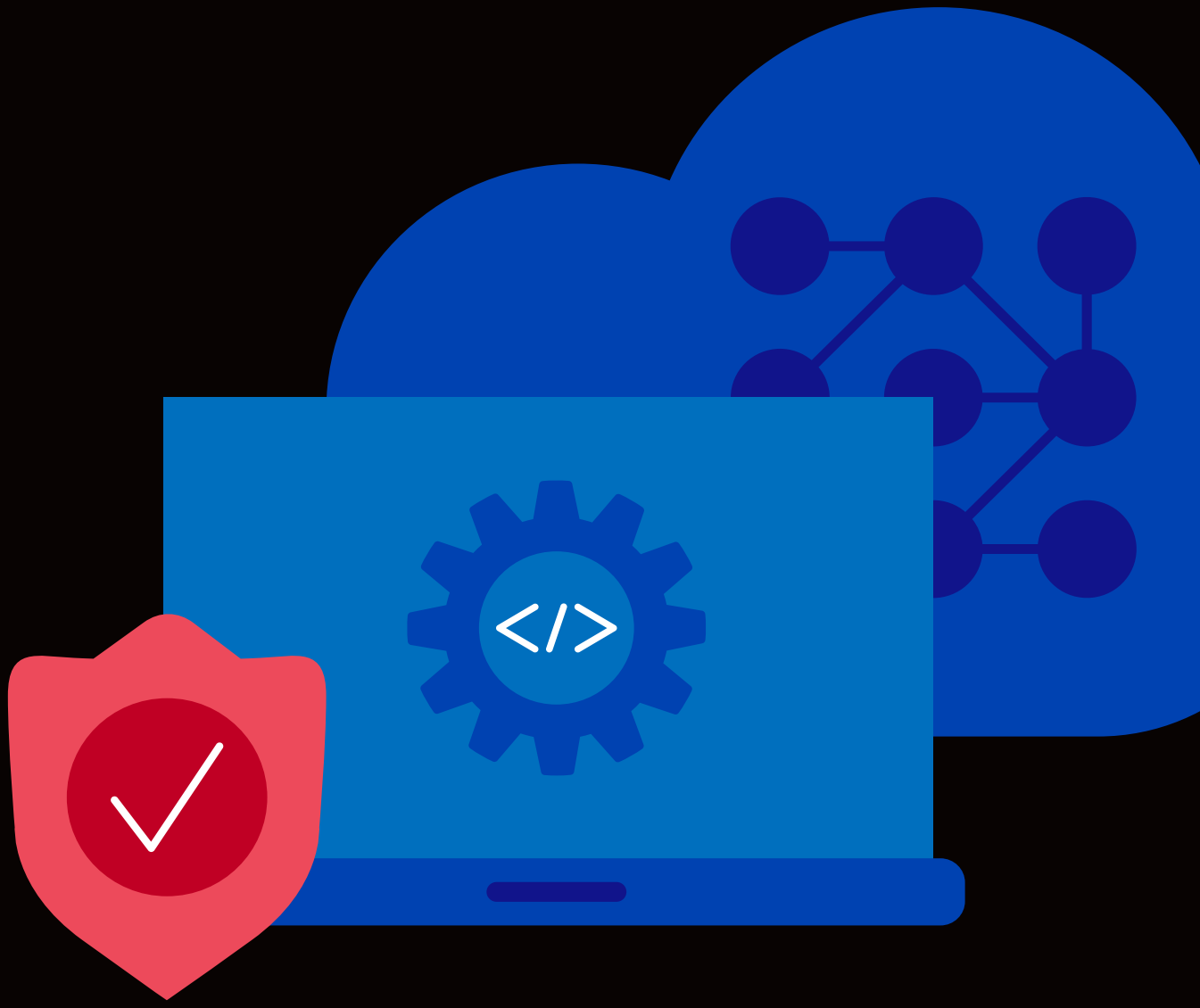
Article

OWASP API Security Top 10 Overview and Best Practices

APIs play a critical role in modern application architectures, and this OWASP project focuses on awareness of common API security weaknesses.



INSERT PARTNER LOGO

Size to be visually equal to F5 logo.
Align to left edge & center vertically.

The goal of the OWASP (Open Worldwide Application Security Project) list of the Top 10 API Security Risks is to educate those involved in API development and maintenance and increasing awareness of common API security weaknesses. APIs have increasingly become a target for attackers and OWASP's API security project focuses on strategies and solutions to understand and mitigate the unique vulnerabilities and security risks associated with APIs.

What Are APIs?

APIs have increasingly become a target for attackers.

APIs (Application Programming Interfaces) are fundamental to the development of modern applications, as they facilitate the ability of applications to communicate and exchange data with other applications, services, or platforms. APIs are a key part of an app modernization strategy and are the foundation of mobile apps. They enable businesses to easily integrate with external platforms and third-party services and build comprehensive solutions by connecting various components. This promotes a modular approach to app development that enables developers to leverage existing services and functionality, promote code reuse, accelerate development cycles, and enhance productivity.

APIs also expand the risk surface and specifically introduce unforeseen risk due to the nature of their interdependencies across multi-cloud architectures. Like web apps, APIs are susceptible to vulnerability exploits, abuse from automated threats, denial of service, misconfiguration, and attacks that bypass authentication and authorization controls.

By their nature, APIs expose critical business logic and sensitive information, such as user data, authentication credentials, and financial transactions, and have increasingly become a target for attackers; in particular, the login, create account, add to cart, and money transfer functions. APIs can become entry points for attackers seeking to exploit vulnerabilities or weaknesses, or to expose underlying infrastructure and resources.

What Are API Security Best Practices?

Robust API security measures are necessary to protect data from unauthorized access, manipulation, or exposure to ensure privacy and maintain the trust of users and stakeholders, as well as ensure the confidentiality, integrity, and availability of APIs. Best practices for API security include the following:

- **Implement strong authentication and authorization.** Enforce proper authorization checks to ensure that authenticated clients have the necessary permissions to access specific resources or perform certain actions. Use granular access controls to limit access to sensitive API endpoints or data, as well as relevant objects and functions.
- **Validate input and output encoding.** Validate and sanitize all input received from API clients to prevent injection attacks and encode output appropriately to prevent the execution of malicious scripts.
- **Use secure communication.** Employ secure protocols for transmitting data between API clients and servers and encrypt sensitive information in transit and at rest to ensure the confidentiality and integrity of data.
- **Implement rate limiting and throttling.** Enforce limits on the number of requests that API clients can make within a specified time frame to prevent excessive usage or unauthorized access attempts, such as Distributed Denial of Service (DDoS) and brute force attacks.
- **Conduct regular security testing and auditing.** Perform regular security assessments, penetration testing, and code reviews to identify and address potential vulnerabilities in your APIs, and conduct security audits to detect weaknesses and ensure compliance with industry standards and mandates. This is especially important due to the interdependencies of APIs and underlying frameworks and libraries.
- **Enforce schema and protocol compliance.** Automatically creating and enforcing a positive security model with OpenAPI specifications is a valuable tool to ensure a consistent security policy.
- **Dynamically discover and continuously assess APIs.** API proliferation has resulted in unaccounted for or unmaintained APIs, including shadow APIs. Security controls need to constantly inventory and protect APIs using zero trust and least privilege access paradigms to mitigate unforeseen risk of third-party interdependencies.
- **Comprehensive threat detection.** APIs are subject to, and need to be protected from, a variety of threats including exploits, misconfiguration, bots, fraud, and abuse.

OWASP API Security Top 10 – 2023

The OWASP API Security Top 10 – 2023 was formulated to increase awareness of common API security weaknesses and to help developers, designers, architects, managers, and others involved in API development and maintenance maintain a proactive approach to API security.

The OWASP API Security Top 10 risks for 2023 are:

1. **Broken object level authorization.** This security vulnerability occurs when an application fails to properly enforce access controls at the object or data level, allowing an attacker to manipulate or bypass authorization checks and grant unauthorized access to specific objects or data within an application. This can happen due to improper implementation of authorization checks, lack of proper validation, or bypassing of access controls. Every API endpoint that receives an ID of an object, and performs any action on the object, should implement object-level authorization checks to validate that the logged-in user has permissions to perform the requested action on the requested object.
2. **Broken authentication.** Authentication mechanisms in an API are often implemented incorrectly, allowing attackers to gain unauthorized access to user accounts or sensitive data, or perform unauthorized actions. It typically arises due to improper implementation or configuration of authentication processes, weak password policies, session management flaws, or other weaknesses in the authentication workflow.
3. **Broken object property level authorization.** This threat occurs when an API fails to properly enforce access controls and authorization checks at the object property level. An API endpoint is vulnerable to these attacks if it exposes properties of an object that are considered sensitive and should not be read by the user, an exploit sometimes referred to as excessive data exposure. An API endpoint is also vulnerable to these attacks if it allows a user to change, add, or delete the value of a sensitive object's property, an exploit sometimes called mass assignment.
4. **Unrestricted resource consumption.** This attack, also called resource exhaustion, involves exploiting weaknesses in the API implementation to intentionally consume an excessive amount of resources, such as CPU, memory, bandwidth, or other system resources. This denial of service (DoS) degrades the performance or availability of the API or the underlying system and can lead to downtime.
5. **Broken function level authorization.** This threat occurs when an API fails to properly enforce authorization checks at the function or operation level, allowing attackers to access unauthorized functionality. Implementing proper authorization checks can be confusing since modern applications can define many types of functional roles and groups and involve complex user hierarchies, which attackers can manipulate.

6. **Unrestricted access to sensitive business flows.** This attack occurs when an API lacks proper access controls or authorization checks, allowing attackers to automate access to sensitive business flows backed by the API. These business flows might support mass purchasing of high-value, low-inventory products such as tickets or sneakers, which can be resold at a mark-up on secondary markets. Attackers often retool their attacks using sophisticated automation toolkits and may pivot to target business logic behind APIs if the target's web apps are adequately protected by anti-automation defenses.
7. **Server-side request forgery (SSRF).** This vulnerability results when an attacker identifies a vulnerable API endpoint that accepts user-supplied URLs or performs server-side requests to external resources. The attacker crafts malicious requests that specify URLs of internal resources or systems that the attacker wishes to target. Unaware of the malicious intent, the server performs the server-side request to the specified URL, potentially exposing sensitive information or services.
8. **Security misconfiguration.** Attackers attempt to find unpatched flaws, common endpoints, services running with insecure default configurations, or unprotected files and directories to gain unauthorized access to the API. This vulnerability can result when appropriate security hardening is missing across any level of the API stack, from the network to the application level, or if there are improperly configured permissions on cloud services. Misconfiguration impacts web apps and APIs, and is an increasing risk as architecture continues to decentralize and become distributed across multi-cloud environments.
9. **Improper inventory management.** APIs are subject to changes and updates over time, but outdated or insecure API versions may remain in production, or older endpoints may be left running unpatched or using weaker security requirements, increasing the risk of security breaches. A lack of proper inventory management makes it difficult to track which versions are in use, which ones are outdated or deprecated, and which vulnerabilities have been addressed. Shadow and Zombie APIs pose significant risks, underscoring the importance of continuous discovery and automated protections.
10. **Unsafe consumption of APIs.** Developers tend to trust data received from third-party APIs, particularly from APIs offered by well-known companies, and adopt weaker security requirements for this data in terms of input validation and sanitization or transport security. Unsafe consumption can also occur when APIs are accessed over insecure protocols or when proper encryption mechanisms are not used, leading to eavesdropping, data interception, and unauthorized access to sensitive information.

F5 addresses the risks identified in the OWASP API Security Top 10 with solutions that protect the growing attack surface and emerging threats as apps evolve and API deployments increase.

The Case for Integrated Security Controls

F5 ADDRESSES OWASP SECURITY RISKS

F5 supports the OWASP Foundation and its dedication to improving software security and raising awareness of web application security risks and vulnerabilities at multiple levels. Indeed, there are security risks common to both apps and APIs that bear consideration when implementing security solutions. For example:

- Weak authentication/authorization controls
- Misconfiguration
- Business logic abuse (credential stuffing, account takeover)
- Server-side request forgery (SSRF).

F5 addresses the risks identified in the OWASP API Security Top 10 with solutions that protect the growing attack surface and emerging threats as apps evolve and API deployments increase. F5 Web Application and API Protection (WAAP) solutions defend the entirety of the modern app attack surface with comprehensive protections that include WAF, API Security, L3-L7 DDoS mitigation, and bot defense against automated threats and fraud. The distributed platform makes it simple to deploy consistent policies and scale security across your entire estate of apps and APIs regardless of where they're hosted, and integrate protections into the API lifecycle and broader security ecosystems.

F5 provides hybrid security architectures that consistently and continuously protect apps and APIs from core to cloud to edge. F5 solutions dynamically discover and automatically protect critical business logic behind APIs using threat intelligence, ML-based security, and zero trust principles, providing the resilience and agility necessary to compete in the API-driven digital economy.

F5 Web Application Firewall solutions also block and mitigate a broad spectrum of risks identified by OWASP Top 10, a widely recognized list of the most critical web application security risks. APIs, like web apps, are susceptible to misconfiguration and automated threats, and can be targeted by vulnerability exploits, SSRF, and attacks that attempt to bypass authentication and authorization controls. F5 WAF solutions combines signature and behavioral protections, including threat intelligence from F5 Labs and ML-based security, to keep pace with emerging threats; they can also be integrated with specialized bot defense controls.

These solutions ease the burden and complexity of consistently securing applications across clouds, on-premises, and edge environments, while simplifying management via a centralized SaaS infrastructure. F5 WAFs also streamline app security by integrating protections into development frameworks and CI/CD pipelines with core security functionality, centralized orchestration, and oversight via a single dashboard with a 360-degree view of app performance and security events across distributed applications.

F5 also offers solutions to address the risks outlined in OWASP's Automated Threats to Web Applications Project. F5 Distributed Cloud Bot Defense prevents fraud and abuse that can bypass existing bot management solutions and provides real-time monitoring and intelligence as well as ML-based retrospective analysis to protect organizations from automated attacks, without inserting user friction or disrupting the customer experience. Distributed Cloud Bot Defense maintains effectiveness regardless of how attackers retool, whether the attacks pivot from web apps to APIs or attempt to bypass anti-automation defenses by spoofing telemetry or using human CAPTCHA solvers.

F5 also offers multi-tiered DDoS protection for advanced online security as a managed, cloud-delivered mitigation service that detects and mitigates large-scale network, protocol, and application-targeted attacks in real time; the same protections are available as on-premises hardware, software, and hybrid solutions as well. F5 Distributed Cloud DDoS Mitigation defends against volumetric and application-specific layer 3-4 and advanced layer 7 attacks before they reach your network infrastructure and applications.

For more information, visit [www.partnername.com] or contact [emailname@partnername.com] to schedule a demo.