CEN429 Secure Programming Week-2

Development Environment Security and Software Development Processes

Author: Dr. UÄŸur CORUH

Contents

List of Figures

List of Tables

1 CEN429 Secure Programming

1.1 Week-2

1.1.0.1 Development Environment Security and Software Development Processes Download

- PDF[1]
- DOC[2]
- SLIDE[3]
- PPTX[4]

1.1.1 Outline

- Development Environment Security and Software Development Processes

[1] pandoc_cen429-week-2.pdf
[2] pandoc_cen429-week-2.docx
[3] cen429-week-2.pdf
[4] cen429-week-2.pptx

- Software Development Process
  - Software Development Flow
  - Configuration Baseline
  - Initiating and Classifying Changes
  - Approving and Releasing Changes
- Software Development Environments
  - Development Environment Security
  - Version Control Systems
  - Source Code Server Security
  - Server Room and Development Computer Security

1.2 Software Development Flow and Change Management

1.2.1 1. Software Development Flow

1.2.1.1 Theoretical Explanation: Software development processes must be controlled through specific flows. Managing changes effectively ensures the success of the project. This flow typically involves version control systems, technical teams, and project management processes.

1.2.1.2 Application:

- **Application:** Start a simple software project and create a process showing how to manage change requests (RFC). Set up an approval mechanism at every step and manage the project through a version control system.

1.2.2 2. Configuration Baseline

1.2.2.1 Theoretical Explanation: Configuration baseline involves defining a specific version of a product or system, ensuring that all changes from this version onward can be tracked. This is a critical step in the development and change management processes.

1.2.2.2 Application:

- **Application:** Create a GIT repository and set up the initial configuration baseline. Establish a structure where all subsequent changes are traceable.

1.2.3 3. Initiating a Change

1.2.3.1 Theoretical Explanation: Change requests (RFC) are made for adding new features or fixing bugs in the project. This process involves defining all requirements and conducting technical meetings before development begins.

1.2.3.2 Application:

- **Application:** Create a change request (RFC) and simulate how this request is communicated to the project team. Show a scenario where decisions are made through meetings and technical reviews.

1.2.4 4. Classifying the Change

1.2.4.1 Theoretical Explanation: Change requests are classified based on cost, time, and technical requirements. If there are no financial or technical barriers, the product owner approves the request for development by the technical team.

1.2.4.2 Application:

- **Application:** Review a change request and manage the process of how it is classified and approved based on specific conditions.

1.2.5 5. Approving and Scheduling the Change

1.2.5.1 Theoretical Explanation: Before starting development, the change request is approved, and a project plan is created. This plan includes sprints and task assignments.

1.2.5.2 Application:

- **Application:** Organize a sprint planning meeting and assign tasks according to the change request. Use planning tools (Jira, Trello, etc.) to structure the process.

1.2.6 6. Releasing the Change

1.2.6.1 Theoretical Explanation: The developed change is deployed to production after testing is completed. This step ensures that the change is implemented successfully.

1.2.6.2 Application:

- **Application:** Pull a developed change from the version control system and deploy it to the production environment. Record the steps and test results during the release process.

1.2.7 7. Validating and Reviewing the Change

1.2.7.1 Theoretical Explanation: After the change is released, it is validated to ensure it has been implemented correctly and meets expectations. Technical and user reviews are conducted.

1.2.7.2 Application:

- **Application:** Test the released change and gather user feedback. Verify whether the change meets expectations.

1.3 Software Development Environments and Source Code Version Control System

1.3.1 1. Development Environments

1.3.1.1 Theoretical Explanation: Software development occurs across different environments: development, testing, and production. Each environment requires different security measures and configurations.

1.3.1.2 Application:

- **Application:** Set up development and testing environments. Develop an application that demonstrates different security configurations for each environment.

1.3.2 2. Version Control Systems

1.3.2.1 Theoretical Explanation: Version control systems (Git, SVN, etc.) are used to track software development processes and revert changes when necessary. Each change is recorded, and developers can switch between versions.

1.3.2.2 Application:

- **Application:** Manage a software development process using GIT. Switch between branches and revert a change.

1.3.3 3. Development Site and Source Code Server Security

1.3.3.1 Theoretical Explanation: The physical and digital security of the development environment is crucial. Protecting source code servers and monitoring systems ensures the integrity of the software.

1.3.3.2 Application:

- **Application:** Demonstrate how to secure a source code server in a development environment. Set up encryption and access control systems.

1.3.4 4. Development Office and Server Room Security

1.3.4.1 Theoretical Explanation: Server rooms and development computers must be protected with security measures to ensure the security of the software. Access controls, encryption, and physical security are part of this process.

1.3.4.2 Application:

- **Application:** Simulate access controls for a server room. Configure security software on development computers and take precautions against potential attacks.

1.4 Summary of the Week and Next Week

1.4.1 This Week:

- **Software Development Flow and Change Management**
- **Configuration Baseline and Change Approval**
- **Development Environments and Version Control Systems**
- **Physical and Digital Security**

1.4.2 Next Week:

- **Data Security and Cryptography**
- **Secure Communication and Key Management**

$$End - Of - Week - 2$$