X-Ware IoT Platform SC Security Target

Version: 2.0
Date: 2018-08-22
Express Logic

Table of Contents

List of Tables

1 Security Target introduction

The ST describes what is evaluated, including the exact security properties of the TOE in a manner that the potential consumer can rely on.

1.1 Security Target reference

- Title: X-Ware IoT Platform SC Security Target
- Version: 2.0
- Date: 2018-08-22
- Author: Express Logic

1.2 TOE reference

- TOE name: X-Ware IoT Platform SC
- TOE version: 5.11
- Developer name: Express Logic

X-Ware IoT Platform SC consists of the following items:

- ThreadX version 5.8sp2: industrial grade IoT RTOS
- NetX Duo version 5.11: advanced dual IPv4 & IPv6 TCP/IP network stack
- NetX Secure TLS version 5.11sp1: TLS cryptographic protocol on top of the NetX Duo network stack
- MQTT version 5.11: Message Queuing Telemetry Transport messaging protocol on top of the NetX Secure TLS TLS implementation

1.3 TOE overview

The TOE consists of an embedded RTOS, an IPv4/IPv6 network stack, a TLS implementation and a MQTT implementation.

The TOE is intended to be used by an integrator that deploys it into an embedded hardware together with their own IoT application in order to provide a full client IoT solution with focus on performance and security.

The main security features of the TOE are as follows:

- Provides a secure IPv4/IPv6 network stack
- Offers a secure TLS versions 1.1 and 1.2 implementation to the integrator

- Offers a secure MQTT over TLS implementation to the integrator

The TOE type is a RTOS for IoT applications.

In order to operate, the TOE requires a general-purpose IC with at least one network interface. The TOE also requires that the integrator implements a device driver layer that provides hardware abstraction.

Since the TOE makes use of cryptography to support some of its functionalities, the device driver layer needs to implement cryptographic operations. Such cryptographic operations can be provided by the underlying IC, provided separately in the form of a software cryptographic library by Express Logic or implemented by the integrator.

In addition to running on a supported architecture with cryptographic support, the TOE needs the following functionalities from the underlying hardware and software:

- An implementation of basic memory management libraries.
- A 10ms resolution timer.

The TOE is capable of testing the underlying cryptography.

The TOE supports the variety of microprocessors including, but not limited to, the following:

- ARM
    - ARM7, ARM9, ARM11
    - Cortex-M, Cortex-R, Cortex-A
    - Cortex-Axx 64-bit
- AndesCore
- Analog Devices
    - Blackfin BF5xx, BF6xx, BF7xx
    - SHARC
- Cadence
    - Xtensa
    - Diamond
- CEVA
    - TeakLite-III
- EnSilica
    - eSi-RISC
- NXP

    - ARM (LPC, i.MX, Kinetis)
    - 68K
    - Coldfire
    - PowerPC
- Imagination
    - MIPS32 4Kx, 24Kx, 34Kx, 1004K
    - microAptiv, interAptiv, proAptiv
    - M-Class
- Intel
    - ARM (Cyclone SOC, Arria 10 SOC)
    - NIOSII
    - x86PM
- Microchip
    - ARM (SAM)
    - AVR32
    - PIC24
    - PIC32
- Microsemi
    - RISC-V
- Renesas
    - ARM (Synergy, RZ)
    - H8/300H
    - RX
    - SH
    - V850
- Silicon Labs
    - EFM32
- ST
    - STM32
- Synopsys
    - ARC 600, 700
    - ARC EM, ARC HS
- Texas Instruments
    - ARM (Tiva-C, Sitara, OMAP)
    - C5xx
    - C6xx
- Xilinx

- ARM (Zynq)
- MicroBlaze
- PowerPC

1.4 TOE description

1.4.1 TOE definition

The TOE is a RTOS for IoT applications.

The TOE is composed of the RTOS for IoT applications source code, delivered as a zip file and the associated guidance documentation, delivered as pdf files:

- NetX Duo MQTT (NetX Duo MQTT) for clients User Guide Revision 5.11
- NetX Duo the high-performance real-time implementation of TCP/IP standards User Guide Revision 5.11
- NetX Secure User Guide Revision 5.11SP1
- ThreadX the high-performance embedded kernel User Guide Revision 5.8
- X-Ware IoT Platform SC Security Guidance 2018-08-22

The TOE and documentation is delivered to the customer via sharefile web site download.

The TOE does not include any physical component.

The TOE scope is depicted in Figure 1. The blue parts are within the evaluation scope and the gray parts are outside of the evaluated scope. The out of scope logical parts compromise the integrator application and device driver software to abstract the underlying hardware to the TOE.

The integrator uses the security functionality provided by the TOE to develop a secure IoT solution.

Finally, the complete software solution runs in a physical integrated circuit with at least a network interface and any extra functionalities that the integrator might need. The integrated circuit and any extra functionalities are also out of the evaluation scope.

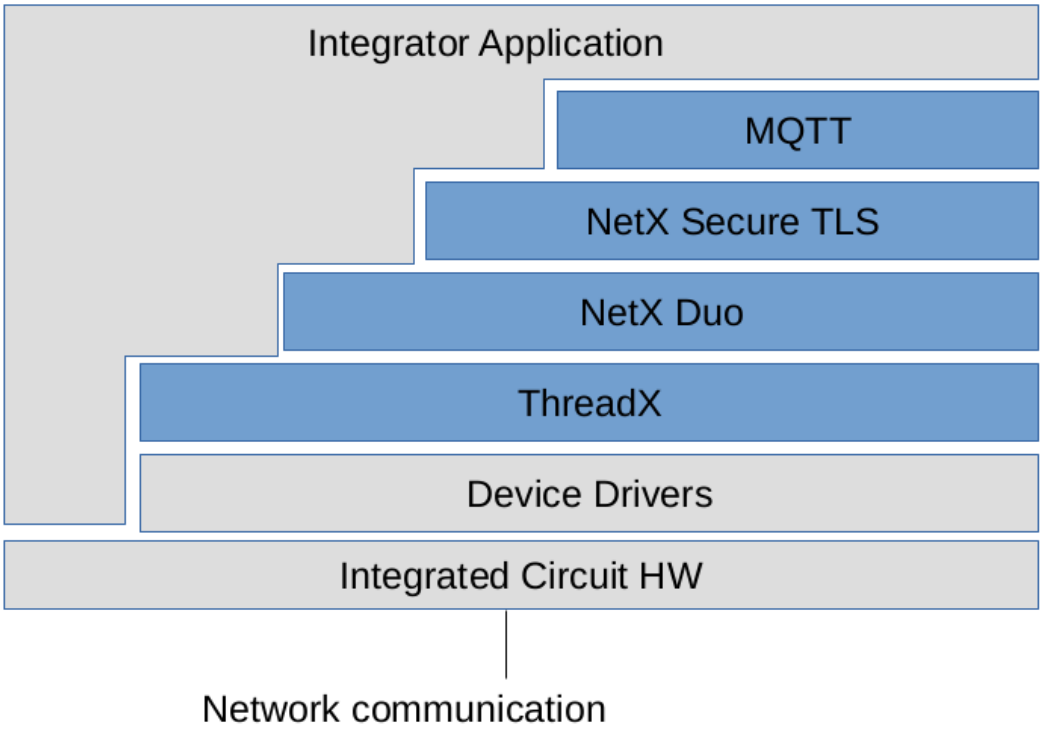

Figure 1 TOE scope

1.4.2 TOE features

THREADX RTOS is *Express Logic's* advanced Industrial Grade Real-Time Operating System (RTOS) designed specifically for deeply embedded, real-time, and IoT applications. THREADX RTOS provides advanced scheduling, communication, synchronization, timer, memory management, and interrupt management facilities. In addition, THREADX RTOS has many advanced features, including its picokernel™ architecture, preemption-threshold™ scheduling, event-chaining,™ execution profiling, performance metrics, and system event tracing. Combined with its superior ease-of-use, THREADX RTOS is the ideal choice for the most demanding of embedded applications. As of 2017, THREADX RTOS has over 6.2 billion deployments, in a wide variety of products, including consumer devices, medical electronics, and industrial control equipment.

NetX Duo is Express Logic's advanced, Industrial Grade dual IPv4 and IPv6 TCP/IP network stack designed specifically for deeply embedded, real-time, and IoT applications. NETX DUO provides embedded applications with core network protocols such as IPv4, IPv6, TCP and UDP as well as a complete suite of additional, higher level add-on protocols.

NetX Secure TLS provides TLS cryptographic protocol versions 1.1 and 1.2 on top of the NetX Duo network stack with legacy support for legacy SSL and TLS versions.

MQTT provides a Message Queuing Telemetry Transport messaging protocol on top of the NetX Secure TLS TLS implementation

All Express Logic's run-time solutions are designed to be small, safe, secure and fast while providing the advanced functionalities needed for deeply embedded applications in an easy and intuitive integration.

2 Conformance claims

2.1 Common Criteria conformance claim

This ST and the TOE claim conformance to Common Criteria (CC) version 3.1, Revision 5, dated April 2017. The claim is CC Part 2 conformant and CC Part 3 conformant conformance.

2.2 Protection Profile claim

This ST does not claim conformance to any PP.

2.3 Package claim

This Security Target claims conformance to Evaluation Assurance Level (EAL) 4 augmented with ALC_FLR.1 Basic flaw remediation.

3 Security problem definition

The security problem definition defines the security problem that is to be addressed in terms of threats, organization security policies and assumptions.

3.1 Threats

T.MITM: An attacker might eavesdrop or tamper with the security sensitive network communication between the TOE and another trusted IT product.

3.2 Organization security policies

P.TRUSTED_PLATFORM: The underlying platform will run in a trusted environment, out of an attacker's physical reach or will be tamper and side channel resistant in a way that is capable of sustaining the TOE's functionality in its operating environment.

P.MQTTTLS: The TOE must provide a secure MQTT implementation that is available over TLS.

P.UNDERLYING_CRYPTO: The integrator will use cryptographic services provided by means of the underlying Device Drivers' layer that fulfill [RFC3447][FIPS197][SP80067][FIPS1804][FIPS1981]. The underlying cryptography will be resistant to timing attacks when operating with secret or private keys.

P.INTEGRATOR_SECRETS: The integrator secrets confidentiality and integrity, including keys, will be preserved by the integrator's application while on integration application's memory space. The keys and temporary secrets that are allocated in memory by the TOE during a secure session with another trusted IT product must be cleared from the TOE's memory space when the session is terminated.

3.3 Assumptions

A.TRUSTED_INTEGRATOR: The integrator is assumed to be competent and will use the security functionalities needed by the complete IoT solution following the TOE guidance documentation, including the usage of secure ciphersuites. The integrator will not attempt to thwart the TOE security functionalities nor bypass them.

4 Security objectives

The security objectives are a concise and abstract statement of the intended solution to the problem defined by the security problem definition. The security objectives show which security concerns are addressed by the TOE, and which are addressed by the environment.

4.1 Security objectives for the TOE

This section identifies and describes the security objectives that are to be addressed by the TOE.

O.TLS: Secure Connection via TLS. The TOE is able to establish, maintain and terminate a secure authenticated connection with a remote server using transport layer security (TLS 1.1 and 1.2) with secure ciphersuites as mentioned in [RFC4346] and [RFC5246]. The TOE must be able to prove its level of trust and identity to the remote server. The TOE must be able to authenticate the remote server.

O.CRYPTO_TEST: The TOE provides a security functionality to test that the underlying device drivers' cryptographic implementation and random number generator are operative.

O.MQTTTLS: The TOE provides a secure MQTT implementation that is available over TLS

O.SECRETS_CLEARING: The TOE clears the keys and temporary secrets that are allocated in memory controlled by the TOE during a secure session with another trusted IT product when the session is terminated.

4.2 Security objectives for the operational environment

This section identifies and describes the security objectives that are to be addressed by the IT domain or by non-technical or procedural means.

OE.TRUSTED_PLATFORM: The underlying platform runs in a trusted environment, out of an attacker's physical reach or is tamper and side channel resistant in a way that is capable of sustaining the TOE's functionality in its operating environment.

OE.UNDERLYING_CRYPTO: The cryptographic services provided by means of the underlying Device Drivers' layer fulfill [RFC3447] [FIPS197] [SP80067] [FIPS1804] [FIPS1981]. The underlying cryptography is resistant to timing attacks when operating with secret or private keys.

OE.INTEGRATOR_SECRETS: The integrator secrets confidentiality and integrity, including keys, is preserved by the integrator's application while on integration application's memory space.

OE.TRUSTED_INTEGRATOR: The integrator uses the security functionalities needed by the complete IoT solution following the TOE guidance documentation, including the usage of secure ciphersuites. The integrator is trusted and does not attempt to thwart the TOE security functionalities nor bypass them.

4.3 Security objectives rationale

	O.TLS	O.CRYPTO_TEST	O.MQTTTLS	O.SECRETS_CLEARING	OE.TRUSTED_PLATFORM	OE.UNDERLYING_CRYPTO	OE.INTEGRATOR_SECRETS	OE.TRUSTED_INTEGRATOR

T.MITM	X							
P.TRUSTED_PLATFORM					X			
P.MQTTTLS	X		X					
P.UNDERLYING_CRYPTO		X				X		
P.INTEGRATOR_SECRETS				X			X	
A.TRUSTED_INTEGRATOR								X

Table 1 Security objectives tracing

O.TLS directly counters **T.MITM** and addresses **P.MQTTTLS** by providing a secure TLS implementation that supports MQTT.

O.CRYPTO_TEST supports **P.UNDERLYING_CRYPTO** by testing the correct functioning of the cryptographic algorithms and RNG.

O.MQTTTLS directly addresses **P.MQTTTLS** by providing an MQTT over TLS implementation**.**

O.SECRETS_CLEARING supports **P.INTEGRATOR_SECRETS** by clearing secrets that are allocated in memory controlled by the TOE.

OE.TRUSTED_PLATFORM directly addresses **P.TRUSTED_PLATFORM**.

OE.UNDERLYING_CRYPTO supports **P.UNDERLYING_CRYPTO** by providing the underlying cryptographic and RNG services.

OE.INTEGRATOR_SECRETS supports **P.INTEGRATOR_SECRETS** by preserving the secrets handled in the integrator's application memory.

OE.TRUSTED_INTEGRATOR directly addresses **A.TRUSTED_INTEGRATOR**.

5 Extended components definition

This ST defines no extended component.

6 Security requirements

6.1 Security functional requirements

This section describes the security functional requirements for the TOE. These requirements are the basis on which the TOE is evaluated.

The operations performed on the SFRs are identified as follows:

- Selection: *chosen selection*

- Assignment: **performed assignment**
- Refinement: Application note: details

FPT_TEE.1 Testing of external entities

FPT_TEE.1.1 The TSF shall run a suite of tests *during initial start-up, **when the integrator software requests it*** to check the fulfillment of **underlying device drivers' cryptographic operations against test vectors and underlying device driver's random number generator providing a non-constant value**.

FPT_TEE.1.2 If the test fails, the TSF shall **notify the integrator software**.

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall provide a communication channel between itself and another trusted IT product that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from modification or disclosure.

FTP_ITC.1.2 The TSF shall permit *the TSF* to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for **transferring and receiving integrator's application user data**.

FPT_RPL.1 Replay detection

FPT_RPL.1.1 The TSF shall detect replay for the following entities: **all remote trusted IT product data within a TLS session**.

FPT_RPL.1.2 The TSF shall perform **notification to the integrator software and termination of the TLS secure channel** when replay is detected.

Application note: All integrator's data that goes through a TLS secure channel must be protected from replay attacks including MQTT over TLS data.

FDP_RIP.1 Subset residual information protection

FDP_RIP.1.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the: *deallocation of the resource from* the following objects: **session keys and intermediate decrypted data**.

6.2 Security assurance requirements

The set of security assurance requirements are those of EAL4 augmented with ALC_FLR.1.

6.3 Security requirements rationale

6.3.1 Security functional requirements rationale

	FPT_TEE.1	FTP_ITC.1	FPT_RPL.1	FDP_RIP.1
O.TLS		X	X	
O.CRYPTO_TEST	X			
O.MQTTTLS		X	X	
O.SECRETS_CLEARING				X

Table 2 Security Functional Requirements tracing

FPT_TEE.1 directly addresses **O.CRYPTO_TEST**.

FTP_ITC.1 addresses the TLS trusted channel of **O.TLS** and **O.MQTTTLS**.

FPT_RPL.1 addresses replay detection for the TLS trusted channel of **O.TLS** and **O.MQTTTLS**.

FDP_RIP.1 directly addresses **O.SECRETS_CLEARING**.

6.3.2 Security assurance requirements rationale

The chosen SARs are the ones of EAL4 + ALC_FLR.1. This set is chosen because it is internally consistent and provides an appropriate level of assurance for a IoT RTOS that will be used by a security-aware integrator.

7 TOE summary specification

7.1 TSF.TLS

The TOE is able to establish, maintain and terminate a secure authenticated connection with a remote server as a client using transport layer security (TLS 1.1 and 1.2) with secure ciphersuites as mentioned in [RFC4346] and [RFC5246]. This is achieved by means of the NetX Secure TLS component

implementation. Additionally the TOE supports MQTT over TLS by using the MQTT component over a TLS channel. This TSF achieves FTP_ITC.1 and FPT_RPL.1.

7.2 TSF.PLATFORM_TEST

The TOE provides the capability to test that the underlying device drivers' cryptographic implementation and random number generator are operative. This is provided by means of an API call that performs a verification against known test vectors and verifies that the RNG is operational. This achieves FPT_TEE.1.

7.3 TSF.TEMPORARY_SECRETS_CLEARING

Session data and keys allocated during a TLS connection and session are cleared from memory once the session is finished or the intermediate data processing is completed. This achieves FDP_RIP.1.

	TSF.TLS	TSF.PLATFORM_TEST	TSF.TEMPORARY_SECRETS_CLEARING
FPT_TEE.1		X	
FTP_ITC.1	X		
FPT_RPL.1	X		
FDP_RIP.1			X

Table 3 TOE Security Functionality tracing

8 Glossary of terms

ST: Security Target

TOE: Target Of Evaluation

IoT: Internet of Things

RTOS: Real Time Operating System

TLS: Transport Layer Security cryptographic protocol

MQTT: Message Queuing Telemetry Transport messaging protocol**.**

CC: Common Criteria

PP: Protection Profile

EAL: Evaluation Assurance Level

SFR: Security Functional Requirement

RNG: Random Number Generator

9 References

[RFC4346] RFC 4346 The Transport Layer Security (TLS) Protocol Version 1.1
[RFC5246] RFC 5246 The Transport Layer Security (TLS) Protocol Version 1.2
[RFC5280] RFC 5280 Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile
[RFC5746] RFC 5746 Transport Layer Security (TLS) Renegotiation Indication Extension
[RFC4279] RFC 4279 PSK ciphersuites for TLS
[RFC6066] RFC 6066 TLS Extensions
[RFC3447] RFC 3447 Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1
[RFC5652] RFC 5652 Cryptographic Message Syntax (CMS)
[FIPS197] FIPS 197 Announcing the ADVANCED ENCRYPTION STANDARD
[SP80067] SP 800-67r1 Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher
[FIPS1804] FIPS 180-4 Secure Hash Standard (SHS)
[FIPS1981] FIPS 198-1 The Keyed-Hash Message Authentication Code (HMAC)
[RFC8017] RFC 8017 PKCS#1 V2.2