cubeSQL

SSL support

cubeSQL SSL

Starting from version 4.3 cubeSQL fully support the SSL protocol. SSL is automatically loaded if OpenSSL is installed on your system. OpenSSL is freely available from: http://www.openssl.org.

SSL support in cubeSQL version 5 has been greatly improved with changes both on server side and client side. This document will briefly summarize all of them.
When not differently specified, validations will use default OpenSSL algorithms. Validations means protocol validation, certificates validation and peer verifications.

Server Side

- By default cubeSQL uses the strongest available protocol with the following order: TLSv1.2, TSLv1.1, TSLv1 and SSLv3 (or the strongest one available in the OpenSSL library).

- If a root certificate is provided then peer verification is performed.

- User can decide which protocol to disable. The following configurations can be used:
  Disable SSLv3
  Disable SSLv3 and TSLv1
  Disable SSLv3, TSLv1 and TSLv1.1
  Disable SSLv3, TSLv1 and TSLv1.1 and TSLv1.2 (to support just a future TSLv1.3 protocol)

- User can specify DH key bits and that key will be generated on the fly by the server.

- User can specify a list of allowed SSL ciphers to accept. Un-allowed ciphers will be refused before a connection will be established.

- By default SSLv2 is disabled.

- By default when performing renegotiation as a server, always start a new session (i.e., session resumption requests are only accepted in the initial handshake). SSL_OP_NO_SESSION_RESUMPTION_ON_RENEGOTIATION flag.

- By default always create a new key when using temporary/ephemeral DH parameters. This option must be used to prevent small subgroup attacks, when the DH parameters were not generated using ``strong'' primes. SSL_OP_SINGLE_DH_USE flag.

- By default apply all available patches and workaround to OpenSSL bugs. SSL_OP_ALL flag.

Client Side

- Added support for loading SSL shared libraries in order to have the possibility to always stay up to date with the latest OpenSSL versions.
- By default client will use the strongest available protocol with the following order: TLSv1.2, TSLv1.1, TSLv1 and SSLv3.
- If a root certificate is provided then peer verification is used.
- User can specify a list of SSL ciphers to use.

Please report your feedback, your impressions or your bug reports directly to me at:
marco@sqlabs.com