	CXX Macro-Compétence Compétence	**BTS CIEL** **TP**

TP2 : Sécurité de l'architecture MQTT

Objectifs

- Mettre en œuvre une architecture MQTT complète (broker / client pub et sub)
- Sécuriser l'architecture MQTT par identifiant/mot de passe
- Analyser une trame MQTT et la sécurité de l'architecture

Rappel

Mosquitto est une implémentation du protocole MQTT pour les deux rôles, broker et client.

- le **broker** (centralise les abonnements et redirige les messages publiés)
- le **client** (qui peut publier / s'abonner)

Lors du TP précédent nous avons utilisé le **client mosquitto**, qui communiquait avec deux brokers

- Le broker de l'enseignant (mosquitto)
- Le site Adafruit (l'interface MQTT est un broker)

Pendant ce TP vous utiliserez votre propre le **broker (mosquitto)**.

Installation du broker MQTT

☐ Démarrer la VM et mettre à jour le système

- Comment installer le **broker** MQTT sur Debian ? (en cherchant sur Internet)

Attention : Il faut bien différencier le numéro de version du logiciel Mosquitto et le numéro de version du protocole MQTT supporté par Mosquitto.

- En tant que logiciel, Mosquitto évolue et est identifié par un numéro de version.
- Par ailleurs, le protocole MQTT évolue également et plusieurs versions coexistent
- En vous aidant de la documentation, indiquer quels sont les **numéros de versions du protocole MQTT** supportées par la version actuelle de Mosquitto

- Installer le **broker** sur votre VM (copie d'écran)

- Quelle est la version de Mosquitto (broker) installée ?

Mise en œuvre du broker MQTT

Démarrage du broker

Le **broker** peut être démarré de deux façons :

- automatiquement en tant que service
- manuellement

broker en tant que service	**broker en commande manuelle**
`sudo service mosquitto` **`start`** `sudo service mosquitto` **`stop`** `sudo service mosquitto` **`status`**	`mosquitto [options]` si le programme n'est pas trouvé, entrer le chemin complet **`/usr/sbin/mosquitto [options]`**

Pour ce TP nous utiliserons le démarrage manuel : ainsi nous aurons l'affichage de l'activité du broker sur le terminal.

À l'installation, le service mosquitto est démarré **automatiquement** en tant que service, ce qui empêche le démarrage manuel d'une nouvelle instance.

- Noter la commande qui permet de savoir si le broker est démarré en tant que service

- Si besoin, arrêter le service mosquitto (noter la commande)

Quelques options pour démarrer le broker en mode manuel

Références en ligne : https://mosquitto.org/man/mosquitto-8.html

`-c <config file>`	Pour spécifier le fichier de configuration à charger Si aucun fichier n'est spécifié, les valeurs par défaut sont utilisées "Load configuration from a file. If not given, then the broker will listen on port 1883 bound to the loopback interface, and the default values as described in mosquitto.conf(5) are used."

-d	Démarre mosquitto en tant que daemon
-p XXXX	Numéro de port (port par défaut 1883)
-v	verbose = mode "verbeux" : le broker affiche son activité détaillée

- Démarrer **manuellement** le **broker** en mode **verbeux** dans un terminal qui restera ouvert (reporter la commande ci-dessous)

- Sur quel port le **broker** MQTT attend-il les requêtes ?

Requêtes acceptées par le broker

Ci-dessous un extrait de la documentation

`-c`,
`--config-file`

Load configuration from a file. If not given, then the broker will listen on port 1883 bound to the loopback interface, and the default values as described in mosquitto.conf(5) are used.

`-p`,
`--port`

Listen on the port specified. May be specified up to 10 times to open multiple sockets listening on different ports.

Important

In version 1.6.x and earlier, the listener defined by `-p` (or the default port of 1883) would be bound to all interfaces and so be accessible from any network. It could also be used in combination with `-c`.

From version 2.0 onwards, the listeners defined with `-p` are bound to the loopback interface only, and so can only be connected to from the local machine. If both `-p` is used and a listener is defined in a configuration file, then the `-p` options are IGNORED.

- Pour votre version du broker installée, y-a-t-il des restrictions de connexions au **broker** ? (justifier)

Requêtes locales

Sous TCP/IP, chaque machine se reconnaît en tant que

- hostname `localhost`
- adresse Ip `127.0.0.1`

On cherche à vérifier si l'architecture MQTT est opérationnelle en local, c'est-à-dire avec des requêtes "publish" et "subscribe" exécutées directement sur le host où tourne le broker.

- Conserver le terminal SSH avec l'activité du broker et ouvrir deux nouvelles sessions SSH sur votre VM
    - Sur la première, écrire une requête "subscribe" sur le topic "ciel" (noter la commande et le résultat)

    - Sur la deuxième, écrire une requête "publish" sur le topic "ciel" (noter la commande et le résultat)

    - Copier et commenter l'affichage sur le broker

- Conclure : Est-ce que les requêtes en local sont fonctionnelles ?

- Conclure : Est-ce que le résultat est conforme à vos attentes ?

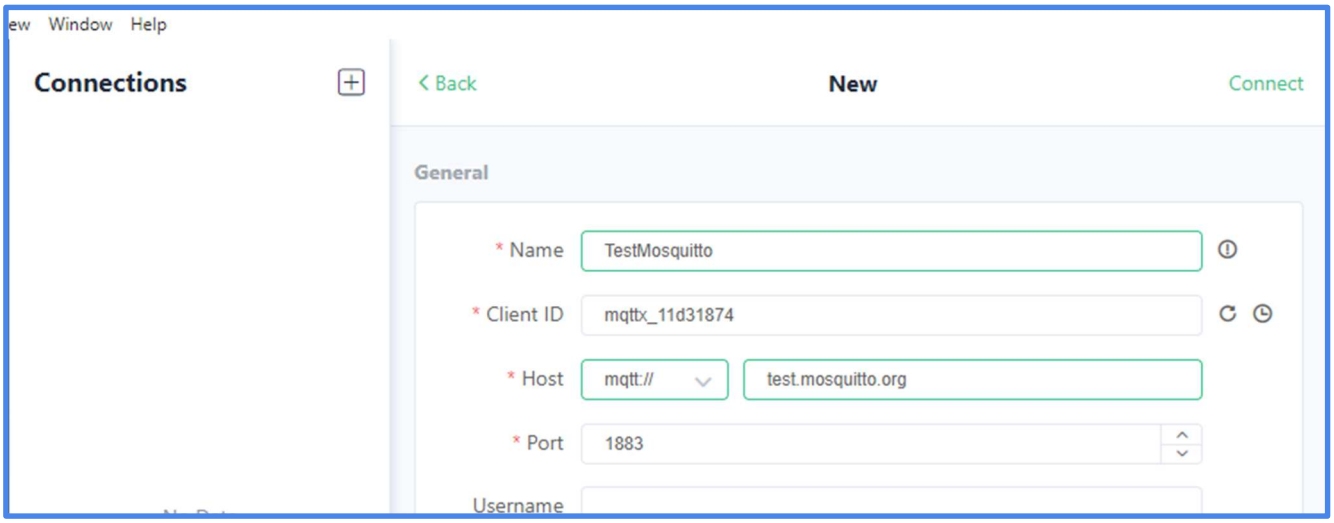
Requêtes distantes

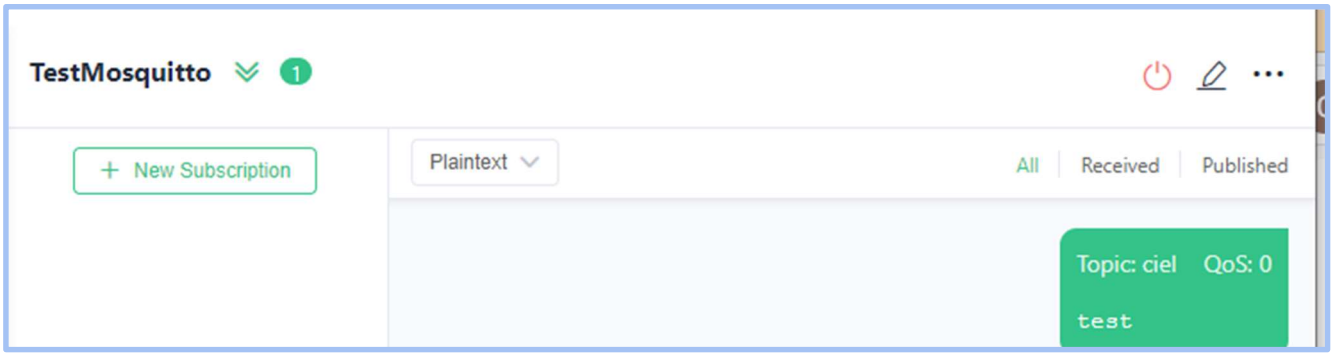
On cherche maintenant à vérifier si l'architecture MQTT est opérationnelle sur le réseau, c'est-à-dire avec des requêtes "publish" et "subscribe" exécutées sur une autre machine du réseau.

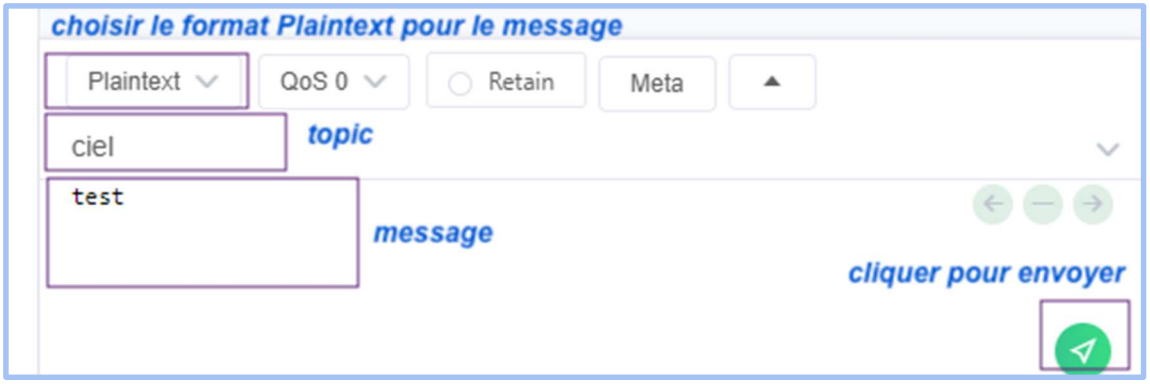
Nous aurons :

- le broker sur la VM Debian
- le client sur le PC Windows

- Installer le client MQTTX sur le poste Windows. Ce logiciel gratuit permet de gérer plusieurs connexions à des brokers et de créer facilement des requêtes PUBLISH et SUBSCRIBE (IHM graphique sous Windows)
- Prise en main du logiciel : vérifier que vous pouvez vous **connecter** au broker de test test.mosquitto.org et que vous pouvez envoyer une requête **PUBLISH**







- Insérer les captures d'écran

- Créer une nouvelle connexion au broker de la VM (copie d'écran)

- Noter et commenter l'affichage sur le broker

- Conclure : est-ce que la connexion distante est autorisée ?

- Conclure : Est-ce que le résultat est conforme à vos attentes ?

Configuration basique du broker MQTT

Fichier de configuration

Un fichier de configuration pour Mosquito est un fichier texte contenant des valeurs pour les paramètres du broker. Il est possible de télécharger un fichier de configuration mosquitto.conf qui contient tous les paramètres avec leurs valeurs par défaut. Tout est commenté.
Pour modifier un paramètre, il suffit de décommenter la ligne et de modifier la valeur.

Extrait :

```
# Config file for mosquitto
#
# See mosquitto.conf(5) for more information.
#
# Default values are shown, uncomment to change.
#
# Use the # character to indicate a comment, but only if it is the
# very first character on the line.

# =================================================================
# General configuration
# =================================================================

# Use per listener security settings.
#
…

# The default behaviour is for this to be set to false, which maintains the
# setting behaviour from previous versions of mosquitto.
#per_listener_settings false
```

- les lignes commençant par # ne sont pas analysées (commentaires)
- les lignes commencent par une variable suivie de sa valeur, les deux étant séparés par une espace

exemple :

`memory_limit` *`limit`*

Pour ce TP nous créerons un fichier de configuration contenant uniquement les paramètres différents des valeurs par défaut pour que ce soit plus lisible.

Le format du fichier ainsi que les variables possibles sont décrits dans la documentation : https://mosquitto.org/man/mosquitto-conf-5.html

Rappel : Si Mosquitto démarre sans fichier de configuration, il utilise les valeurs par défaut.

Configuration : connexion distante

- Créer un fichier “mosquitto.conf” dans votre home directory
- Ajouter les deux lignes suivantes pour que le broker écoute sur le port 1883 pour toutes les interfaces réseaux (et non pas seulement sur l’adresse de loopback) et qu’il s’agit du protocole MQTT

```
listener 1883
protocol mqtt
```

- Arrêter le broker
- Démarrer le broker en lui fournissant le fichier de configuration (noter la commande)

- Vérifier que le fichier de configuration a bien été chargé (copie d’écran)

- Est-ce qu’avec cette configuration il est possible de se connecter depuis le PC ?

- Noter vos observations (analyse des messages du broker)

Configuration du broker : authentification anonyme

`allow_anonymous` [true | false]

Boolean value that determines whether clients that connect without providing a username are allowed to connect. If set to *`false`* then another means of connection should be created to control authenticated client access.

- Modifier le fichier de configuration (paramètre `allow_anonymous`) pour autoriser les connexions anonymes (sans user/mot de passe) sur le broker

- Redémarrer le broker et valider le fonctionnement (connexion au broker depuis le PC) (copie d'écran)

Analyse des trames MQTT

Pour cette partie vous aurez besoin de **trois terminaux SSH sur la VM**

- un terminal sur l'activité du broker
- un terminal pour un SUB
- un terminal pour les requêtes PUB

Vous aurez besoin également du logiciel MQTTX sur le PC

L'objectif est de capturer les trames MQTT qui circulent entre le PC et le broker et de les analyser.

- Déconnecter MQTTX du broker
- Envoyer un "SUBSCRIBE" sur la VM sur le topic "ciel"
- Démarrer Wireshark sur votre carte réseau
- Filtrer l'affichage sur "mqtt"
- Reconnecter MQTTX sur le broker
- Repérer la trame reçue lors de la connexion (copie d'écran)
- à quoi correspond cette trame (quel rôle dans l'échange entre le client et le broker)

C'est l'accusé de réception de la connexion que le broker a envoyé au client ;

Compléter le tableau d'analyse de la trame selon le modèle en couche TCP/IP

- Envoyer un SUBSCRIBE sur le topic "ciel" depuis MQTTX
- Repérer la trame reçue lors du subscribe (copie d'écran)
- à quoi correspond cette trame (quel rôle dans l'échange entre le client et le broker)
- Quelle ligne correspond à cette trame sur le monitoring du broker ? (copie d'écran)
- Envoyer un PUBLISH sur le topic "ciel" depuis le broker

- Vérifier que le message arrive sur les deux clients (terminal sur le broker et MQTTX sur PC)
- Repérer la trame reçue lors du publish (copie d'écran)

- Arrêter la capture et analyser les trames sur Wireshark (copie d'écran)
- Indiquer brièvement les différents types de trames (colonne info)

Compléter le tableau d'analyse de la trame du message publié selon le modèle en couche TCP/IP

N° couche	Nom de la couche	Information
5		Msg Length = Message = Topic length = Topic = (les caractères sont codés en ASCII)
4		port source = port destination =
3		adresse IP destinataire =
2		adresse MAC source =
1		nb d'octets = nb de bits =

- Observer comment retrouver le contenu du message et le topic dans la trame
- Utiliser votre observation pour retrouver le contenu du message envoyé dans cette trame (justifier) (sur fond gris les octets de la couche 5)

```
0000  ac 29 3a e9 79 44 08 00  27 d7 79 d1 08 00 45 00
0010  00 4d ab 29 40 00 40 06  0b ab c0 a8 01 34 c0 a8
0020  01 52 b6 5a 07 5b dc 64  cd fb 90 f9 22 3a 80 18
0030  01 f6 8f bc 00 00 01 01  08 0a 52 11 2a 97 30 dc
0040  dc 12 30 17 00 13 62 61  74 41 2f 65 74 61 67 65
0050  32 2f 32 30 33 2f 6c 75  6d 34 39
```

Configuration du broker avec authentification

Présentation et objectif

Le broker Mosquitto peut être configuré pour nécessiter une authentification du client par username / mot de passe.

L'authentification permet de protéger l'accès au broker (connexion), le couple username /mot de passe peut aussi être utilisé pour restreindre l'accès à des topics.
L'objectif de cette partie est de mettre en place une connexion sécurisée par username /mot de passe et d'évaluer la robustesse de cette authentification.

Cahier des charges

- Le broker sera sécurisé en ajoutant une authentification par user /mot de passe
- Les connexions anonymes seront interdites
- Deux utilisateurs seront créés
    - user ciel / mot de passe ciel
    - user robot / mot de passe robot

La configuration se fait en deux étapes

étape 1 : modifier le fichier de configuration

- Faire une copie de sauvegarde du fichier de configuration dans un fichier nommé "mosquitto.conf.basic". Noter la commande

- Ajouter les lignes suivantes dans le fichier de configuration mosquitto.conf

```
allow_anonymous false
password_file /etc/mosquitto/passwords
```

- Ajouter un commentaire court et explicatif pour chaque ligne

étape 2 : créer le fichier des identifiants

Mosquitto fournit l'utilitaire "mosquitto_passwd" pour ajouter ou supprimer des identifiants/mot de passe : cela évite l'édition directe du fichier des mots de passe.

Référence en ligne détaillée : https://mosquitto.org/man/mosquitto_passwd-1.html

syntaxe en mode interactif (sur la console)

`mosquitto_passwd [ -c | -D ] <passwordfile> <username>`	
sans option	Ajoute le <username> au fichier <passwordfile> Le mot de passe sera demandé au prompt

`option -c`	Ajoute le <username> au fichier <passwordfile> EN PLUS crée le fichier <passwordfile> ATTENTION le fichier existant sera écrasé ! (le faire pour le 1er utilisateur)
`option -D`	Supprime le <username> au fichier <passwordfile>

- Créer le fichier de user/mot de passe avec un utilisateur (noter la commande)

- Redémarrer le serveur et valider que le fonctionnement est conforme au cahier des charges (copie d'écran)

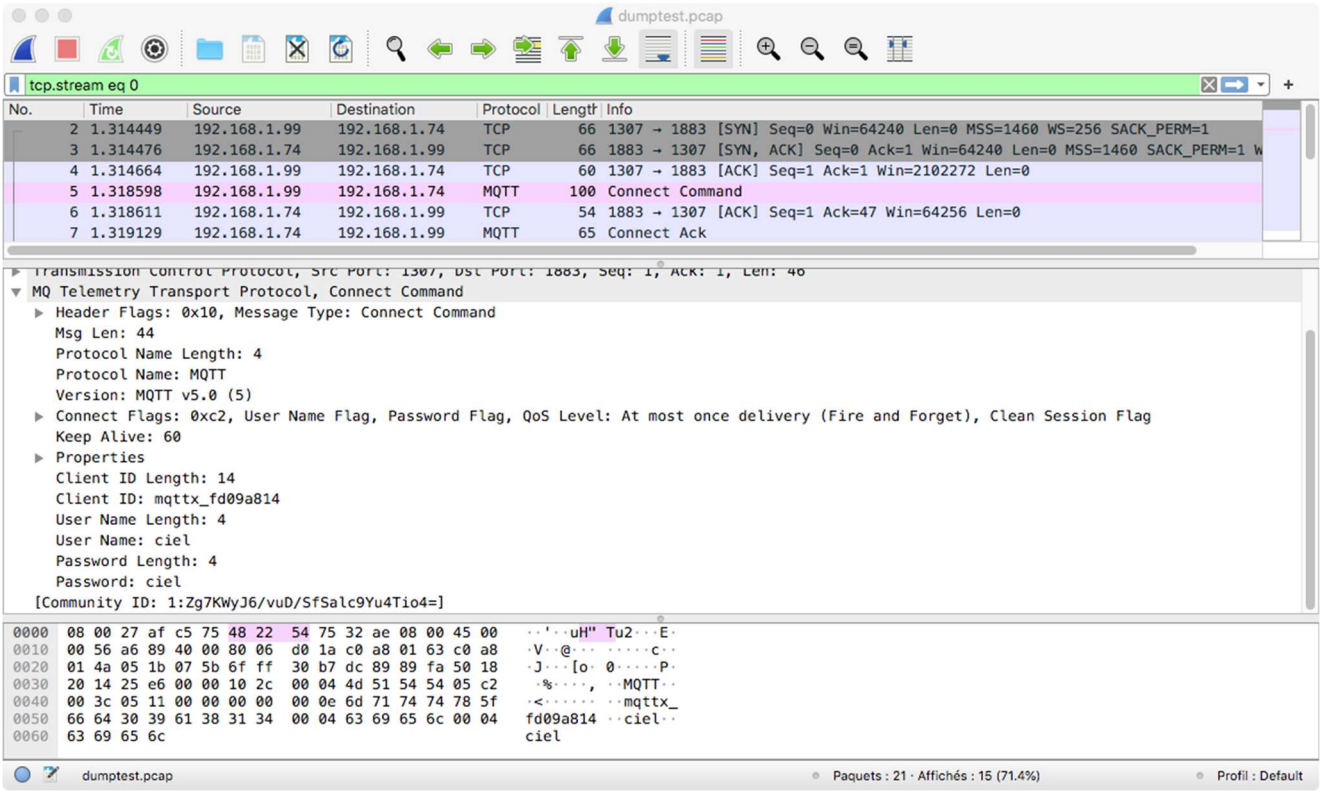
- Ajouter un deuxième utilisateur (noter la commande)

- Redémarrer le serveur et valider que le fonctionnement est conforme au cahier des charges (copie d'écran)

- Ouvrir le fichier des user/mots de passe. Que constatez-vous ?

Niveau de sécurité de l'authentification

Voici un extrait d'enregistrement de conversation par Wireshark



- Que peut-on en déduire sur le niveau de sécurité de cette méthode d'authentification ?

Conclusion

Quels sont les deux niveaux de sécurisation du broker vus dans ce TP et quelles peuvent être leurs utilisations ?

niveau de sécurité	exemple d'utilisation pratique

Bonus

- Quel est peut être l'intérêt de stocker le fichier des user/mot de passe dans /etc/mosquitto plutôt que dans votre home directory ?
- Hacking : Demander à l'enseignante le fichier dump de Wireshark du broker, analyser le fichier et utiliser les informations pour envoyer une information erronée sur le topic (votre nom par exemple)