CSP103 - SECURE SOFTWARE DESIGN

Course Learning Objectives

By the end of this course, you will be able to explain reasons for including security in the design of software, define secure design principles and how they are incorporated into software design and interconnectivity, describe the software design process, identify software security design considerations required for the development of secure software, compare and contrast the architectures that exist for secure software design, and in the end describe the technologies and computing environments and their impact on design decisions regarding security.

Description

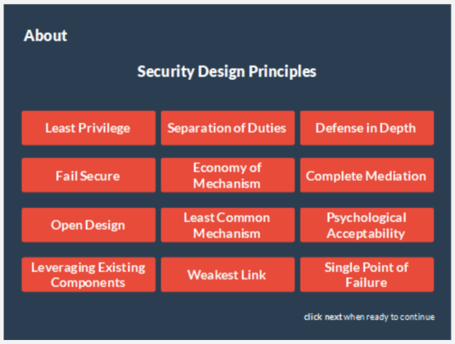
The design phase of software development is one of the most important phases in the Software Development Life Cycle. The Security Software Design domain will provide the learner with an understanding of how to ensure that software security requirements are included in the design of software. Learners will gain knowledge of secure design principles and processes, and be exposed to different architectures and technologies for securing software.

Audience

Certified Secure Software Lifecycle Professional (CSSLP)

Time Required

Tailored learning - 85 minutes total

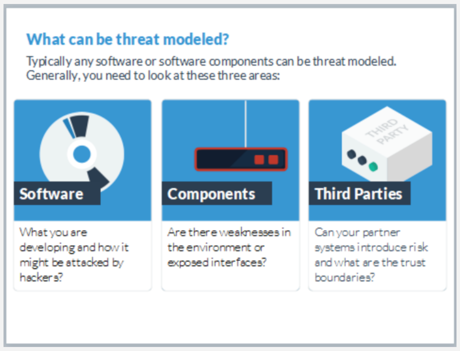


About

Security Design Principles

Least Privilege	Separation of Duties	Defense in Depth
Fail Secure	Economy of Mechanism	Complete Mediation
Open Design	Least Common Mechanism	Psychological Acceptability
Leveraging Existing Components	Weakest Link	Single Point of Failure

click next when ready to continue



What can be threat modeled?

Typically any software or software components can be threat modeled. Generally, you need to look at these three areas:

Software

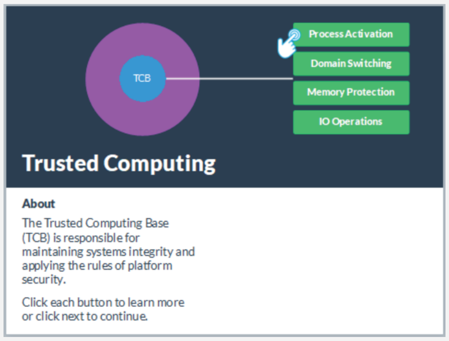
What you are developing and how it might be attacked by hackers?

Components

Are there weaknesses in the environment or exposed interfaces?

Third Parties

Can your partner systems introduce risk and what are the trust boundaries?



Trusted Computing

About

The Trusted Computing Base (TCB) is responsible for maintaining systems integrity and applying the rules of platform security.

Click each button to learn more or click next to continue.

SecurityCompass

CSP103 - SECURE SOFTWARE DESIGN

Course Outline

1. Security design principles

- About
- Least privilege
- Separation of duties
- Defense in depth
- Fail secure
- Economy of mechanisms
- Complete mediation
- Open design
- Least common mechanism
- Psychological acceptability
- Leverage existing components
- About interconnectivity
- Mashups
- Securing interconnectivity

4. Securing common technologies

- About technologies
- Designing authentication
- Newsflash - MGM Data Breach
- Virtualization
- VM security concerns
- Digital rights management
- Trusted computing
- Trusted computing platform
- Newsflash - Secure Boot Bypass
- Databases
- Inference attacks
- Polyinstantiation
- Normalization
- Views for databases

2. The design process and threat modeling

- Attack surface
- RASQ - calculate EAS
- Calculate total RASQ
- Dimensions to root attack vectors
- Importance of threat modeling
- What can be threat modeled?
- Threat model process
- Review of threat modeling

3. Design considerations

- Introduction
- About confidentiality design
- Symmetric encryption
- Key management
- Symmetric encryption speed
- Asymmetric encryption
- How it works
- Symmetric vs. Asymmetric
- Digital certificates
- Transport layer security
- Hashing for confidentiality
- Challenges
- Salts
- Integrity design
- Hashing for data integrity
- Code signing
- Code verification
- Referential integrity
- Resource locking
- Availability design
- Fail-over
- Replication
- Vertical scaling
- Horizontal scaling
- Designing authentication
- Comparing factors
- Multi-factor authentication
- Single sign-on (SSO)
- Access controls
- Entitlement management
- Logging and auditing

SecurityCompass