OWASP TOP 10 API

Security

Mouad Boufas
@mouad-boufas

What is OWASP?

OWASP (Open Web Application Security Project) is a ***non-profit organization*** focused on ***improving web and API security***. It provides free resources, tools, and ***security best practices*** to help developers build secure applications.

Why API Security Matters ?

- APIs are the ***backbone of modern applications***—they connect web apps, mobile apps, and cloud services.
- Unsecured APIs can lead to ***data breaches***, ***account takeovers***, and ***business disruptions***.

#1 Broken Object Level Authorization (BOLA)

- The #1 API security risk—responsible for many data breaches.
- Occurs when an API ***doesn't properly verify user permissions*** for accessing or modifying data.

How To Prevent Bola Attacks ?

- Implement ***proper authorization checks*** for every API request.
- Use ***Role-Based Access Control*** (RBAC) or Attribute-Based Access Control (ABAC).
- Never trust client-side authorization—always ***validate permissions*** on the backend.

#2 Broken Authentification

- Broken authentication occurs ***when an attacker can bypass authentication mechanisms*** to gain unauthorized access.
- Common causes include weak passwords, incorrect session handling, or predictable login paths.

How To Prevent Broken Auth ?

- Enforce ***strong password policies*** (e.g., 2FA, password complexity).
- Use ***OAuth2 or JWT*** for secure token-based authentication.
- Implement ***session expiration*** and ***secure session management***.
- ***Limit login attempts*** and use CAPTCHA to prevent brute force attacks.

#3 Broken Object Property Level Authorization

Broken Object Property Level Authorization ***occurs when an API allows users to access or modify properties of an object they are not authorized to***.

How to Prevent Broken Object Property Authorization ?

- ***Encrypt sensitive properties*** in API responses to minimize exposure.
- Enforce ***strict validation*** for all object properties before processing the request.

#4 Unrestricted Resource Consumption

Unrestricted Resource Consumption ***occurs when an API allows an attacker to consume excessive resources*** (e.g., CPU, memory, bandwidth) without limits.

How to Prevent Unrestricted Resource Consumption ?

- Set ***rate limits*** and request quotas to control resource usage.
- Use ***API throttling*** to limit the number of requests from a user.
- Implement ***request validation*** to avoid resource-intensive queries.

#5 Broken Function Level Authorization

- This occurs when an API ***fails to properly enforce user roles and permissions*** for certain functions.
- Attackers can ***access admin-only or restricted functionalities*** by manipulating API requests.

How to Prevent Broken Function Level Authorization ?

- Implement ***Role-Based Access Control (RBAC)*** to restrict access based on user roles.
- Use ***least privilege principles***, granting users only the permissions they need.

#6 Unrestricted Access to Sensitive Business Flows

- This occurs when an ***API exposes critical business processes*** (e.g., financial transactions, order processing) without proper security controls.
- Attackers can ***exploit these flows*** by ***automating*** or ***abusing*** them at scale, leading to ***fraud***, ***spam***, or ***service disruptions***.

How to Prevent Unrestricted Access to Sensitive Business Flows ?

- Use ***workflow-based security*** (e.g., require multiple validation steps for high-risk actions).
- Implement ***rate limiting*** & ***CAPTCHA*** to prevent automated abuse.

#7 Server Side Request Forgery

- SSRF happens when an API ***allows unauthorized requests*** to internal or external systems.
- Attackers can exploit this to ***access internal services***, ***steal data***, or ***scan networks***.

How to Prevent Server Side Request Forgery ?

- ***Restrict outbound requests*** to only allowed domains or IPs.
- Validate and ***sanitize user input*** to prevent malicious URLs.
- Use ***allowlists*** instead of accepting any URL input.

#8 Security Misconfiguration

- Happens when APIs are ***improperly configured***, exposing sensitive data or functionality.
- Can lead to ***data leaks***, ***unauthorized access***, or ***system compromise***.

How to Prevent Security Misconfiguration ?

- Use ***secure configurations*** for servers, frameworks, and APIs.
- Limit ***error messages*** to avoid leaking sensitive data.
- ***Disable default accounts*** & ***credentials*** before deployment.

#9 Improper Inventory Management

- Happens when APIs ***lack proper tracking and documentation***, leading to ***exposed outdated*** or ***unprotected endpoints***.
- Attackers can ***discover and exploit forgotten APIs*** (e.g., old, unmonitored versions).

How to Prevent Improper Inventory Management ?

- Regularly ***deprecate and remove old APIs*** that are no longer in use.
- Implement ***strict access*** controls for all API endpoints.
- Use ***API gateways*** to manage and monitor API traffic.

#10 Unsafe Consumption of APIs

- Happens ***when an API blindly trusts external APIs without proper validation***, leading to security risks.
- Attackers can ***exploit insecure third-party APIs*** to inject malicious data or gain unauthorized access.

How to Prevent Unsafe Consumption of APIs ?

- ***Validate and sanitize*** data from third-party APIs.
- Use ***authentication*** and ***encryption*** when communicating with external APIs.
- Implement ***strict error handling*** to prevent crashes or data leaks.

Want to Dive Deeper into API Security ?

- https://owasp.org/www-project-api-security -- ***Official OWASP documentation on API security risks.***

- https://www.postman.com/api-security
  -- ***API Security Testing with Postman***

- https://portswigger.net/web-security -- ***PortSwigger Web Security Academy***

Follow me to get more Information and tips like this.

Mouad Boufas

@mouad-boufas