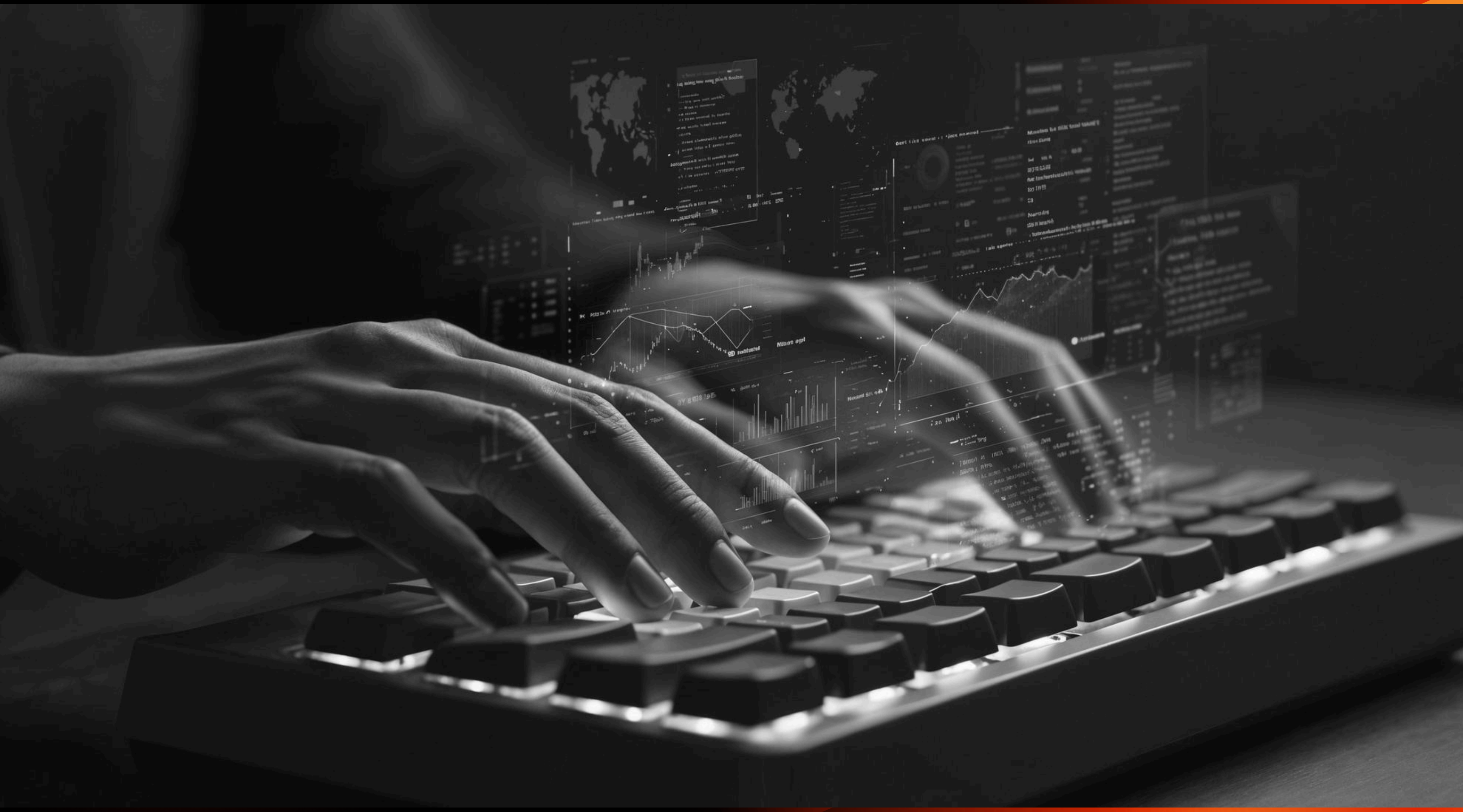
wallarm



Modern APIs Deserve Modern Security:

Beyond WAFs and API Gateways

www.wallarm.com

(415) 940-7077
188 King St. Unit 508, San Francisco, CA 94107

APIs are the backbone of modern digital ecosystems. They enable seamless communication between software, drive business processes, and power digital platforms. However, as APIs grow in number and complexity, they also become prime targets for attackers. Organizations often rely on Web Application Firewalls (WAFs) and API Gateways (API GWs) to protect their APIs. While these tools play an important role, they fall short in addressing the sophisticated threats APIs face today.

API security incidents are no longer hypothetical risks. According to the **2024 Postman State of the API Report:**

- Organizations report managing an average of **459 APIs**—a number that continues to grow annually.
- **89% of developers** say API security is a top concern, especially as complexity and deployment speed increase.
- **Nearly 60% of organizations** deploy changes to their APIs at least once per week, with many doing so daily creating a moving target for traditional security tools.

These figures show that without intelligent and adaptive protection, organizations risk falling behind in their ability to secure their expanding API surface area.

To truly safeguard your APIs, a dedicated API security solution is not just beneficial, it's critical. A dedicated API security solution is no longer optional; it's essential. Here's why.

Business Use Cases for API Security Beyond Legacy Tools

For modern organizations, API security is not just a technical necessity, it's a business enabler. Let's take a look at some of the business use cases that demonstrate why investing in dedicated API security delivers measurable outcomes far beyond what WAFs and Gateways can offer.

Use Case: Secure Digital Transformation and E-Commerce Growth

Scenario	An e-commerce platform scaling its operations across web, mobile, and third-party marketplaces.
Challenge	Exposure of sensitive customer or product data through APIs powering shopping carts, recommendations, and payment processing.
WAF Limitations	Cannot detect logic abuse or account-level anomalies, like gift card fraud or excessive order modifications.
API Security Value	Detects account takeovers, behavior-based fraud, and schema violations in checkout flows.
Business Outcome	Reduced fraud loss, improved customer trust, and PCI/GDPR compliance.

Use Case : Protect Patient or Financial Data in Regulated Industries

Scenario	A healthcare provider or fintech using APIs to share data with apps, insurers, or partners.
Challenge	Prevent data leaks, unauthorized access, or API misuse that could violate HIPAA or SOX/GLBA regulations.
WAF Limitations	Cannot enforce object-level access control or detect deep data exposure.
API Security Value	Enables role-based monitoring, data classification, and anomaly alerts on sensitive endpoints.
Business Outcome	Minimized risk of compliance violations, fines, and legal exposure.

Use Case: Enable Safe Innovation with Agentic AI and Internal Copilots

Scenario	Deploying internal AI agents or developer tools that interact with APIs autonomously.
Challenge	AI-driven interactions may misuse endpoints, trigger destructive operations, or exceed intended privileges.
WAF Limitations	Cannot understand prompt chaining, behavioral anomalies, or intent misalignment.
API Security Value	Provides visibility into API activity from autonomous actors and blocks malicious or unintended actions in real-time.
Business Outcome	Safer AI adoption, governance for LLM integrations, and better control over automation risks.

Use Case : Support Partner/API Monetization Programs

Scenario	Monetizing APIs via third-party developer programs or open platforms.
Challenge	Protect APIs from scraping, abuse, overuse, or exploitation by bad actors.
WAF Limitations	Treats all authenticated traffic equally, offering no usage context or adaptive controls.
API Security Value	Tracks API usage per partner, detects abuse patterns, and applies custom controls without breaking SLAs.
Business Outcome	Accelerated partner growth with reduced overhead and attack surface.

Use Case : Accelerate CI/CD and Shift Security Left

Scenario	DevOps teams rapidly push changes to APIs in agile development cycles.
Challenge	Frequent changes lead to misconfigurations, exposed endpoints, and outdated controls.
WAF Limitations	Requires manual rules and cannot auto-learn or adapt to new versions.
API Security Value	Provides dynamic discovery, schema drift detection, and integration into CI/CD pipelines.
Business Outcome	Higher velocity without increased risk, and security that scales with innovation.

Use Case: Enhance Threat Detection and Incident Response

Scenario	A security team tasked with reducing MTTD (Mean Time to Detect) and MTTR (Mean Time to Respond).
Challenge	Lack of visibility into API behaviors, payloads, or attack chains.
WAF Limitations	Limited to logs at the HTTP layer, no correlation or runtime behavior insight.
API Security Value	Offers forensic-level visibility, attacker fingerprinting, and integration with SIEM/SOAR platforms.
Business Outcome	Faster detection, improved incident resolution, and reduced security operations cost.

Technical Limitations of WAFs and API GWs

Now that we have discussed some of the business use cases that are driving API security adoption, let's review some of the technical limitations of WAF's and Gateway's in addressing the modern API challenges. Both WAFs and API Gateways are vital components in managing and securing APIs, but they have inherent limitations when it comes to handling modern API-specific threats.

1. WAFs Are Built for Web Apps, Not APIs

Web Application Firewalls (WAFs) are designed to protect traditional web applications by filtering and monitoring HTTP traffic. While they can block common threats like SQL injection and cross-site scripting (XSS), their scope is limited when applied to APIs. APIs have unique characteristics that WAFs do not adequately address, including:

- **Complex Data Structures:** APIs often communicate using JSON or XML payloads, which WAFs struggle to analyze beyond surface-level patterns.
- **Context Awareness:** APIs rely on a series of interdependent calls and workflows. WAFs are incapable of understanding or enforcing this context, leaving APIs vulnerable to issues like broken object-level authorization (BOLA).
- **Dynamic Behavior:** APIs often include frequent updates, changes, and feature additions. WAFs, typically rule-based, cannot adapt quickly enough to keep pace with changes.

Why it matters: WAFs are built for web apps, not APIs. They lack the context to understand API workflows, data structures, and logic. This means API-specific threats like BOLA or data exposure often slip through leaving critical gaps in protection.

2. No Schema or Specification Awareness

WAFs do not natively support **OpenAPI, Swagger,** or **GraphQL introspection**. They can't validate whether incoming requests match the intended API contract.

- This opens up risks for **mass assignment, parameter pollution,** and access to undocumented fields.
- WAFs lack the ability to parse or enforce granular schema validation on deeply nested payloads.

Why it matters: APIs exposed via GraphQL or misaligned with schema enforcement are highly prone to abuse—often without triggering alerts in traditional defenses.

3. Blind to Business Logic Attacks

WAFs operate based on pattern-matching rules and signatures. They cannot understand application-specific logic, such as:

- A user updating someone else's profile using a valid token
- Frequent but subtle abuse of rate limits or pricing APIs
- Inventory scraping or competitive intelligence collection

Why it matters: Business logic vulnerabilities (like BOLA or BFLA) are among the top API threats listed in the OWASP API Top 10, yet are invisible to WAFs.

4. Stateless Request Inspection Fails API Workflows

WAFs inspect each request in isolation and do not track multi-step processes. This means:

- They can't link API calls across a session
- They miss chained attack flows and replay patterns

Why it matters: Most modern attacks don't happen in one request—they unfold across time and context.

5. Cannot Detect Shadow and Zombie APIs

WAFs only see what they're explicitly configured to see. As a result:

- **Shadow APIs** (untracked services deployed by individual teams) evade detection
- **Zombie APIs** (deprecated or outdated versions still exposed) remain unmonitored

Why it matters: Attackers routinely probe for forgotten or undocumented endpoints—often the weakest links in an otherwise secured perimeter.

6. No Feedback Loop for Development

Most WAFs lack integration with CI/CD pipelines or developer tooling. This means:

- Developers receive no visibility into blocked API traffic or risks
- Security teams can't shift left to prevent vulnerabilities early

Why it matters: Effective API security needs to be continuous and integrated—not bolted on post-deployment.

7. Rule-Based Models Can't Keep Up with API Velocity

APIs evolve daily, but WAFs are reliant on manual rule updates or static tuning. They struggle to keep pace with:

- Constant addition of new endpoints
- Changes in payload structure and auth flows

Why it matters: The time gap between an API change and the corresponding WAF configuration creates a dangerous window for attackers.

8. API GWs: Built for Management, Not Defense

API Gateways are crucial for traffic routing, rate limiting, and access control—but they aren't security tools:

- **No Deep Payload Analysis:** Gateways don't scan request bodies for malicious payloads.
- **Weak Attack Detection:** Complex injection or logic abuse is undetectable.
- **Lack of Runtime Intelligence:** Once a request is authenticated, Gateways rarely question behavior.

Scenario: If a threat actor gains access tokens, Gateways treat malicious requests as normal, offering no defense once past authentication.

Why it matters: Gateways don't inspect payloads or behavior. Once authenticated, even malicious requests go unchecked.

9. API GWs Are Focused on Management, Not Security

API Gateways are essential for managing API traffic. They handle tasks like rate limiting, authentication, and routing. However, they are not built to detect or prevent sophisticated security threats.

Here's where API Gateways fall short:

- **Surface-Level Protection:** While API Gateways enforce authentication and throttle requests, they don't analyze request payloads for malicious intent.
- **Limited Attack Detection:** Complex attacks like injection or unauthorized access often bypass Gateways because they don't analyze API behavior at a granular level.
- **No Runtime Protection:** Gateways lack runtime protection capabilities, meaning they can't stop threats that manifest after initial authentication.

Consider a scenario where a malicious actor gains valid credentials and sends harmful requests through the API. An API Gateway would likely treat these requests as legitimate, as it focuses on authentication rather than content or behavior analysis. This oversight could lead to attacks such as mass data exfiltration.

Why it matters: Gateways miss complex attacks post-authentication due to lack of runtime protection and behavioral analysis.

10. Gateways Trust Authenticated Requests Too Much

Gateways validate tokens and API keys but assume everything post-authentication is safe.

Why it matters: If credentials are leaked or stolen, attackers can abuse APIs with full access—completely undetected.

11. No Inspection of Request Payloads

Gateways generally inspect headers and routing logic but not payload contents.

Why it matters: Malicious data in JSON or GraphQL bodies (e.g., SQLi, BOLA, data exfiltration) will pass through unnoticed.

12. No Enforcement of Fine-Grained Authorization

Gateways apply basic access control but cannot:

- Validate object-level permissions
- Prevent privilege escalation
- Detect function-level misuse

Why it matters: Attackers can tamper with object IDs or bypass intended workflows, and the Gateway won't stop them.

13. No Behavioral or Anomaly Detection

API GWs don’t detect when:

- A user makes hundreds of requests in seconds
- Bots scrape data across endpoints
- Sequences of API calls resemble abuse patterns

Why it matters: Behavioral abuse is a leading attack vector and Gateways are blind to it.

14. No Visibility into Shadow or Legacy APIs

Gateways only secure what they’re configured to route. Anything outside that config is invisible.

Why it matters: Undocumented or deprecated APIs are some of the most exploited and Gateways won’t even see them.

15. Lack of Threat Intel, Enrichment, and Response

Gateways don’t correlate threats, enrich traffic with external intel, or initiate automated remediation actions.

Why it matters: API threats often span time and services. Without context and correlation, they're easy to miss.

16. Insufficient Logging for Incident Response

Gateways don’t log complete payloads or behavioral context, making forensic analysis nearly impossible after a breach.

Why it matters: Post-incident response is slowed or blocked by incomplete data.

What Makes Dedicated API Security Solutions Different?

Unlike WAFs or API GWs, dedicated API security solutions are purpose-built to meet the unique challenges of securing APIs. Here's how they provide superior protection:

1. Deep API Traffic Analysis

Dedicated solutions deeply analyze API traffic, going beyond surface-level patterns to inspect payloads, headers, and behavior.

For instance, if a user attempts to manipulate an object they’re unauthorized to access, such as editing another user's profile, a dedicated API security tool would detect this unauthorized action. It would recognize the API traffic deviating from expected workflows, even if the request appears legitimate on the surface.

2. Context-Aware Threat Detection

Modern API threats often involve a deep understanding of API workflows, user roles, and expected behaviors. By building a real-time map of API activity, dedicated solutions provide context-aware monitoring and protection.

For example, context-aware tools can identify anomalies like:

- A single user making an unusually high number of data requests in a short period.
- Out-of-order or unexpected API calls that suggest an attacker probing for vulnerabilities.

This contextual analysis is crucial for detecting advanced threats such as BOLA attacks, where attackers manipulate object IDs in API calls to access unauthorized resources.

3. Proactive Runtime Protection

Dedicated API security solutions provide runtime protection, which means they detect and block threats as they occur. This includes actively protecting against injection attacks, data scraping, or brute force attempts on API endpoints.

Imagine a scenario where an attacker injects malicious SQL code into an API field. A dedicated API security tool would identify the unusual data patterns, block the request in real time, and flag it as a threat. Traditional WAFs or API Gateways would either miss such an attack altogether or identify it too late.

4. Adaptive to API Changes

APIs evolve constantly, with new endpoints, workflows, and data structures introduced regularly. Dedicated API security tools use machine learning and automation to adapt to these changes dynamically, reducing the need for manual configuration and updates. For example, if an organization rolls out a new API version with updated features, the security solution will quickly learn the normal traffic patterns and adjust its defenses accordingly.

API Security in the Age of Agentic AI

Why WAFs and Gateways Are ill-Equipped for Autonomously Acting AI Systems

The next generation of applications is increasingly powered by **Agentic AI** autonomous systems that can plan, reason, and take actions across APIs without human input. Examples include:

- AI agents booking flights or generating reports via third-party APIs
- Automated customer service bots issuing refunds or updating user records
- Developer copilots interacting with internal APIs during code generation

These intelligent systems use APIs not as static integrations, but as **dynamic, decision-making interfaces**. This paradigm shift creates new risks and dramatically amplifies existing ones.

What Makes Agentic AI a Security Game-Changer?

1. Unpredictable Workflows

AI agents can compose novel API sequences in real time, bypassing traditional workflows or UI constraints.

Problem: WAFs and Gateways can't track or validate the intent behind autonomous API activity.

2. Increased Attack Surface

More autonomy means more frequent API calls to more endpoints, with wider privileges.

Problem: Without strict behavioral baselining and authorization enforcement, misuse by AI or AI compromised by adversarial prompts can go undetected.

3. Prompt Injection + API Misuse

Malicious prompts or training data can lead AI to misuse APIs (e.g., issuing refunds, deleting data).

Problem: WAFs and Gateways do not analyze prompt-level behavior or detect instruction chaining that could lead to destructive actions.

4. Lack of Contextual and Intent Awareness

Agentic AI makes decisions based on internal reasoning loops, often invisible to security systems.

Problem: Without full context tracking and intent modeling, legacy tools are blind to logic abuse, excessive privilege usage, or misuse of trusted credentials.

Moving Forward with API Security

While WAFs and API Gateways remain important parts of an organization's overall security posture, they are not sufficient to handle the growing complexity and sophistication of API threats. A dedicated API security solution provides the in-depth analysis, runtime protection, and contextual awareness needed to stay ahead of attackers.

By investing in a purpose-built API security approach, you can ensure your APIs remain both agile and secure, enabling innovation without compromise.

As Agentic AI redefines how APIs are consumed and controlled, legacy tools like WAFs and Gateways simply can't keep up. They were built for predictable, human-triggered interactions, not autonomous, logic-generating systems operating at machine speed.

To secure the future of software, API security must evolve with AI itself. A dedicated, context-aware API security solution that employs defense in depth with existing toolsets like WAF's and API Gateway's is a realistic line of defense in an AI-driven world.

wallarm

BOOK A DEMO

www.wallarm.com
(415) 940-7077
188 King St. Unit 508, San Francisco, CA 94107