Cheatography

The Top 12 Practices of Secure Coding Cheat Sheet

by [deleted] via cheatography.com/2754/cs/15063/

Introduction

Developing and managing software is no easy feat. Consider that an operating system can contain over 50 million lines of code. To help developers rise to the software security challenge, enter OWASP, the Open Web Application Security Project. Comprised of thousands of super-smart participants collaborating globally, OWASP provides free resources "dedicated to enabling organizations to conceive, develop, acquire, operate and maintain applications that can be trusted."

It might make good sense then, when evaluating your (or your vendor's) program, to begin by measuring it against OWASP's Software Assurance Maturity Model. The below list provides a quick summary of the top 12 security practices to mitigate risks from internal and third-party software. How many boxes does your program check?

Source: https://www.securitymagazine.com/articles/88600-the-top-12-practices-of-secure-coding

Governance

1. Strategy and Metrics: Establish a unified security roadmap, set corporate risk tolerance and align expenses with asset value.

2. Education and Guidance: Provide role-specific secure software development lifecycle training.

3. Policy and Compliance: Understand compliance drivers, create compliance gates and collect the right types of data to enable audit.

Construction

4. Threat Assessment: Identify, evaluate and mitigate application--specific threats.

5. Security Requirements: Specify necessary security controls, including within supplier agreements, and audit those controls.

6. Secure Architecture: Adopt software development frameworks, identify secure design patterns and embed secure-by-default principles.

Verification

7. Design Review: Assess software design against a comprehensive set of best practices.

8. Implementation: Integrate automated code analysis tools into development processes, customize code review for language-level risks and for application-specific vulnerabilities.

9 Security Testing: Require human penetration testing and automate application-specific testing throughout the development process and, significantly, before deployment.

How to Develop and Review Code Security

Operations

10. Issue Management: Create a vulnerability response team, implement a security issues disclosure process (consider a bug bounty program), conduct root cause analysis and collect per-issue metrics.

11. Environment Hardening: Install critical upgrades and patches, monitor configurations, deploy network protection tools.

12. Operational Enablement: Facilitate communications between development teams and operators, capture critical security information, maintain formal procedures for issuing alerts, create per-release change management procedures and perform code signing.