escrypt
SECURITY. TRUST. SUCCESS.

Security trainings

Secure product design

Training introduction

Attacks on connected systems have become considerably more frequent over recent years – affecting products such as cryptographic libraries, back-end infrastructures, and brand-new vehicles – and have been discussed both in the mass media as well as in the relevant professional journals. The reasons for this increase in attacks may vary, but manufacturers' lack of security awareness and, in this context, their failure to act to protect products and product development processes stand out. The only way to proactively adapt to constantly evolving threats is to implement and deploy suitable security tools and processes.

IT security must be the bedrock for every reliable product development cycle. Without proper security measures, safety gaps can expose companies, their employees, and their customers to significant risks. The earlier that IT security is taken into account, the easier it is to incorporate reasonable measures and to avoid weaknesses and vulnerabilities. This two-day IT security basics training by ESCRYPT is aimed at managers in charge of security, products, and projects as well as system engineers, software engineers, and developers. The training covers organizational as well as technical aspects: participants will learn the basics of IT security and technical cryptography, understand the importance and structure of a secure development process, and gain valuable insights into measures to establish security processes and to harden architectures and implementations.

ESCRYPT 2018/07

escrypt
SECURITY. TRUST. SUCCESS.

Training topics

- Get to know different aspects of security (e.g., theory vs. practice, challenges)
- Learn and understand security basics (e.g., basic terminology)
- Find out how to set up a secure software development lifecycle
- Establish fundamental knowledge about cryptographic tools, algorithms, and protocols
- Understand important aspects of access control (authentication and authorization)
- Learn to apply main security principles
- Secure coding module option: comprehend secure coding techniques
- Risk analysis module option: learn how to apply a risk-based approach towards security (e.g., economic security)

Target group

- Product/project managers who need to establish a solid understanding about general security principles, processes and tools that are necessary for secure product design
- System engineers/architects who are responsible for developing and analyzing security requirements and for defining security concepts

Requirements

- Basic IT knowledge and experience
- Basic computer programming skills
- No security background is required

Duration: 2 days

Languages: German/English

Participants limit: 15

Location: ESCRYPT site/customer site/conference hotel

Course outline

Day 1:

- Security basics
  - Discussion of recent IT security threats
  - Different aspects of IT security
  - Basic terminology
  - Generic security framework
- Secure software development
  - Economic security
  - Security activities for a software development process
  - Conventional software development
  - Agile software development
- Cryptographic primitives
  - Benefits and limitations of cryptography
  - Fundamental cryptographic principles
  - Symmetric and asymmetric tools & algorithms
  - Cryptographic protocols
  - Application & practical advices

Day 2:

- Access control: authentication
  - Password-based authentication
  - Secure password management
  - Multi-factor authentication
  - Implementation aspects
- Access control: authorization
  - Permission management
  - Access control lists, role-based access control
  - Capabilities
  - Secure session management
- Security principles & concepts
  - Security principles (defense in depth, keep it simple, least privilege)
  - Security concepts (trust boundaries, separation of duties, error & exception handling)
- Option A: secure coding
  - Secure coding in the development process
  - Weaknesses, vulnerabilities & attack references
  - Best practices in defensive coding
  - Secure coding guidelines
- Option B: security risk analysis
  - Security asset & objective identification
  - Threat modeling
  - Risk assessment
  - Security requirements: integration into the development process