Dokumentation - Mosquitto Authentifizierungsmethode

Inhaltsverzeichnis

[1. Einführung 2](#_Toc187422966)

[1.1 Vorraussetzungen 2](#_Toc187422967)

[1.2 Zielsetzung 2](#_Toc187422968)

[2 Passwortdatei 2](#_Toc187422969)

[2.1 Beschreibung 2](#_Toc187422970)

[3 Mosquitto-Passwd Utility 3](#_Toc187422971)

[3.1 Beschreibung 3](#_Toc187422972)

[4 Ablaufbeschreibung MQTT-Umgebung absichern 4](#_Toc187422973)

[4.1 User-Accounts Konfigurationsdatei erstellen 4](#_Toc187422974)

[4.2 User-Accounts Konfigurationsdatei bearbeiten 4](#_Toc187422975)

[4.3 Passwortdatei mit Benutzer erstellen 4](#_Toc187422976)

[4.4 Haupt-Konfigurationsdatei überprüfen 5](#_Toc187422977)

[4.5 Mosquitto MQTT-Broker neustarten 5](#_Toc187422978)

[5 Testen 5](#_Toc187422979)

[6 Fazit 6](#_Toc187422980)

# Einführung

## Voraussetzungen

Für eine erfolgreiche Umsetzung des Projekts benötigt man einen Raspberry Pi bzw. eine Linux-Umgebung, eine installierte MQTT-Umgebung, in diesem Fall Mosquitto MQTT-Broker, sowie Grundkenntnisse in der Terminal-Nutzung.

## Zielsetzung

Implementierung einer Authorisierungsmethode zur Absicherung einer MQTT-Umgebung mit zusätzlicher Dokumentation und einem erfolgreichen Testlauf.

# Passwortdatei

## Beschreibung

Die Passwortdatei bei Mosquitto erfüllt den Zweck, die Authentifierzung von Benutzern für den Zugriff auf den MQTT-Broker zu verwalten. Die Datei beinhaltet die Benutzernamen und das zugehörige Passwort, das beim Verbindungsaufbau mit dem Mosquitto-Broker abgefragt werden. Dies vergewissert, dass nur autorisierte Clients und Geräte Zugriff auf den MQTT-Broker erhalten.

Die Passwortdatei ist sehr simpel aufgebaut und beinhaltet eine Liste aller Benutzer mit dem zugehörigen Passwort.



Abb. 1. – Mosquitto Passwortdatei

# Mosquitto-Passwd Utility

## Beschreibung

Das Utility mosquitto\_passwd dient zur Verwaltung der Passwortdatei des Mosquitto MQTT-Brokers. Folgende Optionen stehen zur Verfügung:

<passwd-file>: Die zu modifizierende Passwortdatei

<username>: Benutzername, welcher hinzugefügt/aktualisert/gelöscht werden soll

<password>: Das im Batch-Modus zu vergebende Passwort.

**Batch-Modus:**

Befehl: mosquitto\_passwd -b <passwd-file> <username> <password>

Mit dem Zusatz -b startet der Batch-Modus, welcher die gleichzeitige Vergabe eines Benutzernamen und Passworts in der Kommandozeile ermöglicht. Dieser Befehl sollte jedoch bedacht verwendet werden, da das Passwort bei der Eingabe und in der Historie sichtbar ist.

**Erstellen:**

Befehl: mosquitto\_passwd -c <passwd-file> <username>

Mit dem Zusatz -c wird eine neue Passwortdateie erstellt. Sollte die Datei bereits existieren, wird diese überschrieben

**Löschen:**

Befehl: mosquitto\_passwd -D <passwd-file> <username>

Mit dem Zusatz -D wird ein Nutzer aus der Passwortdatei gelöscht.

**Verschlüsselungsart (Hash):**

Befehl: mosquitto\_passwd -H <hash-method>

Mit dem Zusatz -H kann die Verschlüsselungsart bzw. Hash gewählt werden. Folgende Hash-Methoden stehen zur Verfügung:

* sha512 (empfohlen)
* sha256
* bcrypt
* md5

**Verschlüsseln:**

Befehl: mosquitto\_passwd -U <passwd-file>

Mit dem Zusatz -U konvertiert man eine Passwotdatei mit normalen Passwörtern aus Zeichen in verschlüsselte Passwörter. Wurde die Passwortdatei bereits verschlüsselt, werden neue Hashes generiert.

# Ablaufbeschreibung MQTT-Umgebung absichern

## User-Accounts Konfigurationsdatei erstellen

Im ersten Schritt wird in dem Verzeichnis /etc/mosquitto/conf.d eine neue Konfigurationsdatei namens „010-listener-with-users.conf“ angelegt. In dieser Datei werden alle Einstellungen für die Authentifizerungsmethode gespeichert.



Abb. 2 - User-Accounts Konfigurationsdatei erstellen

## User-Accounts Konfigurationsdatei bearbeiten

Nach erfolgreicher Erstellung der Konfigurationsdatei kann diese nun erweitert werden. Dafür muss die Datei mit einem belibiegen Texteditor geöffnet werden.



Abb. 3 - User-Accounts Konfigurationsdatei öffen

Nun können folgende Einstellungen hinzugefügt werden:

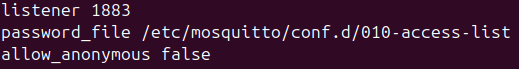


Abb. 4 – User-Accounts Konfigurationsdatei Einstellungen

* listener 1883, gibt den Port an auf dem MQTT laufen soll
* password\_file, gibt den Dateipfad der Passwortdatei an
* allow\_anonymous, erzwingt die Verwendung eines Benutzers (optional)

## Passwortdatei mit Benutzer(n) erstellen

Nun kann die Passwortdatei mit den Benutzern und Passwörtern erstellt werden. Dafür verwenden wir das Utility mosquitto\_passwd.

Im folgenden Schritt wird ein Benutzer namens mosquitto mit dem Passwort mosquitto erstellt. Die Passwortvergabe erfolgt nach der Eingabe des Befehls.



Abb. 6 – Passwortdatei mit Benutzer erstellen

Anschließend müssen die Passwörter in der Passwortdatei noch verschlüsselt werden.



Abb. 7- Passwörter verschlüsseln

Die Passwörter der Benutzer sollten nun verschlüsselt sein.



Abb. 8 – Verschlüsselte Passwörter

## Haupt-Konfigurationsdatei überprüfen

Nach Erstellen der Passwortdatei, sollte noch überprüft werden, ob sich redundante Einstellungen in der Haupt-Konfigurationsdatei befinden. Wenn ja, müssen diese entfernt werden, ansonten kann der Mosquitto MQTT-Broker nicht starten.

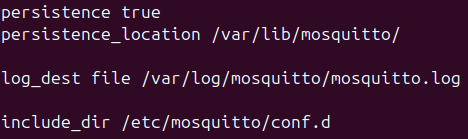


Abb. 9- Haupt-Konfigurationsdatei Muster

## Mosquitto MQTT-Broker neustarten

Nun muss Mosquitto MQTT-Broker neugestartet werden, um die Einstellungen zu übernehmen.



Abb. 10 – Mosquitto MQTT-Broker neustarten

# Testen

Nun kann getestet werden, ob die Authentifizierungsmethode erfolgreich implementiert wurde.

Zunächst wird mosquitto im Debug Modus gestartet. Dies geschiet mit der Option -v (verbose).



Abb. 11- Mosquitto Debug-Modus

Als Nächstes wird versucht, einen mosquitto\_pub Befehl aufzurufen, ohne einen Benutzer und Passwort anzugeben. Dies sollte jetzt nicht mehr funktionieren.



Abb. 12 – Fehlermeldung bei unauthorisiertem Zugriff

Wenn man nun in einem Topic veröffentlichen oder eins subskribieren möchte, muss ein Benutzer mit Passwort wiefolgt spezifiziert werden.



Abb. 13 – Versenden einer Nachricht innerhalb eines Topics

Abb. 14 – Subskribieren eines Topics

Die Tests waren erfolgreich. Die Authentifizierungsmethode ist korrekt implementiert.

# Fazit

Innerhalb weniger Minuten lässt sich eine simple und doch effektive Authentifizierungsmethode implementieren, was die Sicherheit erheblich erhöht. Infolgedessen sollte die Verwendung einer Passwortauthentifizierung als grundlegender Sicherheitsmechanismus in jede Mosquitto-Installation implementiert werden, insbesondere in Umgebungen mit mehreren Benutzern oder öffentlich züganglichen MQTT-Brokern.