Introduction to Software Security

Software Vulnerabilities (2)

Prof. Seong-je Cho

Spring 2020

Computer Security & Operating Systems Lab,

Dept. of Software Science, Dankook University, Korea

Sources / References

- *Information Security: Principles and Practice*, 2nd edition by Mark Stamp, Wiley, 2011
- Software Security: Principles, Policies, and Protection, Matias Payer, Apr. 2019.
- N. Vlajic, CSE 3482: Introduction to Computer Security, Yorku
- Nicholas Weaver, Computer Science 161: Computer Security, Berkeley
- Myrto Arapinis, Computer Security: INFRA10067, University of Edinburgh
- Lecture 12 Program Security, CS 450/650 Lecture

Contents

- Terminology
  - Bug, Vulnerability, S/W Vulnerability
  - Security requirements
  - Security Policy
  - Security mechanism

- Development Life-Cycle (DLC)
  - Software Development Life-Cycle (SDLC)
  - Microsoft SDL

Terminology

- NIST Computer Security Resource Center -- Glossary
- Cybersecurity Glossary of Terms

IT, IT system, Threat

- **IT**: H/W + S/W + Network + Data
- **Information system**: IT + People + Procedure (Policy)

- **IT System**: Any organized assembly of resources and procedures united and regulated by interaction or interdependence to accomplish a set of specific functions

- **Components of security threat:** Asset (Target) + Agent + Event

Measures, Countermeasures, …

- **Measures** (방안, 조치, 결과, 성과)
  - The results of data collection, analysis, and reporting

- **Countermeasures**: (보호조치, 대책)
  - Actions, devices, procedures, techniques, or other measures that reduce the vulnerability of an information system.
  - Protective measures prescribed to meet the security requirements

- **Security control**
  - Anything used as part of a security response strategy which addresses a threat in order to reduce risk. (Also known as countermeasure or safeguard.)

Address: (문제·상황 등에 대해) 고심하다[다루다]

Security control, Assessment

- **Security control** (보안 통제, 보안 제어)
  - A safeguard or countermeasure prescribed for an information system or an organization designed
    - to protect the confidentiality, integrity, and availability of its information and
    - to meet a set of defined **security requirements**.

- **Assessment**
  - The testing and/or evaluation of the management, operational, and technical security controls in an information system to determine the extent to which the controls are

    implemented correctly,

    operating as intended, and

    producing the desired outcome with respect to meeting the security requirements for the system.

Security Goal, Security Service

- **Security Goal**
  - The five security goals are C., I., A., accountability, and assurance.

- **Security Service**
  - A capability that supports one, or more, of the security requirements (Confidentiality, Integrity, Availability).
    - **Capability**: A feature or function. A set of security controls
  - A capability that supports one, or many, of the security goals.
  - Mechanisms used to provide confidentiality, data integrity, authentication or non-repudiation of information.
  - Examples of security services are key management, access control, and authentication.

Security Requirements

- **Security requirement(s) == Security Service(s)**
  - The set of minimum security controls (defined for a low-impact, moderate-impact, or high-impact information system) that provides a starting point for the tailoring process.
  - Requirements levied on an information system that are derived from applicable laws, Executive Orders, directives, **policies**, standards, instructions, regulations, or procedures, or organizational mission/business case needs to ensure the confidentiality, integrity, and availability of the information being processed, stored, or transmitted.
    - **E.g.,** Password requirements (password policy), Privacy policy requirements

Applicable law: 준거법. 국제 사법의 규정에 따라, 일정한 법률관계를 규정하는 데 준거하는, 자국(自國) 또는 외국의 법률.
Executive Orders: 대통령 명령, 행정 명령.
be levied on: ~에 부과되다.
levy: (세금 등을) 부과[징수]하다. (특히 세금의) 추가 부담금

Security Policy

- A set of criteria for the provision of security services.
- The statement of required protection for the information objects.
- A method, tool, or procedure that is the realization of security requirements.

- Security policies define the objectives and constraints for the security program.
  - Policies are created at several levels, ranging from organization or corporate policy to specific operational constraints (e.g., remote access).
  - In general, policies provide answers to the questions “what” and “why” without dealing with “how.”
  - Policies are normally stated in terms that are technology-independent.

Provision: 공급, 제공, 준비, (법률 관련 문서의) 조항[규정/단서]

Security Mechanism

- A device or function designed to provide one or more security services
  usually rated in terms of strength of service and assurance of the design.

- A method, tool, or procedure that is the realization of security requirements.
  - Note 1: A security mechanism exists in machine, technology, human, and physical forms.
  - Note 2: A security mechanism reflects security and trust principles.
  - Note 3: A security mechanism may enforce security policy and therefore must have capabilities consistent with the intent of the security policy.

Bug, Defect

- **Bug**
  - An error or mistake in S/W coding or H/W design or construction.
  - A bug represents a **flaw** or **vulnerability** in a system discoverable by attackers and used as point of compromise.
  - Attacks often use fuzzing technique (i.e. randomize testing tools) to locate previously unknown bugs in order to craft new exploits.

- **Defect**
  - An occurrence of a defect check that failed on an assessment object.
  - It indicates a weakened state of security that increases risk.

Vulnerability

- Weakness in an information system, system security procedures, internal controls, or implementation that could be exploit or triggered by a threat source
- a flaw or weakness that may allow harm to occur to an IT system or activity.
- A flaw or weakness in a computer system, its security procedures, internal controls, or design and implementation, which could be exploited to violate the system security policy.
- A security exposure in an OS or other system S/W or application S/W component.
  - A variety of organizations maintain publicly accessible DBs of vulnerabilities based on the version numbers of software.
  - Each vulnerability can potentially compromise the system or network if exploited.

Vulnerability

- a bug, flaw, weakness, or exposure of an application, system, device, or service that could lead to a failure of confidentiality, integrity, or availability

- An error, flaw, or mistake in **computer software** that permits or causes an unintended behavior to occur.

- Any weakness in an asset or security protection which would allow for a threat to cause harm.
  - It may be a **flaw in coding**, a **mistake in configuration**, a **limitation of scope or capability**, an **error in architecture, design, or logic** or a **clever abuse** of valid systems and their functions.

- **Software Vulnerability**
  - A security flaw, glitch, or weakness found in S/W that can be exploited by an attacker.

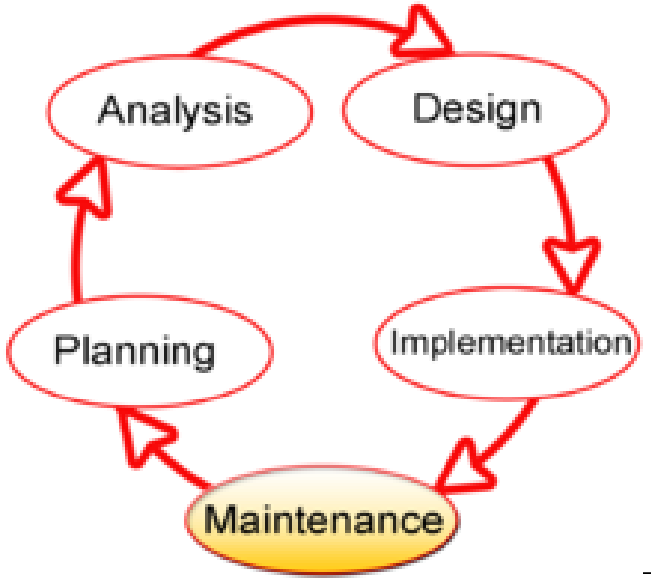
SDLC

- Software Development Life Cycle
- Microsoft SDL

Development Life Cycle (DLC)

- The scope of activities associated with a system, encompassing the system's initiation, development and acquisition, implementation, operation and maintenance, and ultimately its disposal that instigates another system initiation.

- **Systems Development Lifer Cycle (SDLC), Application Development Life-Cycle**
  - a process for planning, creating, testing, and deploying an information system.
  - The systems development life cycle concept applies to a range of hardware and software configurations, as a system can be composed of hardware only, software only, or a combination of both.



Source: Wikipedia
Model of the SDLC, highlighting the maintenance phase

Software Development Life Cycle (SDLC)

Stage1: Planning and requirement analysis

Stage2: Defining Requirements

Stage3: Designing the Software

- Analysis of requirements, UI, I/O, algorithm (abstraction), …

Stage4: Developing the project

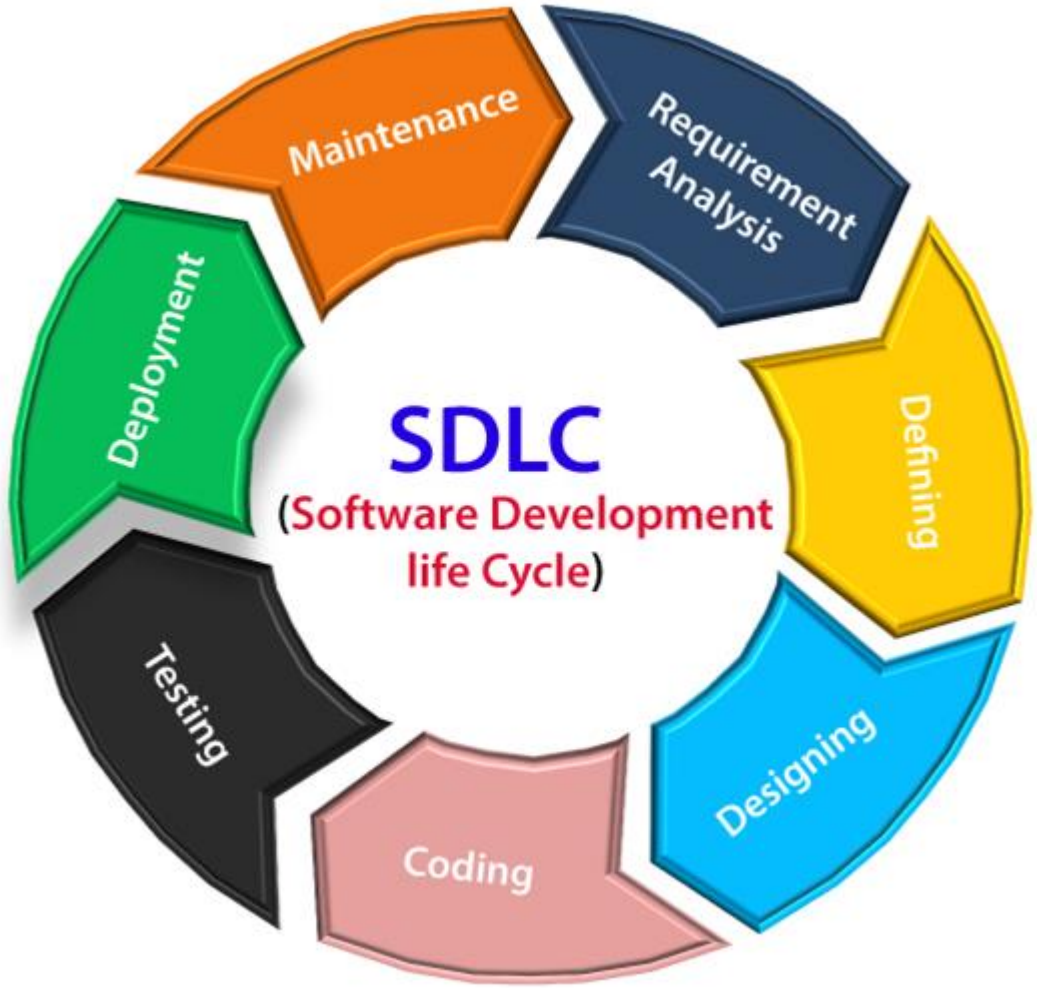
- Coding, Implementation

Stage5: Testing

Stage6: Deployment

- Once the software is certified, and no bugs or errors are stated, then it is deployed.
- After the software is deployed, then its maintenance begins.

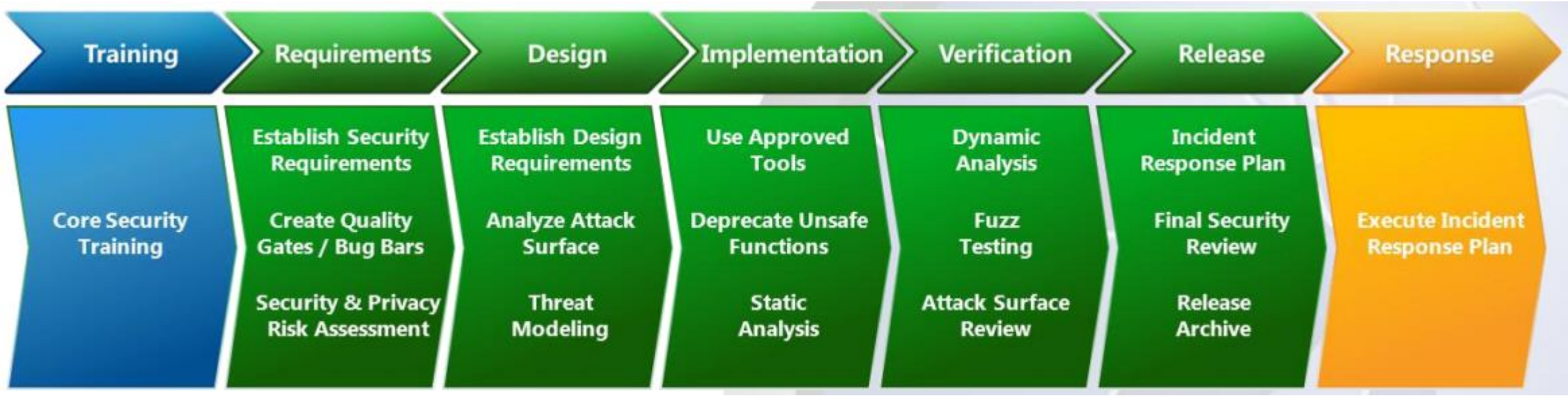
Stage7: Maintenance



Source: https://www.javatpoint.com/software-engineering-software-development-life-cycle

Microsoft Security Development Lifecycle (SDL)

1. **Training:** Core security training
2. **Requirements:** Establish security requirements, Security & Privacy risk assessment
3. **Design:** Establish design requirements, Threat modeling
4. **Implementation:** Deprecate unsafe functions, Static analysis
5. **Verification:** Dynamic analysis, Fuzz testing
6. **Release:** Incident response plan, Final security review
7. **Response:** Execute incident response plan



Source: Microsoft SDL in practice, Alex Thissen

Summary

- Security Requirements
- Bug, Flaw
- Vulnerability, S/W Vulnerability

- SDLC
- Microsoft SDL