s2n(3) 0.1 | s2n Tcl wrapper

Cyan Ogilvie

S2N

s2n Tcl wrapper - layer TLS onto Tcl channels

SYNOPSIS

package require s2n ?0.1?

s2n::push *channelName* ?**-role client**|**server**? ?**-servername** *host*?

DESCRIPTION

This package provides a channel driver that can be stacked on any Tcl channel that supports reading and writing to add TLS support. The TLS implementation uses Amazon's s2n for the TLS implementation and aws-lc for the libcrypto implementation.

COMMANDS

s2n::push *channelName* ?*-opt val* . . . ? Stack a TLS protocol driver onto the channel *channelName*. See **OPTIONS** for the options supported. The TLS protocol driver may be removed with the standard **chan pop** *channelName* Tcl command.

OPTIONS

-role client|server Set the role the TLS driver will play in the TLS handshake. If **client** (the default) the driver will initiate a TLS handshake otherwise it will wait to receive a ClientHello TLS message and handshake as a server.

-servername *host* Set the SNI (Server Name Indication) name to send when handshaking as a client.

EXAMPLES

Connect to an HTTPS server, request /

```
set sock    [socket www.google.com 443]
s2n::push $sock
chan configure $sock -buffering none -translation crlf -encoding ascii
puts $sock "GET / HTTP/1.1\nHost: www.google.com\nConnection: close\n"
puts [read $sock]
close $sock
```

BUILDING

This package has no external dependencies other than Tcl. The s2n and aws-lc libraries it depends on are included as submodules (or baked into the release tarball) and are built and statically linked as part of the package build process.

Currently Tcl 8.7 is required, but if needed polyfills could be built to support 8.6.

From a Release Tarball

Download and extract the release, then build in the standard TEA way:

```
wget https://github.com/cyanogilvie/tcl-s2n/releases/download/v0.1/s2n0.1.tar.gz
tar xf s2n0.1.tar.gz
cd s2n0.1
./configure
make
sudo make install
```

From the Git Sources

Fetch the code and submodules recursively, then build in the standard autoconf / TEA way:

```
git clone --recurse-submodules https://github.com/cyanogilvie/tcl-s2n
cd tcl-s2n
autoconf
./configure
make
sudo make install
```

In a Docker Build

Build from a specified release version, avoiding layer pollution and only adding the installed package without documentation to the image, and strip debug symbols, minimising image size:

```
WORKDIR /tmp/tcl-s2n
RUN wget https://github.com/cyanogilvie/tcl-s2n/releases/download/v0.1/s2n0.1.tar.gz -O - |
    ./configure; make test install-binaries install-libraries && \
    strip /usr/local/lib/libs2n*.so && \
    cd .. && rm -rf tcl-s2n
```

For any of the build methods you may need to pass `--with-tcl /path/to/tcl/lib` to `configure` if your Tcl install is somewhere nonstandard.

Testing

Since this package deals with security sensitive code, it's a good idea to run the test suite after building (especially in any automated build or CI/CD pipeline):

```
make test
```

And maybe also the memory checker `valgrind` (requires that Tcl and this package are built with suitable memory debugging flags, like `CFLAGS="-DPURIFY -Og" --enable-symbols`):

```
make valgrind
```

SECURITY

Given the limitations of a scripting language environment, this package's code does not have sufficient control over freed memory contents (or memory paged to disk) to guarantee that key material or other sensitive material (like decrypted messages) can't leak in a way that could be exploited by other code running on the shared memory (or disk) machine. For this reason, careful consideration should be given to the security requirements of the application as a whole when using this package in a shared execution context, or in a virtual machine.

FUZZING

TODO

AVAILABLE IN

The most recent release of this package is available by default in the `alpine-tcl` container image: docker.io/cyanogilvie/alpine-tcl and the `cftcl` Tcl runtime snap: https://github.com/cyanogilvie/cftcl.

SEE ALSO

This package is built on the s2n library and the aws-lc library.

PROJECT STATUS

This is a very early work in progress.

With the nature of this package a lot of care is taken with memory handling and test coverage. There are no known memory leaks or errors, and the package is routinely tested by running its test suite (which aims at full coverage) through valgrind. The `make valgrind`, `make test` and `make coverage` build targets support these goals.

SOURCE CODE

This package's source code is available at https://github.com/cyanogilvie/tcl-s2n. Please create issues there for any bugs discovered.

LICENSE

This package is placed in the public domain: the author disclaims copyright and liability to the extent allowed by law. For those jurisdictions that limit an author's ability to disclaim copyright this package can be used under the terms of the CC0, BSD, or MIT licenses. No attribution, permission or fees are required to use this for whatever you like, commercial or otherwise, though I would urge its users to do good and not evil to the world.

The s2n and aws-lc submodules are not public domain and have their own licenses, consult the LICENSE files in each project for the details.