SZTPD Alpha-6 Release Notes

Watsen Networks

August 27, 2020

Abstract

This documentation provides release notes for the Alpha-6 release of SZTPD.

Contents

1 Introduction

The sections below cover what is and is not working in this release of the SZTPD product.

Most everything listed as "not working" will be implemented before FCS.

2 Implemented / Tested

This section describes what is working in SZTPD. All of the following has been tested[1]:

- All APIs work: 'native', 'tenant', and 'rfc8572'.
- Each listening port is HTTP with or without TLS[2].
- Client auth can be TLS client-certs and/or HTTP basic auth.
- RESTCONF HEAD, GET, POST, PUT, and DELETE work over entire tree.
- The ./well-known/host-meta, RESTCONF root (i.e., {+restconf}), and YANG-library resources.
- Tested using in-memory, file-base, MySQL (inc. AWS Aurora), and Postgres databases.
- TLS connection to backend RDBMS, with or w/o client certificate.
- Plugin-based callouts for response manager[3].
- Plugin-based callout for ownership verification
- Bootstrapping log (including information about SZTPD's relay to remote systems)
- Audit log (including per-tenant audit log)
- Both product modes, 1 and x, work.
- Webhook-based callouts for notifications
- Database-level transactions.
- Both Python 3.7 and 3.8.

3 Should Work

This section regards things that should work, but haven't been tested yet.

- IPv6: The default port binds to "127.0.0.1", but this can be changed by setting the FIXME "SZTPD_DEFAULT_ADDR" environment variable. Other than this, there is no other IPv4-only code in the product as everything is handled by Python modules.

- UTF-8: Python string types support both ASCII and unicode, like UTF-8. Presumably string handling is ambivalent, but this in isn't tested yet.

- Windows: the software has been developed and tested exclusively in UNIX based systems. Python is very portable, and SZTPD has almost no interaction with the filesystem, so running on Windows might work, but this has not been tested.

[1] There is more than twice the number of lines of test code than code in SZTPD itself.
[2] An external TLS-terminator must be used when an SZTPD listening port is configured to NOT use HTTP.
[3] Plugin-based callouts have been used to test support for draft-kwatsen-netconf-sztp-csr.

4 Upcoming Features/Releases:

The following features are sorted by the expected release they might show up in. Please let us know if there is something out of place or missing.

4.1 Alpha-7

- Completely remove support for more than one API interface per listening port. Currently the 'use-for' leaf-list has 'max-elements' set to '1'. Update Installation and Adminstrator guides.
- init perf / scale / soak tests

4.2 Alpha-8

- Ordering (point + insert) query parameters. These query parameters are described by Section 4.8 of RFC 8040 to enable maintaining user-ordered lists. Workaround is to recreate the entire list in the correct order[4].

4.3 Alpha-9

- Paging (offset + limit) query parameters. It is desirable to introduce proprietary query parameters for interacting with potentially long lists (i.e., audit-log, bootstrapping log, devices, tenants, etc.)

4.4 Alpha-10

- Tenant view device types. Currently, tenants can set a valid 'device-types' value only if provided to them out-of-band. These values are only set at the "host" level, (on purpose) outside the tenant's purview. A tenant's clients should be able use either an RPC/action to get the supported device-types from the host, or have that information exposed as opstate (i.e., read-only "config false" values)[5].
- Validating the end-entity certificates, when being configured, match the associated private key. Currently the system trusts that the user passes in valid key/cert pairs.
- Validating base64 contents (CMS). Currently trusts that valid structures are configured. Only later, when using the structures will SZTPD decode them and potentially find errors. Exceptions will be thrown, but the error messages are not always helpful. Tests would include, e.g., that a CMS containing a sequence of certificates does not contain any spurious certificates.

4.5 Alpha-11

- RESTCONF entity-tag (ETag) Support. The HEAD and GET operations would return the "ETag" HTTP header field. The POST, PUT, and DELETE operations would inspect request headers for the "If-Match", and "If-None-Match" fields. This enables concurrent clients to detect if changes are have been made since their using a HEAD or GET command. Support for Etag is a MUST in RFC 8040 only on the top-level datastore resource, and unspecified for inner-resources[6]

[4] Currently, there are only three "ordered-by" lists: download-uris, bootstrap-servers, and matched-responses (the first two are leaf-lists).

[5] Current plan is to do the latter, but due to limitations in YANG, need to split the sztpd-1 schema into two: one for 'native' view and another for the 'tenant' view.

[6] Note that the top-level resource for the tenant-view is mounted to /tenants/tenant.

4.6 Alpha-X

- Filtering (depth + fields) query parameters. These query parameters are described by Section 4.8 of RFC 8040 to reduce the amount of data returned to the client. Workaround is for the client to discard the unwanted parts of the response itself.
- The RESTCONF/HTTP "OPTIONS" method. This needs to be added in order to notify clients that XML is NOT supported.
- Authorization / access-control. It is planned to implement the admin-account "access" node, which sets an enumerated value being one of "unrestricted", "typical", and "minimal".
- Implicit change tracking. This is needed to support a few of the features listed below. It is also needed to detect when any part of the "transport" configuration changes, so that a SIGHUP can be issued. Currently SIGHUP is issued only when an "endpoint" is added or removed.
- Device record counters. It is planned to track when the device records are created, last modified, and the total number of modifications. Similarly, to track when the bootstrapping device first connected, last connected, and the total number of connections.
- Reference tracking. It is desired to track references to objects in the system. In YANG terms, these are the 'leafref' statements that reference a 'list' statement's 'key' node. Tracking includes the current number of references and the time of last reference.
- Automated purging with notifications. It is desired to send notifications when unreferenced objects have been without a reference, for some configurable amount of time. A series of notifications with escalating urgency can be to sent to configurable notification receivers. A final notification is sent when the purging occurs.
- Password expirations with notifications. This feature goes with the previously mentioned "automated purging with notifications" feature, but has its own "preferences" setting for the timeouts.
- Email-based admin activations. It is intended that SZTPD will send email based notifications to newly created admins when their accounts are first created. It is important to validate the email address because the same email address may be emailed later when the password expiration date is approaching. This is why the email address is the primary key for the admin accounts.
- Password minimum length constraints. Currently the "preference" setting for the admin account password's minimum length constraint is ignored.
- Client certificate based auth to NBIs. Currently SZTPD implements client cert based auth to the SBI (i.e, 'rfc8572-interface'). This is to extend that configuration into the NBIs also (i.e., 'native' and 'tenant'). This would provide 2FA to the NBI.
- Webhook-based callouts for both the "get-conveyed-information" and "verify-device-ownership" callouts. Callback-based callouts can be used as a workaround until support for webhooks is implemented. More important to add webhook for "verify-device-ownership"?
- Plugin-based callbacks for the "relay-notification" callout. Webhook-based callouts can be used as a workaround until support for plugins is implemented.

4.7 Beta

- Run performance and soak tests. Only address issues found.

4.8 FCS

- Nothing new

- Stress tests
- Soak tests

4.9 Post 1.0

- Remove support for mode ‘1’. This change would affect the NBI only (no impact on the device-facing ‘rfc8572’ SBI).
  - Reasons to remove mode ‘1’:
    1) It may be important for non-production use environments (e.g., those used for evals and demos) to exactly mimic production environments.
    2) Eliminates an artificial constraint for Mode ‘1’ deployments, in they can easily configure an additional “tenant” for whatever reason. It also eliminates need to ever have to migrate mode-1 deployments to a mode-x deployment, which would require a database migration..
    3) The tenant-view provided by Mode ‘x’ quite nicely isolates system-level configuration enabling IT organizations to do the system-level install and then pass an “application” view that excludes the system-level install to another organization within the company.
    4) Only supporting Mode-x could greatly simplify the YANG modules. This could be important to developers as they might need to look at the YANG modules in order to understand the data model exposed by the API. While the current YANG is navigable, it could be even more so if only supporting Mode-x.
    5) Further simplify the YANG modules. Note that mode ‘0’ is already almost completely phased out, and yet the YANG modules have yet to be simplified…somewhat due to waiting to determine if also phasing out mode ‘1’.
  - Reasons to NOT remove mode ‘1’[7]:
    1) It is unclear what the market expectations are regarding metered based subscriptions versus flat-rate tiers for enterprises and service providers. Hesitant to change anything without some feedback.
- Run the “verify-device-ownership” callouts at time of bootstrapping event (in addition to when the device record was first created). Seems like something that should be opt-ed into, and hence a feature that can be implemented later.
- RESTCONF “Last-Modified” support. The HEAD and GET operations would return the “Last-Modified” HTTP header field. The POST, PUT, and DELETE operations would inspect request headers for the “If-Modified-Since” and “If-Unmodified-Since” fields. Support for timestamps is a SHOULD in RFC 8040.
- The RESTCONF/HTTP “PATCH” method. Though a MUST in RFC 8040, it seems mostly like a nice-to-have in SZTPD…
- XML-based messages in SBI. RESTCONF enables the Content-Type to be “application/yang.data+xml” signaling that the server supports encoding XML messages. Currently, SZTPD only supports “application/yang.data+json” for a JSON-only SBI interface. There is a question of market-demand as for if needed. All clients (both NBI and SBI) must use JSON-encoded RESTCONF messages.
- Support a callout to retrieve an ownership voucher from an external system. This would implement the “supply-ownership-voucher” RPC defined in the “wn-sztpd-callbacks” module. The RPC is currently protected by a ‘feature’ statement called “supply-ownership-voucher”, thus programmatically signalling that it is not supported, though visible in the YANG.

[7] if decide to keep mode ‘1’, the YANG can still be simplified by removing support for mode ‘0’.

- Support signing conveyed information sent from SZTPD using the private key associated with a configured owner certificate.
- Support encrypting conveyed information sent from SZTPD using the device's public key from its identity certificate (e.g., IDevID).
- Support stapling revocation responses to CMS objects returned to devices.
  - only needed for signed data? (what about the redirect-info's trust-anchor CMS?)
- Private key encryption. It is desirable for SZTPD to have an ability to encrypt private keys via a "root" key. Note that this is above and beyond DB-level encryption; it purpose to to shield keys even from administrators having appropriate access. This root key would be protected by access control and/or an HSM.
- Actions: None of the 'action' statements defined in the YANG modules are implemented yet. Primarily this effects the ability to create keys, and generate certificate signing requests. The workaround is to generate the keys and certs using an external PKI and then pass those values in as configuration.

5 Internal Work

- In DAL, rebase persistence encoding for documents to not include a "self" envelop. Mostly to simplify code, but might improve performance.
- Fully remove mode '0' by 1) restructuring YANG modules (all relative paths can be same) and 2) in DAL, optimize code bits having mode '0' handling. Mostly to simplify code, but might improve performance.
- In DAL, remove "singleton" lookups - replace all with '/' path. This provides more consistency and simplifies code.
- Factor out app-layer into its own modular package.

6 Known Limitations

- Due to issues with both Python (when SZTPD itself terminates TLS connections) and NGINX (when NGINX terminates TLS connections) limit client certificates to containing just the end-entity certificate (no intermediate certificates). That said, it is unusual for identity certificates (e.g., an IDevID or LDevID certificate) to include any intermediates certificates, so may not be an issue in practice. The Python issue is likely to be resolved in an upcoming Python release. It is unclear when the NGINX issue will be resolved.
- Python (currently) is unable to staple OSCP Responses to the TLS Handshake. Workaround, if needed at all, is to front SZTPD with an external TLS Terminator, which is better for perfomance anyway.

7 Change Log

7.1 0.0.6

- Device-ownership verification callout now works using plugin-based callouts.
- The validation-layer's cache is now rolled back when database transactions fail.
- The validation-layer now tests for uniqueness in global keys.

7.2 0.0.5

- Now supports fronting SZTPD with a TLS-terminator. Requires that the SZTP-client's certificate is passed to SZTPD via the HTTP header "X-Client-Cert" as a PEM (urlencoded). Tested using NGINX.

7.3 0.0.4

- Now supports concurrent write requests.

7.4 0.0.3

- all unit tests now pass when SZTPD points to a MySQL database. With or without TLS, with or without client certificate. AWS Aurora MySQL also tested.
- RESTCONF error meesages are now returned on the SBI (RFC 8572) interface.

7.5 0.0.2

- callback-based callouts implemented to support draft-kwatsen-netconf-sztp-csr.
- fixed bugs related to bootstrapping-log and audit-log not cleaning up correctly when deleted.

7.6 0.0.1

- initial public release