[redacted]

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
  - PDF course material (in English) supplemented by a printed version for face-to-face courses.
  - Online courses are dispensed using the Teams video-conferencing system.
  - The trainer answers trainees' questions during the training and provide technical and pedagogical assistance.
- Practical activities
  - Practical activities represent from 40% to 50% of course duration.
  - Code examples, exercises and solutions
  - For remote trainings:
    - One Online Linux PC per trainee for the practical activities.
    - The trainer has access to trainees' Online PCs for technical and pedagogical assistance.
    - QEMU Emulated board or physical board connected to the online PC (depending on the course).
    - Some Labs may be completed between sessions and are checked by the trainer on the next session.
  - For face-to-face trainings:
    - One PC (Linux ou Windows) for the practical activities with, if appropriate, a target board.
    - One PC for two trainees when there are more than 6 trainees.
  - For onsite trainings:
    - An installation and test manual is provided to allow preinstallation of the needed software.
    - The trainer come with target boards if needed during the practical activities (and bring them back at the end of the course).
- Downloadable preconfigured virtual machine for post-course practical activities
- At the start of each session the trainer will interact with the trainees to ensure the course fits their expectations and correct if needed

Target Audience

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

Evaluation modalities

- The prerequisites indicated above are assessed before the training by the technical supervision of the traineein his company, or by the trainee himself in the exceptional case of an individual trainee.
- Trainee progress is assessed in two different ways, depending on the course:
  - For courses lending themselves to practical exercises, the results of the exercises are checked by the trainer while, if necessary, helping trainees to carry them out by providing additional details.
  - Quizzes are offered at the end of sections that do not include practical exercises to verifythat the trainees have assimilated the points presented
- At the end of the training, each trainee receives a certificate attesting that they have successfully completed the course.
  - In the event of a problem, discovered during the course, due to a lack of prerequisites by the trainee a different or additional training is offered to them, generally to reinforce their prerequisites,in agreement with their company manager if applicable.

Plan

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*

Renseignements pratiques

Inquiry : 18 hours