



UNIVERSITE
DE KARA

11/06/2025

SÉCURISATION D'UN MINI-SYSTÈME IoT LOCAL

Licence professionnelle en sécurité
informatique et cybersécurité

Groupe 1

Groupe 1

TP PRATIQUE – SÉCURISATION D'UN MINI-SYSTÈME IoT LOCAL

Membres du groupe

Date : 11/06/2025

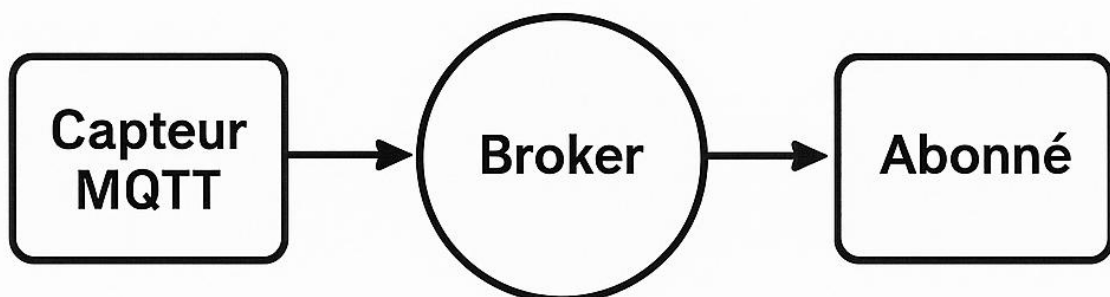
Noms	Prénoms
BINAWAI	Ereke Rachel
KADJADO	M. Pascal
ALEDI	E. Fidèle
KASSA	Luc

1. Schéma logique du système testé

Le système IoT est composé d'un capteur simulé qui envoie des données à un broker Mosquitto, lequel transmet ces données à un client abonné.

Le système mis en place se compose de trois éléments :

- **Capteur simulé** : Il publie des données de température sur un topic MQTT.
- **Broker MQTT (Mosquitto)** : Il reçoit les messages du capteur et les redistribue aux abonnés.
- **Client abonné** : Il s'abonne au topic et reçoit les données.



2. Analyse de la vulnérabilité

Pour analyser le trafic MQTT, nous avons utilisé Wireshark en ciblant l'interface loopback « **lo** ».

Nous avons observé que :

- Les messages publiés sur le topic `capteur/temperature` sont visibles en clair dans les paquets réseau.
- Le champ `Message Payload` affiche les données envoyées (par ex. `temp: 23`).
- Aucun chiffrement n'est appliqué.
- Aucune authentification n'est requise par défaut.

Conclusion

Le protocole MQTT dans sa configuration par défaut présente des failles majeures (pas de confidentialité, pas de contrôle d'accès).

MQTT est un protocole léger conçu pour la rapidité, mais par défaut, il ne chiffre pas les communications ni ne vérifie l'identité des clients. Cela le rend vulnérable :

- Aux interceptions (écoute passive),
- Aux manipulations de messages,
- À l'usurpation d'identité (clients malveillants peuvent se connecter librement).

Mise à jour de la liste des paquets pour préparer l'installation des outils nécessaires.

```
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel# apt-get update  
Réception de :1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]  
Atteint :2 http://tg.archive.ubuntu.com/ubuntu jammy InRelease  
Réception de :3 http://tg.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]  
Ign :4 https://packages.inverse.ca/packetfence/debian stable InRelease  
Réception de :5 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [2 378 kB]
```

Installation du broker Mosquitto et de ses clients pour publier et s'abonner à des messages MQTT.

```

root@rachel:/home/rachel#
root@rachel:/home/rachel#
root@rachel:/home/rachel#
root@rachel:/home/rachel# apt-get install mosquitto mosquitto-clients
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances... Fait
Lecture des informations d'état... Fait
Les paquets supplémentaires suivants seront installés :
  libcjson1 libdlt2 libev4 libmosquitto1 libwebsockets16
Les NOUVEAUX paquets suivants seront installés :
  libcjson1 libdlt2 libev4 libmosquitto1 libwebsockets16 mosquitto mosquitto-clients
0 mis à jour, 7 nouvellement installés, 0 à enlever et 233 non mis à jour.
Il est nécessaire de prendre 648 ko dans les archives.
Après cette opération, 1 967 ko d'espace disque supplémentaires seront utilisés.
Souhaitez-vous continuer ? [O/n] o
Réception de :1 http://tg.archive.ubuntu.com/ubuntu jammy/universe amd64 libcjson1 amd64 1.7.15-1 [15,5 kB]
Réception de :2 http://tg.archive.ubuntu.com/ubuntu jammy/universe amd64 libdlt2 amd64 2.18.6-2 [52,5 kB]
Réception de :3 http://tg.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 libmosquitto1 amd64 2.0.11-1ubuntu1.2 [51,9 kB]
Réception de :4 http://tg.archive.ubuntu.com/ubuntu jammy/universe amd64 libev4 amd64 1:4.33-1 [29,4 kB]
30% [Connexion à tg.archive.ubuntu.com (185.125.190.81)]

```

Redémarrage du service Mosquitto pour appliquer les modifications de configuration

&

Vérification que le service Mosquitto est bien actif et fonctionne correctement.

```

root@rachel:/home/rachel#
root@rachel:/home/rachel# systemctl start mosquitto
root@rachel:/home/rachel#
root@rachel:/home/rachel# systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-06-11 09:35:57 GMT; 2min 58s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Process: 4092 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exited, status=0/SUCCESS)
   Process: 4094 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exited, status=0/SUCCESS)
   Process: 4096 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited, status=0/SUCCESS)
   Process: 4098 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited, status=0/SUCCESS)
  Main PID: 4100 (mosquitto)
    Tasks: 1 (limit: 2269)
   Memory: 1.5M
      CPU: 79ms
   CGroup: /system.slice/mosquitto.service
           └─4100 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

juin 11 09:35:57 rachel systemd[1]: Starting Mosquitto MQTT Broker...
juin 11 09:35:57 rachel systemd[1]: Started Mosquitto MQTT Broker.
root@rachel:/home/rachel#

```

Création d'un script shell

```

oot@rachel:/home/rachel#
oot@rachel:/home/rachel#
oot@rachel:/home/rachel#
oot@rachel:/home/rachel# nano script.sh
oot@rachel:/home/rachel#
oot@rachel:/home/rachel#

```

```
GNU nano 6.2 script.sh *
#!/bin/bash

while true; do
    mosquitto_pub -h localhost -t capteur/temperature -m "temp: $((20 + RANDOM % 5))"
    sleep 2
done
```

Attribution des droits d'exécution au script du capteur.

```
root@rachel:/home/rachel# chmod +x /home/rachel/script.sh
root@rachel:/home/rachel#
root@rachel:/home/rachel#
```

Lancement du script

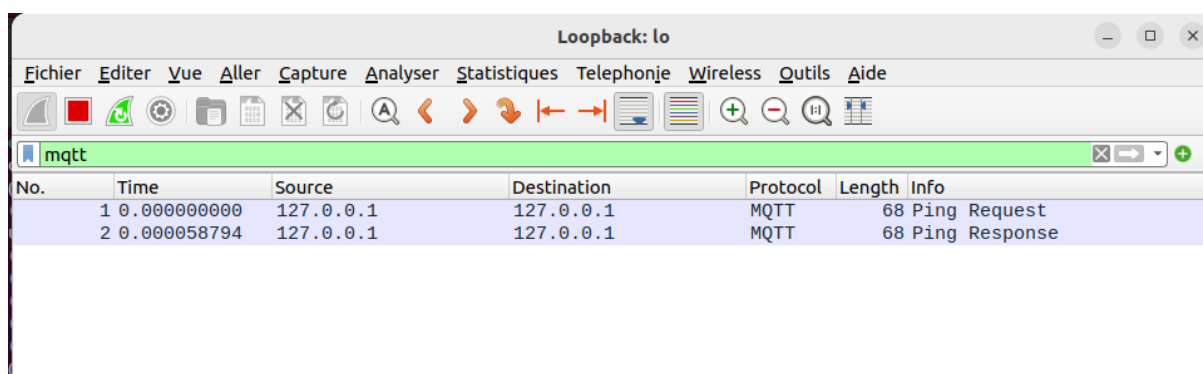
```
root@rachel:/home/rachel# ./script.sh
```

Abonnement au topic pour recevoir les données.

```
root@rachel:/home/rachel#
root@rachel:/home/rachel# mosquitto_sub -h localhost -t capteur/temperature
```

Wireshark (interface lo) : Observation du trafic MQTT en clair.

On travaille en local (localhost), donc on doit sélectionner l'interface lo dans Wireshark



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	MQTT	68	Ping Request
2	0.000058794	127.0.0.1	127.0.0.1	MQTT	68	Ping Response

Loopback: lo

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

mqtt

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	127.0.0.1	127.0.0.1	MQTT	68	Ping Request
2	0.000058794	127.0.0.1	127.0.0.1	MQTT	68	Ping Response
4	60.060279191	127.0.0.1	127.0.0.1	MQTT	68	Ping Request
5	60.060331550	127.0.0.1	127.0.0.1	MQTT	68	Ping Response
7	120.121011181	127.0.0.1	127.0.0.1	MQTT	68	Ping Request
8	120.121077719	127.0.0.1	127.0.0.1	MQTT	68	Ping Response
10	179.180013034	127.0.0.1	127.0.0.1	MQTT	68	Ping Request
11	179.180069325	127.0.0.1	127.0.0.1	MQTT	68	Ping Response

Frame 1: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 46326, Dst Port: 1883, Seq: 1, Ack: 1, Len: 2

MQ Telemetry Transport Protocol, Ping Request

```

0000  00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 36 7b 9f 40 00 40 06 c1 20 7f 00 00 01 7f 00  6{.@.@.
0020  00 01 b4 f6 07 5b 68 ff da 23 3b 5f 3b b8 80 18  ....[h. #;_...
0030  02 00 fe 2a 00 00 01 01 08 0a 14 11 ca 7d 14 10  ...*.....}.
0040  df e0 c0 00  ....

```

*Loopback: lo

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

tcp.stream eq 0

Destination	Protocol	Length	Info
127.0.0.1	MQTT	68	Ping Request
127.0.0.1	MQTT	68	Ping Response
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=3 Ack=3 Win=512 Len=0 TSval=336710269 ...
127.0.0.1	MQTT	68	Ping Request
127.0.0.1	MQTT	68	Ping Response
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=5 Ack=5 Win=512 Len=0 TSval=336770329 ...
127.0.0.1	MQTT	68	Ping Request
127.0.0.1	MQTT	68	Ping Response
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=7 Ack=7 Win=512 Len=0 TSval=336830390 ...

Frame 7: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 46326, Dst Port: 1883, Seq: 5, Ack: 5, Len: 2

MQ Telemetry Transport Protocol, Ping Request

```

0000  00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 36 7b a3 40 00 40 06 c1 1c 7f 00 00 01 7f 00  6{.@.@.
0020  00 01 b4 f6 07 5b 68 ff da 27 3b 5f 3b bc 80 18  ....[h. ';;_...
0030  02 00 fe 2a 00 00 01 01 08 0a 14 13 9f b6 14 12  ...*.....}.
0040  b5 19 c0 00  ....

```

```

root@rachel:/home/rachel#
root@rachel:/home/rachel# mosquitto_sub -h localhost -t capteur/temperature
temp: 20
temp: 20
temp: 21
temp: 22
temp: 24
temp: 23
temp: 20
temp: 20
temp: 21
temp: 24
temp: 21
temp: 23
temp: 23
temp: 23
temp: 24
temp: 24
temp: 20
temp: 23
temp: 23

```

tcp.stream eq 0

Destination	Protocol	Length	Info
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=25 Ack=2288 Win=512 Len=0 TSval=337372...
127.0.0.1	MQTT	97	Publish Message [capteur/temperature]
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=25 Ack=2319 Win=512 Len=0 TSval=337374...
127.0.0.1	MQTT	97	Publish Message [capteur/temperature]
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=25 Ack=2350 Win=512 Len=0 TSval=337376...
127.0.0.1	MQTT	97	Publish Message [capteur/temperature]
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=25 Ack=2381 Win=512 Len=0 TSval=337378...
127.0.0.1	MQTT	97	Publish Message [capteur/temperature]
127.0.0.1	TCP	66	46326 → 1883 [ACK] Seq=25 Ack=2412 Win=512 Len=0 TSval=337380...

Frame 181: 97 bytes on wire (776 bits), 97 bytes captured (776 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

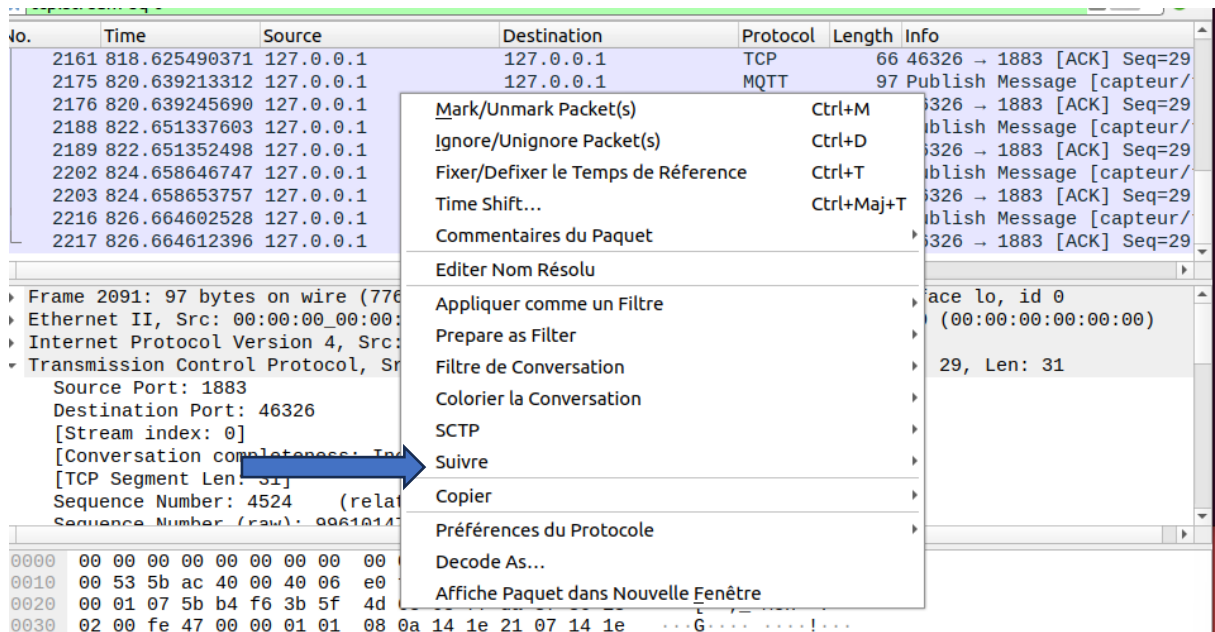
Transmission Control Protocol, Src Port: 1883, Dst Port: 46326, Seq: 329, Ack: 19, Len: 31

Source Port: 1883
Destination Port: 46326
[Stream index: 0]
[Conversation completeness: Incomplete (12)]
[TCP Segment Len: 31]
Sequence Number: 329 (relative sequence number)
Sequence Number (raw): 996097280

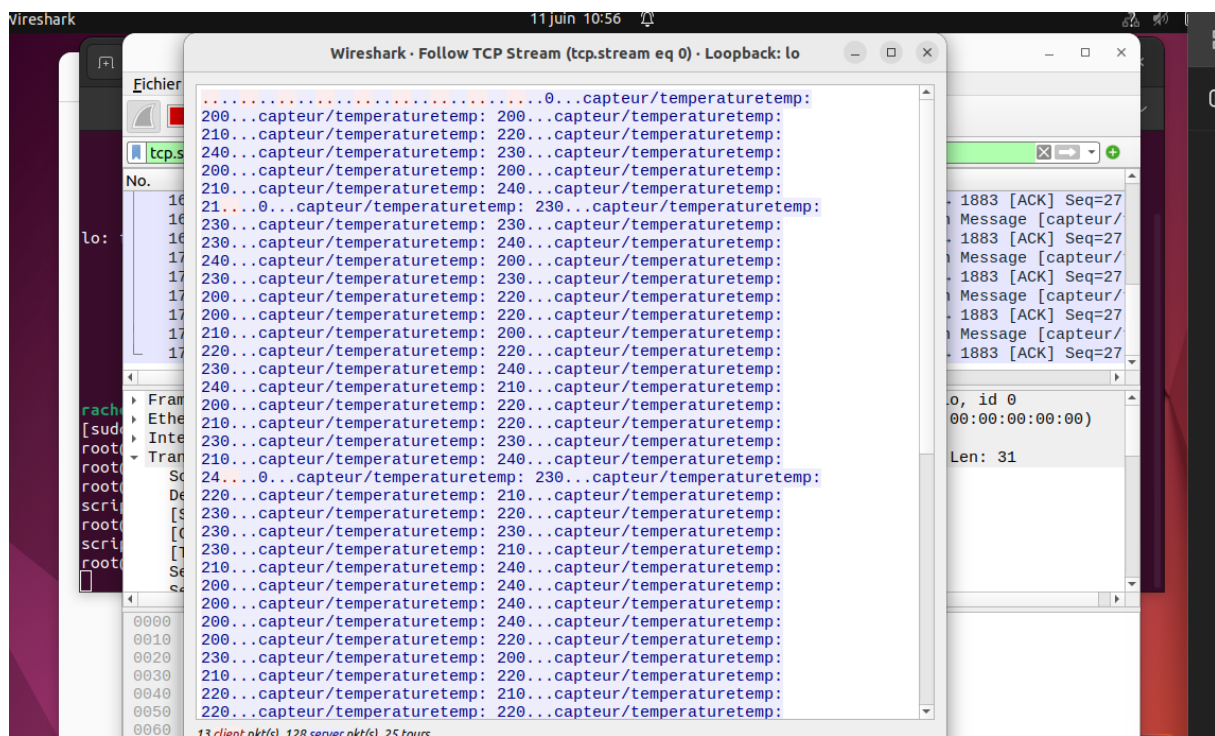
```

0000  00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 53 5b 20 40 00 40 06 e1 82 7f 00 00 01 7f 00  .S[ @. @. ....
0020  00 01 07 5b b4 f6 3b 5f 3d 00 68 ff da 35 80 18  ...[.;_ =.h..5..
0030  02 00 fe 47 00 00 01 01 08 0a 14 19 fe 41 14 19  ...G.....A..
0040  f6 6a 30 1d 00 13 63 61 70 74 65 75 72 2f 74 65  .j0...ca pteur/te
0050  6d 70 65 72 61 74 75 72 65 74 65 6d 70 3a 20 32  mperatur etemp: 2
0060  31 1

```



Message MQTT visible en clair sur le réseau



3. Simulation d'une attaque

Nous avons simulé une attaque de type DoS (Déni de Service) en envoyant 1000 messages falsifiés sur le broker Mosquitto avec la commande :

Effets observés :

- Le terminal du client abonné est saturé par un flux continu de messages inutiles.
- Le broker accepte tous les messages sans contrôle ni filtrage (Impossible de distinguer les vrais messages des faux).
- Il devient quasiment impossible de distinguer les messages légitimes des faux messages.

Conclusion :

Sans mécanismes d'authentification et de filtrage, le broker est vulnérable aux attaques DoS, mettant en péril la disponibilité et la fiabilité du système.

C'est typique d'une **attaque par déni de service (DoS)**. L'objectif est de **surcharger le système** avec des messages inutiles, ce qui bloque le traitement normal. Cela montre que sans filtrage ou authentification, tout acteur malveillant peut perturber le système.

Création du fichier script2.sh

```
root@rachel:/home/rachel#  
root@rachel:/home/rachel# nano script2.sh  
root@rachel:/home/rachel#
```

Commande utilisée :

```
GNU nano 6.2 script2.sh  
#!/bin/bash  
  
for i in {1..1000}; do  
    mosquitto_pub -h localhost -t capteur/temperature -m "false data $i"  
done
```

Attribution des droits d'exécution au script et Lancement du script du capteur

```
root@rachel:/home/rachel#  
root@rachel:/home/rachel# chmod +x /home/rachel/script2.sh  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel# ./script2.sh  
root@rachel:/home/rachel#
```

Saturation du canal MQTT

```
root@rachel:/home/rachel# mosquitto_sub -h localhost -t capteur/temperature
false data 1
false data 2
false data 3
false data 4
false data 5
false data 6
false data 7
false data 8
false data 9
false data 10
false data 11
false data 12
false data 13
false data 14
false data 15
false data 16
false data 17
false data 18
false data 19
false data 20
false data 21
```

```
false data 978
false data 979
false data 980
false data 981
false data 982
false data 983
false data 984
false data 985
false data 986
false data 987
false data 988
false data 989
false data 990
false data 991
false data 992
false data 993
false data 994
false data 995
false data 996
false data 997
false data 998
false data 999
false data 1000
```

*Loopback: lo

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

tcp.stream eq 1

No.	Time	Source	Destination	Protocol	Length	Info
13955	112.640612517	127.0.0.1	127.0.0.1	TCP	66	38144 → 1883 [ACK] Seq=5
13969	112.642604316	127.0.0.1	127.0.0.1	MQTT	103	Publish Message [capteur/
13970	112.642618177	127.0.0.1	127.0.0.1	TCP	66	38144 → 1883 [ACK] Seq=5
13983	112.644002950	127.0.0.1	127.0.0.1	MQTT	103	Publish Message [capteur/
13987	112.644039836	127.0.0.1	127.0.0.1	TCP	66	38144 → 1883 [ACK] Seq=5
13997	112.645232143	127.0.0.1	127.0.0.1	MQTT	103	Publish Message [capteur/
13998	112.645241432	127.0.0.1	127.0.0.1	TCP	66	38144 → 1883 [ACK] Seq=5
14011	112.646515102	127.0.0.1	127.0.0.1	MQTT	103	Publish Message [capteur/
14012	112.646524569	127.0.0.1	127.0.0.1	TCP	66	38144 → 1883 [ACK] Seq=5

Frame 14025: 104 bytes on wire (832 bits), 104 bytes captured (832 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00:00:00:00 (00:00:00:00:00:00), Dst: 00:00:00:00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 1883, Dst Port: 38144, Seq: 36860, Ack: 5, Len: 38

MQ Telemetry Transport Protocol, Publish Message

```

0000  00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 5a 50 e0 40 00 00 06 eb bb 7f 00 00 01 7f 00  .ZP.@.
0020  00 01 07 5b 95 00 cd a8 52 c2 3e 84 c2 dc 80 18  ...[....R>....
0030  02 00 fe 4e 00 00 01 01 08 0a 14 28 f3 c1 14 28  ...N.....(....
0040  f3 c0 30 24 00 13 63 61 70 74 65 75 72 2f 74 65  ..0$.ca pteur/te
0050  6d 70 65 72 61 74 75 72 65 66 61 6c 73 65 20 64  mperatur efalse d
0060  61 74 61 20 31 30 30 30  ata 1000
  
```

Oracle VM VirtualBox

Périphériques Aide

11 juin 11:10

Wireshark - Follow TCP Stream (tcp.stream eq 1) - Loopback: lo

.....0!..capteur/temperaturefalse data 10!..capteur/temperaturefalse data 20!..capteur/temperaturefalse data 30!..capteur/temperaturefalse data 40!..capteur/temperaturefalse data 50!..capteur/temperaturefalse data 60!..capteur/temperaturefalse data 70!..capteur/temperaturefalse data 80!..capteur/temperaturefalse data 90!..capteur/temperaturefalse data 100!..capteur/temperaturefalse data 110!..capteur/temperaturefalse data 120!..capteur/temperaturefalse data 130!..capteur/temperaturefalse data 140!..capteur/temperaturefalse data 150!..capteur/temperaturefalse data 160!..capteur/temperaturefalse data 170!..capteur/temperaturefalse data 180!..capteur/temperaturefalse data 190!..capteur/temperaturefalse data 200!..capteur/temperaturefalse data 210!..capteur/temperaturefalse data 220!..capteur/temperaturefalse data 230!..capteur/temperaturefalse data 240!..capteur/temperaturefalse data 250!..capteur/temperaturefalse data 260!..capteur/temperaturefalse data 270!..capteur/temperaturefalse data 280!..capteur/temperaturefalse data 290!..capteur/temperaturefalse data 300!..capteur/temperaturefalse data 310!..capteur/temperaturefalse data 320!..capteur/temperaturefalse data 330!..capteur/temperaturefalse data 340!..capteur/temperaturefalse data 350!..capteur/temperaturefalse data 360!..capteur/temperaturefalse data 370!..capteur/temperaturefalse data 380!..capteur/temperaturefalse data 390!..capteur/temperaturefalse data 400!..capteur/temperaturefalse data 410!..capteur/temperaturefalse data 420!..capteur/temperaturefalse data 430!..capteur/temperaturefalse data 440!..capteur/temperaturefalse data 450!..capteur/temperaturefalse data 460!..capteur/temperaturefalse data 470!..capteur/temperaturefalse data 480!..capteur/temperaturefalse data 490!..capteur/temperaturefalse data 500!..capteur/temperaturefalse data 510!..capteur/temperaturefalse data 520!..capteur/temperaturefalse data 530!..capteur/temperaturefalse data 540!..capteur/temperaturefalse data 550!..capteur/temperaturefalse data 560!..capteur/temperaturefalse data 570!..capteur/temperaturefalse data 580!..capteur/temperaturefalse data 590!..capteur/temperaturefalse data 600!..capteur/temperaturefalse data 610!..capteur/temperaturefalse data 620!..capteur/temperaturefalse data 630!..capteur/temperaturefalse data 640!..capteur/temperaturefalse data 650!..capteur/temperaturefalse data 660!..capteur/temperaturefalse data 670!..capteur/temperaturefalse data 680!..capteur/temperaturefalse data 690!..capteur/temperaturefalse data

4 client pkt(s), 1 003 server pkt(s), 6 tours.

Entire conversation (36 kB) Show data as ASCII Flux 1

Trouver: Trouver Suivant

Aide Filter Out This Stream Imprimer Sauvegarder sous... Back Fermer

4. Contre-mesures mises en œuvre

Authentification par mot de passe :

Pour empêcher les connexions anonymes, une authentification par utilisateur/mot de passe a été mise en place.

L'authentification empêche tout utilisateur non autorisé de publier ou s'abonner. Seuls les utilisateurs ayant un **mot de passe valide** peuvent interagir avec le broker.

Création d'un utilisateur :

Création d'un fichier de mots de passe et ajout d'un utilisateur

```
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel# mosquitto_passwd -c /etc/mosquitto/passwd rachel  
Password:  
Reenter password:  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#
```

Configuration du fichier `/etc/mosquitto/mosquitto.conf` :

```
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel# nano /etc/mosquitto/mosquitto.conf  
root@rachel:/home/rachel#
```

```
# Place your local configuration in /etc/mosquitto/conf.d/  
#  
# A full description of the configuration file is at  
# /usr/share/doc/mosquitto/examples/mosquitto.conf.example  
  
pid_file /run/mosquitto/mosquitto.pid  
  
persistence true  
persistence_location /var/lib/mosquitto/  
  
log_dest file /var/log/mosquitto/mosquitto.log  
  
include_dir /etc/mosquitto/conf.d  
  
allow_anonymous false  
password_file /etc/mosquitto/passwd
```

Alt-A Aide Alt-E Écrire Alt-W Chercher Alt-K Couper Alt-T Exécuter Alt-C Emplacement Alt-M Appliquer

Redémarrage du broker

```
root@rachel:/home/rachel#
root@rachel:/home/rachel# systemctl restart mosquitto
root@rachel:/home/rachel#
root@rachel:/home/rachel# systemctl status mosquitto
● mosquitto.service - Mosquitto MQTT Broker
   Loaded: loaded (/lib/systemd/system/mosquitto.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2025-06-11 11:19:39 GMT; 14s ago
     Docs: man:mosquitto.conf(5)
           man:mosquitto(8)
   Process: 6675 ExecStartPre=/bin/mkdir -m 740 -p /var/log/mosquitto (code=exited, status=0/SUCCESS)
   Process: 6678 ExecStartPre=/bin/chown mosquitto /var/log/mosquitto (code=exited, status=0/SUCCESS)
   Process: 6679 ExecStartPre=/bin/mkdir -m 740 -p /run/mosquitto (code=exited, status=0/SUCCESS)
   Process: 6680 ExecStartPre=/bin/chown mosquitto /run/mosquitto (code=exited, status=0/SUCCESS)
  Main PID: 6681 (mosquitto)
    Tasks: 1 (limit: 2269)
   Memory: 1.4M
      CPU: 13ms
   CGroup: /system.slice/mosquitto.service
           └─6681 /usr/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf

juin 11 11:19:39 rachel systemd[1]: Starting Mosquitto MQTT Broker...
juin 11 11:19:39 rachel systemd[1]: Started Mosquitto MQTT Broker.
root@rachel:/home/rachel#
```

- **Test :**

Tester la publication avec authentification :

```
root@rachel:/home/rachel#
root@rachel:/home/rachel#
root@rachel:/home/rachel#
root@rachel:/home/rachel# mosquitto_pub -h localhost -t capteur/temperature -m "auth data" -u rachel -P rachel
root@rachel:/home/rachel#
root@rachel:/home/rachel#
```

Tester l'abonnement avec authentification :

```
root@rachel:/home/rachel#
root@rachel:/home/rachel#
root@rachel:/home/rachel#
root@rachel:/home/rachel# mosquitto_sub -h localhost -t capteur/temperature -u rachel -P rachel
auth data

```

5. Mise en place du chiffrement TLS :

Pour garantir la confidentialité des échanges, le chiffrement TLS a été configuré.

TLS chiffre les échanges entre le client et le broker. Même si un attaquant intercepte les paquets, **le contenu restera illisible** sans la clé privée. Cela garantit **confidentialité et intégrité** des messages.

Génération des certificats (CA, serveur) avec OpenSSL.

Déplacement dans `/etc/mosquitto/certs`` et affectation des droits.

```

root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs# openssl genrsa -out ca.key 2048
root@rachel:/etc/mosquitto/certs# openssl req -x509 -new -nodes -key ca.key -sha256 -days 365 -out ca.crt -subj "/CN=MQTT-CA"
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs# openssl genrsa -out server.key 2048
root@rachel:/etc/mosquitto/certs# openssl req -new -key server.key -out server.csr -subj "/CN=localhost"
root@rachel:/etc/mosquitto/certs# openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 365 -sha256
Certificate request self-signature ok
subject=CN = localhost
root@rachel:/etc/mosquitto/certs# sudo chown mosquitto:mosquitto /etc/mosquitto/certs/*
root@rachel:/etc/mosquitto/certs# sudo chmod 600 /etc/mosquitto/certs/server.key
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#

```

Configuration Mosquitto TLS:

```

root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs# nano /etc/mosquitto/mosquitto.conf
root@rachel:/etc/mosquitto/certs#

```

```

allow_anonymous false
password_file /etc/mosquitto/passwd

listener 8883
protocol mqtt

cafile /etc/mosquitto/certs/ca.crt
certfile /etc/mosquitto/certs/server.crt
keyfile /etc/mosquitto/certs/server.key

```

Redémarrer Mosquitto :

```

root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs# sudo systemctl restart mosquitto
root@rachel:/etc/mosquitto/certs#

```

Test sécurisé :

Tester la publication via TLS :

```

root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs# mosquitto_pub --cafile /etc/mosquitto/certs/ca.crt -h localhost -p 8883 -t capteur/temperature -m "test TLS" -u rachel -P rachel
root@rachel:/etc/mosquitto/certs#
root@rachel:/etc/mosquitto/certs#

```

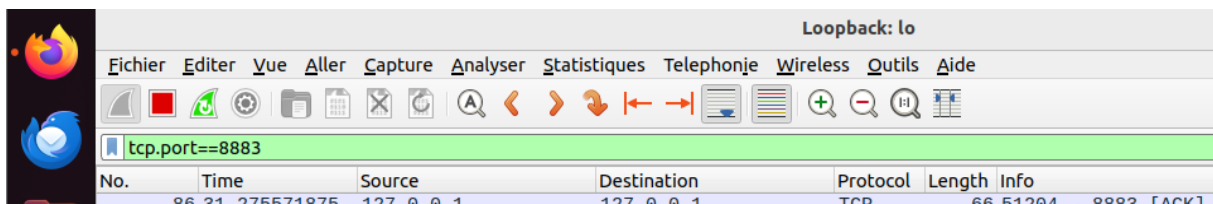
Tester l'abonnement via TLS :

```
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel# mosquitto_sub --cafile /etc/mosquitto/certs/ca.crt -h localhost -p 8883 -t capteur/temperature -u rachel -P rachel  
test TLS
```

Lancement manuel de Mosquitto en mode verbeux pour observer les connexions et messages en temps réel.

```
root@rachel:/home/rachel#  
root@rachel:/home/rachel#  
root@rachel:/home/rachel# mosquitto -v  
1749643461: mosquitto version 2.0.11 starting  
1749643461: Using default config.  
1749643461: Starting in local only mode. Connections will only be possible from clients running on this machine.  
1749643461: Create a configuration file which defines a listener to allow remote access.  
1749643461: For more details see https://mosquitto.org/documentation/authentication-methods/  
1749643461: Opening ipv4 listen socket on port 1883.  
1749643461: Opening ipv6 listen socket on port 1883.  
1749643461: mosquitto version 2.0.11 running  
1749643813: New connection from 127.0.0.1:45084 on port 1883.  
1749643813: New client connected from 127.0.0.1:45084 as auto-EA973992-089F-B36F-72F3-1C4E38CD7B1D (p2, c1, k6, u'rachel').  
1749643813: No will message specified.  
1749643813: Sending CONNACK to auto-EA973992-089F-B36F-72F3-1C4E38CD7B1D (0, 0)  
1749643813: Received SUBSCRIBE from auto-EA973992-089F-B36F-72F3-1C4E38CD7B1D  
1749643813:   capteur/temperature (QoS 0)  
1749643813: auto-EA973992-089F-B36F-72F3-1C4E38CD7B1D 0 capteur/temperature  
1749643813: Sending SUBACK to auto-EA973992-089F-B36F-72F3-1C4E38CD7B1D  
1749643824: Received DISCONNECT from auto-EA973992-089F-B36F-72F3-1C4E38CD7B1D  
1749643824: Client auto-EA973992-089F-B36F-72F3-1C4E38CD7B1D disconnected.
```

Le filtre c'est tcp.port==8883



Loopback: lo

Fichier Editer Vue Aller Capture Analyser Statistiques Telephonie Wireless Outils Aide

tcp.port==8883

No.	Time	Source	Destination	Protocol	Length	Info
86	31.275571875	127.0.0.1	127.0.0.1	TCP	66	51204 → 8883 [ACK]

Loopback: lo

Fichier Éditer Vue Aller Capture Analyser Statistiques Téléphonie Wireless Outils Aide

tcp.port==8883

No.	Time	Source	Destination	Protocol	Length	Info
86	31.275571875	127.0.0.1	127.0.0.1	TCP	66	51204 → 8883 [ACK]
87	31.275584151	127.0.0.1	127.0.0.1	TLSv1.3	92	Application Data
88	31.275596065	127.0.0.1	127.0.0.1	TCP	66	51204 → 8883 [ACK]
89	31.275669321	127.0.0.1	127.0.0.1	TLSv1.3	119	Application Data
90	31.275718994	127.0.0.1	127.0.0.1	TLSv1.3	114	Application Data, A
91	31.276458781	127.0.0.1	127.0.0.1	TLSv1.3	119	Application Data
92	31.276467341	127.0.0.1	127.0.0.1	TCP	66	56282 → 8883 [ACK]
93	31.276506249	127.0.0.1	127.0.0.1	TLSv1.3	90	Application Data
94	31.276515933	127.0.0.1	127.0.0.1	TCP	54	51204 → 8883 [RST]

Frame 1: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 56272, Dst Port: 8883, Seq: 1, Ack: 1, Len: 24

Transport Layer Security

```

0000  00 00 00 00 00 00 00 00 00 00 00 00 08 00 45 00  .....E.
0010  00 4c 0b 69 40 00 40 06 31 41 7f 00 00 01 7f 00  .L.i@.@.1A.....

```

Trafic MQTT chiffré (Wireshark)

Les messages sont chiffrés sur le réseau, protégeant la confidentialité des données.

Wireshark · Follow TCP Stream (tcp.stream eq 4) · Loopback: lo

tcp.stream eq 4

No.	Time	Source	Destination	Protocol	Length	Info
83	31.234921806	127.0.0.1	127.0.0.1	TCP	66	51204 → 8883 [ACK]
84	31.234927427	127.0.0.1	127.0.0.1	TLSv1.3	92	Application Data
85	31.234983379	127.0.0.1	127.0.0.1	TCP	66	51204 → 8883 [ACK]
86	31.275571875	127.0.0.1	127.0.0.1	TLSv1.3	119	Application Data
87	31.275584151	127.0.0.1	127.0.0.1	TLSv1.3	92	Application Data
88	31.275596065	127.0.0.1	127.0.0.1	TCP	66	51204 → 8883 [ACK]
89	31.275669321	127.0.0.1	127.0.0.1	TLSv1.3	119	Application Data
90	31.275718994	127.0.0.1	127.0.0.1	TLSv1.3	114	Application Data, A
91	31.276458781	127.0.0.1	127.0.0.1	TLSv1.3	119	Application Data
92	31.276467341	127.0.0.1	127.0.0.1	TCP	66	56282 → 8883 [ACK]
93	31.276506249	127.0.0.1	127.0.0.1	TLSv1.3	90	Application Data
94	31.276515933	127.0.0.1	127.0.0.1	TCP	54	51204 → 8883 [RST]

Frame 94: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface lo, id 0

Ethernet II, Src: 00:00:00_00:00:00 (00:00:00:00:00:00), Dst: 00:00:00_00:00:00 (00:00:00:00:00:00)

Internet Protocol Version 4, Src: 127.0.0.1, Dst: 127.0.0.1

Transmission Control Protocol, Src Port: 56272, Dst Port: 8883, Seq: 1, Ack: 1, Len: 24

Transport Layer Security

```

...2...P..U.....^..Ky..|...Y...;k...V...l{m.9.@...#...p...S.G.%...
[.gd...e.>.....,0.....+.../...$.(.k.#.'..g.
...9...3.....=<.5./.....localhost.....
.....#.....
%*(.....
.....+.....~.....3.&$....<+OAR....
1.).....i.....<..I..Y...z...v...h)W....."+.2%.y..ugX.x.m...l{m.
9.@...#...p...S.G.%...[.gd...e.....+.....3.$...]......;b.Z.3.1..La}-...
%
X[.....>1>.5a.6h|...h|.....nF...b.i.....>.VF."N..
8..4.....U...T
..V.....[.b.UV.....g,...)....I..D*Xz...G...Q.....M...8.....J....I
..s.9...
.E.t...!..N.U...1...L.....Nq.J.M..J..4o..i...3...hX...AX...ba.
1PY.....p..0.....IH+.....|..V..9...SX[.Y.
x.....0..V...U.M.,\...1.....>T...=m...7M.
.Q...lc
...pY.F..hg...Fo..0.vo..."7T...z.....*7.m...9Z.j...<...
8(7A...0...^...{.i...B...u...JM...w...t...x$.}.yU.V...ldme..
9..a...Qt.....HE8...1...C..."J1...P...5\...z.....
9k..b.D.>..>l.....*..>...!..*..<B.I.7.)...@...#...1...{...[.Vd.n..
8...^..N.$...a.)-sw.L.P...zq=6EF"...Q...@...{<.(Ny0.o...F.Y.7..!
R.E...nE...y3...h.<A.eu.#..6.]~...0.....oc(.J..8T..Ut.....
%..m...T..nY...I..p..n.Ni..g$.A[...B...a.M..54.N....
l...T...i.r;..2:2.&0|.eF..&4JC-...
8...'.*..i.r;..2:2.&0|.eF..&4JC-...
=.0.....0m.G...0..#...
.[...aM>...5.=U.U+
\..m?..
.L.D..
&..6...u...R...!.....%c...3M...Tw..E.....@...x.Ae.....
.a...=.#...x...*.....1.q...t:p.F2...l...6&(PQK.y...
3Q.ac...A..^V..Qso..U
...9...b.A...@...R...C-...%r?.x...$.4MVF..

```

5 client pkt(s), 5 server pkt(s), 7 tours.

Entire conversation (3 131 bytes) Show data as ASCII Flux 4

Trouver: Trouver Suivant

Aide Filter Out This Stream Imprimer Sauvegarder sous... Back Fermer

6. Résultats et conclusion

Résultat :

Grâce à la mise en place de l'authentification et du chiffrement TLS :

- Les connexions anonymes sont refusées.
- Les messages sont chiffrés sur le réseau, protégeant la confidentialité des données.
- Le système est ainsi sécurisé contre les attaques simples comme la publication anonyme ou l'interception passive.
- Le broker refuse toute tentative non authentifiée.

Ce que j'ai appris :

- MQTT est un protocole vulnérable si laissé sans sécurité.
- Des outils simples (Wireshark, scripts bash) suffisent à démontrer les failles.
- Avec peu de configuration, on peut renforcer grandement la sécurité.

Améliorations futures possibles :

- Ajout de règles ACL pour limiter l'accès à certains topics
- Intégration dans un environnement multi-machines
- Activer des **logs et alertes** pour surveiller les activités suspectes.
- Déployer le système sur plusieurs machines pour tester en environnement distribué.
- Mettre en place une interface Web de supervision.