

Common OWASP Top 10 Risk Categories in Open Source Packages

Between April and May, the Checkmarx security research team analyzed the most frequent common vulnerabilities (CVEs) identified across 20,000+ scans of projects on the Checkmarx One platform. The analysis focused on uncovering critical risks in application code and correlating them with the OWASP Top 10 categories for Web Applications. Here are the top three web application security risk categories found:



Trend 01

A01: Broken Access Control – 43.1%

Broken Access Control weaknesses can lead to unauthorized access to sensitive data and functionalities by allowing users to perform actions outside their intended permissions. This can result in unauthorized information disclosure, data modification, or destruction.

To prevent this risk, implement proper access control checks and enforce them consistently across the application. Use role-based access control (RBAC) and least privilege principles to restrict user permissions.

Examples of CVEs:

-  [CVE-2024-28849](#)
-  [CVE-2024-22234](#)

Trend 02

A03:2021: Injection – 29.5%



Injection vulnerabilities can result in unauthorized access, data leakage, and remote code execution by manipulating input data to execute unintended commands or queries.

To prevent this risk, use parameterized queries, stored procedures, and ORM frameworks to keep data separate from commands. Validate and sanitize all user inputs on the server side. Use positive server-side input validation and escape special characters.

Examples of CVEs:

-  [CVE-2024-29041](#)
-  [CVE-2024-31573](#)

More on Injection:

-  [What is SQL Injection, Examples and How to Prevent It](#)
-  [Injection Vulnerabilities – 20 Years and Counting](#)



Trend 03

A08:2021: Software and Data Integrity Failures – 27.4%

Software and Data Integrity Failures can lead to unauthorized code execution, data corruption, and system compromise by relying on untrusted sources for updates and plugins.

To prevent this risk, use digital signatures to verify software and data integrity, ensure trusted sources for libraries and dependencies, and implement robust CI/CD pipeline security measures to prevent unauthorized changes. Use supply chain security tools to verify components and ensure serialized data has integrity checks.

Examples of CVEs:

-  [CVE-2023-46234](#)
-  [CVE-2024-33883](#)

More on Injection:

-  [Software Supply Chain Security \(SSCS\)](#)
-  [The Hidden Dangers of Abandoned Digital Assets in Open-Source Ecosystems](#)

Take Preventive Action
Learn about Checkmarx One

Get Started

Investigate the [Checkmarx Supply Chain Threat Intelligence API](#) that delivers threat intelligence like this directly into your preferred dashboard or integrated development environment.