SI1 - Security for EFR32MG2x Wireless SoCs

TrustZone, Secure Vault, Secure Boot and OTA security on Silicon Labs Series 2

Objectives

- Understand the ARMv8-M TrustZone architecture as implemented on the Cortex-M33 core of the EFR32MG2x Series 2 devices
- Configure Secure and Non-Secure memory and peripheral partitioning
- Understand the Silicon Labs Secure Engine (HSE/VSE)
- Configure and enable Secure Boot with RTSL (Root of Trust and Secure Loader) using the Gecko Bootloader on Series 2 devices
- Understand OTP provisioning
- Understand anti-rollback protection and firmware signing with ECDSA-P256

Course environment

- Target: EFR32MG2x Series 2 board
- OS context: bare-metal / Micrium, Zephyr paths mentioned as reference only

Target audience

- Embedded firmware developers and technical leads with a working knowledge of C/C++.

Prerequisites

- Working knowledge of C programming (functions, pointers, memory management)
- Basic familiarity with embedded systems concepts (MCU, firmware, cross-compilation)
- No prior security expertise required

Course Outline

First Day

EFR32MG2x Security Architecture Overview

- EFR32MG21/MG24 core
- Secure Engine (SE)
- Security building blocks on Series 2
- Secure Vault
- Series 2 SE Firmware
- Cyber Resilience ACT Mapping

ARMv8-M TrustZone Architecture

- TrustZone for ARMv8-M
- Operation states and modes
- Register banking between security states
- Memory model in TrustZone
- System Private Peripheral Bus (PPB)
- Secure and Non-Secure MPU

- Exception handling and the Security Extension

Exercise:
- MPU
- Secure App

Security Attribution Unit and SMU

- SAU (Security Attribution Unit)
- IDAU / ESAU
- SMU (Security Management Unit)
- Memory partitioning on EFR32MG2x
- Peripheral security attribution
- System Security Controller and wrapper components
- Debug access and TrustZone

Exercise: Configure SAU and SMU

Secure and Non-Secure Software Design

- Two-image project structure
- NSC (Non-Secure Callable) veneer functions
- Calling conventions across the security boundary
- Secure world services
- TF-M (Trusted Firmware-M)

Exercise: Implement a minimal Secure monitor

Secure Boot Fundamentals and the RTSL Architecture

- Purpose of Secure Boot
- Root of Trust and Secure Loader (RTSL)
- Boot chain on Series 2 HSE/VSE devices
- Signature algorithm
- Certificate-based Secure Boot

Second Day

OTP

- OTP memory on Series 2
- Public Sign Key
- Secure Boot Enable flag
- Anti-Rollback Enable flag
- Flash page lock settings
- SE OTP provisioning workflow
- Custom Part Manufacturing Service (CPMS)

Exercise: Read current SE OTP configuration with sl_se_read_otp()

Gecko Bootloader Configuration for Secure Boot

- Gecko Bootloader architecture
- GBL (Gecko Bootloader image format)
- Enabling Secure Boot in the SSB
- Generating the ECDSA key pair
- Signing an application image
- Signing a GBL upgrade file
- Bootloader version and anti-rollback
- Upgrading a bootloader without Secure Boot to a bootloader with Secure Boot

Exercise: Build a Gecko Bootloader with Secure Boot

Exercise: Anti-rollback

Secure Debug Lockdown

- Debug access port
- Three debug lock properties
- Standard debug lock
- Secure debug unlock
- TrustZone debug authentication
- Production lockdown checklist

Firmware Update and OTA Security

- Secure OTA pipeline
- GBL file integrity
- OTA path vs Gecko Bootloader integration
- Staged slots
- Encrypted firmware images
- Vulnerability Management (CRA)