INDUSFACE™

MASTERING API SECURITY

Protecting Your APIs Against Modern Threats

INTRODUCTION

As organizations accelerate digital transformation, APIs have become indispensable for delivering innovative services, powering mobile apps, cloud computing, Internet of Things (IoT), and microservices architectures.

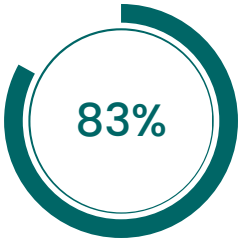
However, this rapid expansion and reliance on APIs have also made them prime targets for cyber attackers. APIs face 68% more threats per host than traditional web applications, based on data from over 2 billion attacks blocked by AppTrana WAAP as reported in the State of Application Security 2025 report.

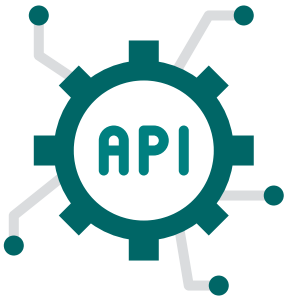
Moreover, API endpoints experience 16 times more DDoS traffic than conventional web applications, reflecting their increased exposure and attractiveness to attackers.

Key Insights

Industry reports and security analyses reveal the following trends:

Over 83% of organizations have experienced API-related security incidents in the past year.





Approximately 40% of API vulnerabilities are due to broken authentication and excessive data exposure.

Attackers increasingly exploit API weaknesses to launch sophisticated attacks, such as business logic abuse, credential stuffing, and injection attacks.

The average cost of an API-related data breach is estimated at $5.2 million, underscoring the financial and reputational stakes involved.

This surge in API threats is driven by several factors: the inherent complexity of APIs, the diverse environments in which they operate, and often, insufficient security practices during API development and deployment. Unlike traditional web applications, APIs tend to expose sensitive data and business logic directly, amplifying the potential impact of any security lapse.

The evolving threat landscape calls for a robust, lifecycle-based approach to API security. This approach must go beyond perimeter defenses and embed security controls at every stage from design and development to deployment and ongoing maintenance.

This whitepaper aims to provide a comprehensive understanding of API security by highlighting common threats, best practices, and innovative approaches, such as those implemented by AppTrana WAAP, to help organizations protect their API infrastructure effectively.

THE FUNDAMENTALS OF API SECURITY

API security refers to the practices, tools, and protocols designed to protect APIs from misuse, abuse, and malicious attacks. The goal is to ensure that only authorized users and applications can access or manipulate the data and services exposed by the API, maintaining the confidentiality, integrity, and availability of the underlying systems. Because APIs often expose sensitive business logic and data, they are a high-value target for attackers aiming to:

- Steal sensitive data such as personal information or payment details
- Manipulate backend services
- Launch denial-of-service attacks
- Exploit vulnerabilities to gain unauthorized access or escalate privileges

Effective API security involves implementing multiple layers of protection including authentication, authorization, input validation, encryption, traffic monitoring, and rate limiting.

In short, API security is crucial for maintaining trust, compliance, and operational continuity in any organization relying on digital applications and services.

TOP 10 API SECURITY RISKS (OWASP 2023)

APIs serve as essential connectors between applications, but their openness and complexity also make them prime targets for attacks. The OWASP API Security Top 10 (2023) outlines the most critical vulnerabilities organizations must address to secure their APIs. Here is a breakdown of each risk, along with real-world examples.

Broken Object Level Authorization (BOLA)

APIs often expose endpoints that reference object identifiers like user IDs. Without strict authorization checks, attackers can manipulate these identifiers to gain unauthorized access to data.

Example:

An attacker modifies the *account_id* in a request URL and retrieves the banking details of another user, bypassing authorization.

Broken Authentication

Weak or improperly implemented authentication can enable attackers to impersonate users, hijack sessions, or escalate privileges.

Example:

Exploiting predictable session tokens or weak passwords, an attacker gains access to another user's account.

Broken Object Property Level Authorization

When APIs fail to enforce access control at the property level, attackers can read or write sensitive object attributes.

Example:

A user API allows profile updates but doesn't restrict access to properties like *isAdmin*. An attacker sets this property to escalate their privileges.

Unrestricted Resource Consumption

Lack of proper rate limiting or resource quotas can lead to denial-of-service (DoS) or abuse of resources.

Example:

An attacker floods the login endpoint with requests, slowing down or crashing the authentication service.

Broken Function Level Authorization

Not all API functions require the same level of access. When authorization isn't enforced at the function level, users may invoke high-privilege operations.

Example:

A regular user gains access to admin-only functions, such as deleting accounts or modifying permissions, due to missing role validation.

Unrestricted Access to Sensitive Business Flows

When APIs expose critical business logic without security barriers, attackers can abuse them for automation or competitive advantage.

Example:

A scalper bot exploits a ticket booking API without CAPTCHA or other verification, purchasing all available tickets before legitimate users.

Server-Side Request Forgery (SSRF)

SSRF occurs when an API fetches a URL without validating it properly, allowing attackers to make requests to internal systems.

Example:

An attacker tricks the API into accessing a cloud metadata endpoint, leaking sensitive cloud credentials.

Security Misconfiguration

Improper configurations like exposed debug modes, default credentials, or verbose errors can leak information or expose weaknesses.

Example:

An API returns full stack traces in error responses, revealing framework details that help attackers craft targeted exploits.

Improper Inventory Management

Failure to properly manage and document APIs, especially old or deprecated versions can increase exposure to known vulnerabilities.

Example:

An outdated API version with known flaws remains accessible, allowing attackers to exploit vulnerabilities that are fixed in newer versions.

Unsafe Consumption of APIs

Trusting third-party or external APIs without validation or sanitization increases the risk of downstream compromise.

Example:

An application directly integrates data from an external API without validating the input, resulting in malicious scripts or data entering the application.

10 ESSENTIAL BEST PRACTICES TO SECURE YOUR APIS

A robust API security strategy goes beyond just setting up authentication. It requires a layered defense involving access control, validation, encryption, and real-time monitoring. Here are 10 proven practices to help safeguard your APIs from evolving threats.

1 Strong Authentication & Authorization

Use standards like OAuth 2.0 or OpenID Connect for secure access. Enforce role-based and object-level permissions, so users only access what they're allowed to.

2 Thorough Input Validation

Validate and sanitize all incoming data to prevent injection attacks (SQL, NoSQL, command injections) and avoid exposing sensitive information.

3 Implement Rate Limiting & Throttling

Limit the number of API requests per user or IP to stop abuse, brute-force attacks, and distributed denial-of-service (DDoS) attempts.

4 Encrypt Data in Transit & Storage

Always use HTTPS (TLS) to encrypt data traveling between clients and servers. Encrypt sensitive data stored in databases or file systems to protect against breaches.

5 Use Security Headers & Configure CORS Properly

Apply headers like Content-Security-Policy (CSP), X-Frame-Options, and Strict-Transport-Security (HSTS). Configure CORS to only allow trusted domains to interact with your API.

6 Maintain API Versioning & Deprecation

Use clear API versioning to manage backward compatibility. Promptly remove or restrict outdated API versions to reduce security risks.

7

Enable Detailed Logging & Monitoring

Log all authentication attempts, errors, and suspicious activities. Continuously monitor logs and configure alerts for anomalous activity to detect attacks early.

8

Conduct Regular Security Testing

Run penetration tests and automated vulnerability scans frequently to identify and fix security flaws before attackers exploit them.

9

Deploy API Gateway or Web Application and API Protection (WAAP)

Use an API gateway for auth, rate limiting, and routing. Add WAAP for comprehensive API protection against DDoS, injection attacks, bots, and abuse.

10

Adopt Zero Trust Security Model

Verify every API request regardless of its origin. Continuously monitor logs and configure alerts for anomalous activity to detect attacks early.

API SECURITY DEPLOYMENT: UNDERSTANDING THE SPECTRUM OF SOLUTIONS

Organizations can implement API security measures at various points in the API infrastructure, each with its benefits and constraints:

1 API Gateways

API gateways act as a central checkpoint for all API calls, enforcing security policies like token validation, rate limiting, and basic input validation. While they provide essential protection integrated directly into the API ecosystem, their capabilities often do not extend to behavioral detection or zero-day threat mitigation.

Strengths: Native integration, centralized policy management

Limitations: Limited advanced threat detection and anomaly response

Best For: Simpler architectures or early-stage APIs

2 WAAP Platforms (Web Application & API Protection)

WAAP platforms offer a unified solution combining web application firewall functionality, API security, DDoS mitigation, bot defense, and vulnerability scanning. They typically operate inline or alongside API gateways, providing broad-spectrum protection against both known and emerging threats.

Strengths: Comprehensive security coverage, visibility, real-time mitigation and reporting

Limitations: Requires initial setup and tuning for optimal performance

Best For: Enterprises with complex or public-facing APIs needing layered defenses

3 Sidecar Proxies & Service Meshes

In microservice-based or containerized environments, sidecar proxies intercept API calls at the service level, enabling granular security controls for each microservice. This model enhances security visibility and enforcement but comes with increased operational complexity.

Strengths: Fine-grained, per-service security controls

Limitations: Requires orchestration expertise and management overhead.

Best For: Large-scale, microservices-driven systems

Deployment Model	Strengths	Limitations	Ideal Use Case
API Gateway	Native API integration, basic validation	Lacks advanced threat detection	Small to mid-sized APIs
WAAP Platform	Comprehensive, real-time security	If fully managed, there are no downsides associated with tuning	Enterprise, public-facing APIs
Sidecar Proxies/ Service Mesh	Granular control at the microservice level	Complex operations, high overhead	Microservices, containerized systems

HOW API SECURITY WORKS

API security functions by applying a set of proactive controls across the entire API lifecycle to detect, prevent, and respond to threats effectively. These controls ensure APIs are discovered, validated, authenticated, monitored, and protected from misuse or attacks.

Key components include:

- Discovery: Identifying all APIs in use, including undocumented or unmonitored ones, often called shadow APIs.
- Validation: Ensuring all inputs, request patterns, and API behaviors strictly conform to expected norms and specifications.
- Authentication & Authorization: Enforcing identity verification and access control policies to ensure that only legitimate users and systems can interact with the API.
- Threat Detection: Continuously monitoring traffic to detect anomalies, misuse, and known attack signatures.
- Protection: Blocking malicious requests in real-time while maintaining minimal disruption to legitimate traffic and avoiding false positives.

These controls are typically implemented using API gateways, reverse proxies, or fully managed Web Application and API Protection (WAAP) platforms. Each is designed to provide layered defenses across the API environment.

AppTrana API Security: Full-Lifecycle Protection for Every API You Expose

AppTrana WAAP offers a unified, intelligent, and fully managed API security solution that doesn't just react to threats but anticipates and adapts to them. Unlike traditional API gateways or rule-based WAFs that rely on static policies, AppTrana's API protection takes a security-first, behavior-aware, and lifecycle-driven approach. It provides both breadth and depth of protection.

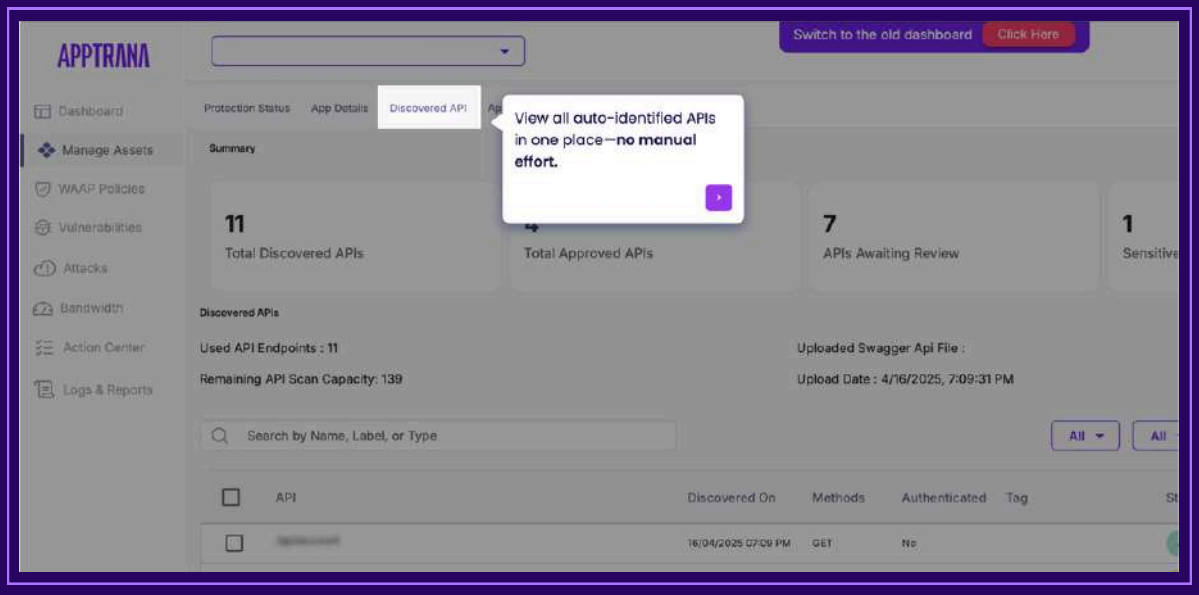
Let's break this lifecycle down and see how each step strengthens your API defense strategy.

1. Discovery: Uncover Every API—Even the Hidden Ones

API growth is inevitable—but it often comes at the cost of visibility. Forgotten, deprecated, or undocumented (shadow) APIs pose a massive threat, especially when they aren't governed by your security policies.

How AppTrana Helps:

- Deep Discovery: Goes beyond surface-level endpoint detection. Captures request body, query parameters, and traffic metadata.
- Crucial Metadata Displayed: Method (GET, POST), authentication status, and tags (e.g., deprecated, PII-handling endpoints).
- Shadow API Identification: Flags APIs not aligned with specifications or not routed through official gateways.

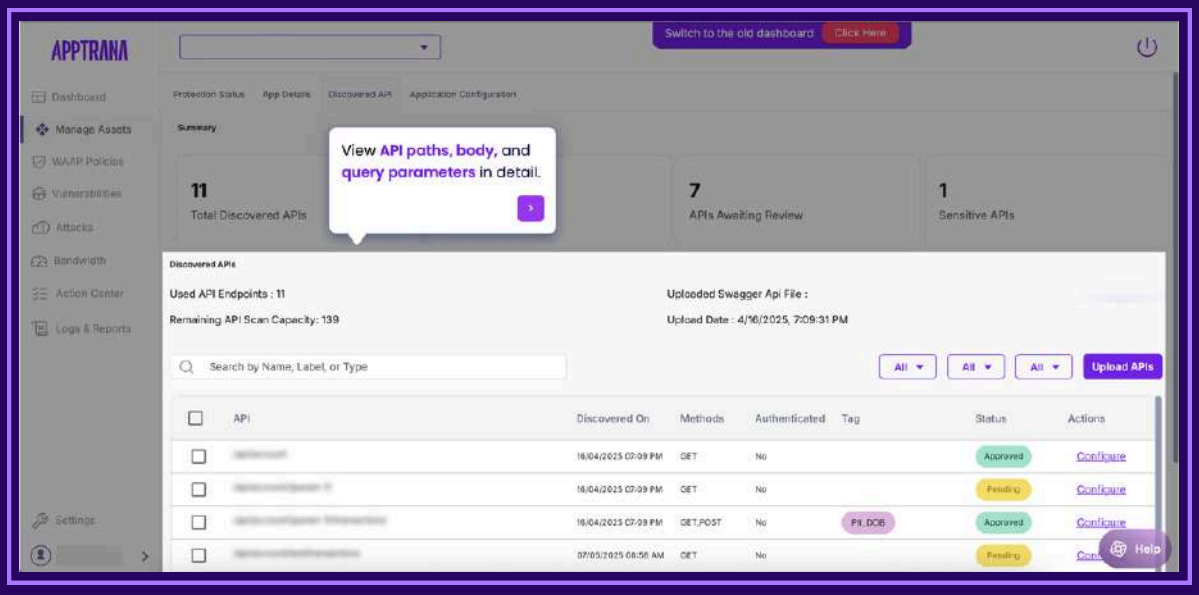


2. Understanding Behavior: Granular Insights for Granular Control

Without knowing how an API behaves or what data it handles, it's difficult to apply the right security rules.

How AppTrana Helps:

AppTrana goes beyond mere detection. It maps API behaviors, correlating query parameters, payload data, and usage patterns to give you a 360° view. It also surfaces metadata like method type (GET/POST), sensitive data tags, and exposure levels.

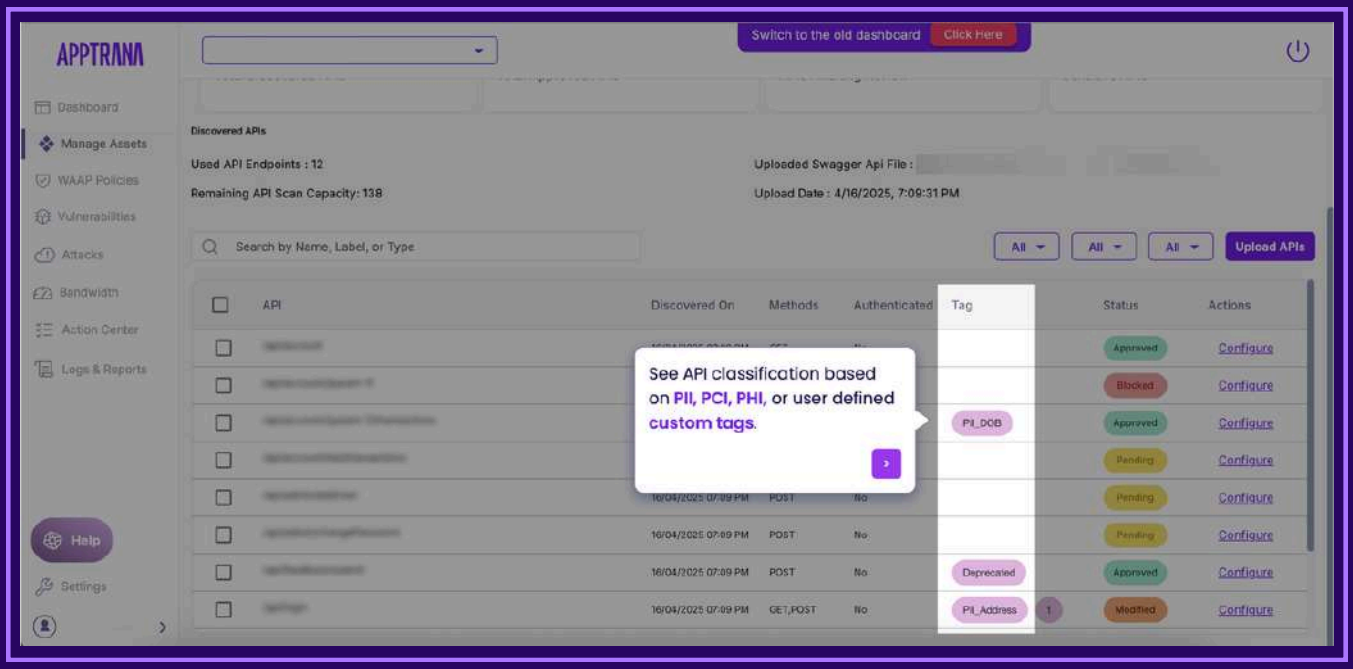


3. Risk-Based Classification: Prioritize What Actually Matters

Not all APIs are created equal. Some simply fetch data, while others process payments or manage PII. Securing all of them with equal rigor is inefficient and costly.

How AppTrana Helps:

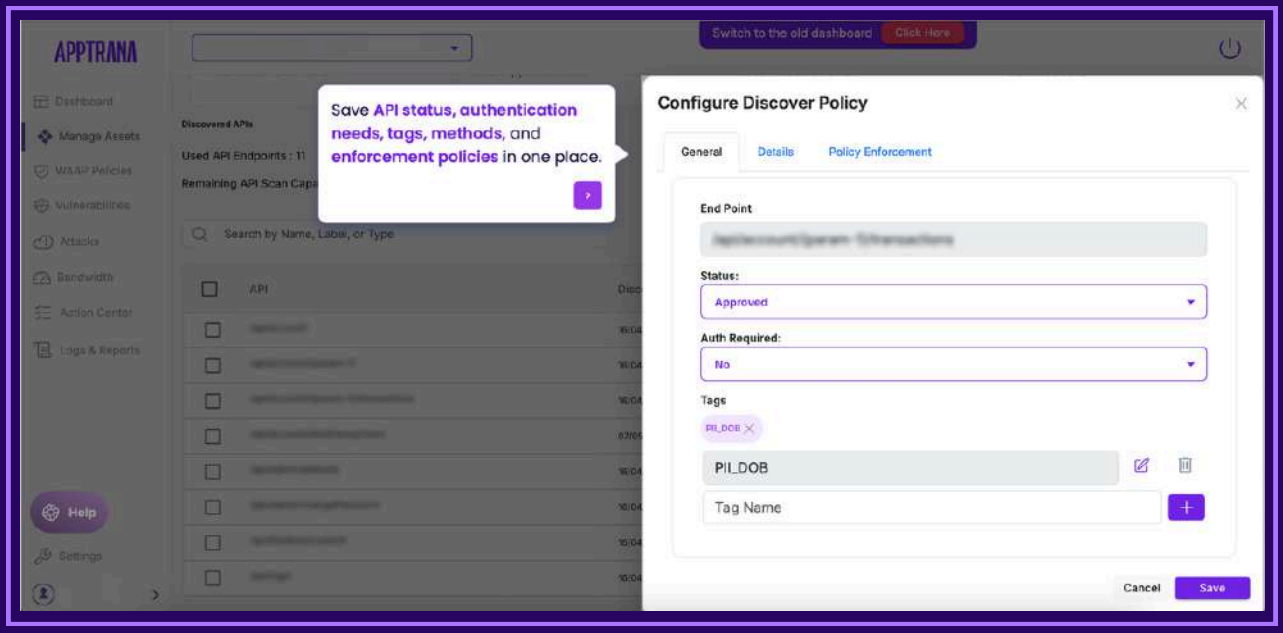
Tagging & Prioritization: Flags high-risk APIs (e.g., internal admin or payment endpoints) and recommends immediate protection.
Compliance-Centric View: Helps security teams focus on APIs tied to PCI DSS, HIPAA, etc.



4. Instant Protection & Policy Enforcement: Secure From the Second They are Discovered

Every second between discovery and protection is a potential window for attackers. Instead of waiting for manual configuration, AppTrana enables immediate enforcement of security policies to protect APIs as soon as they are discovered.

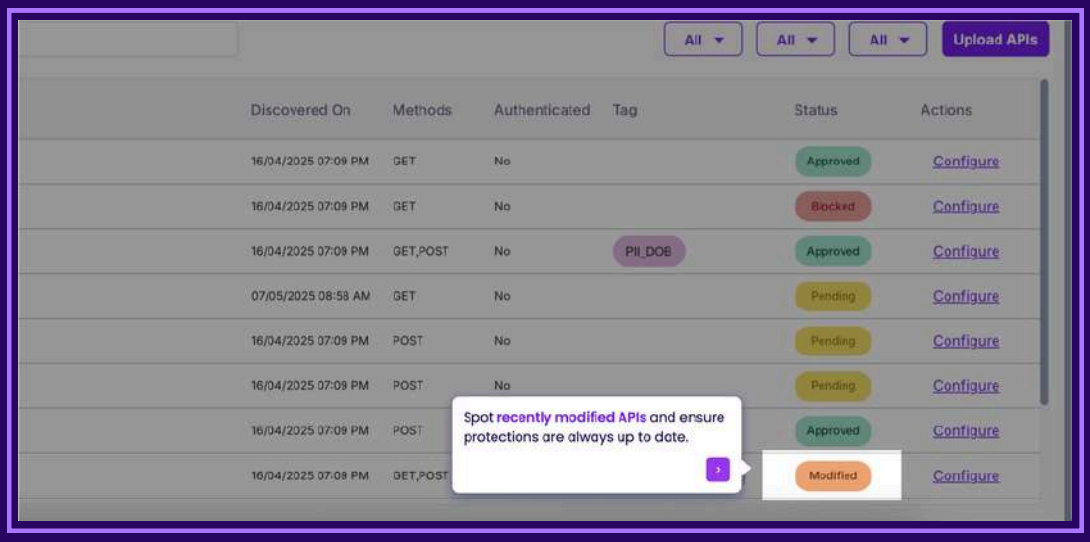
- One-Click Policy Activation: Instantly apply whitelisting rules (positive security model) for newly discovered APIs.
- Automatic Schema Validation: Blocks any requests that deviate from the expected structure (e.g., unknown fields or values).
- Zero Trust for New Endpoints: Treats all newly surfaced APIs as potentially vulnerable until validated. Customers can choose to block these APIs automatically; however, to minimize false positives, AppTrana flags them by default and requires customers to validate policy enforcement.



5. Change Monitoring: Stay Secure as APIs Evolve

APIs change constantly, and even minor modifications can unintentionally introduce security gaps. Proactively monitoring these changes is key to preventing regression vulnerabilities.

How AppTrana Helps:



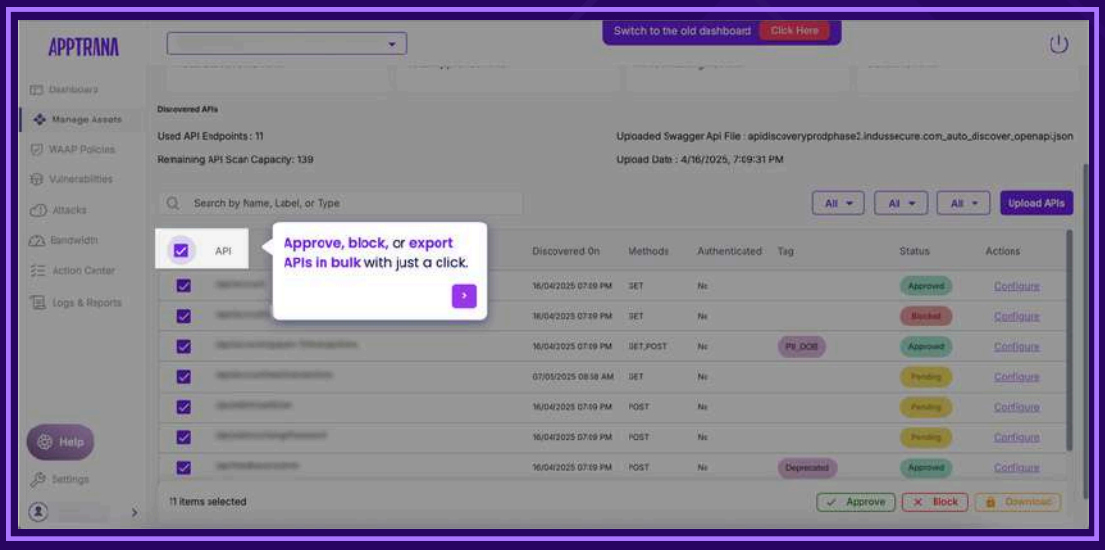
- Continuous Monitoring: Detects changes in request patterns, schemas, and authentication headers.
- Impact Alerts: Flags when an updated API doesn't match existing protection policies.
- Version Awareness: Tracks differences across API versions (v1, v2, etc.), ensuring consistency.

6. Streamlined Management at Scale

Managing hundreds (or thousands) of APIs across development, staging, and production environments is highly complex and practically impossible to do manually. AppTrana eliminates that friction.

How AppTrana Helps:

- Downloadable API Inventory: Enables offline review, documentation, and compliance readiness.
- Scoped Views: Segment APIs by environment or business unit for targeted enforcement.

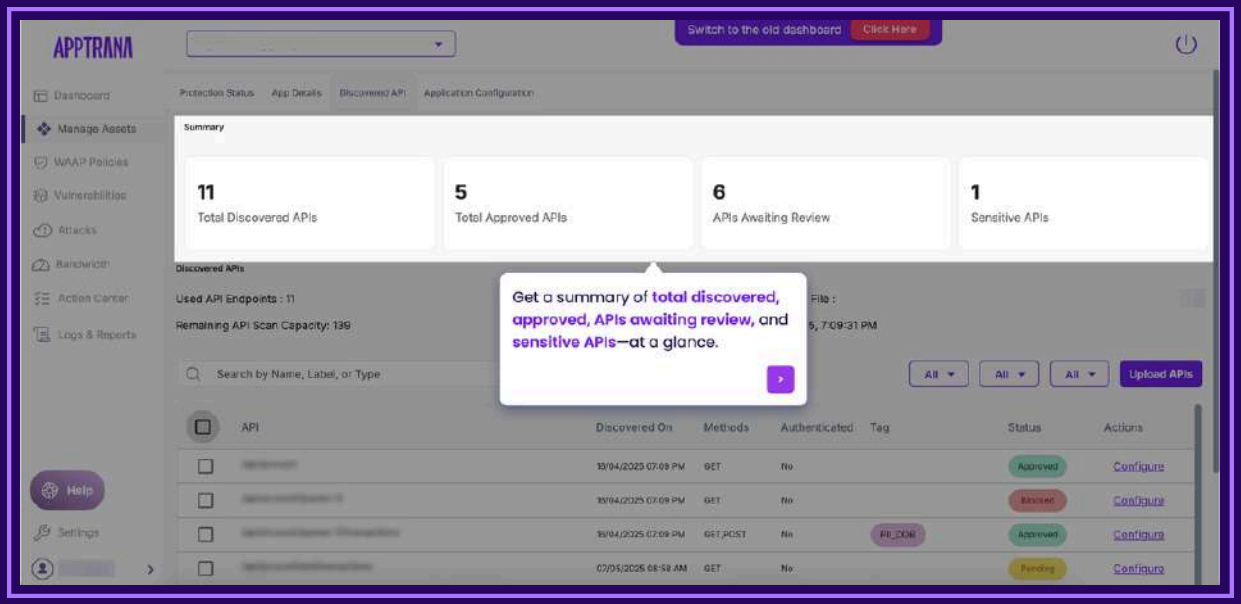


7. Unified Visibility: One Dashboard for Everything

Teams need a consolidated view to maintain situational awareness, collaborate efficiently, and respond swiftly. Siloed tools are no longer sufficient for effective API security management.

How AppTrana Helps:

- 360° API View: Tracks total discovered, pending, protected, and sensitive APIs.
- Real-Time Insights: Visualize threat trends, exposure levels, and policy coverage.
- Collaboration-Ready: Designed to help dev, security, and compliance teams stay aligned.



Core Capabilities and Techniques AppTrana WAAP Uses to Mitigate API Attacks

Method	How It Works
Strict Schema Validation	Rejects malformed requests with unexpected fields or formats. Shields against mass assignment and format-based attacks.
Adaptive Rate Limiting	Dynamically adjusts request thresholds based on traffic patterns, user behavior, and endpoint sensitivity. It stops scraping, brute force, and abuse without blocking legitimate traffic.
Access Control Enforcement	Ensures object-level authorization by checking tokens, scopes, and ownership context.
API Security Testing	Uses automated API scanning and manual penetration testing to identify risks from the OWASP API Security Top 10, such as injection flaws, broken authentication, and data exposure.
Signature-Based Detection	Blocks known exploits (e.g., SQLi, XSS) based on attack payload signatures.
Positive Security Enforcement	Allows only explicitly defined behaviors and values. Stops unknown/malformed payloads by default.
Behavioral Anomaly Detection	Detects usage spikes, sequencing abuse, or geographic mismatches using behavioral baselines.

Token and Session Validation	Validates token integrity, expiration, and misuse to prevent hijacking or replay attacks.
Bot & DDoS Defense	Identifies automated, non-human traffic and applies layered verification challenges or IP blocks.
Real-Time Threat Intelligence	Blocks requests from IPs, ASN, or user agents known for malicious activity.
Expert Management & Continuous Tuning	The managed security team assists with policy creation, threat monitoring, and periodic reviews, ensuring that protection evolves without burdening your internal team.

API Security Operational Modes: Adaptive Enforcement for Every Phase

Understanding that API ecosystems vary widely in maturity and risk tolerance, AppTrana supports flexible enforcement modes to tailor security according to your needs:

Monitor Mode (Log Mode)

This passive mode observes and logs all API traffic without blocking any requests. It's ideal during the initial rollout or when onboarding new or third-party APIs. This mode helps establish a baseline of legitimate API behavior, discover undocumented APIs, and fine-tune detection models. Since it does not interfere with traffic, there is no risk of disrupting application functionality, making it well-suited for early-stage assessment and learning.

Once patterns are well understood and security policies have been refined, block mode actively enforces these policies by rejecting malicious requests. Whether the threat stems from schema violations, suspicious behavior, known signatures, or authorization failures, block mode delivers real-time protection. It is best applied to stable and critical API endpoints, such as login and payment functions, where security is paramount. This mode helps prevent breaches and unauthorized access, safeguarding sensitive data and transactions.

Block Mode

Rate Limiting and Throttling

Designed to control traffic volume, this mode imposes limits on the number of requests from a single user, IP address, or token within a given timeframe. It protects against brute-force attacks, abuse by automated bots, and accidental overloads. Rate limiting is particularly important for public-facing APIs and high-risk endpoints like login or search. By preventing excessive traffic, it helps preserve backend performance and ensures fair usage without requiring complex payload inspection.

Securing the Future of Your APIs

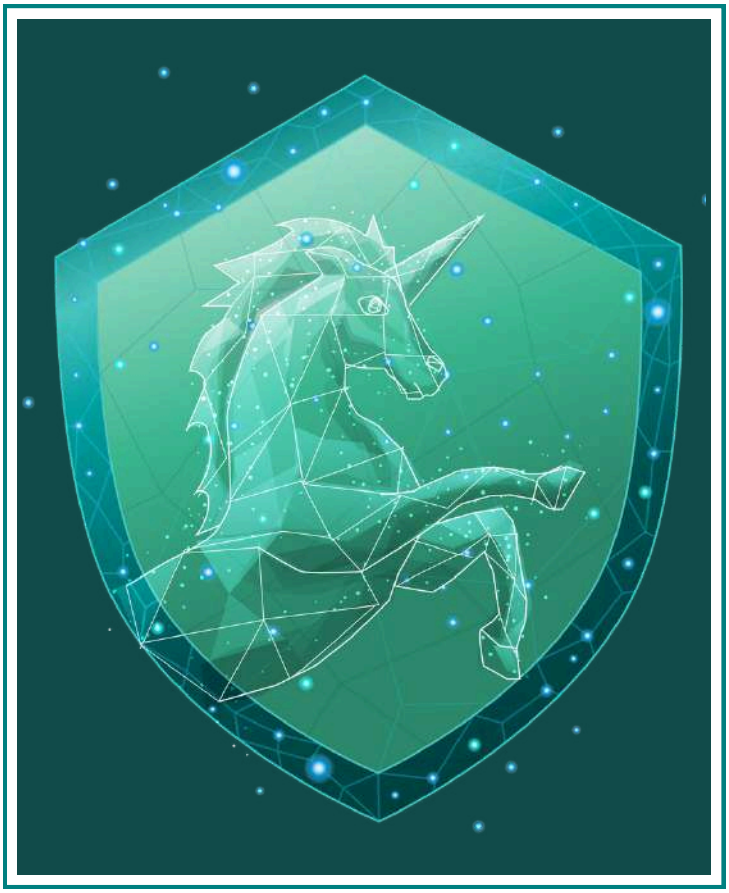
API security requires continuous attention and strategies tailored to your unique business needs. By adopting flexible enforcement and advanced protection techniques, you can build a robust and secure digital foundation.

APIs are the backbone of your digital ecosystem, and they must be protected with the right blend of technology, expertise, and proactive defense.

HOW A FINTECH UNICORN SECURED 6,000+ API ENDPOINTS WITH APPTRANA

SOLUTION HIGHLIGHTS:

- Blocked over 800 million API attacks and 600+ million DDoS attacks every quarter
- Discovered and protected over 6,000 API endpoints, including shadow and undocumented endpoints
- Strengthened protection with AI & custom rules tailored for each API behavior
- Achieved 72-hour SLA-based remediation for all critical, high, and medium API vulnerabilities
- Reduced AWS costs, ensured zero false positives, and maintained an audit-ready posture



ABOUT THE CUSTOMER:

The customer is one of India's leading fintech unicorns, offering digital financial solutions designed for small and medium-sized businesses. Their platform helps businesses manage payments, track transactions, and streamline daily operations through a mobile-first, API-driven SaaS model. As the platform scaled rapidly, securing its growing API infrastructure became critical to ensure performance, reliability, and user trust.

KEY CHALLENGES:

The customer's API infrastructure was expanding rapidly in both volume and complexity. As API usage became central to business operations, performance, security, and cost efficiency had to be tightly aligned. Several key challenges emerged:

- **Zero latency was mission-critical.** APIs powered high-frequency, user-facing features across web and mobile apps. Any added delay could degrade the experience and increase churn risk.
- **Frequent API-targeted DDoS attacks** on critical endpoints were starting to impact availability and infrastructure stability.
- **The existing API gateway lacked the flexibility** to apply traffic limits based on individual endpoint sensitivity. A one-size-fits-all approach was no longer viable.
- **AWS ingress billing was rising sharply** due to a high volume of unwanted traffic, including malicious requests and automated abuse.
- **The security team could not efficiently block bot networks,** especially when traffic originated from entire CIDR (Classless Inter-Domain Routing) ranges. This led to repeated abuse through credential stuffing and scraping.
- **There was no visibility into shadow APIs** that may have been exposed outside the gateway, increasing the risk of unmanaged access points.
- **There was zero tolerance for false positives.** Even a single legitimate request being blocked could disrupt business-critical workflows for thousands of users.

These challenges highlighted the need for a security solution that could provide complete API lifecycle coverage. The customer required a platform that could scale with traffic, eliminate visibility gaps, reduce attack traffic, and adapt quickly to changes, all without compromising performance.

SOLUTION:

To secure their fast-growing API ecosystem, the customer adopted AppTrana's fully managed API security. They received complete coverage, including discovery, classification, real-time protection, monitoring, and continuous improvement. The solution helped address immediate threats while supporting long-term operational scale and stability.

- **Comprehensive API discovery, including shadow endpoints:** Continuous discovery enabled full visibility into all exposed APIs, including undocumented shadow APIs that were not routed through the API gateway. Over 6,000 APIs were discovered and brought under protection, helping eliminate blind spots and ensure complete coverage.

- **Risk-based API onboarding and classification:** APIs were assessed and categorized based on their exposure, sensitivity, and functionality. High-risk endpoints, such as login, payment, and user data APIs, were prioritized for onboarding and stricter security enforcement, followed by a progressive rollout of protection across lower-risk APIs.

- **Positive security enforcement through schema validation:** A positive security model was applied to allow only well-structured and expected API calls. Requests with unknown fields, incorrect formats, or unexpected structures were blocked at the AppTrana level, significantly reducing the attack surface**.**

- **DDoS protection using AI:** AppTrana's AI-powered capability monitored traffic patterns in real time and set appropriate rate limits for each API based on its sensitivity and behavior. High-risk APIs like login and payment were given stricter controls. Smart Mitigation Rules were also used to block entire CIDR ranges linked to malicious traffic, helping stop large-scale DDoS attacks without affecting real users.

- **Bot mitigation:** AppTrana identified and blocked bot attacks targeting login and payment APIs. These included credential stuffing to hijack user accounts, carding attacks using stolen card data, and bots trying to exploit referral programs through fake account creation. By analyzing behavioral patterns, malicious automation was stopped without disrupting genuine users.

- **VPC Peering to reduce AWS ingress billing and improve performance:** Integration with the customer's AWS environment through VPC Peering enabled early filtering of traffic. This reduced ingress bandwidth usage and improved response times by keeping traffic within the cloud backbone.

- **AI-led monitoring with expert oversight:** API traffic was continuously monitored using AI models that detected anomalies and flagged unusual behavior. This was backed by Indusface's 24x7 managed security team, which reviewed alerts, fine-tuned protection policies, and ensured accurate detection without false positives.

- **Continuous discovery, change detection, and protection updates:** Newly exposed APIs were continuously discovered and brought under protection, while existing APIs were monitored for changes in schema or exposure. Based on these insights, security policies were proactively updated, and all critical, high, and medium vulnerabilities were remediated within a 72-hour SLA through AppTrana SwyftComply's autonomous remediation.

This end-to-end approach helped the customer gain complete control over their API ecosystem. It ensured high availability, strong performance, reduced operational overhead, and resilience against evolving threats, without compromising user experience or agility. It also helped them stay audit-ready at all times, with updated protections and zero vulnerability reports for compliance.

RESULTS:

A view of the 90-day attack trend:

- 800+ million API attacks blocked every quarter
- 600+ million DDoS attacks mitigated per quarter
- Over 6,000 APIs discovered and brought under protection, including shadow and undocumented endpoints
- ~30 million attacks blocked using custom rules and positive security models
- Implemented VPC Peering to improve response times and reduce AWS ingress bandwidth billing for the customer
- Zero false positives assurance
- All critical, high, and medium vulnerabilities remediated within a 72-hour SLA
- Stayed audit and compliance ready with complete API visibility, documentation, and protection in place

Whitepaper

INDUSFACE™

DALLAS | BENGALURU | VADODARA | MUMBAI | NEW DELHI

Indusface is a leading application security SaaS company, securing over 5,000 customers across 95 countries with its award-winning platform. Funded by institutional investors, it has been a category leader in Gartner Peer Insights™ for the past three years.

The industry's only AI-powered, all-in-one AppSec platform helps businesses discover, detect, remediate, and protect web applications and APIs at an internet scale, backed by a 100% uptime guarantee.

Contact Us: +1 866 458 3058
Email - sales@indusface.com | Website - www.indusface.com