SZTPD Installation Guide

Watsen Networks

February 15, 2025

Abstract

This documentation is for the "Secure ZTP Deamon (SZTPD)" product by Watsen Networks.

This documentation is still a work-in-progress. Some sections clearly indicate when material is pending, but there are also some missing sections, and other sections may not have enough detail.

SZTPD is currently in its "pre-alpha" stage and details captured in this document may change.

Contents

1 Introduction

SZTPD is an implementation of a "bootstrap server", as defined in the Terminology section of RFC 8572, also known as an "SZTP server", as defined in the Terminology section of RFC 9646".

SZTPD is provided as software, providing both a CLI command called "sztpd" as well as a Python package that could be used in Python "import statements.

In either case, SZTPD presents a northbound API for configuration and monitoring, eastbound hooks for integration with deployment-specific infrastructure, and a southbound API for bootstrapping devices.

This documentation is in preparation for a "1.0.0" release, using the common "major.minor.patch" semantic versioning convention. The use of versions like "0.0.N", where 'N' is an integer, should be read as "the Nth Pre-Alpha, with no statement about if it contains"major" or "minor" changes.

2 Installation

SZTPD is installed using the command:

```
$ pip install sztpd
```

or, if Python 3 is installed separately:

```
$ pip3 install sztpd
```

On some systems it may be necessary to install the dependency pacakges first. The SQLite package in particular can be tricky to get installed. For the Ubuntu and MacOS platforms, the project's GitHub Action workflow shows exactly which commands work.

2.1 Machine Resources

This section is nearly identical to the "Machine Resources" section in the YANGcore installation guide. The following sections attempt to document only where SZTPD use of resources differs.

2.1.1 Processors

SZTPD's use of processors is inherited from YANGcore. Please see the "Processors" section in the YANGcore Installation Guide for details.

2.1.2 Memory

SZTPD's use of memory is inherited from YANGcore. Please see the "Memory" section in the YANGcore Installation Guide for details.

2.1.3 Filesystem

SZTPD's use of the filesystem is inherited from YANGcore. Please see the "Filesystem" section in the YANGcore Installation Guide for details.

2.1.4 Network Interfaces

In addition to the networking demands described by the "Network Interfaces" section in YANGcore's Installation Guide, SZTPD also presents a Southbound interface.

STPD's southbound interface is used for machine-to-machine interactions (i.e., SZTPD and bootstrapping devices), and thus the speed of the network interface needs to be fast enough to not cause timeouts or user frustration[1].

[1] When fully automated, there is no user in the loop and time doesn't matter. Some bootstrapping use cases entail a user waiting for a signal (e.g., an LED changing color), in which case time mattes.

2.2 Operating System

SZTPD's support for operating systems is the same as YANGcore. Please see the "Operating System" section in the YANGcore Installation Guide for details.

2.3 Networking

SZTPD's networking usage is described in the following two sections. The first section regards the networking needs for inbound connections. The second section regards the networking needs for outbound connections.

2.3.1 Inbound Connections

In addition to YANGcore listening for connections from "northbound" clients (i.e., users), SZTPD also listens for connections from "southbound" clients (i.e., bootstrapping devices).

2.3.1.1 APIs

SZTPD adds the "rfc8572" interface type, in addition to YANGcore's "native" interface type. Please see the "APIs" section in the YANGcore's Installation Guide for details about YANGcore's "native" interface type.

The "rfc8572" interface type presents the bootstrap server API defined in Section 7 of RFC 8572.

In order for SZTPD to satisfy its business purpose, there must be at least one listening port presenting the "rfc8572" interface.

More than one "rfc8572" interface may be configured if, e.g., distinct listening ports are needed to listen on different IP addresses and/or present different TLS certificates[2].

2.3.2 Outbound Connections

In addition to the reasons listed in YANGcore's Installation Guide, SZTPD initiates outbound connections for the reasons discussed in this section.

2.3.2.1 Executing Dynamic Callouts In addition to the dynamic callouts supported by YANGcore, SZTPD defines additional dynamic callouts (see the "Dynamic Callouts" section in the User Guide). Each dynamic callout is implemented by a deployment-specific plugin (Python code) and thus can exhibit any behavior necessary.

2.4 Security

SZTPD leverages the Security implemented by YANGcore. Please see the "Security" section in YANGcore's Installation Guide.

[2]Additional listening ports may be used to present a distinct TLS end-entity certificate to different groups of devices, such as may be necessary in order to accommodate devices manufactured using different trust anchors.

2.5 The sztpd Executable

When the installation completes, the executable "sztpd" is installed in your shell's path.

To test running SZTPD and see its "help" page:

```
$ sztpd --help
```

Since SZTPD presents the same CLI as YANGcore, the following output is identical YANGcore's except that the strings "YANGcore" and "yangcore" are replaced with "SZTPD" and "sztpd", respectively.

```
usage: sztpd [-h] [-v] [-C CACERT] [-c CERT] [-k KEY] database-url

SZTPD implements the "bootstrap server" defined in RFC 8572.

positional arguments:
  database-url          see below for details.

optional arguments:
  -h, --help            show this help message and exit
  -v, --version         show version number and exit.
  -C CACERT, --cacert CACERT
                        path to certificates used to authenticate the database
                        (see below for details).
  -c CERT, --cert CERT  path to cert used to authenticate SZTPD to the
                        database (see below for details).
  -k KEY, --key KEY     path to key used to authenticate SZTPD to the database
                        (see below for details).

Exit status code: 0 on success, non-0 on error.  Error output goes to stderr.

The "cacert" argument is a filepath to a PEM file that contains one or more X.509
CA certificates used to authenticate the RDBMS's TLS certificate.

The "key" and "cert" arguments are each a filepath to a PEM file that contains
the key and certificate that SZTPD should use to authenticate itself to the
RDBMS.  These parameters must be specified together, and must be specified
in conjunction with the "cacert" parameter.

The "database-url" argument has the form "<dialect>:<dialect-specific-path>".
Three dialects are supported: "sqlite", "postgresql", and "mysql+pymysql".
The dialect-specific-path for each of these is described below.

For the "sqlite" dialect, <dialect-specific-path> follows the format
"///<sqlite-path>", where <sqlite-path> can be one of:

  :memory:    - an in-memory database (only useful for testing)
  <filepath>  - an OS-specific filepath to a persisted database file

  Examples:

    $ sztpd sqlite:///:memory:                      (memory)
    $ sztpd sqlite:///relative/path/to/sztpd.db     (unix)
    $ sztpd sqlite:////absolute/path/to/sztpd.db    (unix)
    $ sztpd sqlite:///C:\path\to\sztpd.db           (windows)

For both the "postgresql" and "mysql+pymysql" dialects, <dialect-specific-path>
follows the format "//<user>[:<passwd>]@<host>:<port>/<database-name>".

  Examples:

    The following two examples assume the database is called "sztpd" and
    that the database server listens on the loopback address with no TLS.

      $ sztpd mysql+pymysql://user:pass@localhost:3306/sztpd
      $ sztpd postgresql://user:pass@localhost:5432/sztpd

Please see the documentation for more information.
```


2.5.1 Persistence Selection

SZTPD's presents the same CLI parameter to select the persistence strategy as YANGcore. Please see the "Persistence Selection" section in the YANGcore Installation Guide for details.

2.5.2 First-time Initialization

SZTPD's first-time initialization is the same as YANGcore, with the only differences being the Contract displayed and the name of the environment variable used to bybass the prompt. For instance:

```
$ export SZTPD_ACCEPT_CONTRACT="Yes"; sztpd sqlite:///:memory:
```

2.5.3 Factory Defaults

SZTPD's defaults are inherited from YANGcore. Please see the "Factory Defaults" section in the YANGcore Installation Guide for details.

One difference is the name of the environment variables to alter the defaults address and port changed, using the string "SZTPD" instead of "YANGCORE". For instance:

- The default local address may be changed using the "SZTPD_INIT_ADDR" environment variable.
- The default local port may be changed using the "SZTPD_INIT_PORT" environment variable.

2.5.4 Daemonize

SZTPD's daemonization is the same as YANGcore. Please see the "Daemonize" section in the YANGcore Installation Guide for details.

2.5.5 Signals

SZTPD's signal-handing is the same as YANGcore. Please see the "Signals" section in the YANGcore Installation Guide for details.

2.5.6 User Privileges

SZTPD's need for user priviledge is the same as YANGcore. Please see the "User Privileges" section in the YANGcore Installation Guide for details.

2.5.7 Dedicated Cores

SZTPD use of cores is the same as YANGcore. Please see the "Dedicated Cores" section in the YANGcore Installation Guide for details.

2.5.8 Output

SZTPD inherits its output characteristics from YANGcore. Please see the "Output" section in the YANGcore Installation Guide for details.

The ony minor difference being the command used to pipe the 'sztpd' process's output to a file:

```
sztpd sqlite:///:memory: >> /var/logs/sztpd.log 2>&1
```


2.6 High-Availability

SZTPD inherits its availability support from YANGcore. Please see the "High-Availability" section in the YANGcore Installation Guide for details.

2.7 Scaling Guidelines

SZTPD inherits its scaling characteristics for the northbound interface from YANGcore. Please see the "Scaling Guidelines" section in YANGcore's Installation Guide for details.

For the southbound interface, a timing tests on a MacOS laptop having an "M2" chip show that the bootstrapping responses are returned in less than a half-second, suggesting a theorectical maximum of around 200,000 boostraps per day.

2.8 Upgrades

Upgrading SZTPD uses the same mechanism as YANGcore. Please see the "Upgrades" section in YANGcore's Installation Guide for details.

The only difference between the two experiences is the command to upgrade SZTPD references the "sztpd" (not "yangcore") package:

```
pip install --upgrade sztpd
```

3 RDBMS Configuration

Configuring a database for SZTPD is nearly identical to configuring it for YANGcore. The only difference being the name of the database changes from "yangcore" to "sztpd". Please see the "RDBMS Configuration" section in the YANGcore Installation Guide for details. For instance, see "sztpd" on the second line below:

```
$ mysql -u root -e "CREATE USER 'testuser'@'localhost' IDENTIFIED BY 'secret';"
$ mysql -u root -e "GRANT ALL ON sztpd.* TO 'testuser'@'localhost';"
$ mysql -u root -e "FLUSH PRIVILEGES;"
```

