OWASP TOP 10

OWASP Top 10 (2021) — Detailed & Beginner-Friendly Explanation

OWASP (Open Web Application Security Project) publishes the **Top 10 most critical security risks** for web applications. These are industry-standard risks every pentester, bug bounty hunter, or developer must know.

---

1. A01: Broken Access Control

What it means:

Access control ensures users can only access what they are allowed to. When broken, attackers can act as admins, read private data, or modify accounts.

Common issues

- IDOR (Insecure Direct Object Reference)
- Missing role checks (admin panel accessible to normal users)
- Forced browsing to restricted URLs

Impact:

Account takeover, data leaks, privilege escalation.

Example:

Changing `/user/123/profile` → `/user/124/profile` and accessing someone else's data.

---

2. A02: Cryptographic Failures (Sensitive Data Exposure)

What it means:

Improper protection of sensitive data — passwords, tokens, financial data — due to weak or missing encryption.

Common issues

- Not using HTTPS
- Weak hashing (MD5, SHA1)
- Storing credentials in plaintext
- Hardcoded secrets in code

Impact:

Credentials leak, full system compromise, MITM attacks.

Example:

A login page sending username & password over HTTP.

---

3. A03: Injection

What it means:

Unsanitized input is interpreted as code — SQL, NoSQL, LDAP, OS commands, XML, SMTP.

Types of injection

- SQL Injection (most common & most dangerous)
- Command Injection
- NoSQL Injection
- LDAP Injection
- HTML/JS Injection (XSS is separate category now)

Impact:

Database dump, data deletion, full server control (in severe cases).

Example:

`' OR 1=1 --` bypassing login.

---

4. A04: Insecure Design

What it means:

Flaws in architecture before the code is even written. No security thinking during planning.

Common issues

- No rate limiting → brute force attacks
- Admin features in public APIs
- Lack of threat modeling

Impact:

System-wide failures that cannot be patched easily.

Example:

A banking website allowing unlimited OTP attempts.

---

5. A05: Security Misconfiguration

What it means:

Improper configuration of servers, databases, cloud, frameworks, or APIs.

Common issues

- Default credentials (admin/admin)
- Debug mode enabled in production
- Unnecessary services running
- Directory listing enabled

Impact:

Remote code execution, information leaks, easy exploitation.

Example:

Apache showing `/var/www/html` file list to public.

---

6. A06: Vulnerable and Outdated Components

What it means:

Using old libraries, frameworks, OS packages, or dependencies with known security issues.

Common issues

- Outdated WordPress plugins
- Old JavaScript libraries (e.g., jQuery 1.x)
- Using unsupported server software

Impact:

Attackers exploit old, known CVEs to take over systems.

Example:

Log4j vulnerability (Log4Shell).

---

7. A07: Identification & Authentication Failures

What it means:

Weak login mechanisms or session handling.

Common issues

- Poor password policy
- Missing MFA
- Session IDs in URL
- Session not invalidated after logout
- Credential stuffing vulnerability

Impact:

Account takeover, stolen sessions.

Example:

Using predictable session tokens like `SESSIONID=12345`.

---

8. A08: Software & Data Integrity Failures

What it means:

Untrusted code, unverified updates, or insecure CI/CD pipelines.

Common issues

- Unsigned software updates
- Insecure deserialization
- Dependency confusion attacks

Impact:

Server takeover, supply chain attacks.

Example:

Installing a package from a fake NPM repository.

---

9. A09: Security Logging & Monitoring Failures

What it means:

Insufficient visibility into attacks.

Common issues

- No logging for important events
- Logs not stored securely
- No alerting system
- Weak incident response workflow

Impact:

Attacks go unnoticed for months.

Example:

No alert when 5000 failed logins happen in 10 minutes.

---

10. A10: Server-Side Request Forgery (SSRF)

What it means:

Application lets users control URLs that the server fetches → attacker tricks server into making internal network requests.

Common issues

- Image upload that fetches file from URL
- PDF generators pulling external content

Impact:

Internal network scanning, metadata access, cloud instance takeover.

Example:

Sending URL:

`http://127.0.0.1/admin`

→ Server shows admin panel content.

Quick Summary Table

OWASP Risk	What It Means	Impact
A01 Broken Access Control	Wrong access permissions	Account takeover
A02 Cryptographic Failures	Weak/no encryption	Data leaks
A03 Injection	Malicious input interpreted as code	DB/server compromise
A04 Insecure Design	Architect-level flaws	System-level failures
A05 Misconfiguration	Bad server/app setup	Easy exploitation
A06 Outdated Components	Old vulnerable dependencies	Known CVE attacks
A07 Auth Failures	Weak login/session	Account takeover
A08 Integrity Failures	Untrusted updates/code	Supply chain attacks
A09 Logging Failures	No monitoring	Undetected breaches
A10 SSRF	Server fetches attacker URL	Internal network access