Software Security Engineering [WIP]

Sommersemester 2025

✉ mail@nilslambertz.de
nilslambertz

Alle Angaben ohne Gewähr. Keine Garantie auf Vollständigkeit oder Richtigkeit.

1 Introduction

1.1 Motivation

Influencing factors of secure software include

- **Security features**: Authentication, Authorization, Access control, cryptography, etc.
- **Technology**: Programming languages, development tools, etc.
- **Operational environments**: Firewalls, Intrusion detection systems, etc.

1.2 Terminology

- **Asset**: anything that has value to an organization, including human, physical, information, intangible and environmental resources (ISO 22300:2021)
- **Thread**: potential cause of an unwanted incident, which could result in harm to individuals, assets, a system or organization (ISO 22300:2021)
- **Adversary**: any person or a thing that acts (or has the power to act) to cause, carry, transmit, or support a threat Mussa and Malaiya (2015)
- **Vulnerability**: weakness of an asset or control that can be exploited by one or more threats (ISO/IEC 27000:2018)
- **Exploit**: method that identifies and takes advantage of a vulnerability in an asset Mussa and Malaiya (2015)
- **Attack**: attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use of an asset (ISO/IEC 27000:2018)

1.3 Protection Goals

- **Confidentiality**: property that information is not made available or disclosed to unauthorized individuals, entities, or processes (ISO/IEC 27000:2018)
- **Integrity**: property of accuracy and completeness (ISO/IEC 27000:2018)
- **Availability**: property of being accessible and usable on demand by an authorized entity (ISO/IEC 27000:2018)
- **Authenticity**: property that an entity is what it claims to be (ISO/IEC 27000:2018)
- **Possession**: holding, controlling, and having the ability to use information Parker (2015)
- **Utility**: usefulness of information Parker (2015)

2 Security Requirements Engineering

2.1 What makes security requirements special?

- typically **quality/non-functional** requirements
- often **negative** requirements: Describe what the system **should not** do, hard to validate
- often **immeasurable** as there is no clear satisfaction criteria
- often **incalculable**: Cost-benefit ratio hard to determine
- often **uncertain**, hard to identify without knowing the whole system design upfront
- hard to recognize vulnerabilities during productive/test operation, not “implicitly” tested by stakeholders/users when they use the system

2.2 Requirements Engineering - Core Activities

- **Elicitation**: Obtain requirements from stakeholders, refine them
- **Documentation**: Describe requirements adequately using natural language or models
- **Validation and Negotation**: Guarantee that requirements meet quality criteria
- **Management**: Structure requirements, maintain consistency, ensure implementation

2.2.1 Verification and Validation (V&V)

- **Verification**: "Are we building the product right?" Boehm (1981)
  - assurance, that a product, service or system meets the **needs** of the customers and stakeholders
  - often involves acceptance and suitability
- **Validation**: "Are we building the right product?" Boehm (1981)

2.3 Security Requirements - Definition

- **Security needs**: Design objectives concerning the protection of its stakeholders from consequences of (intentional) threats that conflict with stakeholders' goals
- **Security requirements**: Express security needs in a form suitable to be used and make its results verifiable

2.4 Security Needs - Influencing Factors

2.4.1 Goals

Security goals describe the desired protection of a system and its environment:

- Assets (e.g. documents) and their protection needs
- Dreaded consequences (e.g. loss of reputation)
- Organizational policies (e.g. only staff with security clearance can access classified information)
- Legal or business requirements (e.g. data protection laws)

2.4.2 Threats

A **threat** statement describes adversaries - their goals, capabilities or expected behavior:

- Capabilities (e.g. adversary controls a botnet)
- Objectives (e.g. work for profit and aim to evade apprehension)
- Target selection behavior (e.g. opportunistic, targeted)
- Technical attack patterns (e.g. transparent rerouting of network traffic)

2.4.3 Design

Security needs can be described by the **design** of a system or software:

- Required security functions (e.g. logging)
- Design details with security implications (e.g. long random session IDs)
- Common volnerabilities to avoid (e.g. XSS)

2.5 Abuse vs. Misuse

Abuse described **malicious** and **deliberate** acts.

Misuse describes **spontaneous** acts and possibly **careless** use of a system.

2.5.1 Misuse Cases

"A misuse case is the inverse of a use case, a function that the system **should not allow**.". It causes harm to some stakeholder if the sequence is allowed to complete.

- Misuse cases can be related to regular use cases
- **Threaten** relationship: Use case is **exploited** or **hindered** by a misuse case
- **Mitigate** relationship (of a "Security use case"): Use case is **countermeasure** against a misuse case, reduces the misuse case's chance of success

2.6 SQUARE Process Model

Security Quality Requirements Engineering (**SQUARE**) is a process model for categorizing and prioritizing security requirements. The focus is to build security concept into early stages of the development lifecycle.

2.6.1 SQUARE Steps

1. **Agree on Definitions**:
   - ensure that stakeholders agree of definitions of terms
   - Participants: Stakeholders, requirements engineers
2. **Identify Security Goals**:
   - identify protection goals of stakeholders and prioritize them
   - Participants: Stakeholders, requirements engineers
3. **Develop Artifacts**:
   - develop artifacts to support security requirements definition
   - Participants: Requirements engineers
4. **Perform Risk Assessment**:
   - assess risks using a method recommended by a risk expert
   - Participants: Requirements engineers, risk experts, stakeholders
5. **Select Elicitation Technique**:
   - overcome communication issues between stakeholders by selecting a proper elicitation technique
   - Participants: Requirements engineers
6. **Elicit Security Requirements**:
   - actual elicitation process using the techniques selected in the previous step
   - Participants: Stakeholders
7. **Categorize Requirements**:
   - categorize requirements as to level (e.g. system, software) and if they are quality requirements or constraints
   - Participants: Requirements engineers, other specialists
8. **Prioritize Requirements**:
   - possible cost-benefit analysis considering consequences
   - Participants: Stakeholders
9. **Inspect Requirements**:
   - Peer-review of requirements for problems, concerns, workload etc.
   - Participants: Inspection team

References

Boehm, B. (1981). *Software Engineering Economics*. Prentice-Hall advances in computing science and technology series. Prentice-Hall.

International Organization for Standardization (2021). Security and resilience — vocabulary. Technical Report ISO 22300:2021, ISO, Geneva, Switzerland.

International Organization for Standardization and International Electrotechnical Commission (2018). Information technology — security techniques — information security management systems — overview and vocabulary. Technical Report ISO/IEC 27000:2018, ISO/IEC JTC 1/SC 27, Geneva, Switzerland.

Mussa, A. A. Y. and Malaiya, Y. (2015). Comparing and evaluating cvss base metrics and microsoft rating system.

Parker, D. (2015). *Toward a New Framework for Information Security?*, pages 3.1–3.23.