

BACK TO BASICS

PROTECTING AGAINST IDOR VULNERABILITIES

Insecure Direct Object Reference (IDOR) is a vulnerability caused by the lack of server-side verification of a user's authorisation to access a resource. It is widespread in web applications, as well as in all applications relying on [APIs](#) (web, mobile, native, AI, etc.). IDOR vulnerabilities are categorised into subtypes depending on whether they affect access rights to a function ([BFLA](#)), an object ([BOLA](#)), or a property ([BOPLA](#)).

These easily exploitable vulnerabilities rank **first in the [OWASP Top 10 2025](#)** for the most critical risks to web application security. **They can have devastating consequences, enabling attackers to retrieve large volumes of sensitive data** (see [CVE-2025-53640](#)) **or to act on behalf of another user** (see [CVE-2023-3287](#)).

This document outlines the best practices to implement in order to prevent and detect vulnerabilities over the course of the development lifecycle and during production.

1/ IMPLEMENTING « SECURE BY DESIGN » PRINCIPLES

- **Raise developer awareness** of the need to systematically verify the current user's authorisation when accessing a resource or performing an action. User authentication alone is never sufficient.
- **Enforce access control on the server side.** Client-side checks are insufficient.
- **Simplify access right verification** by using a known authorisation framework, and train the development team in its correct use.

- **Perform peer code reviews** using a checklist that includes access right checks.
- **Include negative authorisation test cases** in the testing strategy. They complement positive test cases performed with User A by confirming that User B is denied access.

2/ SEARCHING FOR IDOR VULNERABILITIES THROUGHOUT THE LIFECYCLE

- **Automate testing in [CI/CD pipelines](#)** using multiple analysis tools:
 - > static application security testing (SAST) for early detection ;
 - > dynamic application security testing (DAST) post-deployment in staging ;
 - > some SAST/DAST vendors offer AI-powered tools to better follow business logic and test for IDORs while reducing false positives.
- **Conduct periodic penetration tests and bug bounty programmes.** Because IDORs are often straightforward to discover, these approaches can be particularly effective for detecting them.

3/ PROTECTING SERVICES THROUGH SECURITY BY DESIGN

→ **Implement specific detection rules to identify potential IDOR exploitations:**

- > detect spikes in the usage of an HTTP endpoint within a short timeframe, with thresholds based on known baseline volumes (update these regularly);
- > check if a significant portion of the spike originates from the same IP address and/or authenticated account, then determine if it appears to be a sequential ID enumeration.

→ **Alert and respond to attacks by:**

- > blocking the IP address and/or malicious user account (or compromised account);
- > rapidly patching the IDOR vulnerability.