Wrapper Python

Project Documentation

Release: August 30th, 2019

Proprietary Notice

The following document offers information, which is subject to the terms and conditions described hereafter.
While care has been taken in preparing this document, some typographical errors, error or omissions may have occurred. We reserve the right to make changes to the content and information described herein or update such information at any time without notice. The opinions expressed are in good faith and while every care has been taken in preparing this document, some typographical errors, error or omissions may have occurred. We reserve the right to make changes to the content and information described herein or update such information at any time without notice. The opinion expressed are in good faith and while every care has been taken in preparing this document.

Authors

Nicoló MAUNERO *(CINI Cybersecurity Natinal Lab)* nicolo.maunero@consorzio-cini.it
Paolo PRINETTO *(President, CINI)* paolo.prinetto@polito.it
Antonio VARRIALE *(Managing Director, Blu5 Labs Ltd)* av@blu5labs.eu

Trademarks

Disclaimer

Contents

1 Features

This project is intended to present python wrapper for the **SE***cube*™ Host-Side API. You can find information on what the Host-Side API provides in terms of functionalities and how they are structured in the general wiki that you can find on the **SE***cube*™ website[1]

[1] https://www.secube.eu/resources/open-sources-sdk/

2 Wrapper Python Installation Guide

In order to install and start using the python wrappers download the **SE*cube*™** SDK from the **SE*cube*™** website[2], then perform the following steps:

```
cd SEcube-py/SEcube
make clean
make
make install
cd bin
sudo cp secube-x86.so /usr/lib
```

Now everything is read for running the provided example and starting developing your projects. In order to modify the source code which is used by the python wrapper perform the following steps:

```
cd SEcube-py/SEcube/secube-wrapper
Modify the file you want
cd ..
make clean
make
make install
cd bin
sudo cp secube-x86.so /usr/lib
```

Now your modification are applied.

3 Examples

Here a list of provided example and how to use them.

3.1 SE*cube*™ Initialization

This python script provides an example of how to initialize the **SE*cube*™** after it has been programmed. Remember that the initialization has to be done the very first time and every time a full flash erase is performed on the **SE*cube*™** DevKit, if you just re-program it without a full flash erase performing again the initialization is not required.

```
cd SEcube-py/SEcube/py
python3 secube_init.py
```

3.2 Add Key

This python script provides an example of how to add new keys to the **SE*cube*™** . The script receive as input the key to be added and it has to be in exadecimal format (you can use this website[3] to convert a string).

```
Encryption key (AESWSSBoella201820192020)
cd  SEcube-py/SEcube/py
sudo python3 add_key.py 0 Encryption_key_1 414553575353426
   f656c6c6132303138323031393230323 0
```

[2] https://www.secube.eu/resources/open-sources-sdk/
[3] https://codebeautify.org/string-hex-converter

```
Disgest key (HMACSHAWSSBoella2018SHAWSSBoella)
sudo python3 add_key.py 1 Digest_key_1 484
   d41435348415753534 26f656c6c6132303138534841575353426f656c6c61
```

3.3 Functional Test

This script performs some operations such as logging-in, encrypt and decrypt a string, etc. in order to test if the **SE*cube*™** has been initialized correctly.

```
cd SEcube-py/SEcube/py
python3 provaEncDec.py
```