

プライベートCA Gléas ホワイトペーパー

MQ Telemetry Transport (MQTT) での

クライアント証明書認証

Ver.1.0 2015 年 11 月

Copyright by JCCH Security Solution Systems Co., Ltd., All Rights reserved

- JCCH・セキュリティ・ソリューション・システムズ、JS3 およびそれらを含むロゴは日本および他の国における株式会社 JCCH・セキュリティ・ソリューション・システムズの商標または登録商標です。Gléas は株式会社 JCCH・セキュリティ・ソリューション・システムズの商標です。
- その他本文中に記載されている製品名および社名は、それぞれ各社の商標または登録商標です。
- Microsoft Corporation のガイドラインに従って画面写真を掲載しています。

目次

1. はじる	めに	4
1.1.	本書について	4
1.2.	本書における環境	4
1.3.	【ご参考】インストール手順	5
2. ブロ・	ーカーの設定	5
2.1.	各種証明書・失効リストの準備	5
2.2.	設定ファイルの編集	6
2.3.	ブローカーの起動	7
3. Gléa	s からの証明書の取得	7
4. 機器	間でのメッセージ送受信	9
4.1.	サブスクライバーの接続	9
4.2.	パブリッシャーからのメッセージ送信	9
5. 問い	合わせ	9

1. はじめに

1.1. 本書について

本書では、弊社製品 プライベートCA Gléas で発行したクライアント証明書を利 用して、MQ Telemetry Transport (MQTT)で接続する際にクライアント証明書 による認証をおこなう環境を構築するための設定例を記載します。

本書に記載の内容は、弊社の検証環境における動作を確認したものであり、あら ゆる環境での動作を保証するものではありません。弊社製品を用いたシステム構 築の一例としてご活用いただけますようお願いいたします。

弊社では試験用のクライアント証明書の提供も行っております。検証等で必要な 場合は、最終項のお問い合わせ先までお気軽にご連絡ください。

1.2. 本書における環境

本書は、以下の環境で検証をおこなっております。

- MQTTブローカー: Ubuntu 14.04 / Mosquitto 1.4.4
 以後、「ブローカー」と記載します
- JS3 プライベートCA Gléas (バージョン1.12.96)
 以後、「Gléas」と記載します
- ▶ MQTTパブリッシャー: Windows10 Pro / Mosquitto 1.4.4 (mosquitto_pub) 以後、「パブリッシャー」と記載します
- ▶ MQTTサブスクライバー: Windows8.1 Pro / Mosquitto 1.4.4 (mosquitto_sub) 以後、「サブスクライバー」と記載します

以下については、本書では説明を割愛します。

- Mosquittoの基本的な設定や操作
- Gléasでのユーザ登録やクライアント証明書発行などの基本操作
- 各サーバ・クライアント端末におけるネットワーク設定など

これらについては、各製品のマニュアルをご参照いただくか、各製品を取り扱っている販売店にお問い合わせください。

1.3. 【ご参考】インストール手順

本書での環境におけるMosquitto(ブローカー)のインストール手順を記載します。 tar.gzパッケージはホームディレクトリに事前にダウンロードされているものと し、また以下手順に含まれるコマンドもインストールされているものとします。

```
■インストール
```

- \$ sudo apt-get install gcc make g++ libssl-dev libc-ares-dev libc-ares2
 uuid-dev
- \$ cd ~
- \$ tar zxvf mosquitto-1.4.4.tar.gz
- \$ cd mosquitto-1.4.4
- \$ make
- \$ sudo make install

■デーモン化

\$ sudo vi /etc/init/mosquitto.conf

```
start on net-device-up
respawn
exec /usr/local/sbin/mosquitto -c /etc/mosquitto/mosquitto.conf
```

\$ sudo initctl reload-configuration

2. ブローカーの設定

2.1. 各種証明書・失効リストの準備

次のファイルをブローカーにアップロードします。

ファイル名、アップロード先ディレクトリ共にサンプルとなります。本書では以下 の名前であることを前提に記載します。

	ファイル名	アップロード先ディレクトリ名
CA 証明書	ia1.pem	/etc/mosquitto/ca_certificates/
サーバ証明書	cert.pem	/etc/mosquitto/certs/
サーバ秘密鍵	key.pem	/etc/mosquitto/certs/
失効リスト (CRL)	crl_ia1.pem	/etc/mosquitto/crls/

【CA 証明書】

Gléas では次の URL から取得します。 http://{Gléas のホスト名 or IP アドレス}/crl/ia1.pem ダウンロード後、上記ディレクトリに配置します。

【サーバ証明書・秘密鍵】

Gléas からダウンロードしたサーバ証明書は PKCS#12 という形式になっているため、PEM 形式に変換・分離する必要があります。 ※ここでは Gléas より取得したファイルを broaker.p12 というファイル名と仮定します

1) PKCS#12 ファイルより証明書を取得

\$ openssl pkcs12 -in broaker.p12 -clcerts -nokeys -out cert.pem

2) PKCS#12 ファイルより秘密鍵を取得

\$ openss1 pkcs12 -in broaker.p12 -nocerts -nodes -out key.pem

取得したファイルを上記ディレクトリに配置します。 また、秘密鍵は mosquitto の実行ユーザをオーナーにし、パーミッションを 400 に 変更します。

【失効リスト (CRL)】 Gléas では CRL ファイル (PEM 形式) は次の URL から取得できます。 http://{Gléas ホスト名 or IP アドレス}/crl/crl_ia1.pem 取得したファイルを上記ディレクトリに配置します。

本書の設定では、Gléas で証明書を失効してもブローカー上の CRL が自動的に更新 されることはありません。新しい CRL が発行された後には既存の CRL と置き換え る必要があります。

Gléas では CRL 更新用の ruby スクリプト(copycrl.rb)を準備しております。 corycrl.rb では、以下のとおりにすることで Gléas より CRL を取得・置換すること が可能です。

\$./copycrl.rb [CRL 取得 URL] /etc/mosquitto/crls/crl_ia1.pem

また CRL 更新後にはサービス再起動が必要になります。 \$ sudo service mosquitto restart

2.2. 設定ファイルの編集

設定ファイル(本書環境では以下のファイル)に記述を追加します。

/etc/mosquitto/mosquitto.conf

cafile /etc/mosquitto/ca_certificates/ial.crt			
certfile /etc/mosquitto/certs/servercert.gleas.example.crt			
keyfile /etc/mosquitto/certs/servercert.gleas.example.key			
crlfile /etc/mosquitto/certs/crl_ia1.pem			
require_certificate true			
use_identity_as_username true			

証明書に関係する設定は以下の通りです。

※設定ファイルの詳細な説明はここではおこないません。man などをご参照ください

- cafile には、ルート証明書のファイルパスを設定します
- certfile には、サーバ証明書のファイルパスを指定します
- keyfile には、サーバ秘密鍵のファイルパスを指定します
- crlfileには、失効リストのファイルパスを指定します
- require_certificateには、trueを指定しクライアント証明書認証を有効にします
- use_identity_as_userをtrueにすると、クライアント証明書のサブジェクトCN をログインユーザ名として扱うようになります

2.3. ブローカーの起動

Mosquitto を起動します。 \$ sudo service mosquitto start ※既に起動している場合は、restartを指定します

以上でブローカーの設定は終了です。

3. Gléas からの証明書の取得

Gléas の RA から発行済み証明書をダウンロードします。 RA に管理者ログインし、大メニュー[証明書]よりサブスクライバー用の証明書を検 索します。

 ●作業名: <u>デモ</u> ●管理者: <u>サンプル管</u> 	理者	プライベートCA	Gléås 🖪
[証明書]>証明書	一覧>[フィルタ]>有効>[検索]>subscriber	▶ 認証局 ▶ ログ ▶ 管理者 ▶ ヘルブ ▶ ログアウト	 ● サイドバー
アカウント	証明書(subscriberの検索結果 全 1件中 1-1件目)	<u>全て開く ■全て閉じる ■全てドック</u>	▶ クイックナビ
Account	シリアル番号順 🗸 ソート フィルタ 有効のみ	▼ ■ アドバンス検索 Q サブジェクトで検索	<u> ユーザ証明書</u>
Group	JCCH-SSS demo CA#11174 有効な証明:	書 rsa 2048 sha256 あど1095日 ●開閉 ■ドック ヘ	
★ 証明書 Certificate	subscriber01		■ <u>5 一八証明書</u> ★ 認証局証明書
SEEデバイス			
- テンプレート			
Template			保存

証明書詳細画面より[ダウンロード]リンクをクリックし、証明書ファイル(*.p12) をダウンロードします。

●作業名: <u>デモ</u> ●管理者: <u>サンブル管理者</u>	プライベートCA Gléäs 🖪
[証明書]>詳細	□ 認証局 □ ログ □ 管理者 □ ヘルブ □ ログアウト ●
アカウント 証明書 (subscriber の検索結果 全)	1件中 1件目)
Account	#11174 DE-22 <u>8 ユーザ証明書</u>
♀ Group ★ 証明書情報······	□ <u>▶-ウンへのインボ-ト</u> □ <u>ダウンロ-ド</u> □ <u>タイムライン</u> ▲ <u>■ ユンスのか</u>
Certificate Subscriber01	開始日:2015/11/08 10:21 終了日:2018/11/08 10:21

●作業名: <u>デモ</u>	証明書を保護するためのバスワードを入力してください。,	CIÁNS RA
 ● 管理者: サンゴル管理法 【証明書】>詳細 2 アカウント Account ※ グループ ※ グループ ※ グループ ※ 近明書 	-パスワードの入力 パスワード ダウンロード キャンセル 	フライベートCA GIERS (A) ロー型に直弦 > 24 / 20 / 20 / 20 / 20 / 20 / 20 / 20 /
Certificate subscriber01	開始日:2015/11/08 10:21 終了 subscriber01.p12 を開く	3:2018/11/08 10:21
	次のファイルを開こさとしています:	

1.2 項でサーバ証明書ファイルを証明書と秘密鍵に分離したのと同様にクライアント証明書も分離し、サブスクライバーに配置します。

パブリッシャーにも同じ手順で、証明書を配置します。

Gléasでは、管理者用コマンドラインインターフェースから makeiso コマンドを使 って、クライアント証明書を一括ダウンロードすることも可能です。詳細は最終章 の問い合わせ先までお問い合わせください。 プライベート CA Gléas ホワイトペーパー MQTT でのクライアント証明書認証

4. 機器間でのメッセージ送受信

4.1. サブスクライバーの接続

クライアント証明書を指定してブローカーに接続します。 ※ここではトピックを topic/test01 とします > mosquitto_sub -h [ブローカーのホスト名或いは IP アドレス] -t "topic/test01" -cafile [ルート証明書ファイル] --cert [クライアント証明書ファイル] --key [クライアン ト秘密鍵ファイル]

接続に成功すると、待ち受け状態になります。

4.2. パブリッシャーからのメッセージ送信

クライアント証明書を指定してブローカーに接続しメッセージを送信します。
> mosquitto_pub -h [ブローカーのホスト名或いは IP アドレス] -t "topic/test01" -m
"送信メッセージ" --cafile [ルート証明書ファイル] --cert [クライアント証明書ファイル] -key [クライアント秘密鍵ファイル]

送信成功すると、サブスクライバーにメッセージが表示されます。

5. 問い合わせ

■Gléasに関するお問い合わせ先 株式会社JCCH・セキュリティ・ソリューション・システムズ Tel: 050-3821-2195 Mail: sales@jcch-sss.com