RP2_P1_MQTT

Alumnos

Sergio Semedi Barranco

Lucas Segarra Fernández

Capturando tráfico

Para comprobar el correcto funcionamiento del broker, podemos suscribirnos utilizando los siguientes comandos:

- Nos suscribimos en la Ci40:

  ```
  mosquitto_sub -h <ip_publisher> -p 1883 -t 'topic1'
  ```

- Publicamos desde la máquina virtual:

  ```
  mosquitto_pub -h <ip_publisher> -p 1883 -t 'topic1' -m 'hello'
  ```

Para capturar y analizar el tráfico generado, utilizamos las herramientas tcpdump y wireshark respectivamente. La llamada concreta desde la máquina virtual es la siguiente:

```
tcpdump -i interface -w file
```

La salida obtenida se encuentra en este repositorio.

Ejercicio. Describir los mensajes MQTT que se observan.

Abriendo con wireshark el fichero generado con tcpdump, y aplicando el filtro *mqtt*, observamos los siguientes paquetes:

No.	Time	Source	Destination	Protocol	Length	Info
12	1.788921	192.168.1.91	192.168.1.1	MQTT	101	Connect Command
14	1.789125	192.168.1.1	192.168.1.91	MQTT	70	Connect Ack
16	1.792978	192.168.1.91	192.168.1.1	MQTT	79	Subscribe Request
17	1.793027	192.168.1.1	192.168.1.91	MQTT	71	Subscribe Ack
33	16.310729	192.168.1.1	192.168.1.91	MQTT	81	Publish Message

Como aclaración, la IP terminada en 91 pertenece a la ci40, y la que termina en 1 a la MV (publisher).

Como observamos, cuando la máquina comienza a publicar no se produce tráfico alguno, porque sabe que nadie está suscrito.

Es al suscribirnos desde la ci40 cuando ésta envía el mensaje de conexión que el

publicante confirma (mensajes 12 y 14). Una vez ha recibido la confirmación, el dispositivo suscrito informa al publicante de cuales son los topics a los que está suscrito. En la información relativa al protocolo MQTT del paquete 16 observamos existe un único campo llamado *Suscribe Request* con los siguientes datos:

Field	Data
1000 0010 = Header Flags	0x82 (Subscribe Request)
Msg Len	11
Message Identifier	1
Topic	topic1
.... ..00 = Granted Qos	Fire and Forget (0)

Como observamos en la cabecera MQTT de la confirmación (*Suscribe ACK*), el identificador sirve para determinar qué subscripción se está confirmando.

Field	Data
1001 0000 = Header Flags	0x90 (Subscribe Ack)
Msg Len	3
Message Identifier	1
.... ..00 = Granted Qos	Fire and Forget (0)

Una vez finalizado el proceso de subscripción, el publicante envía al subscrito las publicaciones relativas a los temas pertinentes, como podemos apreciar en el paquete 33.

Ejercicio. Explicar los segmentos TCP y tramas ARP que se observan.

Aunque los protocolos MQTT y SSH funcionan sobre el protocolo de transporte TCP, entendemos que el objetivo de este ejercicio es analizar el resto de paquetes que se intercambian utilizando este protocolo.

Al estar conectados por SSH la MV y el dispositivo periférico, por el mismo puerto que hemos escuchado con tcpdump, hemos capturado también los paquetes derivados de tal comunicación.

Aplicando el filtro *!ssh*, eliminamos tales paquetes de la visualización en wireshark.

Sin embargo, los dos primeros y el último segmento TCP que observamos, paquetes 5, 8 y 36 corresponden precisamente con confirmaciones a los paquetes SSH 4, 7 y 35 respectivamente, como podemos comprobar con el número de secuencia y longitud de estos paquetes SSH, y el de confirmación del TCP que los sucede.

No	Time	Source	Destination	Protocol	Length	Info
5	0.180521	192.168.1.1	192.168.1.91	TCP	66	50066 → 22 [ACK] Seq=53 Ack=117 Win=608 Len=0 TSval=718032781 TSecr=159417
8	1.768380	192.168.1.1	192.168.1.91	TCP	66	50066 → 22 [ACK] Seq=105 Ack=169 Win=608 Len=0 TSval=718033178 TSecr=159575
9	1.785745	192.168.1.91	192.168.1.1	TCP	74	34564 → 1883 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=159577 TSecr=0 WS=256
10	1.785815	192.168.1.1	192.168.1.91	TCP	74	1883 → 34564 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=718033181 TSecr=159577 WS=128
11	1.787807	192.168.1.91	192.168.1.1	TCP	66	34564 → 1883 [ACK] Seq=1 Ack=1 Win=29440 Len=0 TSval=159577 TSecr=718033181
13	1.789051	192.168.1.1	192.168.1.91	TCP	66	1883 → 34564 [ACK] Seq=1 Ack=36 Win=29056 Len=0 TSval=718033183 TSecr=159577
15	1.791930	192.168.1.91	192.168.1.1	TCP	66	34564 → 1883 [ACK] Seq=36 Ack=5 Win=29440 Len=0 TSval=159578 TSecr=718033183
18	1.829736	192.168.1.91	192.168.1.1	TCP	66	34564 → 1883 [ACK] Seq=49 Ack=10 Win=29440 Len=0 TSval=159582 TSecr=718033183
34	16.317661	192.168.1.91	192.168.1.1	TCP	66	34564 → 1883 [ACK] Seq=49 Ack=25 Win=29440 Len=0 TSval=161030 TSecr=718036813
36	16.318393	192.168.1.1	192.168.1.91	TCP	66	50066 → 22 [ACK] Seq=105 Ack=221 Win=608 Len=0 TSval=718036814 TSecr=161030

Figure 1: tcp frames

Los paquetes 9,10 y 11, corresponden con el establecimiento de sesión clásico de TCP (3 vías).

Los paquetes 12-18 corresponden al establecimiento de sesión MQTT. Produciéndose envíos y confirmaciones a dos capas de red distintas (App y Transporte).

Así la conexión MQTT solicitada en el paquete 12, es confirmada a nivel TCP en el 13 y a nivel MQTT en el 14, dicha confirmación MQTT es a su vez confirmada a nivel TCP en el paquete 15. En el paquete 17 en cambio se confirma a ambos niveles la subscripción (al topic1) solicitada en el 16. Dicha confirmación tambien es confirmada a nivel TCP en el 18.

Los paquetes 33 y 34, son la publicación y confirmación del mensaje hello en el topic1.

Los únicos paquetes ARP que hemos capturado:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	CrispAut_3c:0c:99	Imaginat_89:12:36	ARP	42	Who has 192.168.1.91? Tell 192.168.1.1
2	0.015640	Imaginat_89:12:36	CrispAut_3c:0c:99	ARP	60	192.168.1.91 is at 00:19:f5:89:12:36

Figure 2: ARP frames

Corresponden a la MV preguntando por la MAC de la ci40, y a la respuesta a dicha pregunta.

En esta práctica se propone realizar un programa que publique la temperatura del sensor a través de MQTT. Para ello solicitaremos la ip y el topic por consola y realizaremos la publicación desde la Ci40.

Para comprobar el correcto funcionamiento nos suscribiremos desde un cliente MQTT (la misma máquina virtual) y estudiaremos el tráfico generado con wireshark.

La siguiente imagen corresponde con una captura de pantalla de wireshark con la topología propuesta.

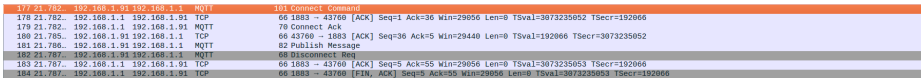

Figure 3: captura mqtt ci40

El fichero obtenido se encuentra en este repositorio.

Como podemos observar la comunicación MQTT es análoga a la estudiada en el ejercicio anterior. Como requisito de este ejercico se publica la temperatura detectada a través del sensor desde la ci40.

Observamos el contenido de la capa MQTT del paquete 181, que contiene un único campo publish mesaje con la siguiente información:

Field	Data
0011 0000 = Header Flags	0x30 (Publish Message)
Msg Len	14
Topic	topic1
Message	27.312

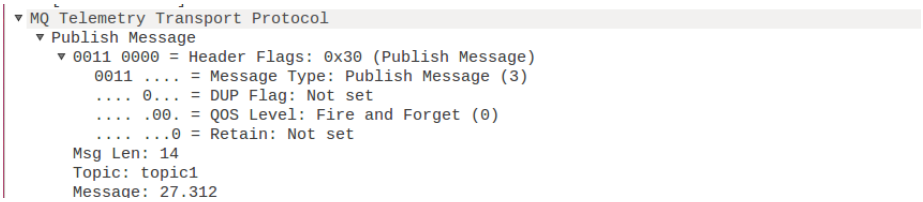

Figure 4: publish message