OWASP API Security Top 10

2023 Edition — Compiled Notes

Jordan Suber

October 19, 2025

Abstract

This document compiles the OWASP API Security Top 10 (2023) into a concise, practitioner-friendly reference with brief descriptions and practical controls for each risk. It is intended for architects, developers, AppSec, and platform teams building or assessing APIs.

Contents

1 Overview

Project overview

The OWASP API Security Project curates and maintains community guidance on the most critical API risks and practices to mitigate them. The 2023 edition lists ten risks that frequently surface in modern API landscapes and backends.

2 Top 10 at a Glance (2023)

Risk list

1. **API1:2023** Broken Object Level Authorization (BOLA)
2. **API2:2023** Broken Authentication
3. **API3:2023** Broken Object Property Level Authorization (BOPLA)
4. **API4:2023** Unrestricted Resource Consumption
5. **API5:2023** Broken Function Level Authorization (BFLA)
6. **API6:2023** Unrestricted Access to Sensitive Business Flows
7. **API7:2023** Server-Side Request Forgery (SSRF)
8. **API8:2023** Security Misconfiguration
9. **API9:2023** Improper Inventory Management
10. **API10:2023** Unsafe Consumption of APIs

3 API1:2023 — Broken Object Level Authorization (BOLA)

What it is

Missing or weak per-object access checks allow attackers to manipulate object identifiers (IDs, UUIDs) to access data they do not own.

Why it matters

Direct object references are common in REST/JSON and GraphQL; BOLA often leads to high-impact data exposure.

Key controls

- Enforce object ownership/tenant checks in business logic (not only at the route layer).
- Use parameterized, server-side lookups; never trust client-provided ownership hints.
- Add centralized authorization policies (ABAC/RBAC) and test with negative cases.

4 API2:2023 — Broken Authentication

What it is

Flaws in credential, token, or session handling enable account takeover or identity spoofing.

Key controls

- Strong authentication (passkeys/WebAuthn or MFA), short-lived tokens, secure refresh flows.
- Validate JWTs (aud/iss/exp/alg), rotate keys, and pin token audience.
- Lockouts, device binding, and anomaly detection (IP/UA/geo velocity).

5 API3:2023 — Broken Object Property Level Authorization (BOPLA)

What it is

Insufficient attribute/field-level checks expose or allow modification of sensitive properties.

Key controls

- Server-side allowlists for readable/writable fields; enforce output filtering.
- Separate read vs. write DTOs; validate patch/merge semantics.
- Contract tests to prevent sensitive fields (e.g., `isAdmin`) from being exposed.

6 API4:2023 — Unrestricted Resource Consumption

What it is

Endpoints allow unbounded use of CPU, memory, storage, or network resources.

Key controls

- Rate limiting, quotas, pagination, and maximum payload limits.
- Timeouts, circuit breakers, and bounded concurrency.
- Cost-aware designs for expensive operations (exports, reports, search).

7 API5:2023 — Broken Function Level Authorization (BFLA)

What it is

Users can invoke functions/actions their role should not access (e.g., privilege escalation).

Key controls

- Enforce RBAC/ABAC on every operation; avoid “hidden” admin routes.
- Align HTTP verbs and scope with least privilege; verify on backend, not UI.
- Policy-as-code + test suites for authorization matrices.

8 API6:2023 — Unrestricted Access to Sensitive Business Flows

What it is

Automation abuses high-value flows (e.g., checkout, coupon/credit, password reset) at scale.

Key controls

- Bot detection, rate shaping, proof-of-work or step-up auth on risky flows.
- Idempotency keys, nonce/replay protection, and transaction limits.
- Telemetry with alerting for anomalous volumes and sequences.

9 API7:2023 — Server-Side Request Forgery (SSRF)

What it is

API backend makes attacker-controlled outbound requests, reaching internal services/metadata.

Key controls

- Deny-by-default egress; strict URL/IP/hostname allowlists and DNS pinning.
- Block link-local and internal address ranges; disable HTTP redirects.
- Use metadata service v2/IMDSv2 and network segmentation.

10 API8:2023 — Security Misconfiguration

What it is

Insecure defaults or drift across gateways, runtimes, and cloud services.

Key controls

- Baseline hardening, CIS benchmarks, and configuration-as-code.
- Secure CORS, headers, TLS, error handling; remove debug endpoints.
- Continuous compliance checks and drift detection.

11 API9:2023 — Improper Inventory Management

What it is

Unknown or stale APIs/versions, missing documentation, and shadow endpoints.

Key controls

- Maintain authoritative API catalog (specs, versions, data classifications).
- Decommission legacy versions with sunset headers and routing controls.
- Discover shadow APIs via gateways, WAF logs, and traffic analysis.

12 API10:2023 — Unsafe Consumption of APIs

What it is

Trusting upstream/third-party APIs without validation or defense-in-depth.

Key controls

- Validate and sanitize all upstream responses; schema-validate inputs/outputs.
- Constrain permissions, timeouts, retries, and fallbacks per dependency.
- Threat-model suppliers; isolate and monitor integrations.

Appendix: 2019 Top 10 (for reference)

Risk list (2019)

1. API1:2019 Broken Object Level Authorization
2. API2:2019 Broken User Authentication
3. API3:2019 Excessive Data Exposure
4. API4:2019 Lack of Resources & Rate Limiting
5. API5:2019 Broken Function Level Authorization
6. API6:2019 Mass Assignment
7. API7:2019 Security Misconfiguration
8. API8:2019 Injection
9. API9:2019 Improper Assets Management
10. API10:2019 Insufficient Logging & Monitoring