SECURING MONGODB WITH A CERTIFIED SSL CERTIFICATE

Cognitive Assistant for Networks (CAN) Release 5.5

JULY 22, 2021

AVANSEUS TECHNOLOGY PVT. LTD.

REVISION HISTORY

Version	Date	Change description	Created by	Updated by	Reviewed by
V 1.0	July, 2021	Initial Release	Sunil	Sandeep Singh	Chiranjib

Table of Contents

avanseus

1. Introduction

This document focuses on configuring MongoDB to use SSL/TLS certificates. Existing customer environments with VM based MongoDB setup can use this documentation for encrypting the data transfers between client applications and MongoDB server.

2. Steps to configure MongoDB using a certified SSL certificate

Follow the steps:

1. Generate the Let's Encrypt SSL certificate using certbot package. A separate document on how to "generate "Let's Encrypt SSL certificate" is available.
2. Login as a root user and perform the following.

```
sudo su
```

3. Combine the privkey and cert into a single file mongo.pem

```
cd /etc/letsencrypt/live/<domain>
cat fullchain.pem privkey.pem > /etc/ssl/mongo.pem
```

4. Create the CA file. Save the content available in this link https://letsencrypt.org/certs/trustid-x3-root.pem.txt to the file name called ca.crt. Then run:

```
printf "\n" >> ca.crt
cat /etc/letsencrypt/live/<domain>/chain.pem >> /etc/ssl/ca.crt
```

5. Convert the crt file to a pem file using openssl:

```
openssl x509 -in /etc/ssl/ca.crt -out /etc/ssl/ca.pem -outform PEM
```

Just to make sure that everything is setup correctly run:

```
openssl verify -CAfile /etc/ssl/ca.pem /etc/letsencrypt/live/<domain>/chain.pem
```

You should get:

```
mongo.pem: OK
```

Note: Even if the certificate is correct, sometimes verification can give an error.

6. Configure MongoDB
Specify the SSL locations in the MongoDB configuration file. Specify PEMKeyPassword if the cert is from a different source than Let's Encrypt.

```
sudo vi /etc/mongod.conf
```

For MongoDB 4.0 and lower versions

```
# network interfaces
net:
  port: 27017
  bindIp:0.0.0.0  #set this to 0.0.0.0 only
  ssl:
    mode: requireSSL
    PEMKeyFile: /etc/ssl/mongo.pem
    PEMKeyPassword:   #optional
    CAFile:     /etc/ssl/ca.pem
```

For MongoDB 4.2 and later versions

```
# network interfaces
net:
  port: 27017
  bindIp: 0.0.0.0  #set this to 0.0.0.0 only
  tls:
    mode: requireTLS
    certificateKeyFile: /etc/ssl/mongo.pem
    CAFile:     /etc/ssl/ca.pem
```

7. Start MongoDB service

```
mongod --config=<MongoDB_configuration_path>
```

8. Connect to mongo shell using the below command:
   For MongoDB 4.0 and lower versions

```
mongo --ssl -sslCAFile /etc/ssl/ca.pem --sslPEMKeyFile /etc/ssl/mongo.pem
<domain>:<port_number>/<db_name> -u <username> -p <password>
```

   For MongoDB 4.2 and later versions

```
mongo --tls --tlsCAFile /etc/ssl/ca.pem --tlsCertificateKeyFile /etc/ssl/mongo.pem
<domain>:<port_number>/<db_name> -u <username> -p <password>
```

   From inside the shell, you can check SSL is running:

```
db.serverStatus().security;
```

   You should get the output as:

```
{
      "SSLServerSubjectName" : "CN=<domain>",
      "SSLServerHasCertificateAuthority" : true,
      "SSLServerCertificateExpirationDate" : ISODate("2021-06-13T14:27:52Z")
}
```

9. Create a Keystore in pkcs12 format using the pem file generated in the previous steps. It will ask for the password during Keystore generation. Please note the password that you give. This Keystore file and password are required for the Java Mongo driver.

```
openssl pkcs12 -export -out mongo.pkcs12 -in /etc/ssl/mongo.pem
```