WHITE PAPER

10 Critical Capabilities of API Detection and Response

Evolving your API security strategy

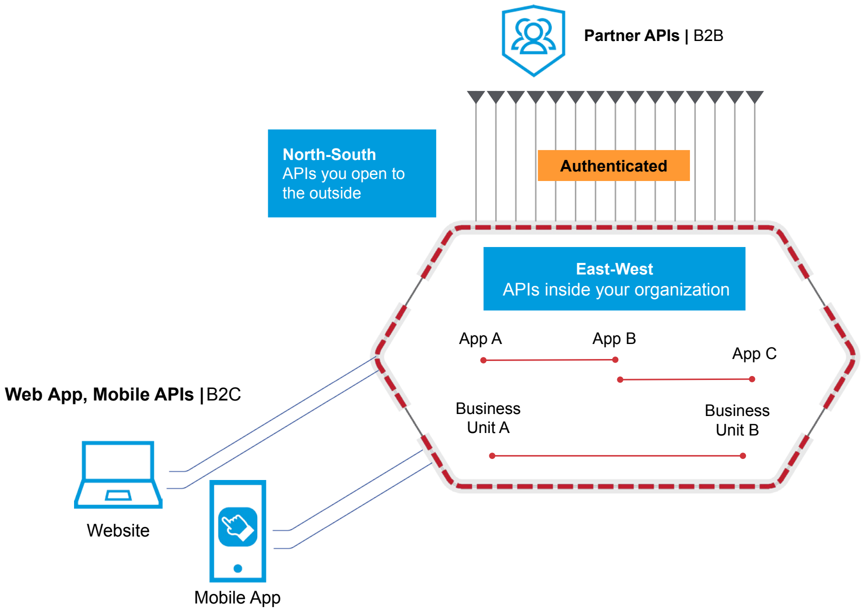
Introduction

APIs are the key building blocks that drive innovation, and business-to-business (B2B) and business-to-consumer (B2C) applications are at the center of this transformation. This means it is essential to protect critical — and often sensitive — communications internally among microservices, and externally among clients and partners. Most organizations now recognize that a sound application security strategy is necessary for long-term business success, and they use security technologies like web application and API protection (WAAP) platforms, cloud security features and products, and security testing tools to reduce application security risk. It's important to recognize how attacks have evolved to circumvent WAAPs and target APIs within organizations. It is time to discuss how to adjust your API security strategy in advance of these threats.

Where does API detection and response fit into an API security strategy?

Over the past few years, organizations have created many more API channels than web application interfaces, and these APIs include increasing volumes of core business data and business logic. APIs have changed the way businesses operate since they enable more use cases, accelerate change, carry more sensitive data, and are open to more users.

What is your API landscape?



While most security product categories support APIs in some way because of their increasing prevalence, APIs are a different asset class and even appear as a different asset in some compliance frameworks. Adding API threat protection capabilities to a legacy security product, such as a WAAP platform, doesn't address the new challenges introduced by API assets. Security organizations must treat APIs as a separate asset class and recognize the critical capabilities that fully protect APIs at scale.

Let's start with the fundamentals of how API protections have changed to meet emerging threats. In the past, if an organization had a full inventory of their APIs and a robust WAAP, API threats could be generally avoided. Now, attacks are targeting APIs within organizations and their partner organizations in ways designed to circumvent the WAAP.

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

The good news is that the critical capabilities needed to protect APIs from emerging trends, specifically detection and response, are already available at scale for organizations. The following pages offer careful consideration of the critical capabilities that make these platforms effective against an ever-changing API threat landscape.

Critical Capability #1
Platform-neutral protection

API services are generally implemented by different groups in an organization, often using a diverse collection of platforms and technologies. For example, some APIs may be implemented on-premises while others may run in the public cloud. There may also be intermediary technologies in use, such as reverse proxies, API gateways, web application firewalls (WAFs), and content delivery networks (CDNs), which create complexity for API visibility.

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

- All departments, acquired companies, and environments
- Both sanctioned and shadow APIs, regardless of whether they utilize the API gateway
- Extended visibility beyond north-south APIs, including 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.

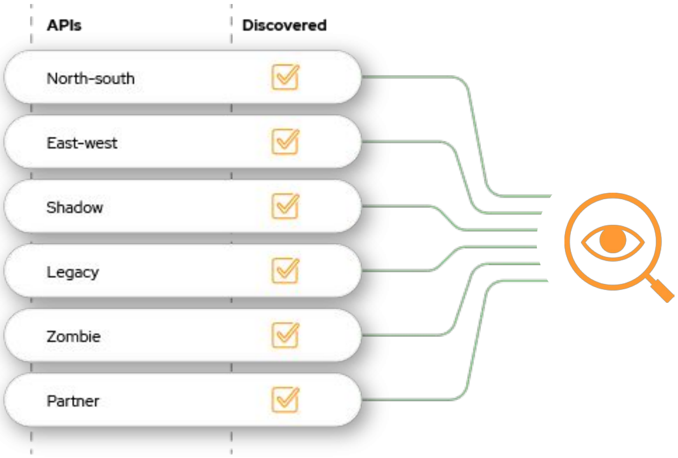
Critical Capability #2
Continuous API discovery and posture management

A comprehensive and continuously updated inventory of all 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 it has in its environment. Many API security products claim to perform some level of API discovery but are limited to on-demand or daily operation. 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 all 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
- Detection of instances of known API vulnerabilities, such as those outlined in the OWASP API Top 10

Improved visibility

Never lose sight of your API inventory ever again



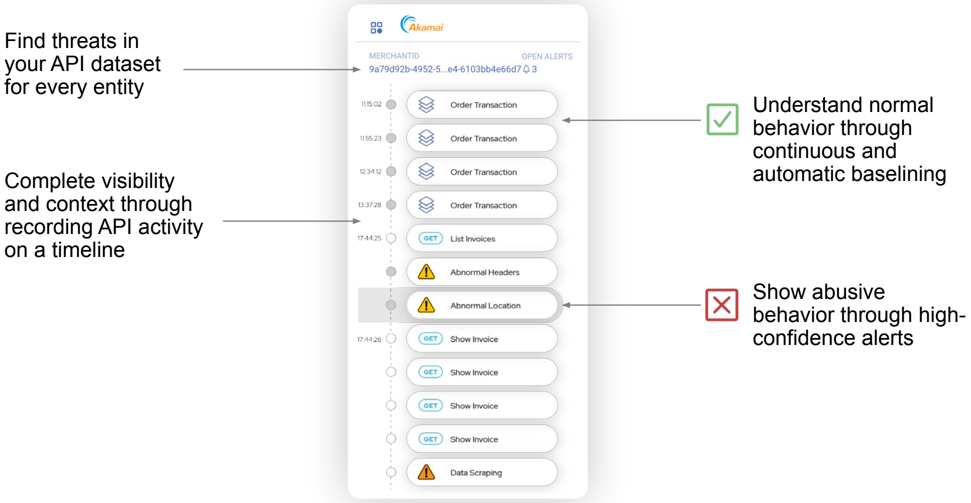
APIs	Discovered
North-south	☑
East-west	☑
Shadow	☑
Legacy	☑
Zombie	☑
Partner	☑

Critical Capability #3
Visualization of API behavior

The ability to show and visualize the actual API behavior (API calls) is a fundamental capability of an API security platform. This capability is required to enable key stakeholders from security, development, and operations departments 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.
- **Threat hunting:** Historical data should extend to at least a rolling 30-day view, including the ability to see all API activity and query for time frames and calls beyond specific alerts. This capability also helps with compliance.
- **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**: Although data fidelity is important, sensitive data can't be stored at rest. Tokenization is required to preserve data richness without storing sensitive data.
- **Timeline visualization:** Users should be provided with a view that makes it easy to move forward and backward through activity sequences.

Detect threats using behavioral analytics



MERCHANTID 9a79d92b-4952-5...e4-6103bb4e66d7
OPEN ALERTS 3

11:15:02 Order Transaction
11:55:23 Order Transaction
12:34:12 Order Transaction
13:37:28 Order Transaction
17:44:25 GET List Invoices
Abnormal Headers
Abnormal Location
17:44:26 GET Show Invoice
GET Show Invoice
GET Show Invoice
GET Show Invoice
Data Scraping

Find threats in your API dataset for every entity

Complete visibility and context through recording API activity on a timeline

✓ Understand normal behavior through continuous and automatic baselining

☒ Show abusive behavior through high-confidence alerts

Critical Capability #4
Tracking of multiple user entities

Understanding the entity and being able to see related API activity offers context for any use or abuse, so it's critical that your API protection platform has the sophistication to track each of these entities individually. This provides essential context, since normal activity for one category of users may be a warning sign of abuse for another user. The ability to view each entity's activity on a timeline provides vital visibility and contextual understanding. For example:

API activity	Participants	Entities	Business-process entities
Examples	internal users, B2B partners, external users	IP address, API token, merchant ID, session ID, tenant ID	payment ID, invoice ID

Critical Capability #5
B2B and east-west API coverage

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

While 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 the WAAP.

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 #6
Behavioral analytics and detection

Detection of sophisticated API threats is not possible by analyzing individual API calls — or even individual sessions. API detection and response requires deep understanding and learning from behavioral contexts. To know if an API's behavior is abnormal, which indicates it might be compromised, it's necessary to analyze API usage over longer periods. The technique of behavioral analytics determines a baseline normal user behavior and continuously monitors that behavior to detect anomalies.

The storage and compute resources required to perform this level of analysis for a typical enterprise's API activity make it impractical to deliver using scale-constrained, on-premises API security tools. EDR and XDR solutions led the way by showing that an architecture based on software as a service (SaaS) is required to perform meaningful behavioral analytics. The power and scale of the cloud allows for storage of data over time, while enabling analysis that determines normal user behavior over time, to detect the needle in the haystack revealing abuse. A SaaS approach has other benefits, such as faster and simpler implementation, and improved scalability and elasticity as your API usage grows.

Critical Capability #7
Meaningful alerts with context

Once an organization has visibility into all API activity and behavioral analytics at scale, alerts on API activity become much more meaningful. Organizations have then eliminated the need to anticipate every possible attack method by making the security monitoring approach more abstract. Baselining normal behavior and detecting anomalies also makes it possible to detect API abuse, which often can't be detected by any patterns or signatures. Additionally, being able to rewind the attack and see what happened before an alert provides valuable insights into the use and abuse of an API estate.

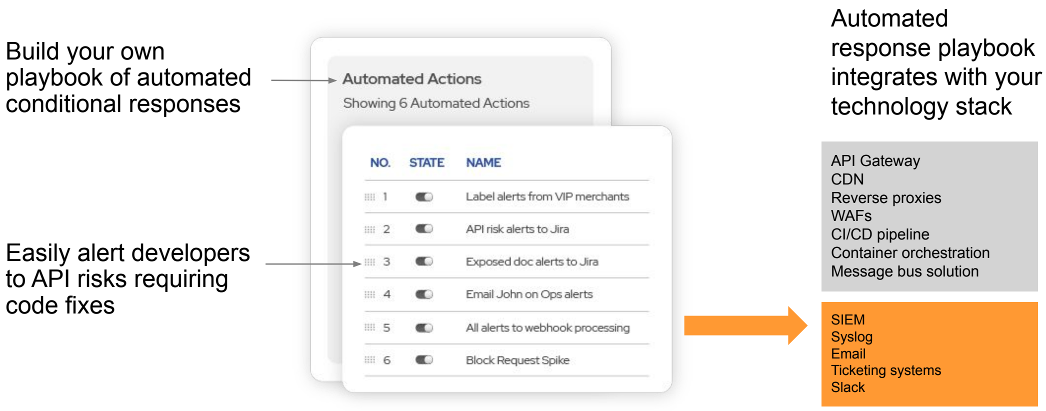
Critical Capability #8

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 attacks. 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 CDN edge filters
- Email notifications for security and business stakeholders
- Ticket creation for developers
- Triggering of webhooks

Responses are customizable for your business processes



Build your own playbook of automated conditional responses

Easily alert developers to API risks requiring code fixes

Automated Actions

Showing 6 Automated Actions

NO.	STATE	NAME
1	●	Label alerts from VIP merchants
2	●	API risk alerts to Jira
3	●	Exposed doc alerts to Jira
4	●	Email John on Ops alerts
5	●	All alerts to webhook processing
6	●	Block Request Spike

Automated response playbook integrates with your technology stack

API Gateway
CDN
Reverse proxies
WAFs
CI/CD pipeline
Container orchestration
Message bus solution

SIEM
Syslog
Email
Ticketing systems
Slack

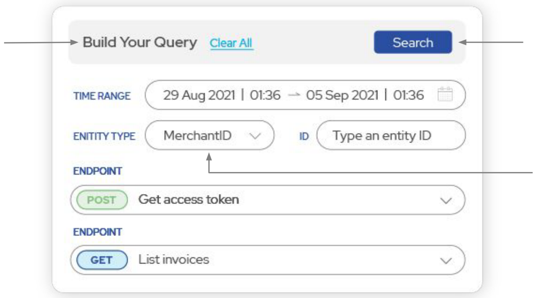
Critical Capability #9

Proactive investigation and threat hunting

Many organizations don’t have the luxury of waiting for an active security incident to occur before acting. A more effective approach is to identify unwanted situations and actively hunt for them. For example, an alert that detected abuse on one API might be identified performing the same behavior on another API through proactive threat hunting. Therefore, an API threat protection platform should include the capability to search for specific types of behavior beyond the alerts generated in response to active incidents. Threat hunting capabilities require access to historical data to find the abuse hiding within the API activity data. Single request solutions that don’t enrich the data to provide context are unable to stitch together a coherent story. Threat hunting and investigations are built upon the foundations of historical data.

The power to investigate and threat hunt at your fingertips

Investigate threats easily using advanced querying capabilities within the entire API dataset



TIME RANGE 29 Aug 2021 | 01:36 → 05 Sep 2021 | 01:36

ENTITY TYPE MerchantID ID Type an entity ID

ENDPOINT

POST Get access token

ENDPOINT

GET List invoices

Speed up investigations into alerts

Proactively hunt for abuse across different partners

Critical Capability #10

Observable data lake

Across all capabilities for a robust API security strategy, context is key to protecting any API over a long period. The best way to maintain sufficient context for observing threats, identifying potential vulnerabilities, and troubleshooting in the event of an attack is by logging all API behavior and keeping a backlog of this activity. This can be accomplished by having a data lake associated with the API security solution. Look for a data lake that provides the highest amount of historical detail to inform your strategy. While feeding basic request data to machine learning models can be helpful, having details like the request parameters allows organizations to actually act on their historical data in ways that will protect them from future threats and attacks.

#1 Platform-neutral protection	Ensuring that your API threat protection platform's visibility is as broad as possible will protect your organization against threats and abuse.
#2 Continuous API discovery and posture management	A comprehensive and continuously updated inventory of all APIs in use across the organization is crucial because organizations cannot protect what they do not know they have in their environment.
#3 Visualization of API behavior	Visibility is required so key stakeholders from security, development, and operations can view and understand how APIs are being used or abused and can communicate among teams and investigate cases.
#4 Tracking of multiple user entities	Understanding the entity and being able to see related API activity offers context for any use or abuse, so it's critical that your API protection platform has the sophistication to track each entity individually.
#5 B2B and east-west API coverage	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.
#6 Behavioral analytics and detection	To know if an API's behavior is abnormal, which indicates it might be compromised, it's necessary to analyze API usage over longer periods. The technique of behavioral analytics determines a baseline normal user behavior and continuously monitors that behavior to detect anomalies.
#7 Meaningful alerts with context	Once an organization has visibility into all API activity and behavioral analytics at scale, alerts on API activity become much more meaningful. Organizations have then eliminated the need to anticipate every possible attack method by making the security monitoring approach more abstract.

#8 Customized, automated responses	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.
#9 Proactive investigation and threat hunting	Many organizations don't have the luxury of waiting for an active security incident to occur before acting. A more effective approach is to identify unwanted situations and actively hunt for them.
#10 Observable data lake	The best way to maintain sufficient context for observing threats, identifying potential vulnerabilities, and troubleshooting in the event of an attack is by logging all API behavior and keeping a backlog of this activity. This can be accomplished by having a data lake associated with your API security platform.

If you found this helpful, the next step is to explore the API Security solution from Akamai to ensure you have the most robust API security strategy possible.