Ingeniería En Desarrollo Y Gestión De Software

Nombre:

García Arreola Howard Isai

Subject:

Desarrollo Movil Integral

Actividad:

Secure Coding Principles Specification

Grupo: 10-B

Profesor:

Ray Brunett Parra

Galaviz

Fecha de Realización:

January 15th, 2025

OWASP Top Ten

The OWASP Top Ten is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications. The list is regularly updated to reflect the evolving landscape of web security threats.

Broken Access Control:

Exploitation of access control mechanisms, allowing unauthorized actions or access to sensitive data.

Examples: URL manipulation to access restricted pages, improper enforcement of user roles.

Cryptographic Failures (formerly Sensitive Data Exposure)

Weak or misconfigured cryptography leads to data being insufficiently protected.

Examples: Insecure storage of passwords, lack of encryption for sensitive data in transit.

Injection

Malicious input is sent to an interpreter, executing unintended commands or accessing data without proper authorization.

Examples: SQL Injection, NoSQL Injection, Command Injection.

Insecure Design

Inadequate security controls in the design phase of software development.

Examples: Absence of threat modeling, secure design principles not applied.

Security Misconfiguration

Incorrectly configured systems or lack of secure settings.

Examples: Default accounts left active, verbose error messages exposing sensitive data.

Vulnerable and Outdated Components

Using components with known vulnerabilities or unsupported software versions.

Examples: Using outdated libraries, frameworks, or software.

Identification and Authentication Failures (formerly Broken Authentication)

Flaws in authentication mechanisms leading to credential theft or unauthorized access.

Examples: Weak passwords, improper session management, and lack of multi-factor authentication.

Software and Data Integrity Failures

Insecure software updates, CI/CD pipelines, or use of unverified third-party libraries.

Examples: Deployment of malicious updates, code tampering.

Security Logging and Monitoring Failures

Insufficient logging or monitoring delays in detecting and responding to breaches.

Examples: Lack of audit logs, failure to monitor access to sensitive resources.

Server-Side Request Forgery (SSRF)

Attackers manipulate the server to make HTTP requests to unintended destinations.

Examples: Exfiltration of sensitive data from internal systems via manipulated URLs.

OWASP Secure Coding Practices Checklist

1. **Input Validation**: Ensure all external inputs are validated to prevent malicious data from causing harm. This involves checking data length, format, and type before processing.
2. **Output Encoding**: Encode data before sending it to external systems to prevent injection attacks, such as cross-site scripting (XSS).
3. **Authentication and Password Management**: Implement robust authentication mechanisms and manage passwords securely to prevent unauthorized access. This includes enforcing strong password policies and securely storing credentials.
4. **Session Management**: Properly manage user sessions to prevent hijacking and fixation attacks. This involves generating unique session identifiers and securely handling session tokens.
5. **Access Control**: Enforce the principle of least privilege by granting users only the permissions necessary for their roles. Regularly review and update access controls to maintain security.
6. **Cryptographic Practices**: Use strong, industry-standard cryptographic algorithms to protect sensitive data both at rest and in transit. Avoid developing custom cryptographic solutions.

7. **Error Handling and Logging**: Implement comprehensive error handling to prevent the exposure of sensitive information. Log security-relevant events for monitoring and auditing purposes, ensuring that logs do not contain sensitive data.
8. **Data Protection**: Safeguard data through proper encryption and access controls, ensuring compliance with data protection regulations.
9. **Communication Security**: Secure communication channels using protocols like TLS to protect data integrity and confidentiality during transmission.
10. **System Configuration**: Maintain secure system configurations by disabling unnecessary services and applying security patches promptly.
11. **Database Security**: Protect databases against SQL injection and other attacks by using parameterized queries and stored procedures.
12. **File Management**: Handle file operations securely by validating file paths and restricting file permissions to prevent unauthorized access.
13. **Memory Management**: Prevent memory-related vulnerabilities, such as buffer overflows, by performing proper memory allocation and deallocation.
14. **General Coding Practices**: Adopt secure coding standards and perform regular code reviews to identify and mitigate potential security issues.

References:

OWASP TOP 10 Vulnerabilities 2024 (UPdated) | *WattleCorp Cybersecurity Labs*. (2024, September 27). Wattlecorp. https://www.wattlecorp.com/owasp-top-10/

Secure Software Development | *StoryBlok*. (n.d.). Storyblok. https://www.storyblok.com/trust-center/secure-software-development

Secure Software Development | *StoryBlok*. (n.d.). Storyblok. https://www.storyblok.com/trust-center/secure-software-development