SEC3 - wolfSSL for Embedded Security

Objectives

- Understand how SSL/TLS Works
- Establish fundamental knowledge about cryptographic, algorithms, and protocols.
- Learn how to implement secure authentication with wolfSSL
- Learn how to effectively configure and Compile wolfSSL for target platforms (NXP, STMicro, Xilinx SoCs, Ti, …)
- Learn effective wolfSSL debugging strategies
- Add wolfSSL to ANSI-C based client and server applications
- Understand how to use wolfSSL's cryptography library (wolfCrypt)
- Learn how to use a TPM to authenticate hardware devices (wolfTPM)

Prerequisites

- C programming
- Experience with embedded systems development.
- Some security concepts are desirable (see our training OSEC1 and OSEC2)

Duration

- Total: 18 hours
- 3 sessions, 6 hours each (excluding break time)
- From 40% to 50% of training time is devoted to practical activities
- Some Labs may be completed between sessions and are checked by the trainer on the next session

Course Environment

- Theoretical course
  - Printed and PDF material in English.
  - Trainer assistance throughout.
- Practical activities (40-50% of duration)
  - Code examples, exercises and solutions.
  - One PC (one per two beyond six trainees) with target board if needed; install manual provided on-site.
- Downloadable preconfigured VM to redo the labs afterwards.
- Each session starts with a trainee check-in.

Target Audience

- Any embedded systems engineer or technician with the above prerequisites.

Course Outline

First Session

Introduction to wolfSSL

- Introduction to embedded security
  - Embedded Security Trends
  - Security policies
- Secure Embedded System Hardware/Software Architecture Overview
- Securing legacy Systems
- Cryptography Overview
- wolfSSL Products and Library overview

wolfSSL embedded SSL/TLS library (1st part)

- Building wolfSSL
- Features
- Portability
- Callbacks
- Keys and Certificates
- Library Design
- SSL/TLS History and Protocol
- WolfSSL Basic Library usage

Exercise: wolfSSL TLS integration
Exercise: SSL/TLS Tutorial
Exercise: wolfSSL Examples

Second Session

wolfSSL embedded SSL/TLS library (2nd part)

- Debugging
- wolfSSL TLS usage
- wolfSSL DTLS Usage
- wolfSSL PSK Usage
- wolfSSL Session Resumption
- wolfSSL with Non-Blocking I/O
- wolfSSL and TLS 1.3

Exercise: Wireshark
Exercise: Convert TCP/IP Client and Server to TLS
Exercise: Extracting Certificate Fields via API
Exercise: Convert simple UDP Client and Server to DTLS
Exercise: Convert simple TCP client and Server to PSK
Exercise: Session Resumption Client
Exercise: Write a Non-Blocking Client and Server
Exercise: TLS 1.3 Client and Server
Exercise: TLS 1.3 Early Data

wolfCrypt (1st part)

- PRNG(Pseudo-Random Number Generator) and RNG(Random Number Generation)
- HASH Functions
- Block Ciphers

    - AES
    - DES and 3DES
    - Camellia
- Stream Ciphers
    - ARC4
    - RABBIT
    - HC-128
    - ChaCha

Exercise: PRNG and RNG
Exercise: Creating a Hash of a File
Exercise: Block Ciphers
Exercise: Block Ciphers

Third Session

wolfCrypt (2nd part)

- Public Key Cryptography
    - RSA
- PKCS Public Key Cryptography Standards
    - PKCS#7 and RFC 3369 : Cryptographic Message Syntax (CMS)
- Cryptographic Certification
    - X.509 Certificates
- Key and Certificate genration

Exercise: Sign and Verify data with ECC
Exercise: Sign and Verify data with Ed25519
Exercise: Key Agreement with ECDH and Curve25519
Exercise: PKCS#7 and CMS bundle Generation and Verification
Exercise: WolfCrypt Certificate Manager
Exercise: Creating Keys and Certificates

WolfTPM

- Overview of TPM Architecture
- The Root-of-Trust
- Key Hierarchy and Key Management
- TPM command Message Overview
- Command Authorization (Typical/Atypical)
- TPM Signature Command
- TPM Capability and Self-Test
- Building WolfTPM
- wolfTPM Library Design

Exercise: Building and Testing wolfTPM