The OWASP Top 10: A Complete Guide for Engineering Teams

Author: Kamakshaiah Musunuru | **Published:** 11 September 2026 | **Category:** Tech Update | **Read time:** 2 min read

Abstract

The OWASP Top 10 remains the cornerstone of web application security awareness. This guide explains the key risks every engineering team must understand, including prevention strategies and modern tooling.

Keywords: OWASP Top 10, web application security, access control, injection, cryptographic failures, SSRF

Tags: OWASP, security, web application, vulnerabilities, best practices

The OWASP Top 10 is the most widely adopted awareness document for web application security. Despite being regularly updated, the fundamental risks remain the primary vector for real-world breaches.

Key Risks for Engineering Teams

1. Broken Access Control The most common and damaging class of risk. Server-side enforcement, deny-by-default, and per-object ownership checks are essential. Never rely on client-side or hidden-UI controls for security.

2. Cryptographic Failures Data exposure often stems from weak crypto. Use modern, vetted libraries, encrypt data at rest and in transit, and never implement custom cryptography or hardcode keys.

3. Injection SQL and command injection remain prevalent. Use parameterised queries, allow-list validation, and proper output encoding to neutralise untrusted input.

4. Insecure Design Prevent flaws at the design stage with threat modelling, secure design patterns, and security requirements written into the architecture before code is written.

5. Security Misconfiguration Harden default configurations, remove verbose errors, patch promptly, and use automated configuration scanning to close misconfiguration gaps.

6. Vulnerable and Outdated Components Track dependencies and their known vulnerabilities. Run dependency scanning in CI, remove unused packages, and stay current with security releases.

7. Identification and Authentication Failures Enforce strong passwords, mandatory MFA, and framework-based session management. Protect credential stores and rotate leaked secrets immediately.

8. Software and Data Integrity Failures Protect CI/CD pipelines, sign code and artifacts, pin dependencies, and avoid unsafe deserialisation of untrusted data.

9. Security Logging and Monitoring Failures Without telemetry, incidents go unnoticed. Log security-relevant events, alert on anomalies, and store logs with integrity protection.

10. Server-Side Request Forgery (SSRF) Validate and allow-list URLs, restrict outbound network access, and disable automatic redirects to stop servers from probing internal resources.

Embed security into the delivery pipeline. Threat model during design, scan in CI, and run regular penetration tests alongside automated security checks.

Codingfigs offers application security assessments grounded in the OWASP framework. Contact our platform engineering team for an audit.

References

OWASP Top 10 (2021); OWASP Testing Guide; CWE Top 25; NIST Secure Software Development Framework.