WHITE PAPER

Mitigating API Security Threats

How FortiWeb Protects against the OWASP Top 10 API Security Risks

Executive Summary

With the increased reliance on application programming interfaces (APIs) comes the need to secure these vital connectors. The Open Web Application Security Project (OWASP) has long been a trusted authority in identifying and addressing security vulnerabilities in web applications, and the OWASP API Security Top 10 provides guidance on today's most critical API security risks. Fortinet FortiWeb API security capabilities protect against the risks detailed in the OWASP API Security Top 10 list.

Why API Security Matters

APIs are the backbone of modern software development. They are necessary for integration between services, popular in hybrid and public cloud operations, and essential for Software-as-a-Service (SaaS) to link the software supply chain. Using APIs, developers can create complex, multi-dimensional technology solutions by seamlessly exchanging data and functions between applications. APIs make it possible for people to use smartphone applications for everything from booking flights to checking the weather or ordering a pizza.

Over 40% of organizations experienced vulnerability exploitation or credential dumping attempts targeting their APIs.[1]

Although APIs provide unprecedented agility and functionality, they also introduce new cybersecurity challenges. APIs are used to connect to services, retrieve information, and execute actions, and this connectivity can expose them to security threats. As APIs continue to proliferate across industries, security is no longer simply a technical concern. Organizations that fail to protect their APIs face risks from data leaks, which could include financial losses, damaged reputations, and legal consequences. Customers and partners expect organizations to protect their data and interactions, so a lapse in API security can erode trust quickly.

What Is the OWASP API Security Top 10?

Since its inception in 2001, the OWASP has been at the forefront of identifying and mitigating vulnerabilities in web applications. Recognizing the increasing importance of APIs, OWASP introduced the API Security Top 10 as a companion to its Web Application Top 10. This curated list details critical security risks to APIs, which range from inadequate authentication and authorization to data exposure and injection attacks. The Top 10 serves as a guide for developers, security professionals, and organizations to understand, prioritize, and address API-specific vulnerabilities.

The OWASP API Security Top 10 is a collaborative effort that involves security experts, developers, and organizations from around the world. The process begins with the collection of real-world data and experiences to ensure the Top 10 reflects the most current and pressing threats. The final list is carefully curated based on criteria such as prevalence, exploitability, detectability, and business impact. It provides clear insights into the vulnerabilities that organizations should prioritize in their security strategies.

2023 OWASP API Security Top 10 Vulnerabilities

The OWASP API Security Top 10 for 2023 highlights critical vulnerabilities that pose significant risks to API security. Understanding these vulnerabilities and taking proactive measures to mitigate them is crucial for safeguarding applications and data.

OWASP API Security Top 10	Description	Fortinet FortiWeb Protection
API1:2023 Broken Object Level Authorization (BOLA)	APIs tend to expose endpoints that handle object identifiers, creating a wide attack surface of object-level access control issues. Object-level authorization checks should be considered in every function that accesses a data source using an ID from the user. Broken object-level authorization (BOLA) occurs when an application fails to validate a user's permissions correctly, allowing unauthorized access to objects. This vulnerability can lead to data leaks, unauthorized updates, or data destruction. To prevent BOLA, implement robust authorization mechanisms, validate user actions, and conduct thorough security testing before deploying changes.	The responsibility to address this exploit is usually with the application server, which introduces additional security through an API gateway to limit the attack surface scope. Enforce a strong authentication schema using dynamic or preferably JSON Web Token (JWT) keys. Restrict HTTP referrers and source IPs if possible. Enable API discovery and protection with personally identifiable information (PII) identification to help identify the highest priority API endpoints to further audit access.
API2:2023 Broken Authentication	Authentication mechanisms are often implemented incorrectly, allowing attackers to compromise authentication tokens or to exploit implementation flaws to assume other user's identities temporarily or permanently. Compromising a system's ability to identify the client/user, compromises API security overall.	Enable client management for session tracking. Use brute force detection to limit the number of requests together with credential stuffing protection, session fixation protection and session timeout. Use cookie security and man in the browser (MiTB) protection and consider using API gateway capabilities such as API key verification and allow user groups.
API3:2023 Broken Object Property Level Authorization	This category combines API3:2019 Excessive Data Exposure and API6:2019 - Mass Assignment, focusing on the root cause: the lack of or improper authorization validation at the object property level. This issue leads to information exposure or manipulation by unauthorized parties. This newly added vulnerability covers cases in which users can access an object's properties without proper authorization. It encompasses aspects of Excessive Data Exposure and Mass Assignment from the previous OWASP API Security Top 10 (2019). Prevent this vulnerability by scrutinizing user access privileges, avoiding generic methods, and enforcing schema-based validation.	Enable machine learning (ML)-based API discovery and protection to discover all APIs endpoints and identify which holds PII data. Further audit PII API endpoints with development to verify they're well validated. Enable additional protocol limits and other HTTP best practice standards in JSON, XML, and SOAP protocol policies.
API4:2023 Unrestricted Resource Consumption	Satisfying API requests requires resources such as network bandwidth, CPU, memory, and storage. Other resources such as emails/SMS/phone calls or biometrics validation are made available by service providers via API integrations and paid for per request. Successful attacks can lead to denial of service or operational cost increases. Unrestricted resource consumption can lead to performance degradation when system resources are needlessly consumed. This vulnerability was previously known as Lack of Resources and Rate Limiting. Mitigate it by enforcing rate limiting, setting maximum input payload sizes, and implementing server-side request validations.	Use the FortiWeb API gateway to limit users who can access the API and define the specific API endpoints that are allowed. Enforce rate-limiting rules and enable distributed denial-of-service (DDoS) and bot rules to further limit abuse.
API5:2023 Broken Function Level Authorization	Complex access control policies with different hierarchies, groups, and roles, and an unclear separation between administrative and regular functions, tend to lead to authorization flaws. By exploiting these issues, attackers can gain access to other users' resources and administrative functions. Broken function-level authorization occurs when APIs grant normal users access to administrative functions. Proper access control policies and role-based authorization checks are essential to prevent this vulnerability.	Use the FortiWeb API gateway to enforce authorized URLs, methods and headers, verify keys, and define allowed user groups. Use additional block/allow IP rules with geo IP, IP reputation, and quarantine IPs to limit access to APIs.

OWASP API Security Top 10	Description	Fortinet FortiWeb Protection
API6:2023 Unrestricted Access to Sensitive Business Flows	APIs vulnerable to this risk expose a business flow, such as buying a ticket or posting a comment without compensating for how the functionality could harm the business if used excessively in an automated manner. This issue doesn't necessarily come from implementation bugs. This new addition to the OWASP API Security Top 10 emphasizes the need to secure and limit access to sensitive business flows exposed through APIs. Automated attacks can bypass traditional security measures, impacting the experience for both businesses and users. To address this issue, employ techniques like device fingerprinting, integrate CAPTCHA solutions, and block Tor requests to detect and mitigate automated attacks.	Use bot detection policies to identify malicious bots and automated tools used to abuse APIs. Enable known bot detection and threshold-based detection policies together with DDoS policies to identify and limit the number of requests per client.
API7:2023 Server-Side Request Forgery (SSRF)	SSRF flaws can occur when an API is fetching a remote resource without validating the user-supplied URI. This situation makes it possible for an attacker to coerce the application to send a crafted request to an unexpected destination, even when it is protected by a firewall or a VPN. SSRF vulnerabilities occur when an API fetches internal server resources without proper URL validation. Attackers exploit this vulnerability to access sensitive data. Prevent SSRF by implementing input data validations, maintaining allow lists, and disabling HTTP redirections.	Use the FortiWeb ML-based API discovery and protection to automatically build and enforce the API schema and automatically build mathematical models for every key pair value in API endpoints to make sure only acceptable values are passed through. Enable attack signatures to block SSRF attacks.
API8:2023 Security Misconfiguration	APIs and the systems that support them typically contain complex configurations, which are meant to make the APIs more customizable. Software and DevOps engineers can miss these configurations or may not follow security best practices when it comes to configuration, which opens the door for different types of attacks.	Security misconfiguration can result from overlooking security best practices. Ensure proper configuration, apply security patches, follow secure communication practices, and set up cross-origin resource sharing (CORS) policies correctly to prevent this vulnerability. Enable ML-based API protection to discover API endpoints and identify PII data within them. Enforce blocking on both schema violations and threat protection violations to block anomalous requests and unsupported methods and request structure. Enable the FortiWeb information disclosure signatures to detect sensitive data that should not be accessible by users. Also enable FortiWeb signatures to protect against known vulnerabilities. Enable FortiWeb HTTP protocol constraints and XML limits to protect against anomalous requests trying to bypass the HTTP RFC definitions. Use allow methods and protected hosts to further limit how APIs are accessed and implement a CORS policy.
API9:2023 Improper Inventory Management	APIs tend to expose more endpoints than traditional web applications, so proper and updated documentation is important. And inventory of hosts and deployed API versions are also important to mitigate issues such as deprecated API versions and exposed debug endpoints. Improper inventory management arises when organizations lack clarity on their APIs and fail to document them adequately. Maintain up-to-date documentation detailing API hosts, environment, and access control. Secure exposed API versions and conduct risk analyses when new versions become available.	Introduce a new procedure to continuously scan APIs, identify all API endpoints and compare them against existing documentation to make sure all API endpoints are documented. Audit all API endpoints to verify whether it is necessary for them to be exposed externally. Confirm the API endpoints that include PII data and introduce additional security measures to protect those APIs. Use the FortiWeb ML-based API discovery and protection to introduce a new procedure to continuously scan and discover all API endpoints and to identify which API endpoints have PII. Enable and enforce schema protection and threat protection to secure APIs. Make sure all discovered API endpoints are documented and audited to verify that they're correctly exposed externally.

OWASP API Security Top 10	Description	Fortinet FortiWeb Protection
API10:2023 Unsafe Consumption of APIs	Developers tend to trust data received from third-party APIs more than user input, and so they tend to adopt weaker security standards. To compromise APIs, attackers go after integrated third-party services instead of trying to compromise the target API directly. Unsafe consumption of APIs is a newly added vulnerability that covers aspects of API8:2019 injection. It occurs when data from third-party APIs is not properly sanitized. To prevent this issue, ensure encrypted API interactions, thoroughly evaluate and sanitize API data, and use allow lists to prevent unnecessary redirections. Insecure utilization of APIs can manifest when the back-end systems or API implementations, connected to external or third-party APIs, fail to scrutinize their endpoints or adequately check and sanitize incoming data. These back-end systems or API implementations also might heedlessly follow redirections from such external or third-party APIs without due validation and neglect to impose resource constraints while handling their responses. These unsafe practices create opportunities for malicious actors to exploit vulnerabilities in APIs and services linked to the target API.	The responsibility for this specific threat should primarily be with the backend application server that is calling the third-party API. Use ML-based API discovery and protection to identify PII data and implement additional audits on the third party for any requests using this sensitive data.

A Multi-Layered Approach to API Security

API and API endpoint discovery, schema validation, access management, data exposure prevention, account takeover detection, and DoS mitigation are all crucial aspects of securing APIs in today's dynamic application development landscape. To protect against these threats, organizations must adopt a multi-layered approach to API security. This approach should include thorough documentation, proper schema validation, secure token management, real-time monitoring, and proactive threat detection. Only by addressing these issues can organizations continue to reap the benefits of APIs without falling victim to ever-present cybersecurity risks. As the digital landscape evolves, so do threats, and robust API security needs to be an ongoing priority.

Consolidated cloud application protection from Fortinet FortiWeb, on-premises or in the cloud, provides consistent, simple, and extensive security for hybrid and multi-cloud environments. ML algorithms deliver accurate threat detection and analysis capabilities to protect web applications and APIs from human-generated attacks and malicious bots. Fortinet cloud application protection can be deployed as hardware, virtually, or as a service and can be purchased using a number of different financial models: as a capital expense, subscription, pay-as-you-go cloud computing, or bring your own license. FortiWeb integrates with the Fortinet Security Fabric, which provides centralized management, visibility, and consistent application security.

[1] Verizon, 2023 Data Breach Investigations Report.

FORTINET

www.fortinet.com

December 15, 2023 2:01 PM

2471369-0-0-EN