Segurança em MQTT

André Luiz Almeida Cardoso
Orientado por: Dr. Francisco José da Silva e Silva

Laboratório de Sistemas Distribuídos Inteligentes (LSDi)
Universidade Federal do Maranhão (UFMA)
http://www.lsdi.ufma.br

Setembro de 2019

Sumário Normal


Sumário


MQTT - Message Queuing Telemetry Transport [ibm]

- Histórico
  - O MQTT foi criado pela IBM no fim da década de 1990;
  - Finalidade de vincular sensores em pipelines de petróleo a satélites;
  - Assíncrono;
  - Utiliza o modelo Publish/Subscribe;

- Por que usar MQTT em IoT?
  - Leve;
  - Flexivel;
  - Permite uso de dispositivos com capacidade de memoria e processamento limitados;
  - Escalável;

MQTT - Funcionamento

- CONNECT message
- PUB: /sensor/data
- SUB: /sensor/data

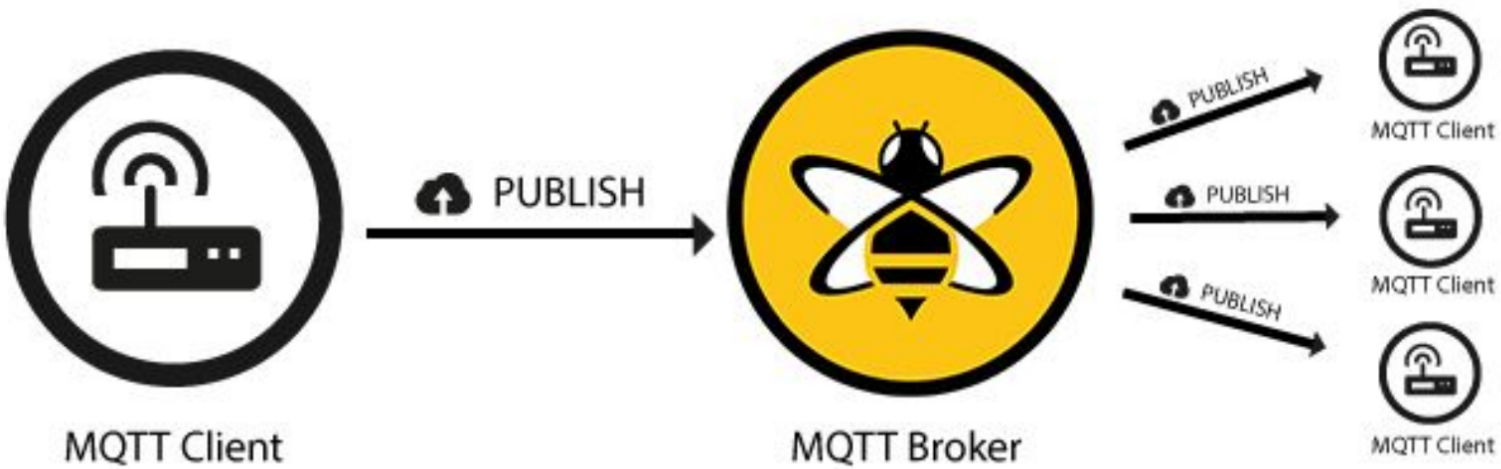


Figura: MQTT example. [hiveMQ]

Segurança - Conceitos Iniciais

- Confiabilidade:
  - Confidencialidade;
  - Integridade;

- Criptografia;
- Autenticação;
- Autorização;
- Auditoria;

Segurança - Criptografia Simetrica [tanembaum]

- Sistemas de chaves compartilhadas;
- Mesma chave é usada para cifrar e decifrar;
- Chave deve ser mantida em segredo;
- Limitações: escalabilidade;
- Algoritmos conhecidos: DES, DES triplo;

Segurança - Criptografia Assimétrica [tanembaum]

- Sistema de chaves publicas;
- Chaves diferentes para cifrar e decifrar;
- Chave publica para cifrar;
- Chave privada para decifrar;

Segurança - Assinatura digital [tanembaum]

- Forma de garantir a integridade e não repudio;
- Usa-se funções hash (MD5,SHA,etc);

Segurança - Certificado digital [tanembaum]

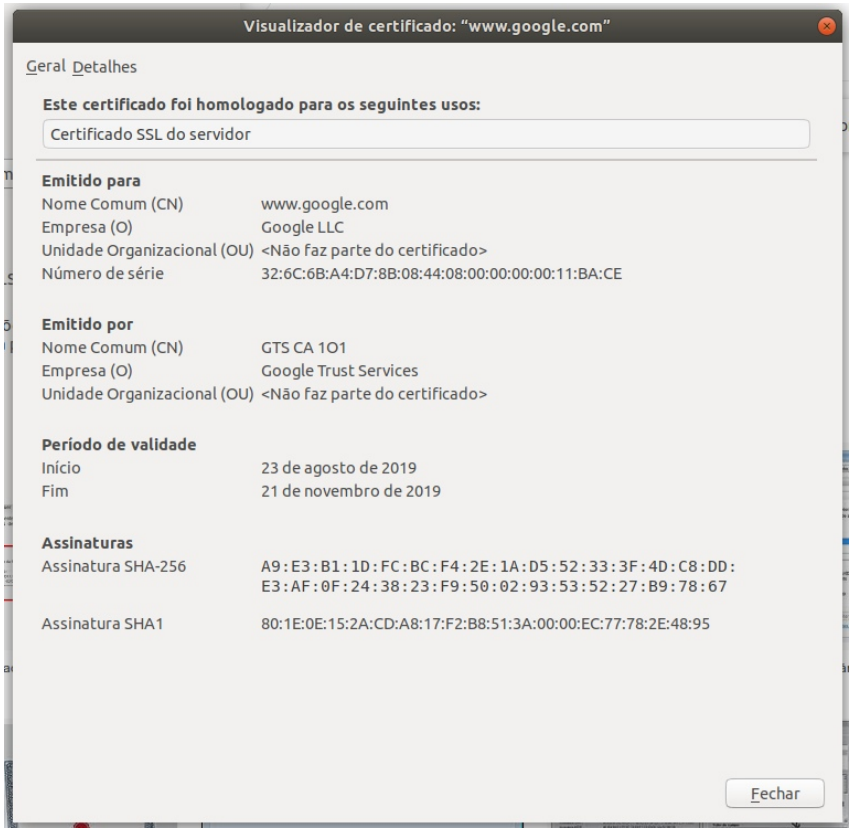


Figura: Certificado Digital: google.com

Segurança - Autoridade Certificadora

Figura: Fonte: `https://bit.ly/2kdvRkW`

Sumário

Segurança no MQTT - Nível de Rede

- Utilizar uma rede fisicamente segura:
  - Acesso fisico ao servidor;
  - Controle das chaves;
  - Utilizar racks com cadeados;
  - Trancar a sala do computador;
- VPN para comunicação entre clientes e brokers:
  - Adequado para aplicações gateway;

Segurança no MQTT - Nível de Transporte

- Confidencialidade;
- Integridade;
- Autenticação;
- Autorização;
- Auditoria;

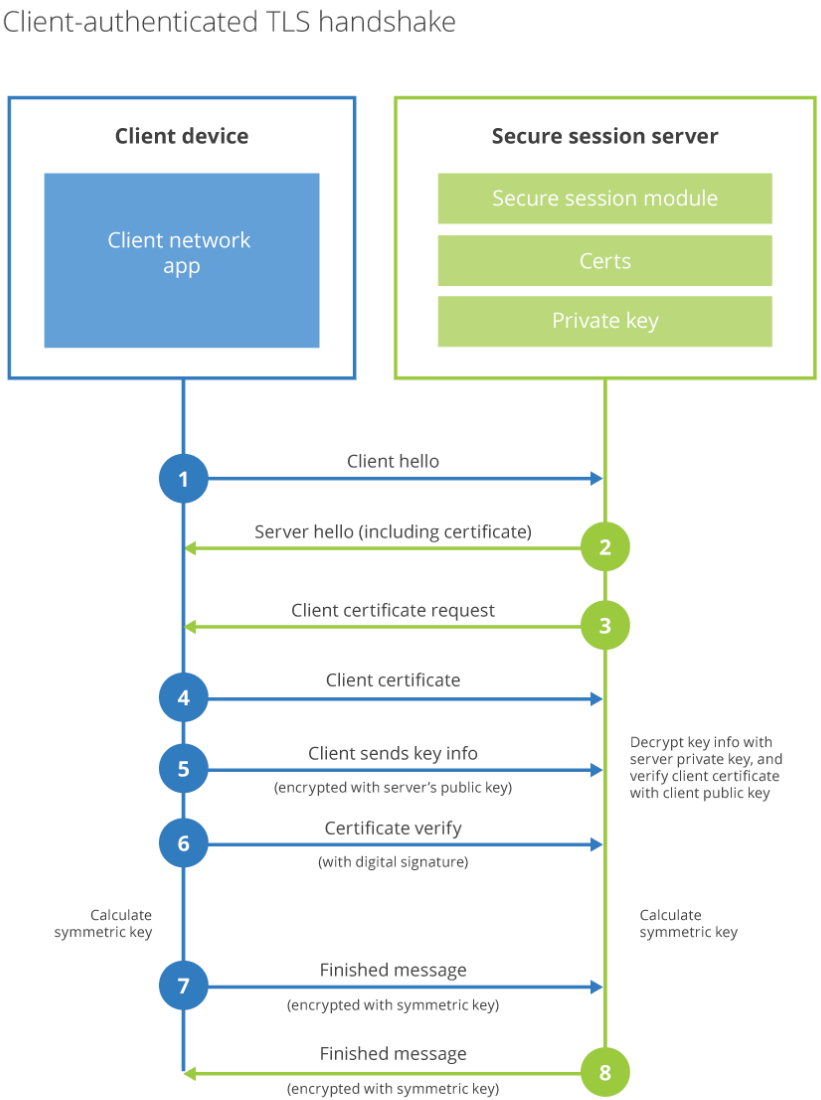
Segurança no MQTT - Nível de Transporte - TLS/SSL

Definition

At the core, TLS and SSL are cryptographic protocols which use a handshake mechanism to negotiate various parameters to create a secure connection between the client and the server. [1]

- MQTT já possui suporte ao TLS, ao utiliza-lo, obtemos:
  - Confidencialidade;
  - Integridade;
- Autenticação:
  - TLS via certificado (cliente e servidor);
  - Broker via usuario e senha;
- Auditoria:
  - Feita no broker via Logs;

Segurança no MQTT - Nível de Transporte - TLS/SSL



Segurança no MQTT - Nível de Transporte - Autenticação

- CONNECT message:
  - Autenticação via usuario e senha no Broker MQTT

MQTT-Packet:

CONNECT

contains:	Example
clientId	"client-1"
cleanSession	true
username (optional)	"hans"
password (optional)	"letmein"
lastWillTopic (optional)	"/hans/will"
lastWillQos (optional)	2
lastWillMessage (optional)	"unexpected exit"
lastWillRetain (optional)	false
keepAlive	60

Segurança no MQTT - Nível de Aplicação

- Autenticação
- Criptografia simétrica

Sumário

Configurando Broker - Mosquitto

- https://mosquitto.org/download/
- TLS;
- Autenticação cliente e servidor via certificado;
- Autenticação cliente via usuário e senha;
- Autorização via ACL no broker;

Configurando Broker - Mosquitto - TLS

Requisitos do Broker:

- Certificado da CA;
- Certificado do servidor;
- Chave privada;

Requisitos do Cliente:

- Certificado da CA;

1. Par de chaves para CA;
2. Certificado da CA;
3. Chaves para o broker;
4. certificado do broker;
5. Assinar o certificado do broker com a CA;

Configurando Broker - Mosquitto - TLS

Gerando arquivos necessários:

- $ openssl genrsa -des3 -out ca.key 2048
- $ openssl req -new -x509 -days 1826 -key ca.key -out ca.crt
- $ openssl genrsa -out server.key 2048
- $ openssl req -new -out server.csr -key server.key
- $ openssl x509 -req -in server.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out server.crt -days 360

Output:

- ca.crt
- ca.key
- ca.srl
- server.csr
- server.crt
- server.key

Configurando Broker - Mosquitto - TLS

Mova os arquivos para:

/etc/mosquitto/certs:

- server.crt
- server.key

/etc/mosquitto/ca_certificates

- ca.crt

Configurando Broker - Mosquitto - TLS

- Altere o arquivo de configuração;
- /etc/mosquitto/conf.d/ local.conf

```
port 8883

allow_anonymous true

password_file /etc/mosquitto/passwordfile

cafile /etc/mosquitto/ca_certificates/ca.crt

keyfile /etc/mosquitto/certs/server.key

certfile /etc/mosquitto/certs/server.crt

require_certificate false

use_identity_as_username false
```

Configurando Broker - Mosquitto - TLS

- TLS já configurado;
- Apenas o cliente autentica o servidor;
- Cliente necessita do certificado da CA que assinou o certificado do broker;
- $mosquitto -c /etc/mosquitto/conf.d/local.conf
- $mosquitto_sub -p 8883 -h localhost -t /hello --cafile ca.crt
- $mosquitto_pub -p 8883 -h localhost -t /hello -m "ola TLS" --cafile ca.crt

Configurando Broker - Mosquitto - Autenticação

- Autenticação via usuário e senha:

  allow_anonymous false
  password_file /etc/mosquitto/passwordfile

- passwordfile.txt

```
andre:changeme
igor:123456
pablo:654321
```

- $mosquitto_passwd -U passwordfile

```
andre:$6$Y97FBWT4pKzYH1Fv...
igor:$6$3E5ZgxX6pwYPikJr...
pablo:$6$O1+MiKq+BfhhMzJg...
```

Configurando Broker - Mosquitto - Autenticação

- $mosquitto -c /etc/mosquitto/conf.d/local.conf
- $mosquitto_sub -p 8883 -h localhost -t /hello −−cafile ca.crt -u andre -P "123456"
- $mosquitto_pub -p 8883 -h localhost -t /hello -m "ola TLS"−−cafile ca.crt -u andre -P "123456"

Configurando Broker - Mosquitto - Autenticação

Autenticação via certificado do cliente;

require_certificates true
use_identity_as_username false
crlfile /path/to/crlfile

Criando o certificado do cliente:

1. Criar chave do cliente;
2. Criar o certificado do cliente usando a chave;
3. Assinar o certificado do cliente com a mesma CA que assinou o servidor;

Configurando Broker - Mosquitto - Autenticação

- Criando o certificado do cliente:
  1. $openssl genrsa -out client.key 2048
  2. $openssl req -new -out client.csr -key client.key
  3. $openssl x509 -req -in client.csr -CA ca.crt -CAkey ca.key -CAcreateserial -out client.crt -days 360
- Output:
  - client.key
  - client.csr
  - client.crt

Configurando Broker - Mosquitto - Autenticação

Autenticação via certificado do cliente;

- $mosquitto_sub -p 8883 -h localhost -t /hello --cafile ca.crt --cert client.crt --key client.key

- $mosquitto_pub -p 8883 -h localhost -t /hello -m "olá tls" --cafile ca.crt --cert client.crt --key client.key

Configurando Broker - ACL

- acl_file path/to/ acl.txt
  - Afeta os clientes sem usuario:

    topic read /topic
  - Afeta o username:

    user andre
    topic write /topic
  - Afeta todos os clientes:

    pattern readwrite /topic/%u/#

Configurando Broker - CRL

1. https://www.hivemq.com/mqtt-security-fundamentals/

Configurando Broker - Auditoria

- Via logs no broker;
- Alterando arquivo de configurações:
  - log_type [debug, error, warning, notice, information, subscribe, unsubscribe, websockets, none, all.]
  - log_dest file /path/to/mosquitto.log

Sumário

Configurando Broker - Moquette

1. `https://github.com/moquette-io/moquette`;
2. Implementado em Java;
3. Arquivos de configuração semelhantes ao Mosquitto;
4. Possivel utilizar keystores;
5. Fornece Interfaces para configurações customizadas;

Referências

1. https://www.hivemq.com/mqtt-security-fundamentals/
2. https://github.com/moquette-io/moquette