocmjxt	university/department12/room1/temperature

Table 1.5: Specified topics

For what concerns the other three Last Will Topics, we filter all PUBLISH messages, with message type 3, from the broker, with IP 5.196.78.28.

```
mqtt && ip.src == 5.196.78.28 && mqtt.msgtype == 3
```

We don't find any result, which means that the broker doesn't publish any message, nor LWM.
We conclude that three clients receive a LWM from a subscription without a wildcard, all of them from the first topic.

TODO Però non ha retain true!

```
mqtt && mqtt.msgtype==3 && mqtt.topic == "university/department12/room1/temperature" &&
6572726f723a20612056495020436c69656e74206a7573742064696564 && mqtt.retain == 1
```

Nessun risultato Devono avere il retain? Se sì, allora non sono last will, altrimenti??? Non penso. Il retain dovrebbe essere sul messaggio di connect, non su quello di last will!

1.6. CQ6

Question

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

Answer

TODO

Explanation

Domanda 6:

utilizzando il filtro: dns.qry.name == "test.mosquitto.org" trovo l'indirizzo ip collegato al dominio richiesto che poi mi servirà per filtrare i pacchetti, l'IP è: 5.196.78.28. quindi filtro i pacchetti con: mqtt.msgtype == 3 and mqtt.qos == 0 and mqtt.retain == 1 and ip.dst == 5.196.78.28 e trovo tutti quelli che rispettano la richiesta: 208 pacchetti.

1.7. CQ7

Question

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

Answer

TODO

Explanation

Domanda 7: Il protocollo MQTT-SN non è riconosciuto nativamente da Wireshark ma sappiamo da teoria che è un protocollo udp. Filtrando quindi con udp.port == 1885 non trovo nessun pacchetto e quindi non sono stati inviati pacchetti sulla porta 1885.

References:

`https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html#_Toc3901022`