Secure coding

ISO 27002 Control 8.28

Control

Secure coding principles should be applied to software development

Purpose

To ensure software is written securely thereby reducing the number of potential information security vulnerabilities in the software

Why is it important?

- Secure coding prevents the introduction of security flaws (e.g., database injection, cross-site scripting) that lead to successful attacks
- Governance ensures consistency across the organization and third-party components
- Reduces the cost and effort of fixing vulnerabilities later in the life cycle

Implementation

- Establish organization-wide governance and a minimum secure baseline
- Provide secure coding training and qualification for developers
- Enforce secure coding standards specific to programming languages
- Prohibit insecure design techniques (e.g., hard-coded passwords)
- Protect source code against unauthorized access and tampering
- Manage and update external tools and libraries regularly

Concepts & Examples

By defining contractually :

- Input validation and sanitization
- Parameterized queries/prepared statements
- Output encoding (XSS prevention)
- Secure session management
- CSRF token implementation
- Error handling without information disclosure
- Secure dependency management (SCA)
- SAST/DAST integration in CI/CD

Link with other frameworks

- NIST 800-53 rev5 : SA-4(3)*, SA-8, SA-11(1)*, SA-15(5)*, SI-10
- NIST CSF 2.0 : NA

Renaud Dardenne
Asphalia Consulting