Security Best Practices (12)

B1 Identify security requirements.

Identify security requirements for your application during the initial planning stages. The security of the application throughout its different stages should be evaluated based on its compliance with security requirements.

B2 Design for security.

Aim for simple designs because the likelihood of implementation errors increases with design complexity. Architect and design your software to implement security policies and comply with security principles such as: secure defaults, default deny, fail safe, and the principle of least privilege.

B3 Perform threat modelling.

Use threat modelling to analyze potential threats to your application. The result of threat modelling should inform security practices in the different SDLC stages, e.g., for creating test plans.

B4 Perform secure implementation.

Adopt secure coding standards for the programming language you use, e.g., validate input and sanitize data sent to other systems, and avoid using unsafe or deprecated functions.

B5 Use approved tools and analyze third-party tools’ security.

Only use approved tools, APIs, and frameworks or those evaluated for security and effectiveness.

B6 Include security in testing.

Integrate security testing in functional test plans to reduce redundancy.

B7 Perform code analysis.

Leverage automated tools such as SATs to detect vulnerabilities like buffer overflows and improper user input validation.

B8 Perform code review for security.

Include security in code reviews and look for common programming errors that can lead to security vulnerabilities.

B9 Perform post-development testing.

Identify security issues further by using a combination of methods, e.g., dynamic analysis, penetration testing, or hiring external security reviewers to bring in a new perspective.

B10 Apply defense in depth.

Build security in all stages of the SDLC, so that if a vulnerability is missed in one stage, there is a chance to eliminate it through practices implemented in the remaining stages. 290 Fourteenth Symposium on Usable Privacy and Security USENIX Association

B11 Recognize that defense is a shared responsibility.

Address software security as a collective responsibility of all SDLC entities, e.g., developers, testers, and designers.

B12 Apply security to all applications.

Secure low risk applications and high-risk ones. The suggested effort spent on security can be derived from assessing the value of assets and the risks, however, security should.