SALT

Tackling API Threats: A Buyer's Guide to Purpose-Built Security Solutions

Table of Contents

Note: Unless otherwise specified, statistics and findings cited throughout this guide are sourced from the Salt Security State of API Security Report, Q1 2025

01 API Security Challenges in the Age of AI & Digital Transformation: Understanding the Modern Threat Landscape

APIs are the engines of modern digital transformation, driving innovation, integration, and efficiency. However, their rapid proliferation and constant evolution have created a complex and expanding attack surface that traditional security measures struggle to protect. Organizations face an explosion in API counts, with many seeing over 50-100% growth annually, yet visibility remains a critical issue. Only 15% of organizations are very confident in their API inventory's accuracy, and just 20% monitor their APIs in real-time, leaving significant blind spots.

15%

of organizations are very confident in their API inventory's accuracy

This lack of visibility translates directly into risk. Nearly all organizations (99%) experienced API security issues in the past year, and a majority (55%) have delayed application rollouts due to these concerns. Common problems include vulnerabilities like misconfigurations (a top issue at 37%), sensitive data exposure (34%), and authentication flaws (29%). The impact is significant; Gartner's warning emphasizes the urgency: "**The average API breach results in at least 10 times more leaked data than the average security breach.**"* Compounding the issue, a vast majority of attacks (95%) target authenticated users, bypassing simple perimeter defenses, while existing security tools are often ineffective against API attacks (only 17% rate them "very effective").

The reality is clear: securing APIs requires a dedicated, modern approach. Standard security tools weren't built for the unique challenges of API logic and behavior. Organizations need a best-of-breed API security platform offering continuous discovery, proactive posture management, and intelligent, AI-driven threat detection to navigate this evolving threat landscape and protect critical assets.
AI-based discovery.

02

The Modern API Security Platform Checklist

A true API security platform delivers comprehensive capabilities. Use this checklist to assess core requirements:

- □ **Complete API Discovery:** Continuously discovers and inventories all APIs (internal, external, shadow, zombie) across all environments.
- □ **Rich Context:** Provides deep understanding of each API, including functionality, data sensitivity, risk posture, and ownership.
- □ **Sensitive Data Identification:** Accurately identifies and classifies sensitive data exposure within API traffic, supporting custom definitions.
- □ **Posture Governance Engine:** Enforces security policies (custom and standard-based, e.g., OWASP) consistently across all APIs.
- □ **Automated Compliance:** Monitors and reports on compliance with relevant regulations and standards (PCI, HIPAA, etc.).
- □ **AI-Driven Threat Detection:** Uses advanced AI/ML and behavioral analysis to detect the full spectrum of API attacks (OWASP Top 10, BOLA, logic abuse).
- □ **Attacker Intent Analysis:** Accurately distinguishes malicious attackers from benign anomalies, minimizing false positives.
- □ **Authenticated Threat Focus:** Effectively detects attacks originating from authenticated users (the vast majority of attempts).
- □ **Low-and-Slow Attack Detection:** Identifies subtle reconnaissance and attack patterns over time.
- □ **Actionable Remediation:** Provides clear guidance and correlates attacker activity for efficient response.
- □ **Ecosystem Integration:** Seamlessly integrates with SIEM, SOAR, ticketing, CI/CD pipelines, and enforcement points (WAFs, Gateways).
- □ **Shift-Left Capabilities:** Integrates security insights and checks early in the development lifecycle.
- □ **Enterprise Scalability:** Scales to handle high traffic volumes across complex, hybrid environments.
- □ **Flexible Deployment:** Offers easy deployment options, including agentless methods where possible.
- □ **Privacy Architecture:** Protects sensitive data through methods like metadata analysis.
- □ **Platform Manageability:** Includes RBAC, audit logs, and an intuitive interface.

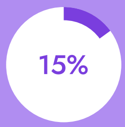
03 Key Questions Your API Security Solution Must Answer: Defining Your Needs

Before engaging with vendors, clearly define the critical questions your API security solution must address. These questions map directly to the core challenges of securing modern API ecosystems, often highlighted by significant gaps in current practices.

What APIs do I actually have, and where are they?

Robust API logging and monitoring are crucial for maintaining API security, providing a detailed audit trail, and prompt security and violation alerts, enabling swift incident response, and minimizing the risk of unauthorized access

The first hurdle for any API security program is achieving complete visibility. Can the solution answer this fundamental question by providing a comprehensive, accurate, and continuously updated inventory of all APIs across your landscape? This must include the shadow, rogue, and zombie APIs often missed by manual documentation or basic scans. A simple list isn't enough; the solution must provide rich context for each API – identifying ownership, the type of data handled, and its potential risk profile. Without this foundational visibility, effective security monitoring and policy enforcement are impossible.



Only 15% Very Confident: Complete visibility is foundational, yet most organizations lack confidence in their current API inventory.

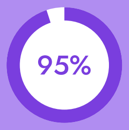
Are my APIs secure, compliant, and properly configured?

Beyond just knowing what APIs exist, you must understand and control their security state. Can the solution proactively assess your APIs against defined security standards and compliance mandates? It must identify security misconfigurations, policy deviations, excessive permissions, and instances where sensitive data is exposed unnecessarily. Can it enforce your desired security posture consistently, providing automated checks and alerts? This capability is vital for preventing breaches stemming from common posture issues and avoiding costly application rollout delays experienced by many organizations.

Only 10% Have Strategy: Effective posture governance is rare, leaving APIs vulnerable to misconfigurations and non-compliance.

How are attackers attempting to exploit my APIs?

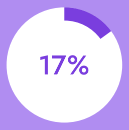
Effective protection demands understanding how attackers target APIs today. Since most attack attempts originate from authenticated sources, solutions relying solely on authentication checks are insufficient. Your platform must detect sophisticated attacks that exploit API-specific vulnerabilities and business logic flaws – techniques often missed by signature-based tools. Can it reliably identify OWASP API Top 10 threats like BOLA and injection, credential stuffing attacks, and subtle reconnaissance patterns that signal an impending attack?



95% Authenticated Attacks: Attackers overwhelmingly use legitimate credentials, bypassing traditional defenses to target API logic.

Which API risks pose the greatest threat and require immediate action?

Security teams are frequently inundated with alerts, making it hard to discern real threats from benign anomalies. With most organizations finding their current tools only somewhat effective, a key requirement is intelligent prioritization. Can the solution analyze behavior, data context, and attacker activity over time to identify malicious intent accurately? It must cut through the noise to highlight the specific users and API interactions that pose the most significant risk, allowing your limited security resources to focus where it matters most.



Only 17% Find Tools "Very Effective": Current tools often overwhelm teams with noise, making prioritization difficult.

How can my teams efficiently fix vulnerabilities and stop active attacks?

Discovering a threat or vulnerability is only the first step. The solution must empower your teams to act swiftly and effectively. Does it provide clear, context-rich, and actionable remediation guidance suitable for both security analysts and developers? Crucially, how well does it integrate with your existing ecosystem? Look for seamless connections to SIEM, SOAR, ticketing systems (Jira, ServiceNow), and enforcement points (WAFs, Gateways) to automate workflows, track remediation progress, and enable rapid blocking of active threats, maximizing efficiency despite common resource limitations.

30% Cite Budget Obstacles: Effective remediation requires clear guidance and seamless integration, especially with budget and resource constraints.

04 Identifying Limitations in Traditional API Security Approaches

Evaluating API security requires recognizing that many existing tools simply weren't built for the job. Relying on repurposed or legacy technologies creates dangerous gaps and a false sense of security. Don't let these common limitations undermine your protection

Static or Incomplete API Visibility

Are you flying blind? Relying on point-in-time scans, outdated documentation, or limited network views means you are. In today's dynamic environments where APIs change constantly, this static approach guarantees you'll miss critical assets. Shadow APIs, internal east-west traffic, and forgotten zombie APIs remain invisible – creating an unknown, unprotected attack surface that attackers will find. You cannot protect what you cannot see in real-time

Weak or Inflexible Posture Governance

Visibility without control is useless. Generic security controls and manual oversight utterly fail to govern the specific nuances of API configurations. Traditional approaches lack the robust, flexible policy engines needed to define and enforce rules for API authentication, authorization, or sensitive data handling. This inevitably leads to critical misconfigurations, compliance failures, and vulnerabilities stemming directly from inconsistent or non-existent API-specific governance.

Inadequate Scalability & Performance

Modern API traffic can overwhelm tools not explicitly designed for its volume and speed. Legacy security technologies often choke, introducing latency, demanding excessive resources, or simply failing under load. Workarounds like data sampling might seem practical, but they mean you're analyzing incomplete data, inevitably missing subtle attacks hidden within the noise of high-throughput interactions. If a tool can't keep pace, it can't protect your enterprise.

Superficial Threat Detection (High Noise/Missed Attacks)

Don't be fooled by basic alerts. Traditional signatures are blind to attacks targeting unique API logic – the most common and damaging threats. Basic anomaly detection fares little better, drowning security teams in a flood of false positives while missing sophisticated, low-and-slow attacks that cleverly mimic legitimate behavior. These superficial methods lack the context to understand true malicious intent, leaving your APIs exposed to advanced threats.

05 Key Considerations for a Best-of-Breed API Security Solution: Evaluating Critical Capabilities

Choosing the right API security platform requires thoroughly evaluating its core capabilities. A best-of-breed solution provides deep visibility, proactive control, intelligent threat detection, and seamless integration. Use the following tables to structure your evaluation with API security vendors:

a) Comprehensive API Discovery & Inventory

Salt Security provides unparalleled visibility by leveraging AI/ML to continuously discover and contextualize the entire API landscape, including often-missed shadow and undocumented APIs across all environments. This ensures you have the complete, accurate foundation needed for effective security.

What to look for	Why it matters	Questions to ask
Cloud-native architecture; Rapid, often agentless deployment; Continuous, automated discovery (all types: shadow, zombie, internal, external); AI/ML-powered accuracy & context.	Complete visibility is the foundation. You can't protect what you can't see. Addresses low inventory confidence. Understand attack surface, prioritize risks, and apply policies. Vital for dynamic environments.	• How do you ensure the discovery of all APIs (undocumented, internal) across our specific environments? • How is the inventory automatically updated? • What context is provided (data, risk, owners)? • How does AI/ML enhance discovery/classification accuracy?
Rich, dynamic inventory: endpoints, risk analysis, sensitive data mapping, functionality classification, potential owners.	Enables effective security management and prioritization.	• Can you demonstrate discovering APIs missed by our current tools in a POC? • What level of detail does the inventory provide regarding sensitive data exposure?

Automated compliance monitoring, alerting on violations, Integration into remediation workflows, and Shift-left integration support.	Complete visibility is the foundation. You can't protect what you can't see. Addresses low inventory confidence. Understand attack surface, prioritize risks, and apply policies. Vital for dynamic environments.	• How is compliance status tracked and reported? • How configurable are alerts for policy violations (severity, destination)? • What ticketing systems are supported for remediation workflows? • What feedback is provided to developers pre-production? • Can it analyze API design files?

b) Proactive API Posture Governance

Salt Security enables proactive risk reduction through its industry-leading posture governance engine. Featuring the unique Policy Hub and flexible controls, Salt allows organizations to easily define, automate, and enforce security standards consistently across all APIs.

What to look for	Why it matters	Questions to ask
Robust, flexible policy engine; Pre-built templates (OWASP, PCI, HIPAA, etc.); Custom, complex rule creation; Automated enforcement & auditing.	Prevents breaches from misconfigurations (top OWASP risk) & design flaws. Ensures compliance. Reduces risk proactively. Critical as only 10% have a posture strategy, though many plan one. Avoids rollout delays (affected 55%). Fosters secure development culture.	• How flexible/granular is the policy engine? • Can we implement our specific complex rules? • Pre-built templates for OWASP, NIST, PCI DSS? • How is policy enforcement automated/monitored? • How do you integrate findings into dev workflows/CI-CD? • Is there a Policy Hub?

c) AI-Driven Behavioral Threat Protection

Salt Security moves beyond basic detection with cloud-scale AI and patented Intent Analysis. By analyzing behavior and context over time on the industry's largest API data lake, Salt accurately identifies and stops sophisticated attacks while minimizing false positives.

What to look for	Why it matters	Questions to ask
Sophisticated AI/ML engine; Training on massive data lake (trillions of calls); Accurate baselining; Contextual analysis; **Intent Analysis** to differentiate malicious behavior.	Modern API attacks exploit unique logic & mimic users, invisible to traditional tools. Intent analysis stops attacks missed by basic anomaly detection & signatures. Reduces false positives for SOC efficiency.	• Explain your AI/ML approach. Data source? Baselines? • How do you specifically differentiate malicious intent from anomalies? Examples? • How do you detect BOLA, logic abuse, credential stuffing,, especially from authenticated users? • Typical FP rate?
Proven detection of OWASP Top 10 (80% of attacks), BOLA, logic abuse, low-and-slow attacks; Adaptive learning; Automated response/blocking options; SOC-friendly interface.	Protects against unknown & unique API vulnerabilities. Vital given high impact of API breaches. Adaptive AI is needed for evolving threats (including GenAI risks). Clear interface aids rapid response.	• How is threat info presented to SOC analysts? • Automated response/blocking capabilities?

d) Enterprise Readiness & Integrations

Designed for complex enterprise environments, Salt Security offers seamless integration, proven scalability, and a privacy-first architecture. This ensures operational efficiency, trust, and the ability to manage API security effectively at scale.

What to look for	Why it matters	Questions to ask
Scalable architecture; Flexible deployment (multi-cloud, hybrid, on-prem, K8s); Easy setup/maintenance; Privacy-first design (e.g., metadata analysis); Robust integrations (SIEM, SOAR, ITSM, WAFs, CI/CD).	Must fit existing infrastructure & workflows. Needs to scale reliably as API usage grows. Strong integrations enable efficient response & automation. Privacy controls are crucial for compliance/trust. ROI depends on integration. Addresses program complexity & tooling gaps.	• What deployment options are there for our environments? • How easy is the setup/maintenance? • How is data privacy/compliance (GDPR) handled? • Where is the data processed/stored? • Specific integrations with our key tools (list them)? • How does it scale for high traffic? • Enterprise features like
Enterprise-grade features (RBAC, auditing); Strong vendor support & services.	Ensures secure, manageable operation in large organizations. Provides necessary assistance for success.	• What levels of Role-Based Access Control (RBAC) are available? • Can roles be customized? • What platform activities and configuration changes are included in audit logs? • What are the standard support SLAs and options (e.g., premium support, TAM)? • What training resources are provided?

06 Salt Security: The Leading API Security Solution

Salt Security takes a fundamentally different, API-centric approach to security. By applying cloud-scale big data, AI, and machine learning, Salt provides the deep context and real-time analysis needed to protect the entire API lifecycle. Because Salt focuses holistically on discovering all APIs, continuously improving posture, and understanding attacker intent, we empower security teams to effectively mitigate risk, ensure compliance, and enable the business to innovate fearlessly. Salt is the only solution offering the necessary visibility, adaptive intelligence, and robust protection required to defend against today's sophisticated API threats.

Take the Next Step:

Don't rely on inadequate tools for your most critical assets.
See the Salt difference for yourself

Get a demo

*Gartner, Market Guide for API Protection, Dionisio Zumerle, Aaron Lord, Esraa ElTahawy, Mark O'Neill, 29 May 2024