Practical DevSecOps

UPDATED LIST

OWASP TOP 10 API Security Risks - 2023

Also, Understand Best Practices to prevent it!

practical-devsecops.com | #certifiedapisecurityprofessional

1. Broken Object Level Authorization

Attackers can exploit vulnerable API endpoints by manipulating object IDs within requests. Object IDs can be sequential integers, UUIDs, or generic strings and are easily identifiable in the request target, headers, or payload.

BEST PRACTICES TO PREVENT IT:

- Implement a strong authorization mechanism based on user policies and hierarchies.
- Perform authorization checks for every action on records.
- Use random and unpredictable GUIDs as record IDs.
- Test the authorization mechanism thoroughly before deploying changes.

2. Broken Authentication

Broken authentication and session management can enable attackers to impersonate valid users and compromise data privacy and infrastructure.

BEST PRACTICES TO PREVENT IT:

- Implement two-factor authentication.
- Secure session management.
- Enforce strict password policies.
- Account lockouts and brute-force protection:
- User account monitoring.
- Security incident response.

3. Broken Object Property Level Authorization

Broken object property level authorization allows unauthorized access to sensitive object properties, which can lead to data exposure, loss, corruption, and potential privilege escalation or account takeover.

BEST PRACTICES TO PREVENT IT:

- Implement strong object property level authorization controls.
- Regularly review and validate access permissions.
- Apply the principle of least privilege.
- Conduct thorough security testing and code reviews.

4. Unrestricted Resource Consumption

Unrestricted resource consumption occurs when an API allows excessive or uncontrolled use of system resources, leading to a degradation of service or a complete service disruption for legitimate users.

BEST PRACTICES TO PREVENT IT:

- Implement proper rate limiting and throttling mechanisms.
- Set resource consumption limits and enforce them.
- Conduct regular performance testing and monitoring.
- Employ caching and optimization techniques.

5. Broken Function-Level Authorization

Broken function-level authorization involves unauthorized access to sensitive functions or data due to misconfigured or weak access controls. This potentially allows actors to perform escalated actions, leading to data breaches or application hijacking.

BEST PRACTICES TO PREVENT IT:

- Implement strict access controls to ensure appropriate role-based access to sensitive data.
- Use an automated access control mechanism.
- Implement regularly scheduled device updates.
- Stay current with information and vulnerability feeds and exploit databases.

6. Unrestricted Access to Sensitive Business Flows

Unrestricted access to sensitive business flows is a significant API security vulnerability, enabling unauthorized users to manipulate critical operations, bypass business rules, and compromise sensitive data.

BEST PRACTICES TO PREVENT IT:

- Implement strong access controls and permissions.
- Input validation to prevent attacks.
- Enforce business logic checks.
- Encrypt sensitive data.
- Monitor access logs.

7. Server Side Request Forgery

Server-Side Request Forgery (SSRF) is an API security vulnerability where attackers manipulate a server to make unintended requests to internal or external resources. It can lead to unauthorized data exposure, service disruption, and further exploitation.

BEST PRACTICES TO PREVENT IT:

- Validate inputs rigorously.
- Use whitelisting to limit server access.
- Employ network-level protections.
- Enforce access controls strictly.
- Regularly update server software.

8. Security Misconfiguration

Improperly configured systems and software pose risks to APIs. Common security misconfigurations include insufficiently secured cryptography protocols, incorrect file permission configuration, and poor endpoint protection.

BEST PRACTICES TO PREVENT IT:

- Follow the OWASP secure coding principles and guidelines.
- Enforce strict access controls.
- Use a secure configuration management. process that reduces the attack surface of the API.

9. Improper Inventory Management

Improper inventory management in an API creates security vulnerabilities, allowing attackers to breach data, manipulate inventory, and cause financial losses by exploiting weaknesses like insufficient validation and access controls.

BEST PRACTICES TO PREVENT IT:

- Strong access controls.
- Thorough input validation.
- Secure authentication and authorization.
- Regular monitoring and auditing.
- Encryption for sensitive inventory data.

10. Unsafe Consumption of APIs

Unsafe API consumption occurs when developers trust third-party API data more than user input, leading to weaker security standards. Attackers exploit this vulnerability by targeting integrated third-party services instead of directly attacking the API.

BEST PRACTICES TO PREVENT IT:

- Validate and sanitize API data.
- Apply consistent security standards.
- Assess and update third-party service security.
- Implement strict access controls.
- Stay informed about API security.

Become an API Security Expert with Us!

Certified API Security Professional

Link in the description

practical-devsecops.com | #certifiedapisecurityprofessional

Practical DevSecOps

Making Product Security Accessible to Everyone