# A Holistic Review of Tools for API Security, Testing and Vulnerability Detection

Rajika Jain
Department of Computer Science:
Cyber Security
COEP Technological University
Pune Maharastra, India

***Abstract*— With the widespread adoption of microservices and cloud-native applications, APIs have become the foundational building blocks of software ecosystems. They enable seamless data exchange, business automation, and service interoperability across diverse platforms. However, this rapid proliferation of APIs has significantly expanded the attack surface, making them a primary target for malicious actors. APIs frequently expose sensitive data and critical business logic, and can lead to severe consequences such as data breaches, financial losses, and non-compliance. Despite growing awareness, many organizations struggle with limited visibility into their API inventory and usage, along with limited real-time monitoring of their security posture. Furthermore, manual testing often fails to catch sophisticated threats outlined in the OWASP API Top 10, emphasizing the need for automated, continuous, and intelligence-driven API security testing. As DevSecOps practices mature, securing APIs with minimal overhead becomes essential and complex. In response to these growing challenges, this study presents a comprehensive review of two leading API security tools—Akto and Wallarm —each offering unique capabilities to enhance API observability, vulnerability detection, and runtime protection. Evaluated across key dimensions such as detection accuracy, alignment with OWASP API Security Top 10, AI integration, shift-left support, and DevOps compatibility, these tools demonstrate varied strengths in real-time monitoring, automated inventorying and vulnerability prioritization. This analysis serves as a practical resource for security teams to select the most suitable toolset for proactive and scalable API protection.**

***Keywords*— *API Security Testing, API Risk Assessment, API security tooling solutions, Comparative Security Analysis.***

## I. INTRODUCTION

Software being used today uses APIs for communicating with each other. Without the need to know how a software system is implemented, API's allows different systems to communicate with each other. Due to the continuously increasing usage of API's, API security had become a necessity. Protection of API's from various threats and vulnerabilities is a challenging task. Traditional forms of web and application security such as a Web Application Firewall (WAF) or API gateway are not able to detect API threats. As a result, we've seen a noticeable increase in API-specific security threats like Broken Object Level Authorization (BOLA), Excessive Data Exposure, Security Misconfigurations, and Improper Asset Management. These issues are not just theoretical—they're real challenges that development and security teams face every day, and many of them now included in the OWASP API Security Top 10.

Even with these known threats, many organizations still rely heavily on manual methods like penetration testing and code reviews. These approaches are helpful but they are quite slow and can't keep up with today's fast-paced development. More critically, manual approaches lack visibility into real-time API behavior and runtime threats, due to which anomalies and misconfigurations can be found in applications.

In order to overcome these challenges there is a need for automated, and easy to integrate API security solutions via which human intervention can be minimized. A proper API security tool or security solution should be able to offer API monitoring at runtime along with automated vulnerability scanning, and mechanisms for alerting and threat response.

There are a lot of API security tools present and choosing the right tool is not an easy task. A lot of options are present in the market and they offer different kinds of features like discovery features, testing capabilities, and ability to detect threats. However, the factors like ease of setup and performance also plays an important role.

After doing a proper research, testing and analysis of a wide range of API security tools present in the market, two popular tools are found that provides some great features, and performance. The practical usability of these tools also seems to be good. This analysis focuses on their strengths, trade-offs, and how well they adapt to real-world development workflows, aiming to guide teams toward a more complete and effective API security strategy.

## II. BACKGROUND ON API SECURITY

### A. What are API Security Tools?

API security tools enhances the security posture of APIs and protects them from different kinds of threats and vulnerabilities.

These kind of tools provide various features which can be used to enhance the security posture of an application such as real time threat detection, API discovery etc.. They offer real-time monitoring and threat detection capabilities, which can be helpful for organizations to be able to detect and respond in a timely manner to potential breaches.

### B. Why are API Security Tools Important?

*Growing API Attacks*: An API connects services and transfers sensitive data, use of APIs had increased significantly and so the attacks on them. A compromised API can cause serious data breaches, which can leak Personally Identifiable Information(PII) and other sensitive information.

*Complexity of Security Needs*: APIs do not follow traditional security measures, instead they require specific security frameworks to detect and respond to vulnerabilities, including insecure authentication and inadequate access controls. API security tools can be helpful in the implementation of these kind of security measures.

*Real-Time Threat Detection*: Many API security tools uses AI and machine learning to analyze traffic patterns and detect vulnerabilities in real-time. This can be helpful for security team to respond to potential threats, minimizing the impact of any security breaches.

*Compliance and Regulatory Requirements*: Organizations must adhere to several security and data protection regulations. API security tools helps in achieving these compliance as they enable APIs to adhere to the security standards.

*Secure Development Practices*: API security tools can be integrated with the development lifecycle, allowing for early detection of vulnerabilities. This approach helps to prevent developers from deploying security flaws into production environments.

*Mitigation of Logic Flaws*: Many security breaches occur due to issue in the logic of API code rather than traditional vulnerabilities. Specialized tools can identify these logic vulnerabilities before attackers can exploit them.

### C. *Selection Criteria for API Security Tools*

From a vast range of API security tools, two particular tools were selected based on the robust security measures offered by them including authentication, encryption, and compliance with standards such as GDPR. The selection criteria emphasized on the solutions that provide real-time threat detection and prevention, along with effective incident response capabilities.

Tools that work well with existing software and can be deployed in a variety of environments—like cloud, hybrid, or on-premises—are generally preferred. Scalability and the ability to maintain performance under high loads were also considered.

In addition, having responsive and knowledgeable customer support is crucial. Tools that allow customization for specific needs, provide detailed analytics and reporting, and use AI or machine learning for advanced threat detection are especially valued.

## III. Experimental Setup And Evaluation Criteria

To evaluate the selected API security tools, a controlled test environment was established that closely mirrored real-world API interactions. A virtual machine was provisioned as the primary testing platform , where Docker and Kubernetes were installed for application deployment and manage traffic.

For testing, we used **crAPI (Completely Ridiculous API)**—an intentionally vulnerable, microservice-based application designed to simulate real-world API attacks [1]. It was used as a benchmark environment to evaluate each tool's ability to detect OWASP API Top 10 [6] vulnerabilities and other common threats. To further validate their effectiveness in real-world settings, we also tested the tools on open-source, on-premises applications like the Delivery Excellence Platform (DEP) Project.

Each security tool was deployed, and tested against crAPI. The evaluation considered the following criteria:

- Automated Vulnerability Scanning: Detection of known flaws, misconfigurations, and logical errors.
- Customization & Integration: Flexibility to configure, fine-tune, and integrate with existing CI/CD pipelines and DevSecOps workflows.
- Detailed Reporting & AI Remediation: Quality of generated reports and ability to suggest actionable remediations using AI or automation.
- Performance Impact: Measurement of added latency or throughput degradation.
- Detection Accuracy: Precision in identifying OWASP API Top 10 vulnerabilities and other security risks.
- Traffic Monitoring: Ability to observe and log API traffic in real time.
- Feature Coverage: Breadth of capabilities including API visibility, fuzzing, access control, and runtime protection.
- DevOps Compatibility: Ease of automation and workflow integration.
- Ease of Use: UI/UX, documentation, and setup experience.
- Deployment Environment: Support for various OS and runtime platforms.

## IV. API Security Tools Experimentation

For the API Security Tools selected for this paper as per the selection criteria, a thorough experimentation was performed for each of them, focusing on dynamic testing, attack simulation, input validation, automated scanning etc. Below is the detailed walkthrough on the experimentation of each tool and it's outcome :

## V. Akto:API security platform

### A. *Tool Overview:*

A tooling solution designed to analyze APIs at runtime. It offers continuous API inventorying, vulnerability assessment of APIs, and identification of runtime issues aligned with OWASP API Top 10 categories [2]. (see Image 1).

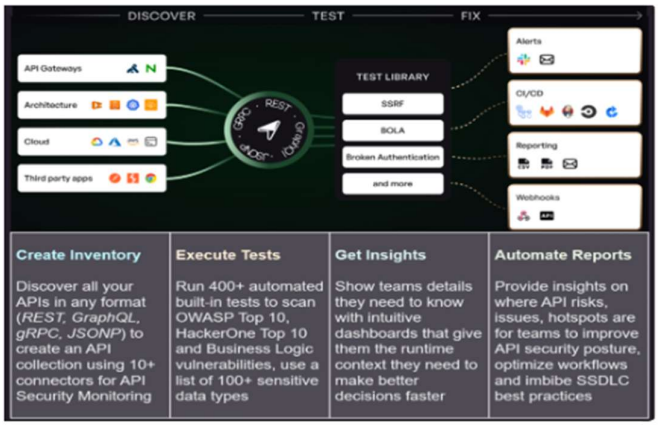


Image1. Akto – Detailed Workflow

### B. *Key features:*

- Automated API Scanning: Automatically scans for comprehensive security vulnerabilities on APIs.

- Real-Time Vulnerability Detection: Alerts users of possible threats as they occur.
- CI/CD Integration: CI/CD can easily integrate into continuous integration and deployment workflows.
- API Inventory Management: It automatically discovers and catalogs all APIs, shadow APIs included, ensuring full visibility.
- Compliance Reporting: Generate in-depth compliance reports meeting the requirements of standards such as OWASP, GDPR, or PCI DSS to make regulatory compliance easier.
- Scalability: The Akto platform is designed for large-scale API deployments. It can handle multiple API endpoints and manage and monitor them efficiently.

### C. *Experimentation Process: DEP Project*

An ubuntu server was setup and the open source Delivery Excellence Platform(DEP) application was deployed on it in order to assess its runtime API security posture. To perform the experiment, a Postman collection of API Endpoints (JSON file with valid authentication token) was created to be shared with Akto. Once the Postman collection file was uploaded onto the Akto platform, it automatically verified the saved responses or replayed the endpoints for successful API responses (2xx status codes) to discover and populate the complete inventory collection.

The Akto platform contains over 250 predefined templates which were then used to perform the testing of the API endpoints. These automated tests covered OWASP API Top 10 category checks, including Broken Object Level Authorization (BOLA), Broken Authentication, Server Side Request Forgery (SSRF), Mass Assignment, Excessive Data Exposure, and many others. In a matter of minutes, the dashboards were generated containing data like API risk scores, vulnerabilities, and hotspots based on the real time state of the application (as seen in Image 2). This data can be used to improve the API security posture of the application.


Image 2. API security dashboard generated by Akto

The risk score of the API collection and its underlying endpoints was examined to understand the security implications and potential vulnerabilities. This API collection has a risk score of "4" and its underlying endpoints have risk score values ranging from "0 to 4" Using the risk score provided by Akto, it can be prioritized which API endpoints require additional security measures. Endpoints with higher risk scores may necessitate stricter security controls, such as stronger authentication, rate limiting, or increased monitoring [7]. ( see image 3)

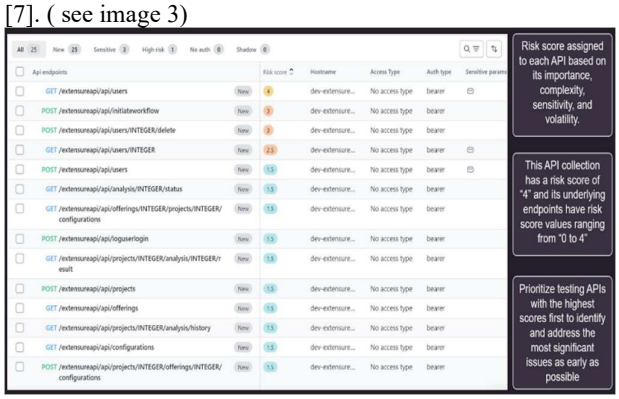


Image 3. Risk Score across API Inventory

### D. *Advantages Identified via experimentation:*

- User-Friendly Interface: Akto's system is developed based on user-friendliness to create accessibility for API security.
- Quick Deployment: Easy to deploy, so security teams can get up and running with minimal downtime.
- Comprehensive Coverage: Covers OWASP API Security Top 10 vulnerabilities, ensuring robust protection against common threats.
- Proactive Security Posture Management: Continuous monitoring for maintaining the strength of a security posture for an evolving API.
- Advanced Authentication Testing: Supports different types of authentication, allowing for proper security checks on APIs with complex access controls.

### E. *Disadvantages identified via experimentation*

- Dependency on live Traffic: For automatic API discovery, Akto requires access to live traffic. However, In some organizations such access is restricted.
- Out-of-Scope Endpoints: There is no easy way to exclude irrelevent api endpoints from getting analysed by the tool.

### F. *Conclusion*

The akto tool uses the mirrored traffic for analysis, which ensures that there is minimum performance impact on the application. As per the experimentation it can be concluded that it is an easy to use tool which can be used for detailed security analysis of API's and to identify the key areas where security enhancements need to be done, prioritized on the basis of Risk Scores. The data generated by the tool also categorizes the issues identified in the API's on the basis of API security breaches/possible attacks, which can also be used to prioritize the further actions based on the organization need.

## VI. WALLARM: ADVANCED API SECURITY.

### A. *Overview:*

Wallarm is a tool which provides advanced API security measures, it can not only perform detection, but it is also able to block API-based attacks in real time [3].

Wallarm provides a complete view of all the APIs exposed by an application. It also identifies protocols which are in use, performs the evaluation of existing security controls, and do the verification of API configurations.

The API discovery feature of wallarm analyzes API traffic to identify all exposed endpoints within an application. It also determines whether any of these endpoints are disclosing sensitive data.

Below are the 4 pillars of wallarm [4]:

*Discover*: Continuous API discovery that provides visibility of APIs and data flows.

*Protect*: Protection of APIs via real-time blocking of attacks.

*Respond*: Provides integrations via which response on incidents can be taken.

*Test*: Conduct automated API security testing so that remediation can be prioritized.

Wallarm nodes are deployed locally and they operates within the environment. They analyses API calls, traffic, and ensure that no data leaves the infrastructure.

### B. *Key features*

- Wallarm performs the blocking of attacks in real-time.
- It uses AI based threat detection mechanisms for the detection of attacks.
- It is easy to integrate wallarm with any CI/CD workflow.
- It protects AI applications against injection attacks, data leakage, and unauthorized access.

### C. *Experimentation Process*

In order to evaluate evaluate the features of the Wallarm Tool, a test environment was set up on the Google Cloud Platform. A vulnerable version of a java application was deployed on a GCP virtual machine. Wallarm's cloud-based connectors specifically istio was used to integrate the tool with the application so that it can monitor the API traffic in real time. Once the connection was established, Wallarm started generating data about the vulnerabilities present in the api's exposed by the application.

In the second phase of the experiment i.e to simulate real-world threats, a number of malicious API requests were sent to the application. SQL injection attack and log4j attacks were simulated on the application. Wallarm was able to identify these requests as attacks, with detailed data on the dashboard like traffic volume, breakdowns by API endpoint, attack trends, and geolocation of threat origins as seen in dashboard.(See Image 4)

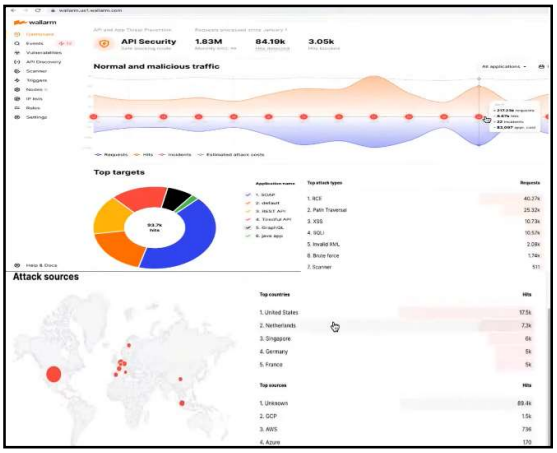


Image 4: Wallarm Dashboard

It grouped related malicious requests into unified attack chains, which helps in reducing noise generated by the tool and helps in focusing on the issues. Each request generated on the wallarm tool contains data such as raw requests, HTTP headers, and payloads, which helps to investigate and trace the attack.(See Image 5)

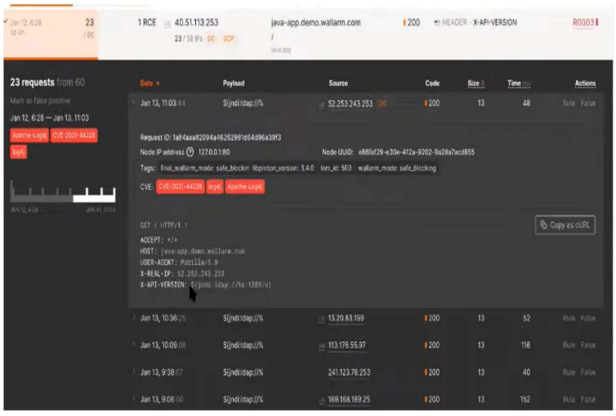


Image5: Deep API request Inspection

Wallarm active threat verification feature scans every attack that was detected, After scanning if the tool is able to block the attack, green check mark is provided (as seen in image 6) on the right side. For certain cases depending upon the severity of the attack the tools creates incidents.

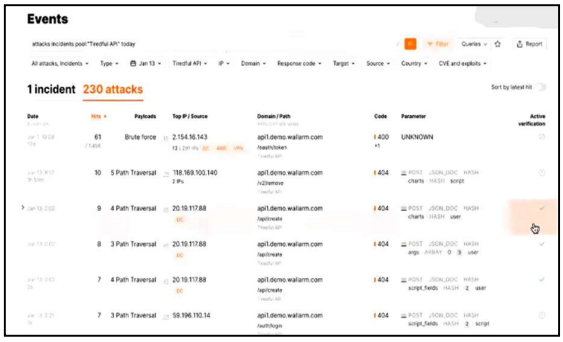


Image6: Active threat verification by wallarm

*D. Advantages identified via experimentation*

- Automatic Discovery of API's: Wallarm is able to analyze traffic and generate API specifications without the need of any user provided specification.
- Detection of Shadow API's: Wallarm was also able to identify deprecated or undocumented APIs present in the application.
- In-Depth Vulnerability Detection: Critical vulnerabilities present in the application like Log4j, RCEs, exposed Git repos, outdated libraries, etc were detected by the tool and provide details for each issue.
- Threat Response: Specific automated responses can be created in Wallarm (e.g., blacklisting based on request patterns, user agents, IP thresholds).
- Integration with DevOps & SIEM Tools: Wallarm can be integrated with Slack, SIEMs, and other DevOps tools for the purpose of alerting and incident creation.
- Deployment Options: It can be easily integrated with IAC tools like terraform and puppet.
- Low-Latency Performance: Even when the tool runs in full blocking mode it almost provides zero or negligible latency which makes it suitable for environments where performance is critical.

*E. Disadvantages identified via experimentation*

- Complex Configuration for Some Features: For some of the granular controls offered by the tool require complex technical tuning.
- False Positives: Although it is rare, false positives were present in the tool.

*F. Conclusion*

Based on the experimentation it can be clearly said that Wallarm is able to perform real-time threat detection. It is also able to perform prevention of attacks without any manual intervention. It also provides in depth data regarding the incident which can be used for the incident response process and further investigation purposes. However, it was also identified that a few false incidents were generated by the tool i.e. it wrongly identified legitimate traffic as malicious.

## VII. Comparative Result Analysis

This paper compares the two tools: Akto and Wallarm. The Akto tool offers a developer-first approach with a focus on fast API inventory, modular testing, and seamless CI/CD integration, which makes it suitable for teams which have focus on speedy delievery and automation. On the other hand, Wallarm is a tool which focuses on enterprise level security, this tool offers deeper threat detection and response as well, which is suitable for organizations which deal with complex applications and which require higher security.

| Feature | Akto | Wallarm |
|---|---|---|
| **Primary Focus** | Developer-first API security testing and inventory | Enterprise-grade API security |
| **Deployment Style** | Docker-based lightweight deployment | Cloud-native, hybrid, and scalable for large environments |
| **API Discovery** | Automated via traffic mirroring tools | Automated via live traffic analysis |
| **Runtime Protection** | Real time threat blocking is not supported | Real-time threat detection and blocking |
| **Scalability** | Ideal for startups and mid-sized teams | Enterprise scale performance |
| **AI Usage** | Basic detection, and recommendations | Advanced AI for protection |
| **Pricing Model** | Free | License based, with some of the features free to use |

## VIII. Conclusion

Through this research, a detailed comparative analysis was done between the two API security tools i.e. Akto and Wallarm, both tools are designed to be able to serve different organizational needs and security requirements. Akto offers a user-friendly interface, rapid API discovery, and seamless integration with CI/CD pipelines. The solution offered by wallarm is more focused for enterprises, it offers real-time threat detection and blocking, performs deep analysis of traffic, and advanced AI-powered detections which are suitable for large applications.

The experimentation results show that both the tools fulfill critical roles in API security, but with different operational needs — Akto focuses on early-stage testing and proactive posture management, while Wallarm is better in terms of live threat response and runtime protection.

## References

[1] OWASP Foundation, *OWASP crAPI Project: Completely Ridiculous API*, [Online]. Available: https://owasp.org/www-project-crapi/.

[2] Akto, *Akto API Security Platform*. [Online]. Available: https://www.akto.io/.

[3] Wallarm, *Wallarm API Security and Protection Platform*. [Online]. Available: https://www.wallarm.com/.

[4] Wallarm, *Wallarm Documentation: Platform Overview*, Wallarm, Inc., [Online]. Available: https://docs.wallarm.com/about-wallarm/overview/.

[5] Wallarm, *Wallarm API Security Platform Comparison*, Wallarm, Inc., [Online].Available: https://hubspot.wallarm.com/hubfs/Comparison%20pdf.pdf.

[6] OWASP Foundation, *OWASP API Security Project*, [Online]. Available: https://owasp.org/www-project-api-security/.

[7] Akto, "Risk Score - API Inventory," *Akto API Security Documentation*, [Online]. Available: https://docs.akto.io/api-inventory/concepts/risk-score.