Certificate Creation with OpenSSL

Complete OpenSSL TLS certificate guide: generate CA key/cert and server key/CSR, sign X509 with openssl x509 -CA/-CAkey, and verify certificates.

Warning: the instructions below generate version 1 certificates only. These work fine with servers and clients using OpenSSL, but fail if WolfSSL is used instead, as is the case for our Windows MSI packages and our binary tarballs for Linux.

WolfSSL requires version 3 certificates instead when using TLS v1.2 or higher, and so won't work with certificates generated as shown here when using two-way TLS with explicit client certificates.

Generating version 3 certificates requires a few more minor steps, we will upgrade the instructions below soon to include these.
See also: MDEV-25701 ↗

In order to secure communications with the MariaDB Server using TLS, you need to create a private key and an X509 certificate for the server. You may also want to create additional private keys and X509 certificates for any clients that need to connect to the server with TLS. This guide covers how to create a private key and a self-signed X509 certificate with OpenSSL.

Certificate Creation

The OpenSSL library provides a command-line tool called openssl, which can be used for performing various tasks with the library, such as generating private keys, creating X509 certificate requests, signing X509 certificates as a Certificate Authority (CA), and verifying X509 certificates.

Creating a Certificate Authority Private Key and Certificate

The Certificate Authority (CA) is typically an organization (such as Let's Encrypt) that signs the X509 certificate and validates ownership of the domain. However, when you would like to use self-signed certificates, you need to create the private key and certificate for the CA yourself, and then you can use them to sign your own X509 certificates.

To start, generate a private key for the CA using the openssl genrsa command. For example:

```
# openssl genrsa 2048 > ca-key.pem
```

After that, you can use the private key to generate the X509 certificate for the CA using the openssl req command. For example:

```
# openssl req -new -x509 -nodes -days 365000 \
      -key ca-key.pem -out ca.pem
```

The above commands create two files in the working directory: The `ca-key.pem` private key and the `ca.pem` X509 certificate are both are used by the CA to create self-signed X509 certificates below.

Creating a Private Key and a Self-signed Certificate

Once you have the CA's private key and X509 certificate, you can create the self-signed X509 certificates to use for the MariaDB Server, client, replication and other purposes.

To start, generate a private key and create a certificate request using the openssl req ↗ command. For example:

```
# openssl req -newkey rsa:2048 -days 365000 \
      -nodes -keyout server-key.pem -out server-req.pem
```

After that, process the key to remove the passphrase using the openssl rsa ↗ command. For example:

```
# openssl rsa -in server-key.pem -out server-key.pem
```

Lastly, using the certificate request and the CA's private key and X509 certificate, you can generate a self-signed X509 certificate from the certificate request using the openssl x509 ↗ command. For example:

```
# openssl x509 -req -in server-req.pem -days 365000 \
      -CA ca.pem -CAkey ca-key.pem -set_serial 01 \
      -out server-cert.pem
```

This creates a `server-cert.pem` file, which is the self-signed X509 certificate.

Certificate Verification

Once you have created the CA's X509 certificate and a self-signed X509 certificate, you can verify that the X509 certificate was correctly generated using the openssl verify ↗ command. For example:

```
# openssl verify -CAfile ca.pem server-cert.pem
server-cert.pem: OK
```

You can add as many X509 certificates to check against the CA's X509 certificate as you want to verify. A value of `OK` indicates that you can use it was correctly generated and is ready for use with MariaDB.

Previous
Data-in-Transit Encryption

Next
Enabling TLS on MariaDB Server

Last updated 28 days ago

Was this helpful?

MariaDB

Products

Enterprise Platform

Community Server

Download MariaDB

Pricing

Support

Customer Login

Technical Support

Remote DBA

Professional Services

Resources

MariaDB Blog

Webinars

Customer Stories

MariaDB Events

Documentation

Developer Hub

Company

About MariaDB

Newsroom

Leadership

MariaDB Careers

Legal

Privacy Policy