ISSECO
SECURE SOFTWARE ENGINEERING

ISSECO
Syllabus
Public Version
v1.0

ISSECO Certified Professional for Secure Software Engineering

Date: October 16th, 2009

This document was produced by the "ISSECO Working Party Syllabus"

Introduction to this Syllabus

Purpose of this Document

This syllabus forms the basis for the Certified Professional for Secure Software Engineering. Training providers will produce courseware and determine appropriate teaching methods for accreditation, and the syllabus will help candidates in their preparation for the examination. In the accreditation process, the training material will be checked against this syllabus.

This version represents a public, short version of the syllabus. The full version is available to accredited training providers.

Certified Professional for Secure Software Engineering

The qualification scheme is aimed at anyone involved in software engineering. This includes people in roles such as security professionals, project managers, software designers, programmers and consultants. Certificate holders will be able to control and design security aspects across the entire software development lifecycle.

Learning Objectives

All terms listed under “Terms” right below the chapter headings shall be understood, even if not explicitly mentioned in the learning objectives.

Cognitive levels are given for each section in this syllabus:

- K1: remember, recognize, recall
- K2: understand, explain, give reasons, compare, classify, summarize
- K3: apply

The Examination

The examination will be based on this syllabus. Answers to examination questions may require the use of material based on more than one section of this syllabus. All sections of the syllabus can be subject to examination. The format of the examination is multiple choice. Exams may be taken as part of an accredited training course or independently (e.g. at an examination center).

Accreditation

Training providers whose course material follows this syllabus may be accredited by an accreditation board. Accreditation guidelines can be obtained from the board that performs the accreditation. An accredited course is recognized as conforming to this syllabus, and is allowed to perform an examination as part of the course.

Level of Detail

The level of detail in this syllabus allows internationally consistent teaching and examination. In order to achieve this goal, the full version of the syllabus consists of:

- General instructional objectives describing the intention of the syllabus
- A list of terms that students must have understood and be able to recall
- A description of the key concepts to teach, including sources such as accepted literature or standards

This syllabus does not contain a description of the entire knowledge in the areas it covers; it merely reflects the level of detail to be covered in training courses.

How this Syllabus is Organized

The syllabus is split up into 16 major chapters. The top-level heading names the covered topic, and specifies the allotted time for the chapter. This time is a guideline for training providers and also reflects the relative importance of the chapter. Sub-areas are specified as part of the major chapter without any time allottments. Each major chapter contains the most important terms of the area covered. These terms are subject of the final examination.

1. Introduction & Motivation 45 mins

Goals

Understand why security is an important topic.

Content

To design and build secure IT systems, all elements of the system need to be secure. The focus on perimeter and infrastructure security is not sufficient in coping with rapidly spreading attacks against the data and information stored and processed by IT systems.

Today, as more and more security breaches are explored due to insecure software systems, secure software applications are urgently needed.

This course gives an insight into the design, development and testing of secure software systems and will answer the following questions:

- Why is secure software urgently needed?
- Why is secure software development often neglected?
- What are the consequences of shipping insecure software to customers?
- Why is security often neglected during software development ?
- How important is the security of software systems to their stakeholders?

This course enables software architects, developers, quality managers, testers, project managers and other software development stakeholders to understand the principles of secure software engineering and to apply them in their company.

2. View of the Attacker 45 mins

Goals

Understand the differences between hacker and cracker, their motivation, skill level and how they gather information.

Terms

Meaning Hacker vs. Cracker, Historical Background, Hardware and Software Knowledge, Skill Level, Mode of Ethical Hacking, Hacker Motive, Reaching Goals, Gathering Information

Content

- Explain the Definition of the Terms Hacker, Cracker and Ethical Hacker (K2)
- Show Examples of Famous Hackers and their Attacks (K1)
- Point Out the Different Skill Levels (K2)
- Explain Different Modes of Hacking (K2)
- Recall Motives of Hackers (K1)
- Explain the Process of Hacking (K2)
- Outline Common Hacking Tools (K2)

3. View of the Customer 45 mins

Goals

Understand customers' expectations regarding software security and be able to classify requirements accordingly.

Terms

Secure Software, Compliance Requirements, C-Level Language, Assets, Threats and Risks, Security Requirements, Confidentiality, Integrity, Availability

Content

- Recall how important the security of software systems is to different stakeholders (K1)
- Explain Why Customers Expect Secure Software (K2)
- Recall What Customers Expect in Terms of Software Security (K1)
- Classify the Customer's Security Requirements (K3)
- Describe Assets, Threats and Corresponding Risks (K2)
- Recall Customer's Compliance Requirements with Regards to Software Security (K1)
- Map Customer-Specific Security Needs to Technical Requirements (K3)
- Help Customers Understand What They Want to Protect (K3)

4. Methodologies 90 mins

Goals

Understand the reasoning and approaches for different secure software development methodologies; apply them in participant's company / project context.

Terms

Secure Software Development Lifecycle (SDLC), Secure Development Lifecycle (SDL), Security Practices, OWASP, ISO 15408, Common Criteria (CC), build-in-security

Content

- Recall the typical elements of a Methodology for Developing Secure Software (K2)
- Describe an Overview of Development Methodologies or Life Cycle Models (K2)
- Explain Security Practices (K2)
- Describe Existing Models (K2)

5. Requirements Engineering 45 mins

Goals

Understand how to develop security requirements for technologies and applications.

Terms

Availability; Authenticity; Confidentiality; Efficiency; Integrity; Maintainability; Portability; Reliability; Requirement; Requirements Engineering; Requirements Management; Trustworthiness; Usability; Asset; Actor

Content

- Recall what is a Requirement (K1)
- Describe How to Define "Good" Requirements (K2)
- Identify Security Requirements (K2)
- Know Requirement Areas Relevant to Security (K1)

6. Threat Modeling 45 mins

Goals

Understand threat modeling.

Terms

Asset, Threat, Attack, Dataflow Diagram (DFD), Threat Tree (Attack Tree), STRIDE, DREAD.

Content

- Recall Reasons for Threat Modeling (K1)
- Execute the Threat Modeling Stages (K3)
- Identify Threats (K3)
- Perform Attack Classification (K3)
- Rate Threats (K3)

7. Secure Design 90 mins

Goals

Understand the principles for secure design; be able to apply them in context.

Terms

Threat Modeling, STRIDE, Security Principles, Guidelines for Secure Software Development, Security Architecture, Software Attack Surface, Secure Software Development Lifecycle (SDLC), Mandatory Access Control (MAC), Discretionary Access Control (DAC), Role-based Access Control (RBAC), Access Matrix

Content

- Recall the role of architecture and design in creating a secure system (K1)
- Explain important Security Design Principles (K2)
- Explain General Concept of Security Design Patterns (K2)
- Recall Some Important Security Patterns (K1)
- Describe the Need for an Overall Security Architecture (K2)
- Recall Access Control Models (K1)
- Explain Trust Models (K2)
- Explain Security Architecture and Design Reviews (K2)

8. Secure Coding 90 mins

Goals

Participants shall understand vulnerabilities in coding, identify, and remediate them.

Terms

Vulnerabilities, Vulnerability Patterns, Secure Coding Practices, Code Checking Tools, Cross Site Scripting, Injection Flaws, Cross Site Request Forgery, Denial of Service.

Content

- Recall vulnerability classification patterns (K1)
- Explain basic vulnerability patterns (K2)
- Explain Secure Coding Practices (K2)
- Assign the most important vulnerabilities to vulnerability patterns and identify the correct remediation (K3)
- Recall Helper Tools for Secure Coding (K1)
- Identify examples of vulnerabilities in code samples (K3)
- Recall online resources for secure coding (K1)

9. Security Testing 90 mins

Goals

Understand how to coordinate and perform security testing, understand the basic principles and how to interpret security testing results.

Terms

Test Cases, Security Test Plan, White Box Test, Black Box Test, Penetration Testing, Code Review, Test Report

Content

- Recall different Cases of Tests (K1)
- Recall the parts of a test case specification (K1)
- Develop a Security Test Plan from Threat Models (K2)
- Explain Hacking Techniques as Test Methods (K2)
- Explain test patterns (K2)
- Explain Code Review (K2)
- Recall Tools for Code Analysis (K1)
- Writing Test Reports (K3)

10. Secure Deployment 45 mins

Goals

Understand why and how secure deployment is important and how this impacts development practices.

Terms

Secure Default Configuration, Product Life Cycle, Automated Deployment Process, Secure Target Environment, Secure Delivery of Code, Trusted Origin, Code Signing, Least Privilege Permissions, ITIL Release and Deployment Management

Content

- Recall productive IT operations and related security activities (K1)
- Explain the Phases that are part of Software Deployment (K2)
- Describe Security Activities in Release Phase (K2)
- Recall Secure Software Configuration in Target Environments (K2)
- Describe Activities during Software Activation (K2)
- Describe activities during adaptation and update (K2)

11. Security Response 45 mins

Goals

Understand and be able to implement a security response process, making sure that security issues in software installations are fixed and communicated responsibly.

Terms

Security Response, Security Bulletins, Vulnerabilities, Security Patches, Disclosure, Full Disclosure, Responsible Disclosure, „Patch Tuesday“, Security Response Policy, Security Response Process, Common Vulnerability Scoring System, CVSS

Content

- Explain the Difficulties of Fixing Security Issues via Standard Maintenance Processes (K2)
- Explain why Security Patches Should be Different from Standard Software Patches (K2)
- Recall the Dangers of Trading Vulnerability Information (K1)
- Describe Success Criteria for Vulnerability Communication (K2)
- Discuss Vulnerability Disclosure Strategies (K2)
- Recall Standards for Vulnerability Description and Their Main Elements (K1)
- Develop Patch and Security Response Policies (K3)
- Define and Implement a Security Response Process (K3)

12. Security Metrics 45 mins

Goals

Understand how to measure software security across the life cycle.

Terms

Measuring Security, Security Metrics, Software Development Lifecycle, Categories of Software Metrics, Quality Criteria of Metrics, Goal–Question–Metrics method (GQM

Content

- Explain the Need for Measuring Security (K2)
- What is a Software Metric? (K2)
- Recall Limitations of Software Metrics (K1)
- Where Can You Measure Security? (K2)
- Categorize Software Metrics (K2)
- Recall Quality Criteria of Metrics (K2)
- Developing Security Metrics (K2)
- Apply Existing Software Metrics to Security (K3)
- Use Examples for Software Security Metrics (K3)

13. Code & Resource Protection 45 mins

Goals

Understand how to protect code against malicious behavior from developers or intruders; “code assurance”, vs. “software assurance”.

Terms

Back Door, Time Bomb, Four-Eyes Principle, Confidentiality Classification, Background Screening, Security Clearance, Offline and Online Licensing Mechanisms, Code Obfuscation

Content

- Recall Reasons for Code and Resource Protection (K1)
- Recall the Technical Types of Risks in the Product (K1)
- Recall the risk of configuration manipulation (K1)
- Explain Basic Protection Measures Against the Malicious Manipulation of Software Code (K2)
- Apply Additional Procedural Security Measures Due to Political Risks (K3)
- Recall technical Security Measures (K1)
- Explain Usage Protection (K2)
- Recall how to protect sensitive pieces of code (K1)
- Explain Code Obfuscation Techniques (K2)