Secure Coding Implementation

This playbook describes the sequential steps for writing secure code that defends against common web security vulnerabilities, including SQL injection, XSS (Cross-Site Scripting), and CSRF (Cross-Site Request Forgery) attacks.

Step 1: **Input Validation**

Enforce strict input validation to ensure that all incoming data adheres to expected formats. Use whitelisting for allowed characters and data types to reduce the risk of malicious input slipping through.

Step 2: **Parameterized Queries**

Avoid SQL injection by using parameterized queries or prepared statements. This method separates the SQL logic from the data, preventing attackers from manipulating queries with harmful input.

Step 3: **Output Encoding**

Prevent XSS attacks by encoding data before output. Convert special characters in output into safe encoded representations, especially in user-generated content.

Step 4: **Authentication Controls**

Strengthen authentication mechanisms. Implement multi-factor authentication, secure session management, and encrypted password

storage with hashing and salting to protect against unauthorized access.

Step 5: **CSRF Tokens**

Defend against CSRF by including a unique, secret token in forms that change server state. This token should be validated on the server side before any state-change operation is allowed.

Step 6: **Code Audits**

Regularly perform code reviews and security audits. Use static and dynamic analysis tools to detect potential vulnerabilities that may have been missed during development.

Step 7: **Update Libraries**

Keep third-party libraries and dependencies up-to-date. Security vulnerabilities are often found in outdated software components, so apply updates and patches promptly.

Step 8: **Security Training**

Invest in security training for developers. Ensure that the development team is aware of secure coding practices and emerging threats to write inherently secure code from the start.

General Notes

Continuous Learning

Security is an evolving field, and attackers constantly develop new methods. It's important to stay informed about the latest security threats and defenses.

Security Standards

Consider adhering to security standards and frameworks such as OWASP Top 10, CERT Coding Standards, and Common Weakness Enumeration for additional guidance on secure coding.