libosdp Test Programs

Table of Contents

Program Structure

t.b.d.

Configuration

Roles Summary

the OSDP role (CP or PD) is based on the config file value. In the networking cases either the PD or the CP can be listening for a network connection from the other end.

CP Configuration

run in /opt/open-osdp/run/CP. run the program /opt/open-osdp/bin/osdp-net-client. it writes to open_ospd.log, it reads it's config from open-osdp-params.json. There has to be something listening at the destination address or else the program fails.

in /opt/open-osdp/run/CP
  file open-osdp-params.json is the config file
  set it to be the CP
  set the verbosity
  set the fqdn to the CN field of the other end's certificate.
  if it's the client set the ip address of the server

An example configuration is in the doc directory as `open-osdp-params-CP.json`

PD configuration

run in /opt/open-osdp/run/PD. run the program /opt/open-osdp/bin/osdp-net-server. it writes to open_ospd.log, it reads it's config from open-osdp-params.json.

  set it to be the PD
  set the verbosity
  set the fqdn to the CN field of the other end's certificate.
  if it's the client set the ip address of the server

set the test card data

An example configuration is in the doc directory as `open-osdp-params-PD.json`

Configuration Parameters

These are specified in open-osdp-param.json.

address - PD address to use or to talk to. must be valid OSDP address value. Value is in hext.

bits - number of bits in RAW response. Value is in decimal.

disable_checking - disable certificate checking ("1") or not ("0")

fqdn - DN field of peeir's certificate if cert checking is turned on.

init_command - command to initialize serial device.

network_address - ipv4 address to use to connect (for TLS or TCP client.)

poll -- poll frequency (CP polling the PD.)

raw_value = value is hex, this is the card data. Note for 26 bit it is left justified (bottom 6 bits of last octet are not used.)

role - CP PD or MON

serial_device -- name of serial device for RS-485. Typically /dev/ttyUSB0

slow_timer -- in TLS (or TCP) causes the CP to wake up if PD traffic arrives before the poll interval.

timeout

verbosity -- logging verbosity. 1-3 are normal, 9 is loud, >9 is very loud.

Use with TLS

osdp-net-server and osdp-net-client are the TLS server and client respectively. Each one runs the open-osdp code. Configuration details are read from open-osdp-params.json.

Certificates go in /opt/open-osdp/etc. For the server side it uses ca_keys.pem, key.pem, cert.pem. the key is not encrypted. For the clietn side it uses ca_keys.pem, client_key.pem, clietn_cert.pem. The key is not encrypted.

Certificates

Both sides need certificates and keys. Each side has to be configured to trust the root that issued the other end's cert. For the test program, the Common Name field of the Subject Name must be set in the config file as GnuTLS checks this.

Building

"make release" makes a file structure starting at opt, built to be loaded from the root. It is assumed

you would extract the tar file as appropriate. It's intended to not overwrite existing configuration files e.g. in run/CP and run/PD or certificates in etc.

Directory Structure

The directory structure is

```
/opt/open-osdp/bin
/opt/open-osdp/etc
/opt/open-osdp/run/CP
/opt/open-osdp/run/PD
```

bin - programs osdp-net-client, osdp-net-server
etc - certificate files ca_keys.pem (for root key), key.pem (private key for server), cert.pem (certificate for server), client_key.pem (private key for client), client_cert.pem (cert for client.)
run/CP - directory where CP is run
run/PD - directory where PD is run
run/MON - directory where Monitoring is run

Demo VM

See separate documentation. This is a Debian 7 vm running the code out of /opt/open-ospd. with a web server on port 80 running a UI.

Appendix

b. doc fragments

from readme.txt

```
there's a "json" file, see doc directory for sample.

commands are:
  dump_status
  send_poll

request pd ident
rquest capabilities
request local status report
request reader status
```