TLS Support

SSL/TLS is supported by Redis starting with version 6 as an optional feature that needs to be enabled at compile time.

Getting Started

Building

To build with TLS support you'll need OpenSSL development libraries (e.g. libssl-dev on Debian/Ubuntu).

Run `make BUILD_TLS=yes`.

Tests

To run Redis test suite with TLS, you'll need TLS support for TCL (i.e. `tcl-tls` package on Debian/Ubuntu).

1. Run `./utils/gen-test-certs.sh` to generate a root CA and a server certificate.
2. Run `./runtest --tls` or `./runtest-cluster --tls` to run Redis and Redis Cluster tests in TLS mode.

Running manually

To manually run a Redis server with TLS mode (assuming `gen-test-certs.sh` was invoked so sample certificates/keys are available):

```
./src/redis-server --tls-port 6379 --port 0 \
    --tls-cert-file ./tests/tls/redis.crt \
    --tls-key-file ./tests/tls/redis.key \
    --tls-ca-cert-file ./tests/tls/ca.crt
```

To connect to this Redis server with `redis-cli`:

```
./src/redis-cli --tls \
    --cert ./tests/tls/redis.crt \
    --key ./tests/tls/redis.key \
    --cacert ./tests/tls/ca.crt
```

Certificate Configuration

In order to support TLS, Redis must be configured with a X.509 certificate and a private key. In addition, it is necessary to specify a CA certificate bundle file or path to be used as a trusted root when validating certificates. To support DH based ciphers, a DH params file can also be configured. For example:

```
tls-cert-file /path/to/redis.crt
tls-key-file /path/to/redis.key
tls-ca-cert-file /path/to/ca.crt
tls-dh-params-file /path/to/redis.dh
```

TLS Listening Port

The `tls-port` configuration directive enables accepting SSL/TLS connections on the specified port. This is **in addition** to listening on `port` for TCP connections, so it is possible to access Redis on different ports using TLS and non-TLS connections simultaneously.

You may specify `port 0` to disable the non-TLS port completely. To enable only TLS on the default Redis port, use:

```
port 0
tls-port 6379
```

Client Certificate Authentication

By default, Redis uses mutual TLS and requires clients to authenticate with a valid certificate (authenticated against trusted root CAs specified by `ca-cert-file` or `ca-cert-dir`).

You may use `tls-auth-clients no` to disable client authentication.

Replication

A Redis master server handles connecting clients and replica servers in the same way, so the above `tls-port` and `tls-auth-clients` directives apply to replication links as well.

On the replica server side, it is necessary to specify `tls-replication yes` to use TLS for outgoing connections to the master.

Cluster

When Redis Cluster is used, use `tls-cluster yes` in order to enable TLS for the cluster bus and cross-node connections.

Sentinel

Sentinel inherits its networking configuration from the common Redis configuration, so all of the above applies to Sentinel as well.

When connecting to master servers, Sentinel will use the `tls-replication` directive to determine if a TLS or non-TLS connection is required.

Additional Configuration

Additional TLS configuration is available to control the choice of TLS protocol versions, ciphers and cipher suites, etc. Please consult the self documented `redis.conf` for more information.

Performance Considerations

TLS adds a layer to the communication stack with overheads due to writing/reading to/from an SSL connection, encryption/decryption and integrity checks. Consequently, using TLS results in a decrease of the achievable throughput per Redis instance (for more information refer to this discussion).

Limitations

I/O threading is currently not supported with TLS.

This website is open source software. See all credits.

Sponsored by redislabs HOME OF REDIS