DragonBoot Interactive & Secure Bootloader

What is it ?

DragonBoot is a secure bootloader with a user-friendly command-line interface developed for demonstrating :

- How a bootloader could be developed for a barebones microcontroller.
- How security measures could be adopted to ensure that applications can't be reverse-engineered and replaced on the board.
- How a shell-like interface could be used for testing peripherals / performing a peek-and-poke of memory contents as a debugging aid while developing applications. This project was developed on the TI TM4C123GH6PM microcontroller for demonstration purposes only.

Project Contents

This project contains files required for building the DragonBoot bootloader for the TI TM4C123GH6PM board. It also contains test applications which can be downloaded onto the board via DragonBoot as well as an encryption tool for encrypting & digitally signing an application binary before it can be flashed onto the board.

Folders description

- **bootloader** - Contains all modules & build system necessary to cross-compile the DragonBoot bootloader for the target microcontroller.
- **docs** - Houses useful documentation for the project.
- **cryptograf** - Tool which takes an application binary cross-compiled for the microcontroller & performing encryption as well digital signature operations to make it fit for download to the target device via DragonBoot.
- **bearssl** - Source code of the original BearSSL software modified for use with DragonBoot.
- **testapp** - Test application build-tree which can be used for developing applications that would be launched via DragonBoot through the interactive console.

Pre-requisite packages to be installed

1. Install the ARM (arm-none-eabi) toolchain :

```
sudo apt-get install gcc-arm-none-eabi
```

Ensure that the toolchain is appended to the PATH variable to ensure that you can invoke the tools from a shell without prefixing the full pathname for the same.

2. Install the lm4flash utilities from Github :

```
sudo apt-get install libusb-1.0-0-dev
git clone https://github.com/utzig/lm4tools.git
cd lm4tools/lm4flash
make
```

Add the built lm4flash tool's directory to the PATH environment variable / copy the binary to /usr/bin to ensure that it can be invoked with its name from a shell without prefixing the full pathname.

3. Install minicom or equivalent terminal emulator (please ensure that it has support for file transfer via XMODEM protocol)

```
sudo apt-get install minicom
```

Build instructions

1. To compile the bootloader, navigate to the parent directory of the project and run the below commands :

```
cd bootloader/;
make;
```

2. To flash the built bootloader onto a Tiva TM4C123GH6PM board, continue from same directory as above :

```
make flash
```

3. To compile the "cryptograf" encryption tool, navigate to the parent directory of the project and run the below commands :

```
cd cryptograf;
make;
```

4. To compile the test application, navigate to the parent directory of the project and run the below commands :

```
cd testapp;
make;
```

Testing instructions

0. Fire up Minicom & configure the serial port settings to the following :

- Interface name : /dev/ttyACM0 for the TM4c123GH6PM board
- 115200bps,
- 8-bits data,
- No parity,
- No hardware flow control.

1. In order to test software download via DragonBoot, we must first encrypt & digitally sign the test application :

```
cp testapp/build/main.bin cryptograf;
cd cryptograf;
./cryptograf main.bin;
cp image.bin ~;
```

The above steps copy the built test application to cryptograf's directory, and invoke the tool to generate a new binary "image.bin"; which is a secured version of the original testapp. This secured image is copied to the user's home directory for ease of locating the same for upload to DragonBoot.

2. Next, boot the microcontroller and run the below command on the shell :

```
flash flash0.app0
```

You'll be greeted with the following message : "Alright! Kindly use an XMODEM file transfer utility on your terminal to send a new application image. Please ensure that the packet size is set to 128 bytes. Shall wait till file transfer is initiated..."

3. Next, press Ctrl+A followed by Z, then Shift+S to initiate a file transfer from minicom. Select the "xmodem" option and key-in the name of the secure application image that we'd created earlier ("image.bin").

4. Minicom should have started file transfer while XMODEM, wait until it finishes. Ensure that no keyboard keys are pressed during file upload, since the tool is still sensitive to key presses (which might interfere with the UART file transfer).

5. Once the file transfer is completed, you'll get a message like below :

"Transfer complete. READY : press any key to continue"

Wait for another 3-5s and then press the "Enter" / "Return" key.

6. Next, run the following command to boot the application that you'd just downloaded :

```
boot flash0.app0
```

You should see your application run (hopefully!) and see LED(s) blink if the default test app has been used.

7. To exit the application, you'll have to hard-reset the microcontroller to reboot the same (and greet you with the DragonBoot console once again!).

8. Feel free to play around with other commands implemented on DragonBoot, you can get a full list of the same along with their syntax by simply typing the below command on the DragonBoot console :

```
help
```

Also, you can use the TAB auto-complete feature to help you with the syntax of the commands implemented for the console.

For additional details, take a look at the document "docs/DragonBoot_UserGuide.pdf".

Cheers, have fun !