Developing update applications with SUIT

IETF112 Hackathon

Brendan Moran

Overview

- Components
  - SUIT Manifest Generator: builds SUIT manifests
  - SUITloader: Bootloader that verifies and executes manifests using the SUIT Parser
  - SUIT Parser: Uses a minimal CBOR parser to execute SUIT manifests
  - Build scripts: Builds a bootable image with the SUITloader, and two applications at different offsets
  - Application: simply blinky program with
- Warning! The SUITloader is targeted at NRF52840, may require some porting effort for other platforms.

Getting Started

- Prerequisites
  - Python 3
  - arm-none-eabi-gcc
- First Steps
  - $ git clone https://github.com/ARMmbed/suit-manifest-generator.git
  - $ cd suit-manifest-generator
  - $ python3 -m pip install --user --upgrade .
  - $ pip3 install mbed-cli
- More Details:
  - https://github.com/ARMmbed/suit-manifest-generator/blob/master/README.md
  - https://github.com/ARMmbed/suit-manifest-generator/blob/master/parser_examples/README.md

SUIT Manifest Generator

- Builds SUIT manifests
- Two commands used to build a manifest:
  - Create
    - Uses a JSON input file to construct a manifest
    - suit-tool create -i input.json -o output.mfst
  - Sign
    - Signs an existing manifest with the supplied private key
    - suit-tool sign -i input.mfst -o output.mfst -k private.key

SUITloader

- Simple bootloader
  - Cryptographic functions
    - ECDSA
    - SHA256
  - UART for debug output
  - SUIT Parser

SUIT Parser

- Processes a SUIT manifest by:
  - Checking basic metadata:
    - Sequence number
    - Manifest version
  - Processing the common section
    - Noting the offset of each part of the common section
  - Processing each command sequence by:
    - Processing each command in the common sequence
    - Processing each command in the current command sequence

Build Scripts

- Create a bootable image with two copies of the application:
  - Builds the suitloader
  - Prepares Slot A:
    - Builds the application for slot A
    - Build a manifest for the application in slot A
    - Signs the manifest for the application in slot A
  - Prepares Slot B:
    - Builds the application for slot B
    - Build a manifest for the application in slot B
    - Signs the manifest for the application in slot B
  - Uses srec-cat to pack suitloader, both applications and both manifests into one binary image
- For more information: https://github.com/ARMmbed/suit-manifest-generator/blob/master/parser_examples/README.md

Adding update support

- The application doesn’t support updates.
- First add the SUIT parser to the application
- Next add a way to get a manifest
- Then, pass the manifest to suit_do_process_manifest
- Finally, replace Line 710 of suit_parser.c:
  - -CBOR_KPARSE_ELEMENT(SUIT_DIRECTIVE_FETCH, CBOR_TYPE_UINT, NULL, "Fetch"),
  - +CBOR_KPARSE_ELEMENT_H(SUIT_DIRECTIVE_FETCH, CBOR_TYPE_UINT, fetch_handler"Fetch"),

Writing a fetch handler

```
PARSE_HANDLER(fetch_handler)
{
  uint64_t image_size;
  size_t component_index = 0;
  suit_reference_t *sz;
  int rc = key_to_reference(SUIT_PARAMETER_IMAGE_SIZE, &sz, ctx);
  const uint8_t *np = sz->ptr;
  rc = rc ? rc : cbor_get_uint64(&np, sz->end, &image_size);
  const uint8_t *image;
  rc = rc ? rc : suit_platform_get_image_ref(NULL, &image);
  suit_reference_t *uri;
  rc = rc ? rc : key_to_reference(SUIT_PARAMETER_IMAGE_URI, &uri, ctx);
  // rc = rc ? : fetch_from_uri(image, image_size, uri->ptr, uri->end);
  return rc;
}
```

Additional porting work:

- Some functions in suit_bootloader.c will need porting.

Known issues

- Not up to date with latest manifest specification
  - Missing support for manifest_envelope tag
  - Missing support for SUIT_digest in SUIT_Authentication