WHITE PAPER

11 Critical Capabilities of API Detection and Response

Evolving your API security strategy

Introduction

APIs play a critical role in every application your organization builds for customers, uses internally, and makes available to vendors and suppliers. Their job: exchange information (often sensitive data) between technologies. Their location: not only in your applications but also across your cloud migrations, generative AI tools, and digital supply chain.

The challenge is, APIs have also taken a prominent spot in your organization's attack surface.

As enterprises rush to innovate, APIs are often hastily developed, insufficiently tested, and released into production with misconfigurations and missing security controls. What's more, these APIs have amassed into a sprawl-like dynamic to the point security teams lack visibility into a major portion of their API estates. And without proper visibility, organizations:

1. Cannot detect APIs that are unmanaged, forgotten, and lingering with unchecked exposure to sensitive data, to the internet, and to attackers
2. In turn, cannot assess APIs' risks — for example, only 27% of enterprises with full API inventories know which of their APIs return sensitive data, down from 40% in 2023
3. End up with an attack surface full of API-centric vulnerabilities that attackers frequently — and often easily — exploit

Until recently, organizations have felt comfortable relying on a commonly used roster of tools for managing APIs and gaining baseline of protection. However, with 84% of organizations having experienced an API security incident in the past 12 months, up from 78% in 2023, something needs to change.

As API attacks increase in number and sophistication, it's time to explore adding new layers of protection to tools such as API gateways, web application firewalls (WAFs), and web application and API protection (WAAP) platforms.

These new layers should provide greater visibility into all APIs in your environment and their risks — including the large portion of APIs that are unmanaged, such as:

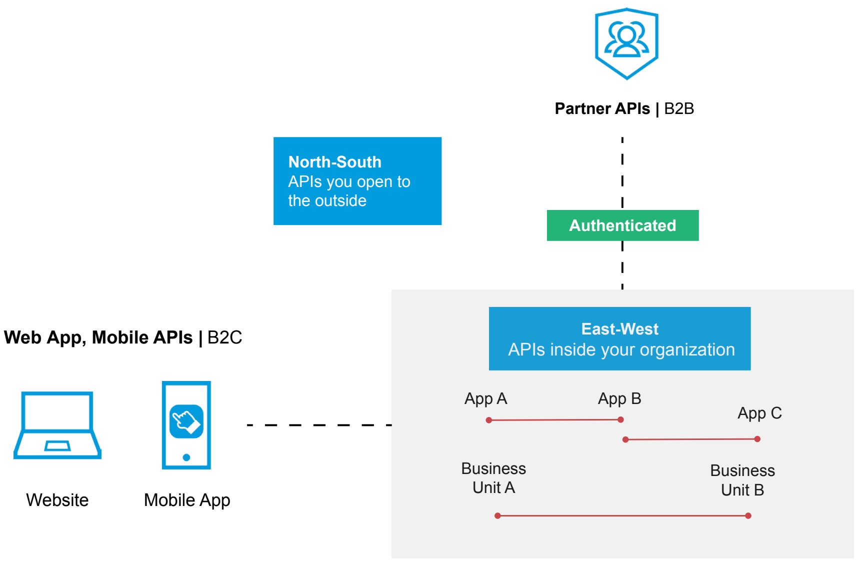
- Zombie APIs that should have been depreciated but remain active
- Shadow APIs that are undocumented and should either be eliminated or brought into formal governance processes

Organizations also need deeper capabilities for detecting and addressing API abuse and attacks, including every threat detailed in the OWASP Top 10 API Security Risks. And with an eye toward finding and remediating vulnerabilities throughout an API's full lifecycle, enterprises should adopt rigorous and real-time security testing for APIs — from early development stages through production.

Does this mean piling on a new tool for every problem that arises? No. Instead, it's more like you're ensuring an orchestra has the right players for the job, playing the right notes at all the right times — and in precise coordination with their counterparts.

When you think about how to add new layers to your API protection stack, consider the defense-in-depth approach that security teams apply to other threats — for example, deploying a gauntlet of controls to detect, prevent, and mitigate the effects of a ransomware attack. This is exactly how organizations should think about APIs.

In this white paper, we'll explore 11 critical capabilities that you can embed into your API security strategy, focusing on API threat detection and response.


Partner APIs | B2B

North-South
APIs you open to the outside

Authenticated

Web App, Mobile APIs | B2C

East-West
APIs inside your organization

Context is key

Where do API threat detection and response fit into your API security strategy?

As you've likely seen firsthand, APIs have changed how businesses operate by enabling more use cases, accelerating change, carrying more sensitive data, and being open to more users. It's not surprising that organizations have created many more API channels than web application interfaces. And the risk compounds as these proliferating APIs are embedded with increasing volumes of core business data and business logic.

Given the prevalence of APIs across the myriad technologies that security teams are already protecting (i.e., applications), most categories of security products support APIs in some fashion. However, APIs and applications are not the same; they even appear as different assets in some compliance frameworks. It's not enough to add piecemeal API threat protection capabilities to, for instance, an existing application security product. APIs merit more focus than they typically receive at most organizations. Today's security teams should view APIs as a separate asset class with a distinct set of risk attributes and look for critical capabilities for seeing and securing every API at scale.

In the past, if an organization had an API inventory and some baseline tools for API management and protection, they'd have a good shot at preventing a known range of common API attacks. Unfortunately, today's attackers often innovate like companies do, with a similar eye on continuous improvement.

- Malicious actors are logically evolving their tactics to circumvent tools they know that most organizations rely on to defend APIs.
- Similarly to how most enterprises use AI, attackers are augmenting their limited human capabilities with around-the-clock help from generative AI capabilities.
- Increasingly, attackers are looking for weak links in an enterprise's API-connected digital supply chain, such as a company's B2B partners who may not be prioritizing API protection.

For example, some forms of API abuse originate from customers and partners who have been granted API credentials but use them in unauthorized ways. There are also ways to hijack seemingly legitimate API credentials or security tokens. Hidden vulnerabilities in API client implementations are another attack vector that threat actors may exploit to abuse APIs in ways that traditional security tools cannot detect.

The good news is that the critical capabilities needed to protect APIs from fast-evolving attack methods are available at scale for organizations. Read on for details on 11 key capabilities that your team can start with as you take action to protect your APIs — and the data they exchange — from attacks.

Critical capability #1

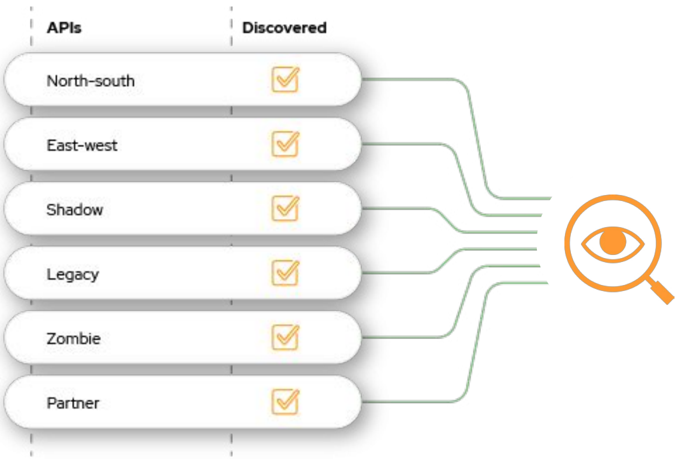
Continuous API discovery and posture management

A comprehensive and continuously updated inventory of APIs in use across the organization is a crucial foundation for any API security strategy. This is for the simple reason that an organization cannot protect what it does not know is in its environment. Many API security products claim to perform some level of API discovery but are limited to on-demand or daily operations. It's important to ensure that your platform's API discovery capabilities include:

- Automated and continuous discovery of APIs around the clock, including discovery of APIs that are only used once (on-demand or daily discovery is insufficient)
- Discovery of APIs across different technologies and infrastructure
- Discovery of newly deployed APIs and comparison with well-documented APIs to identify shadow APIs
- Risk scoring of each API service and endpoint — this helps both security and development teams cut through the noise and prioritize APIs with the biggest potential impact, if compromised
- Detection of instances of known API vulnerabilities, such as those outlined in the OWASP Top 10 API Security Risks

Improved visibility

Never lose sight of your API inventory ever again



APIs	Discovered
North-south	☑
East-west	☑
Shadow	☑
Legacy	☑
Zombie	☑
Partner	☑

Critical capability #2
Visualization of APIs' behavior

The ability to visualize actual API behavior (API calls) is fundamental to an API security platform. This capability is required to enable key stakeholders from security, development, and operations to view and understand how APIs are being used or abused so they can communicate among teams and investigate cases. Specific visualization capabilities to look for include:

- **Investigation:** Any alert should include the ability to inspect the original API activity call by call to identify the specific trigger for the alert.
- **Data fidelity and enrichment:** For every API call, it should be possible to tell who the user is, what operation they used, what records they accessed or manipulated, what headers and parameters were used, etc.
- **Data privacy:** While data fidelity is important, sensitive data can't be stored at rest. A solution should analyze the traffic and only send relevant metadata to update dashboards.

Critical capability #3

Uncovering API abuse attempts via context on user entities

Security teams need the ability to track malicious activity to entities such as IP addresses and business process entities like payment IDs. This can be extremely valuable when combined with capabilities for correlating attacks from different IPs in instances when other relevant identifiers can offer context into API abuse.

Let's say an unknown user is calling a retail company's API using /api/getpaymentID/50 as their ID. In this scenario, the retailer's security team knows that every other user in the company's platform is tied to one type of payment ID. If a security analyst sees that suddenly, the unknown user is making repeated calls, each time slightly adjusting the ID number (/api/getPaymentID/51 … 52 … 53 … 54), this is a key indicator of attempted API abuse.

Having real-time insight into atypical user behavior can be the difference between a thwarted breach attempt and a successful API attack.

$943,162

The average cost to remediate API security incidents, according to U.S.-based CISOs, CIOs, and CTOs who reported experiencing such events in the past 12 months.

Learn more about your peers' views and experiences in the 2024 API Security Impact Study.

Critical capability #4
Behavioral analytics and detection

While analyzing individual API calls from user entities — or even individual sessions — can help security teams, it's important to have comprehensive API threat detection focusing on the big picture. Seek out capabilities for gaining a deep understanding of behavioral patterns and anomalies across the full API estate. To determine if an API's behavior is abnormal, indicating that it might be compromised, API use should be analyzed over longer time periods and with a foundation of context built by thorough behavior tracking over long time periods. This provides security teams with a reliable baseline as they continuously monitor behavior to detect anomalies.

Critical capability #5
API spec drift detection

APIs are in constant flux amid shifting market demand and business requirements. As a result, organizations are continuously releasing new endpoint implementations to meet fast-evolving enterprise needs, fix bugs, and introduce technical improvements. Updating API documentation in lockstep with these changes, based on API specs, is critical, and special attention should be dedicated to ensuring API traffic always aligns with its specifications.

To make APIs resilient against abuse and attacks, organizations should seek out capabilities for detecting API specification drift. This helps enterprises pinpoint any discrepancies or gaps in API documentation by continuously comparing real-time API traffic to defined specs.

If the API spec drift function uncovers any mismatches or undocumented endpoints being accessed in production, it can alert developers and security teams, allowing them to:

- Stay ahead of issues before they become critical
- Ensure APIs operate as intended
- Bolster security for the applications these APIs support
- Maintain the integrity of the enterprise's API ecosystem

Critical capability #6
B2B and east-west API coverage

The biggest growth area in API use is in B2B use cases — both internally and externally facing. API security must cover B2B, machine-to-machine APIs, including both north-south (externally facing) and east-west (internally facing) instances.

Although B2C web applications are afforded protection from WAAP and WAF platforms, some of the most sensitive types of API activity, such as internal east-west APIs or proprietary application functionality exposed to partners through B2B APIs, can still be compromised even when passing through WAAPs.

Often, once a user is authenticated on a B2B partner API, they are assumed safe, and no further monitoring is performed. This creates a critical gap in many organizations' API security posture. To provide a complete picture of API activity and the broader threat landscape, organizations must use an approach that provides effective visibility, observability, and monitoring for all use cases.

Critical capability #7
Meaningful alerts with context

Once an organization has visibility into its API activity and behavioral analytics at scale, alerts on API activity become much more meaningful. But how can you make sure you're focusing attention and resources on true API threats? An attacker confidence engine can use advanced machine learning algorithms trained to evaluate external and internal signals — including API behavior, network traffic patterns, geolocation data, threat intelligence feeds, and other contextual factors — to determine the confidence level that a detected runtime incident is the result of malicious activity. This capability can help a security team quickly zero in on critical threats and should be complemented by functions that create automatic remediation and notification flows for high-probability attacks.

Critical capability #8
Customized, automated responses

Traditional inline API approaches can take automated actions to block suspected API attacks, with the catch that organizations must be able to identify the attack. Because behavioral analytics and anomaly detection on APIs are performed over time with much greater business context, the depth of detection allows for anomalies to surface. This enables a wide range of automated and customized responses, which can be performed with high accuracy. Examples include:

- Blocking or throttling traffic at supported API gateways and content delivery network (CDN) edge filters
- Emailing notifications for security and business stakeholders
- Creating tickets for developers
- Triggering of webhooks

What can organizations do to help stretched security teams maximize their team and energy as API threats grow? Look for automation capabilities that improve efficiency and productivity by simplifying the creation and management of multiaction workflows. The right automation capabilities should offer a no-code visual designer interface that can create complex event response processes and synchronize incident-related data between your core API security solutions and myriad third-party services, including ServiceNow, Jira, and Azure DevOps.

Critical capability #9
API traffic analysis

Organizations need always-on capabilities for recording, visualizing, and analyzing API traffic in their environments without deploying a data lake. By recording API data flows that match specific criteria across application environments — including typical and anomalous API activity — organizations can hunt for threats more effectively while managing the risk exposure of suspicious users and unusual API behaviors. It's important to have API traffic audit functions that can be custom tailored to a particular use case, allowing organizations to capture and retain traffic according to predetermined filters and rules.

Critical capability #10
Rigorous, real-time API testing

In the rush to innovate, organizations are releasing APIs into production with vulnerabilities and design flaws that often go undetected. Organizations can prevent these issues by adopting a shift-left approach to API testing in development. Core capabilities include:

- Running automated tests that simulate malicious traffic, including the types covered in the OWASP Top 10 API Security Risks
- Inspecting API specifications against established governance policies and rules
- Testing APIs on demand or as part of a CI/CD pipeline

Critical capability #11
Platform-neutral protection

API services are generally implemented by different groups in an organization, who often use a diverse collection of platforms and technologies. For example, some APIs are implemented on-premises, while others run in the public cloud. Often, organizations use intermediary technologies — such as reverse proxies, API gateways, WAFs, and CDNs — which offer business value but create complexity for API visibility.

The ability to access API activity data from each of these technologies is imperative. A platform-neutral API threat protection approach ensures that your organization always has a comprehensive picture of API activity, regardless of the implementation details or infrastructure in use. This will provide protection coverage for:

- All departments, acquired companies, and environments
- Both sanctioned and shadow APIs, whether they use the API gateway or not

A platform-neutral approach will also extend visibility beyond north-south APIs and include public, partner, and internal east-west APIs.

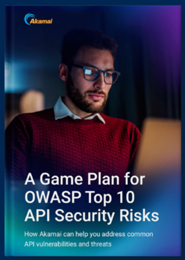
Ensuring that your API threat protection platform’s visibility is as broad as possible will protect your organization against insider threats and abuse of APIs by partner organizations — in addition to risks from external threat actors.

Conclusion

APIs are a key component of organizations’ ability to serve customers, generate revenue, and operate efficiently in today’s digital and cloud-centric economy. However, their continuous growth, proximity to sensitive data, and lack of security controls make APIs a significant source of risk.

Akamai API Security provides all 11 of the critical capabilities covered in this white paper, helping organizations build on their existing approaches with essential functions, such as:

- Discovering APIs
- Assessing risks (including exposure to sensitive data)
- Detecting API abuse and attacks
- Testing APIs for security risks and vulnerabilities



A Game Plan for OWASP Top 10 API Security Risks

How Akamai can help you address common API vulnerabilities and threats

Learn more about how to protect against the OWASP Top 10 API Security Risks.

Learn how we can help you by scheduling a customized Akamai API Security demo.

Published 01/25.