Curriculum for

Certified Professional for Software Architecture (CPSA)®
Advanced Level

Module
WEBSEC

Web Security

Version 2020.1-EN; September 30, 2020

Table of Contents

List of Learning Goals

- LG 5-4: Significance and functioning of denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks
- LG 5-5: Attacks via the runtime environment/application platform
- LG 5-6: Identifying security vulnerabilities via fuzzing
- LG 5-7: Man-in-the-middle attacks
- LG 5-8: Important sources for current threats and attacks
- LG 6-1:Function/operation and processes of firewalls
- LG 6-2: Web Application Firewalls
- LG 6-3: Intrusion Detection / Prevention systems
- LG 6-4: Validation with feedback from operations
- LG 6-5: Use of TLS

Introduction: General information about the iSAQB Advanced Level

What is taught in an Advanced Level module?

- The iSAQB Advanced Level offers modular training in three areas of competence with flexibly designable training paths. It takes individual inclinations and priorities into account.
- The certification is done as an assignment. The assessment and oral exam is conducted by experts appointed by the iSAQB.

What can Advanced Level (CPSA-A) graduates do?

CPSA-A graduates can:

- Independently and methodically design medium to large IT systems
- In IT systems of medium to high criticality, assume technical and content-related responsibility
- Conceptualize, design, and document actions to achieve quality requirements and support development teams in the implementation of these actions
- Control and execute architecture-relevant communication in medium to large development teams

Requirements for CPSA-A certification

- Successful training and certification as a Certified Professional for Software Architecture, Foundation Level® (CPSA-F)
- At least three years of full-time professional experience in the IT sector; collaboration on the design and development of at least two different IT systems
  - Exceptions are allowed on application (e.g., collaboration on open source projects)
- Training and further education within the scope of iSAQB Advanced Level training courses with a minimum of 70 credit points from at least three different areas of competence
  - existing certifications (for example: Sun/Oracle Java architect, Microsoft CSA) can be credited upon application
- Successful completion of the CPSA-A certification exam

Essentials

What is taught in the WEBSEC module?

Security requirements are among the key challenges when designing and developing software. There are often a variety of potential attack points in IT systems that could be successfully exploited by potential attackers (with appropriate effort).

The lack of basic knowledge on security, high time pressure or carelessness frequently leads to seemingly small errors, which can then be exploited with fatal consequences in terms of security. Web applications, in particular, often have a potentially large, globally distributed user group with access via the Internet. As a result of this, the circle of attackers increases massively and so, too, does the likelihood of errors being discovered and exploited. In addition, web applications are often victims of automated attack attempts shortly after implementation. Information systems may only be used by the company's own employees and are thus exposed to other attack scenarios. After all, embedded systems can now be found almost everywhere, so security issues can have a massive impact. Updates are not always possible in embedded systems.

If you take a look at the most common attack methods, they can usually be prevented by a "clean" architecture and clear communication. This curriculum aims to combine the somewhat academic world of security in software development with common technical practice.

Security cannot be considered independently of the context in which the systems are used. The reference to web applications, information systems, or embedded systems limits the thematic focus and ensures that the relevant information for the security of the respective systems is communicated. The curriculum focuses on web applications, but content about embedded systems or information systems can be inserted at the relevant points instead.

Demarcation

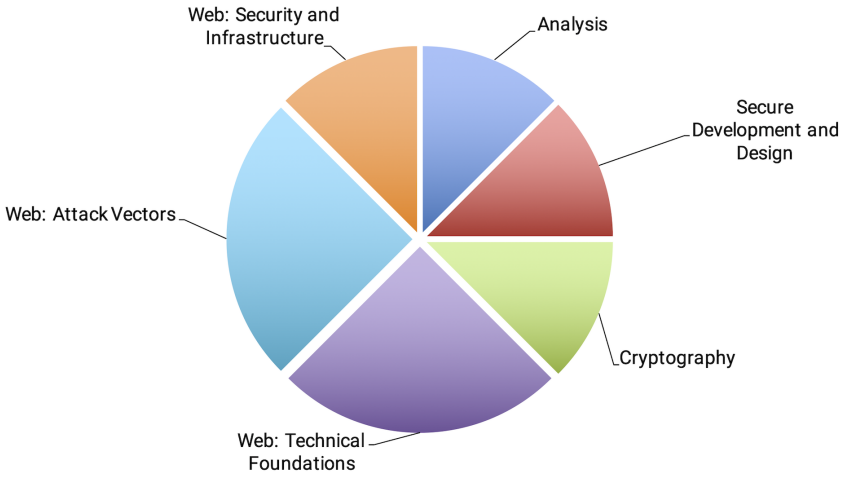
This curriculum excludes the following topics:

- Security measures such as organizational access and entry control systems
- Structural measures to increase the security of IT infrastructure (e.g., fire protection, locking systems)
- Detailed presentation or discussion of the legal foundations
- Hardware (physical attacks, biometrics, emissions)

Curriculum Structure and Recommended Durations

Content	Recommended minimum duration (minutes)
1. Analysis	135
2. Secure Development and Design	135
3. Cryptography	135
4. Web: Technical Foundations	270
5. Web: Attack Vectors	270
6. Web: Security and Infrastructure	135
Total	1080 (18h)

Allocation of time for the topic areas



Duration, Teaching Method and Further Details

The times stated below are recommendations. The duration of a training course on the WEBSEC module should be at least 2 days, but may be longer. Providers may differ in terms of duration, teaching method, type and structure of the exercises and the detailed course structure. In particular, the curriculum provides no specifications on the nature of the examples and exercises.

Licensed training courses for the WEBSEC module contribute the following credit points towards admission to the final Advanced Level certification exam:

Methodical Competence:	10 Points
Technical Competence:	20 Points
Communicative Competence:	0 Points

Prerequisites

The requirements for training depend on the area on which the training focuses. For example, the requirements for security in the web area are:

- Basic knowledge of network communication
- Basic knowledge of web technologies such as HTML, CSS, and JavaScript
- Basic knowledge of the creation of web applications

There are other requirements for information systems and embedded systems. However, basic knowledge of the architecture and implementation of such systems is sufficient.

Structure of the Curriculum

The individual sections of the curriculum are described according to the following structure:

- **Terms/principles**: Essential core terms of this topic.
- **Teaching/practice time**: Defines the minimum amount of teaching and practice time that must be spent on this topic or its practice in an accredited training course.
- **Learning goals**: Describes the content to be conveyed including its core terms and principles.

This section therefore also outlines the skills to be acquired in corresponding training courses.

Supplementary Information, Terms, Translations

To the extent necessary for understanding the curriculum, we have added definitions of technical terms to the iSAQB glossary and complemented them by references to (translated) literature.

1. Analysis

Duration: 105 min	Practice time: 30 min

1.1. Terms and principles

Security in software development often involves greater effort when designing, developing, maintaining, and operating an application. These measures are intended to increase the security of the application and thus support fundamental architectural objectives. This includes an analysis of the following aspects: - data and system functions worth protecting - potential attack vectors - dealing with problems caused by potential security vulnerabilities. In addition, there are common guidelines and standards. These analyses serve as the basis for all other security concepts.

Important terms:

Risk management, risk management framework, threat modeling, threat management, attack tree, attack vector, operational concept, audit, guidelines, standards, business context analysis, risk appetite

1.2. Learning Goals

LG 1-1: Risks and models

A comprehensive software architecture includes:

- Analysis of risks from a security perspective
- Threat model
- Attack trees
- Threat analysis

Classification of risks related to architecture:

- Adaptation of quality objectives
- Additional non-functional requirements
- The possible application of secure design patterns

Understanding the importance of risk management and possible actions via the phases:

- Requirement analysis
- Architectural design
- Development process
- Acceptance and testing
- Operations and infrastructure

These phases are independent of process models. Agile processes and a waterfall process both pass through these phases, only at different frequencies.

LG 1-2: The fundamental security goals

Security goals include:

- Confidentiality
- Integrity
- Authenticity
- Availability
- Liability
- Anonymity
- Imputability

Following the training, participants should know the security goals and be able to assess their importance.

LG 1-3: Asset identification and access concepts

Participants should be able to identify and classify assets worth protecting (for example, sensitive or personal data). They should also be able to handle access concepts. To do this, it is necessary to understand access principles and know different types of principles (ACL, role-based, etc.).

LG 1-4: Identify criteria for acceptance and auditing

After identifying assets worth protecting, measures can be derived from this to protect these assets. The implementation of these measures should be reviewed in the event of acceptance according to previously known criteria.

LG 1-5: Tradeoff of security against other quality attributes

Participants should understand the tradeoff of security against quality attributes (e.g., usability, ISO 25010 formerly 9126) or their own business purpose (business context analysis).

LG 1-6: Understand security as a process, not as a single measure

LG 1-7: Understand security as the responsibility of all stakeholders

LG 1-8: Know common guidelines, standards, and recommendations

- ISO 27000 (Information Technology - Security Techniques)
- ÖNORM A 7700 (Sicherheitstechnische Anforderungen an Webapplikationen)
- BSI Grundschutz
- Common Criteria for Information Technology Security Evaluation (ISO 15408)
- OWASP (Open Web Application Security Project)
- PCI-DSS (Payment Card Industry Data Security Standard)
- GDPR (General Data Protection Regulation)
- Regulations on security for commissioned data processing
- Legal liability for security issues

LG 1-9: Know common classification systems for security issues

- CVSS (Common Vulnerability Scoring System)
- OWASP Rating

LG 1-10: Categorize common certifications

2. Secure design and development process

Duration: 105 min	Practice time: 30 min

2.1. Terms and Principles

Security must be taken into account when an application is created. This extends across all stages of development. Security-specific design decisions often relate to specific requirements. However, there are established tools and process models to make design and implementation fundamentally more secure.

Important terms: Documentation security concept, access concept, secure development infrastructure, handling 3rd party code, supporting tools, application security lifecycle, security by design

2.2. Learning Goals

LG 2-1: Basic concept 'validation of all inputs and escaping of all outputs'

Most attacks are based either on the lack of validation of user input or the lack of escaping of output from the application. A secure application design makes the implementation of these basic rules easy.

LG 2-2: Principle of security gates, review, and 'trust no one'

Further principles lay the foundation for secure application in design, development, and team culture. This includes the classic 'two-man rule' as well as 'trust no one – document the exceptions', 'a chain is only as strong as its weakest link'.

LG 2-3: Indicators of secure application design

- Compliance with documented architectural specifications
- Compliance with a documented development process
- Regular reviews (for instance, the two-man rule)
- Measures derived from the threat analysis

LG 2-4: Basic patterns for secure coding guidelines

- Find your own security patterns

More examples: - Secure Factory - Secure State Machine - Secure Logger - …

LG 2-5: Content of a secure development process and example framework

- OWASP SAMM
- MS SDL
- BSIMM
- Own derivations of best practices suitable for your own company

LG 2-6: Access concepts for system landscape, artifacts, and source code

- Creating access concepts

- The design of security-relevant acceptance criteria
- How changes to an application are fully documented
- How to handle 3rd party code securely (dependency management)
- How does penetration testing work as an audit method?
- What test options are there?
- Influence of a secure development process on infrastructure and project planning

LG 2-7: Which tools and infrastructure components support the secure development process

- Basic tools to support a more secure development process
- Basic security-relevant infrastructure components

LG 2-8: Demarcation of analytical methods

- SAST (Static Application Security Testing)
- DAST (Dynamic Application Security Testing)
- IAST (Interactive Application Security Testing)
- SCA (Software Composition Analysis)

LG 2-9: Incident management

- Acknowledgment of security vulnerabilities
- Application security lifecycle
- Patch management

3. Cryptography

Duration: 135 min	Practice time: 0 min

3.1. Terms and principles

Hash procedures, random numbers, entropy, symmetric encryption, asymmetric encryption, common standards and their lifetime, general recommendations

3.2. Learning Goals

LG 3-1: Basics

- Cryptography (encryption) to keep information confidential
- Basic terms:
  - Integrity
  - Authenticity
  - Confidentiality
  - Liability
  - Protection against unintentional/unauthorized access
- Public versus confidential algorithms (Kerckhoffs's principle)
- Current state of research: public algorithms, secret keys

LG 3-2: Hashing

- The use of hashing methods
- Common procedures
- Known attacks against hashing methods (e.g., rainbow tables)
- Salting

LG 3-3: Encryption procedures

- The influence of secure random number generation on encryption (entropy)
- The use cases and use of symmetric encryption
- The use cases and use of asymmetric encryption
- Common procedures and their recommended lifetime
- 'Perfect forward secrecy' principle

LG 3-4: Trust concepts

- CA principles and PKI
- Web of trust

LG 3-5: Practical use

- Use of cryptographic libraries and frameworks
- Principle of the crypto-provider (separation of interface and implementation)
- Certificates (X.509)
- Digital signatures

4. Web: Technical foundation

Duration: 240 min	Practice time: 30 min

4.1. Terms and principles

Specific security issues such as authentication or secure data transmission are often solved with technically similar measures. Some of these solutions require a specific application design. Subsequent refactoring in the direction of these designs is often associated with high expenses and is prone to errors at these critical points.

This section relates to the security of web systems. For information systems or embedded systems, other content must be conveyed accordingly.

Important terms:

Authentication principles, multifactor, single sign-on, identity management, federation, tools, secure protocols, authorization concepts

4.2. Learning Goals

LG 4-1: Common 'good practices'

- 'Don’t do it yourself'
- External security reviews
- Penetration testing

LG 4-2: Authentication types

- Multifactor
- HTTP Auth
- Single Sign On

LG 4-3: "Security through obscurity" security measures

‘Security through obscurity’ measures alone are not enough to implement a secure system, but can be helpful to reduce attack vectors and provide the attacker with as little information as possible.

LG 4-4: Security-related protocols (e.g., TLS)

LG 4-5: Common authorization concepts and relevant implementations

- Stateless vs. stateful
- OAuth (Open Standard for Authorization) / OpenID Connect
- OpenID
- SAML (Security Assertion Markup Language)
- JWT (JSON Web Tokens)

LG 4-6: Supporting tools

- Review
- Code/project analysis
- Integration build process
- If necessary, further tools can be explained in the training

5. Web: Known attacks and attack vectors

Duration: 240 min	Practice time: 30 min

5.1. Terms and principles

Although attack methods such as SQL injections have been known for several decades, their prevalence shows that this problem still exists. A recurring pattern of attacks is evident, especially for web applications. An understanding of common attacks is required for designing and developing secure applications.

This section relates to the security of web systems. For information systems or embedded systems, other content must be conveyed accordingly.

Important terms:

Attack vectors, network attacks, common web attack methods, injection, fuzzing, hijacking, cross-site attacks, social engineering, denial of service (DoS), credential stuffing

5.2. Learning Goals

LG 5-1: Attack vectors and classification

Participants should know the usual attack vectors and be able to classify them in an architecture-specific manner

- Application layer
- Operating system and container layer
- Network layer
- Design layer
- Process layer

LG 5-2: Specific dangers of social engineering in web applications

LG 5-3: Injection attacks

- How do injection attacks work?
- What design decisions hamper injection attacks?

LG 5-4: Significance and functioning of denial-of-service (DoS) and distributed denial-of-service (DDoS) attacks

- How do DoS and DDoS attacks work?
- Which design decisions can hamper DoS and DDoS attacks?
- How can such attacks be detected during operation?
- Importance of botnets

LG 5-5: Attacks via the runtime environment/application platform

- Encryption of (temporary) data
- Importance of swapping
- Backdoors
- In-memory attacks

LG 5-6: Identifying security vulnerabilities via fuzzing

- What problems can be found by means of fuzzing?
- Fuzzing as a black box test
- Countermeasures and effects

LG 5-7: Man-in-the-middle attacks

- How do MITM attacks work?
- What possible countermeasures can hamper and/or prevent MITM attacks?
- How can MITM attacks be detected?

LG 5-8: Important sources for current threats and attacks

There are a number of well-known vendor-neutral websites that publish and explain current attacks and threats.

- SANS25
- OWASP Top Ten

6. Web: Security and infrastructure

Duration: 135 min	Practice time: 0 min

6.1. Terms and principles

Web applications do not run as completely isolated systems, but are surrounded by infrastructure. In order to increase security, the same components are used again and again. This is not about specific products from a manufacturer, but about the basic ideas and principles behind them.

This section relates to the security of web systems. For information systems or embedded systems, other content must be conveyed accordingly.

Important terms: WAF, DMZ, firewall, IDS, IPS, logging, monitoring, ongoing feedback

6.2. Learning Goals

LG 6-1:Function/operation and processes of firewalls

- Packet filtering
- Zone principle in operation
- Use and architectural impact of a DMZ

LG 6-2: Web Application Firewalls

LG 6-3: Intrusion Detection / Prevention systems

LG 6-4: Validation with feedback from operations

Feedback channels can be, among other things:

- Logging
- Monitoring
- Fixed feedback processes/interfaces

LG 6-5: Use of TLS

- Principle of transport layer security
- Use even within closed networks

References

This section contains references that are cited in the curriculum.

A

- [Anderson 2001] Ross Anderson: "Security Engineering", O'Reilly 2001, Methodology

B

- [BSI Grundschutz] BSI IT-Grundschutz: "Umfassende Ein- und Ausgabevalidierung bei Webanwendungen und Web-Services", https://www.bsi.bund.de/DE/Themen/ITGrundschutz/ITGrundschutzKataloge/Inhalt/_content/m/m04/m04393.html

C

- [CERT] CERT: "Top 10 Secure Coding Practices", https://www.securecoding.cert.org/confluence/display/seccode/Top+10+Secure+Coding+Practices

F

- [FIRST] FIRST, Common Vulnerability Scoring System, https://www.first.org/cvss

M

- [McGraw 2006] Garry McGraw: "Software Security - Building Security in", Addison Wesley 2006
- [Mitnick 2002]Kevin Mitnick, Steve Wozniak: "The Art of Deception", John Willey & Sons 2002

O

- [OWASP TM] OWASP Wiki: "Threat Modelling", https://www.owasp.org/index.php/Application_Threat_Modeling
- [OWASP RRM] OWASP Wiki: "Risk Rating Methodology", https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology
- [OWASP SCP] OWASP Wiki: "Secure Coding Practices", https://www.owasp.org/index.php/OWASP_Secure_Coding_Practices_-_Quick_Reference_Guide

S

- [Schneier 1996] Bruce Schneier: "Applied Cryptography", John Wiley & Sons 1996
- [Schneier 1999] Bruce Schneier (1999), https://www.schneier.com/academic/archives/1999/12/attack_trees.html
- [Security Patterns] Schumacher, Fernandez-Buglioni, Hybertson, Buschmann, Sommerlad: "Security Patterns", "Integrating Security and Systems Engineering", Wiley 2005