Premier League News

with

MQTT

- 1. Introduzione
- 2. **MQTT**
- 3. Strumenti di sviluppo
- 4. Wireshark
- 5. <u>News</u>
- 6. Considerazioni finali

Cipelli Lorenzo - 305784

&

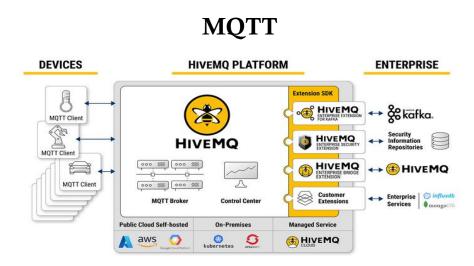
Cresci Nicholas - 310259

Introduzione

Il progetto realizzato consiste in un sistema di condivisione delle news centralizzato, ottenuto tramite l'utilizzo di un **broker MQTT** per la sincronizzazione e consegna dei messaggi agli utenti iscritti ai topic di interesse. I due broker utilizzati per questo progetto sono stati:

- mqtt.eclipseprojects.io, utilizzato per le connessioni senza uso del TLS;
- ff4612ecc8c647d7b6128392b277b68a.s1.eu.hivemq.cloud, utilizzato per le connessioni con uso di TLS;

Generalmente è corretto usare TLS, ma per scopo illustrativo abbiamo usato maggiormente la connessione senza TLS per poter leggere il contenuto del payload dei pacchetti su Wireshark. Il codice sviluppato riguarda una prima parte per ottenere le ultime news dal sito ufficiale della lega di calcio inglese: https://www.skysports.com/football/teams, riguardanti ogni squadra attualmente militante in Premier League. La seconda parte riguarda l'implementazione del codice del **publisher**, che si occuperà di pubblicare le news ricavate dalla prima parte sul broker. La terza parte riguarda la stesura del codice del **subscriber**, che potrà iscriversi a uno o più topic per ricevere le news della squadra di calcio che intende seguire.



Il protocollo **Message Queue Telemetry Transport**, è un protocollo con architettura di tipo publish/subscribe, che permette di avere un uso della rete ridotto dovuto all'assenza di polling e dovuto alla dimensione dei messaggi, sensibilmente ridotto rispetto a quelli HTTP, queste caratteristiche lo rendono perfetto per essere sfruttato all'interno di sistemi embedded e piattaforme mobili. I topics sono strutturati in maniera gerarchica, in uno stile auto-descrittivo. Una funzionalità molto importante dell'MQTT e quella del **Quality of Service**, QoS, realizzabile in tre diversi livelli:

- QoS 0: fire-and-forget, nessun controllo sulla trasmissione avvenuta;
- QoS 1: at-least-once, controllo della trasmissione avvenuta, possibilità di ripetizione della consegna del messaggio, opzioni più utilizzata;
- QoS 2: at-most-once, controllo più pesante della trasmissione, opzione meno utilizzata, si controlla che il pacchetto arrivi esattamente una volta;

Altre funzionalità testate nel codice sono state quelle di Retained Message, Will and Testament, Keep-Alive, Persistent Session.

Strumenti di sviluppo

Il progetto è stato realizzato in Python, con l'utilizzo della libreria **paho-mqtt**: https://pypi.org/project/paho-mqtt/, facilmente installabile seguendo le istruzioni del link. La libreria in questione fornisce gli strumenti per testare tutte le modalità di funzionamento dell'MQTT. Altra libreria utilizzata è **pandas**: https://pandas.pydata.org/docs/user_guide/, utilizzata indipendentemente dal protocollo MQTT, serve per creare il file csv agilmente e poter di conseguenza leggerlo nel codice del publisher per pubblicare le relative news immagazzinate.

Infine per il "web scraping" della prima fase, e l'analisi del contenuto dei file HTML dai cui sono state reperite le news si usa **BeautifulSoup**:

https://www.crummy.com/software/BeautifulSoup/bs4/doc/, questa libreria è molto "potente" per le situazioni in cui è necessario avere la struttura di un documento HTML riprodotta in un oggetto, saremo quindi in grado di poter sfruttare la programmazione ad oggetti per l'analisi dei tag del documento.

Wireshark

"Wireshark è un software per analisi di protocollo o "packet sniffer" utilizzato per la soluzione di problemi di rete, per l'analisi e lo sviluppo di protocolli o di software di comunicazione e per la didattica" (it.wikiperdia.org, s.d.).

Il software in questione è utilizzato per mostrare il contenuto dei pacchetti MQTT invitati rispettivamente tra publisher-broker e broker-subscriber. Con Wireshark siamo andati a verificare la differenza tra connessione sicura (TLS) e in chiaro, mostrando come nella connessione sicura non ci era possibile verificare il contenuto dei payload, mentre al contrario con la connessione non cifrata abbiamo verificato quanto studiato durante il corso sulla struttura dei pacchetti e opzioni sulle trasmissioni realizzabili con MQTT.

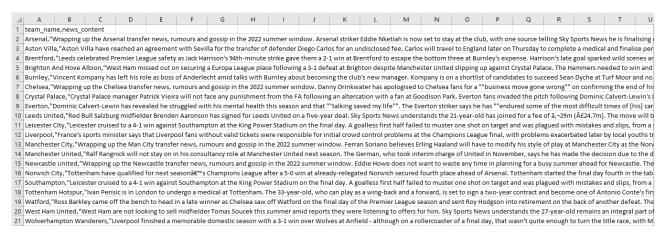
```
Length Info
                                              Destination
                                                                    Protocol
    714 116 562548
                       192.168.0.134
                                              137.135.83.217
                                                                    MOTT
                                                                                56 Ping Request
    716 116.676827
                       137.135.83.217
                                              192.168.0.134
                                                                    MQTT
                                                                                60 Ping Response
  Frame 716: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF_{59E01F2B-8D21-4CE9-B627-56ACB7E6A0C0}, id 0
Ethernet II, Src: SkyUk_29:80:f1 (6c:a0:b4:29:80:f1), Dst: HonHaiPr_64:1d:b9 (dc:a2:66:64:1d:b9)
 Internet Protocol Version 4, Src: 137.135.83.217, Dst: 192.168.0.134
  Transmission Control Protocol, Src Port: 1883, Dst Port: 50232, Seq: 3, Ack: 5, Len: 2

✓ MQ Telemetry Transport Protocol, Ping Response
    [Expert Info (Note/Protocol): Unknown version (missing the CONNECT packet?)]

→ Header Flags: 0xd0, Message Type: Ping Response
        1101 .... = Message Type: Ping Response (13)
        .... 0000 = Reserved: 0
     Msg Len: 0
                                                            ··fd··l· ·)····E·
·*··@·/· ·?··S···
0000 dc a2 66 64 1d b9 6c a0 b4 29 80 f1 08 00 45 00
0020 00 86 07 5b c4 38 b7 99 82 d9 62 13 f7 64 50 18 0030 01 f6 df c5 00 00 d0 00 00 00 00 00
                                                            ···[·8···b··dP
```

News

Le news raccolte nella prima parte dello sviluppo sono state raccolte all'interno di un file *csv*, il file in questione contiene sulla prima colonna il nome delle squadre, mentre sulla seconda colonna il contenuto delle ultime news per ognuna di queste squadre.



Considerazioni finali

Chiaramente il progetto non ha un peso elevato in quanto questa è una dimostrazione di funzionamento di un protocollo relativamente semplice e leggero. Nulla toglie che il sistema sviluppato possa essere testato in condizioni più stressanti, con diversi publisher e diversi subscriber. Pensiamo che questo progetto possa essere utilizzato a scopo didattico per una facile e intuitiva introduzione all'MQTT, data anche dal tema calcistico alla base del progetto, da anteporre ad uno studio più approfondito del protocollo.

La guida sull'esecuzione del codice è disponibile nel file papers/Instruction.txt fra i file sorgente del codice, oppure nel README.md della repository git al seguente link: https://github.com/lorenzocipelli/telematica