Akamai

API Security Buyer's Guide

Rising to meet the API security challenge

As organizations become increasingly cloud-centric and digital, their APIs grow in scope and scale, increasing their value. APIs now:

- Operate at the heart of applications and services that serve your customers and partners, including the latest AI innovations
- Are embedded across cloud environments, from the services your developers use to the workloads your engineers lift and shift
- Represent revenue streams themselves, helping to grow your business and build a developer ecosystem

However, if you're like the 78% of IT and security professionals who have experienced API security incidents,[1] you've also seen firsthand that APIs are a growing risk. Exposed or misconfigured APIs are prevalent, unprotected, and easy to compromise. Many organizations often don't even know about all of their APIs, leaving them unmanaged. These dormant, or zombie, APIs are key attack vectors.

The stakes are high. Attacks on your APIs can jeopardize an enterprise's revenue, resilience, and regulatory compliance. Most organizations don't yet have the right controls and capabilities in place to prevent API attacks. Certainly, many companies have API tools in their existing stack — including API gateways and web application firewalls. But while these tools can offer some protection, they aren't designed to provide the degree of visibility, real-time security, and continuous testing to defend against modern API attacks.

1. Akamai Technologies, "API Security Disconnect Report," 2023

So what does it take to fully protect your API estate? Although a host of API security products have emerged over the past few years, navigating the growing scope of vendors and their capabilities can be difficult.

Today's threats call for a complete API security solution, encompassing four critical areas: API discovery, posture management, threat detection and remediation, and security testing. This buyer's guide describes the key capabilities a comprehensive API security solution requires, defining the features and security controls you need to develop and maintain secure APIs, while locating and protecting every API in your ecosystem.

Key capabilities for comprehensive API security

To determine the API security capabilities you need, it's important to understand the nature of the challenges you face.

APIs are often spread across multiple environments, from on-prem to hybrid cloud. Adding to the complexity, your API ecosystem likely extends far beyond your own network and cloud presence. Think about the myriad connections your APIs have forged with apps, services, and systems belonging to third parties, who may or may not prioritize API security.

Moreover, it's difficult to gain real-time insights into:

- Where your APIs are routed
- How they're configured
- What sensitive data they move
- What risks they pose

As enterprises rapidly develop and roll out new applications and APIs, the attack surface grows exponentially. As for older APIs, your organization may have a cluster of them — built and produced years ago, before API security surfaced as a critical need.

The lack of visibility leads to troubling findings: Only 4 in 10 security professionals with full API inventories know which of their APIs return sensitive data when called. Many of these API calls come from malicious actors testing for vulnerabilities — and once they spot a gap, the attacks are often relentless.

When you're vetting security vendors who say they can fully secure your API, it's important to make sure they have established, in-production controls and capabilities across four critical areas.

Read on, for a series of buyer's checklists you can use to vet vendors' capabilities.

01

API discovery

It’s not uncommon to have APIs that no one knows about. However, without an accurate inventory, your enterprise is exposed to a range of risks. To effectively inventory your APIs, you need to be able to:

☑ Locate and inventory your APIs regardless of configuration or type

☑ Detect dormant, legacy, and zombie APIs

☑ Identify forgotten, neglected, or otherwise unknown shadow domains

☑ Eliminate blind spots and uncover potential attack paths

02

API posture management

Simple API misconfigurations can open the door to attackers. Once inside, they can quickly access and exfiltrate sensitive data. To understand how all your APIs are configured, you need to be able to:

☑ Automatically scan infrastructure to uncover misconfigurations and hidden risks

☑ Create custom workflows to notify key stakeholders of vulnerabilities

☑ Identify which APIs and internal users are able to access sensitive data

☑ Assign severity rankings to detected issues to prioritize remediation

03

API threat detection and remediation

API attacks are reaching the point of inevitability. To effectively detect and remediate threats, you need to be able to:

☑ Monitor for data tampering and leakage, policy violations, suspicious behavior, and API abuse

☑ Analyze API traffic from all sources and integrate with existing workflows (ticketing, security information and event management, etc.) to alert security operations teams

☑ Prevent attacks and misuse in real time with partially or fully automated remediation

04

API security testing

Speed is essential for every application your developers build — but this makes it easier for a vulnerability or design flaw to go undetected. To properly test your APIs, you need to be able to:

☑ Run a wide range of automated tests that simulate malicious traffic and follow the underlying API business logic

☑ Discover vulnerabilities before APIs enter production, reducing the risk of successful attacks

☑ Inspect your API specifications against established governance policies and rules

☑ Run API-focused security tests that run on demand or as part of a CI/CD pipeline

API discovery: Deep dive into key capabilities

Many organizations operate both legacy and new APIs. It's not uncommon to have unmanaged APIs in production that no one on the operations or security teams knows about, exposing the business to a range of cybersecurity risks and operational difficulties. Rogue APIs can arise from factors such as shortcuts, process failures, and not being shut down when decommissioned. On the next page, we provide key examples to watch for.

Commercial APIs

Some commercial software packages include APIs to connect with other applications and external data sources. These APIs may get activated without anyone noticing.

Failure to deactivate

APIs can also be officially decommissioned but remain in operation due to operational oversights. These are sometimes called zombie APIs.

Old API versions

Sometimes an older version of an API never gets decommissioned. An old version may have to coexist with a new version for a period of time while the software is updated. But what if the person who's responsible for deactivating the API leaves the company, is reassigned, or simply forgets to shut down the old version?

Shortcuts and process failures

Some rogue APIs are the result of failing to inform the right people. For example, a line of business (LOB) team might create APIs to address specific needs without informing IT, or developers may be more concerned with execution than procedure. APIs that have been "inherited" as part of an acquisition are also frequently overlooked. These types of rogue APIs are often referred to as shadow APIs.

When you speak with vendors, ask them to explain how they ensure that rogue, legacy, zombie, and shadow APIs are identified and dealt with before they can be exploited. Legacy and zombie APIs are often the weakest link in API security. It's therefore critical to discover APIs that are not managed by an API gateway and locate them, inventory them, and determine whether they require remediation or decommissioning.

Key API discovery features

An API security solution should incorporate the following discovery features.

API asset discovery and granular inventory

An API discovery tool must be able to locate and identify the APIs you have, regardless of configuration or type — including RESTful, GraphQL, SOAP, XML-RPC, JSON-RPC, and gRPC. It should also create an inventory that is updated automatically to prevent it from getting stale, and provide the ability to search, tag, filter, assign, and export APIs based on any attribute.

Detection of dormant, legacy, and zombie APIs

Legacy and zombie APIs can predate your organization's API security initiatives. These APIs typically lack ownership and function without any visibility or security controls. It's critical that an API discovery tool is able to locate these APIs.

Shadow domain discovery

In addition to shadow APIs, you may have entire shadow domains — API domain names that you know nothing about. API discovery tools must be able to identify forgotten, neglected, or otherwise unknown shadow domains that could pose a security risk.

Automatic scans

Scanning is essential to eliminate blind spots and identify critical issues, including:

- Leaked API keys and credentials
- API code and schema exposure
- Infrastructure misconfigurations
- Vulnerabilities in documentation, GitHub repositories, Postman workspaces, etc.

Identifying these and other sources of exploitable intelligence can also help teams understand potential attack paths that could be exploited by cybercriminals.

Limited custom development

Finally, with the right API discovery tool, you shouldn't need custom development for traffic sources. These tools should come with pre-built integrations for major infrastructure components. Custom development is typically time-consuming, and if there are changes in the source origin, an integration would likely need to be reworked, which isn't scalable for stretched IT security teams.

API posture management: Deep dive into key capabilities

Threats to your API estate are growing rapidly due to trends such as the shift from centralized IT to decentralized LOB operations, the increased use of cloud resources, and the transition to microservices-based architectures.

Robust discovery (as described in the previous section) is the first step in securing your API estate. You need to discover and inventory APIs of all types that are currently in use.

There are several additional capabilities that are essential for managing your security posture across your APIs. You need to be able to identify which APIs access and transmit sensitive data and classify those APIs accordingly — because APIs that touch data such as customer information definitely need to be authenticated. It's also important to identify infrastructure vulnerabilities that will make any API more vulnerable.

Configuration assessment

Many cyberattacks succeed as the result of simple misconfiguration of the networks, API gateways, or firewalls that broker and protect API traffic.

An API security solution should scan infrastructure and software configurations regularly, including log files, replays of historical traffic, configuration files, and more. This allows you to uncover misconfigurations and vulnerabilities, and eliminate risk from configuration drift.

Customizable severity

As the solution identifies new vulnerabilities in your environment, it should also assign a severity level to the issues that were uncovered so they can be prioritized for remediation. Severity levels should be customizable to align with your organization's risk tolerance, regulatory requirements, and internal policies.

Custom workflows

Along with customizable severity, the ideal posture management tool should allow you to create custom workflows to take action immediately when you identify vulnerabilities. These workflows could range from creating tickets to notifying key stakeholders to updating network configurations.

Autogenerated documentation

API documentation tells consumers of an API what it does and how to use it. Organizations must evaluate secure APIs for compliance against specifications and accurate documentation. Poor or nonexistent documentation makes security testing more difficult, increasing the risk that an API reaches production with an undetected vulnerability. This problem is often exacerbated by outsourcing API development. Regardless of the source of the problem, out-of-date, incomplete, and missing documentation is unacceptable if you want your API security program to be successful.

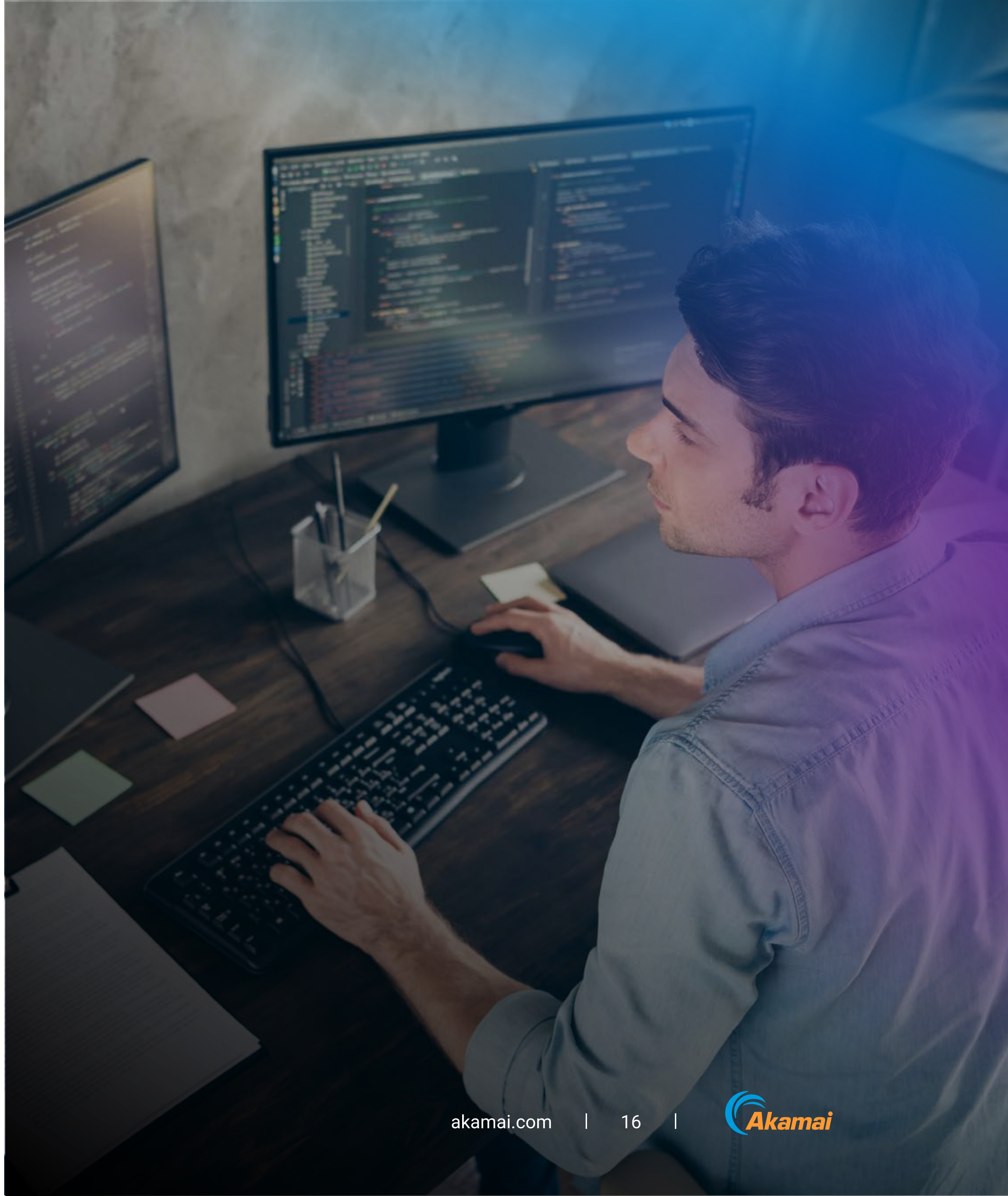
The OpenAPI specification defines standard interface descriptions. An API security solution should be able to:

- Compare API specs to actual observable traffic and identify differences, which enables organizations to see which of their deployed APIs are out of spec, and potentially a risk.
- Automatically generate complete OpenAPI documentation based on the API's current and future state to help ensure that all APIs are properly documented and that documentation is up to date. Identifying these and other sources of exploitable intelligence can also help teams understand potential attack paths that could be exploited by cybercriminals.

API threat detection and remediation: Deep dive into key capabilities

Attacks that attempt to exploit API vulnerabilities are now a fact of life. It’s no longer a question of if your organization will be attacked, it’s a question of when and how. It’s become imperative to detect attacks quickly and block them before they can do significant damage — like exfiltrating private customer data. Even if your APIs are as secure as you can make them, you need active runtime protection to detect data leakage, data tampering, data policy violations, suspicious behavior, and API security attacks. This should include logging API traffic, monitoring sensitive data access, detecting threats, and blocking or remediating attack vectors.

On the following two pages, we explain the key features that an API security solution should include.



Out-of-band monitoring in real time

API security monitoring shouldn't impact or slow down API traffic. Look for vendors that can provide an agentless approach that allows companies to deploy faster and see more traffic. However, in circumstances where appropriate (e.g., complex on-prem environments), the solution should have enough flexibility to support agents as well.

An API security solution should mirror traffic from identified data sources and perform analysis on that traffic data in the background, with real-time alerting of any issues discovered.

API anomaly and exploitation detection

Passive data collection is not enough, especially as the number of APIs and the total volume of API traffic continues to scale. API activity must be analyzed continuously to detect anomalous events and alert security and operations teams.

Advanced tools incorporate AI and machine learning capabilities to analyze traffic in real time and leverage contextual insights to identify anomalous activity that can indicate data leakage, data tampering, data policy violations, and other API security attacks.

API attack prevention

Once an anomaly or other problem has been identified and an alert generated, time is of the essence. Unauthorized movement of sensitive data via APIs or other suspected misuse of APIs must be detected and blocked. An API security solution should not only block misuse through integration with your existing firewalls and API gateways — it should partially or fully automate remediation. Semiautomated remediation should be available to address some types of alerts. For previously identified, recurring issues, you should have the option to provide a fully automated response.

Scoring for attack confidence

Some solutions in the market use machine learning algorithms trained to evaluate external and internal signals, including API behavior, network traffic patterns, geolocation data, and threat intelligence feeds. Using contextual factors like these, a solution can determine the confidence level that a detected runtime incident is the result of malicious activity.

Integrations for incident response

When an incident occurs, an API security solution must include the necessary integrations to ensure remediation tasks are assigned to appropriate teams. If misconfigurations, data policy violations, or suspicious behaviors are detected, they should be reported to the API gateway, SIEM system, and other information security engines to ensure the right level of awareness.

As a general rule, an API security solution should integrate easily with the other security, monitoring, and management tools your organization uses.

API security testing: Deep dive into key capabilities

A mistake that many development teams make is waiting too long to begin API testing, allowing testing to become a bottleneck. Teams need to take a shift-left approach to ensure that testing begins early enough in the development process to ensure it is comprehensive. The benefits of effective API security testing are significant:

- **Prevent attacks**
  - By discovering vulnerabilities before APIs enter production, you reduce the risk of successful attacks
- **Improve compliance**
  - Comprehensive testing will help you ensure compliance, and avoid fines and reputational damage
- **Boost confidence**
  - Rigorous and effective testing can increase your organization's confidence in APIs — and help ensure your developers' releases happen on time

Some vendors in the market can offer recommendations to enterprises on how to remediate issues in their environments, as well as how to enable comprehensive API testing configurations. Recommendations could include action steps to configure proper authentications or fix API dependencies. The benefit: If you can address business logic issues within your environment, you can increase the number of APIs optimized for testing, resulting in greater testing coverage.

The whole concept of API security testing, however, remains somewhat nebulous. Development teams may not fully understand what it entails. Shift-left API testing is a three-step process:

1. **Understand the API:** Understanding the use case for the API informs testing, especially for tricky business logic issues.

2. **Ensure you can interact with the API correctly:** Make sure you can use the API as it was intended. This is essential to validate that your understanding of the API matches how the API works.

3. **Send attack traffic to the API:** This could include manually manipulating requests to the API, inserting fuzzing strings into requests, or using an automated tool to perform API security testing. As with everything in modern IT, automation is often the best way to get the job done at scale without sacrificing velocity.

Key API security testing features

API security testing should include static, dynamic, and pen testing. An API security solution should include tools to facilitate thorough testing, automating testing processes to the greatest extent possible. Look for the following API testing features in an API security solution:

Proactive automated API security testing

Automated security testing significantly reduces risk and cost by identifying misconfigurations, vulnerabilities, and noncompliance — before an API enters production.

API governance

It is essential to think through governance issues like roles, responsibilities, and policies. This includes execution-level responsibilities for developers, security engineers, and platform engineers, as well as policy oversight and decisions about risk. An API security solution should allow you to inspect your API specifications against established governance policies and rules.

CI/CD pipeline and code repository integration

DevSecOps is a variant of DevOps that adds security to the software development workflow. API security needs to be part of DevSecOps initiatives. An API security solution should provide a suite of API-focused security tests that run on demand or as part of a CI/CD pipeline. CI/CD integration is essential because it enables the continuous, rapid API security testing needed to keep up with application development.

Bringing it all together: Identify and address your API security gaps

APIs are an essential component of organizations' ability to serve customers, generate revenue, and operate efficiently in an increasingly digital and cloud-centric economy. However, their continuous growth, proximity to sensitive data, and lack of security controls make APIs an appealing target for today's attackers.

The existing tools that many organizations use to manage APIs and gain baseline protection do provide a degree of risk reduction. But it's not nearly enough to take on today's API threats. They can't be relied on as sole sources of protection.

Instead, organizations should seek a comprehensive API security solution that can provide all four of the components discussed in this buyer's guide: discovery, posture management, threat detection and remediation, and security testing. You do not have to jettison existing tools that have proven effective in certain areas — just look for a solution that can seamlessly integrate with your existing tools.

Getting started on API security does not mean you need to allocate significant resources. You can begin by committing to a smaller-scale, measurable pilot that addresses specific gaps in your security stack. Or you can start your API security journey through a comprehensive update. Every organization is different.

With API-focused attacks on the rise, your most important step is the decision to take action. We hope you found this buyer's guide helpful.

Read more about API attack methods, common API vulnerabilities, and how to secure your organization.

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

Akamai