Akamai

# Definitive Guide to API Runtime Protection

# Table of contents

# Introduction

## Why API security is imperative

In the race to meet customers' needs, organizations face pressure to quickly develop and produce and enhance applications, services, and GenAI tools. This need for speed unfortunately results in a hidden risk: The APIs working behind the scenes for all these innovations are often built with misconfigurations, coding errors, and missing security controls. And when these APIs reach the production stage, it's not just end users interacting with them; attackers are constantly testing out ways to compromise the APIs and access the data they exchange.

Misconfigured and compromised APIs are increasingly a key driver of significant data breaches, and yet few organizations are able to keep tabs on the thousands of API calls within their digital ecosystems. Fewer still are fully protected against runtime API threats.

For example, in 2021, a fitness retail company found a bug in an API for user account data allowing anyone to make unauthenticated requests for data including age, gender, city, weight, and birthdate. While this vulnerability was thankfully detected and reported to the company by a security researcher, bugs like this can go unnoticed and be exploited for weeks or months.

When it comes to securing APIs, the traditional tools that organizations typically rely on — for example, API gateways and web application firewalls — can provide a baseline of protection. However, today's security teams require additional security layers, as API attacks grow in number and sophistication. The key is augmenting existing controls with deeper insights into vulnerabilities, potential attack paths, malicious activity, and API behavior.

Organizations can attain these capabilities through a comprehensive API security solution, encompassing four areas:

1. API discovery
2. API posture management
3. API runtime protection
4. API security testing

## What this guide covers

API runtime protection is the process of securing APIs as they operate and manage requests during their normal functioning. This guide discusses the key requirements for API runtime protection — including API monitoring in defense of misconfiguration and exploitation, and API attack prevention. It explores runtime prevention basics and introduces the runtime prevention capabilities offered by Akamai API Security.

# Why runtime protection?

API runtime protection secures APIs throughout the production stage of their lifecycle, when the API is operational and available for interaction with your intended end users — and with attackers. With capabilities that help organizations quickly identify and address malicious API requests, effective runtime protection capabilities can secure APIs against a range of post-deployment threats, including:

- An attacker pulling large volumes of sensitive data from an API
- Privilege escalation attacks that exploit security bugs
- Deployment of unauthorized APIs outside normal processes

Blocking runtime API threats requires an understanding of the context of operations for each individual API, including API access, usage, and behavior. To begin, you need to know the scope of your API estate. Our Definitive Guide to API Discovery explains the importance of an API inventory. With a complete API inventory, you can monitor all API traffic and build a baseline understanding of "typical" behavior for each API that can be used to recognize anomalous behavior. API runtime protection should detect:

- Data leakage
- Data policy violations
- API security attacks
- Data tampering
- Suspicious behavior

In addition, runtime protection should log API traffic, monitor sensitive data access, detect threats, and block or remediate attacks.

## Monitoring API traffic for attacks

Observing API traffic behavior is essential to identify risks. Deploying a monitoring solution without an accurate picture of your API estate only provides limited visibility. After your API footprint is inventoried, API runtime protection should continually monitor traffic and API consumption, and look for vulnerabilities and misconfigurations.

## Detecting anomalous behavior

Having a baseline of normal API behavior makes it possible to identify anything out of the ordinary. Replaying historical data can help identify anomalous behavior, which may also reveal an attacker's intent.

Any potential anomalies should be examined further in the context of other actions taking place within the application or network. For instance, if data requests are generally of a certain size, and an API call requests data outside the range of the usual requests, it should be flagged. It may or may not be malicious, but the anomaly requires further inspection.

## Detecting data exposure

Some of the APIs in your estate likely send and receive sensitive data. Sensitive information that is exposed due to a security vulnerability allows an attacker to escalate privilege or other improper access control configurations. AI and machine learning can be instrumental in real-time traffic analysis and anomaly detection, providing contextual insights into data leakage, data tampering, data policy violations, suspicious behavior, and API security attacks.

One type of attack that's become increasingly common is for cybercriminals to get their hands on valid API keys. Once an attacker has valid keys in hand, just about the only way to protect against improper API use and potential data breach is the ability to detect and block anomalous behavior and data exposure.

## API security auditing

API security auditing tools should monitor traffic in real time and alert you of attacks and other malicious intent. At a minimum, API security auditing should:

- Perform continuous monitoring to identify attackers and malicious requests
- Passively scan APIs, internally and externally, for misconfigurations and oversights that could enable or worsen a breach or weaken defenses
- Enforce policies on what data should (and shouldn't) be sent or received by APIs

API runtime protection should also be complemented by API posture management, which identifies misconfigurations and known vulnerabilities. Check out our **Definitive Guide to API Posture Management** for more insights.

# Runtime protection features you can't live without

If your organization is actively developing and deploying APIs, robust runtime protection needs to be a part of your API security program. Here are key features that your runtime protection tools must include.

## Out-of-band monitoring in real time

API security monitoring shouldn't impact, slow down, or add latency to API traffic. It should run completely out of band with no required network changes and no cumbersome, difficult-to-install agents. Runtime protection tools 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.

Akamai runs out of band and agentless by default, but we provide the options for agent-based detection and inline blocking if needed.

## 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. State-of-the-art platform tools incorporate AI and machine learning capabilities to analyze traffic in real time and leverage contextual insights into data leakage, data tampering, data policy violations, suspicious behavior, and API security attacks.

## API attack prevention and risk remediation

Once an anomaly or other problem has been identified and an alert generated, time is of the essence. Unauthorized movement of sensitive data via API or other suspected misuse of APIs must be detected and remediated. Runtime protection should not only prevent API misuse through integration with your existing firewalls and API gateways, it should provide remediation options — automated when possible. Look for capabilities that include attacker confidence scores that help your team determine if signals of abuse, attacks, or breaches are legitimate and in need of escalating.

## Integrations for incident response

As a general rule, runtime protection tools should integrate easily with the other security, monitoring, and management tools your organization uses. For example, when an incident occurs, runtime protection tools 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. Having an attacker confidence scoring capability can allow teams to filter out the noise and focus their attention on true API security priorities.

## Rapyd

Rapyd, a global payment processing and fintech company, operates payment systems in more than 100 countries. Lacking granular visibility into API usage and behavior, the company needed a better way to secure public-facing APIs — and hundreds of internal APIs — in a highly complex, global system operating from the AWS cloud. Rapyd needed a granular inventory of all of its APIs, visibility into misconfigurations and vulnerabilities, and intelligently prioritized alerts for a more logical remediation approach.

Akamai API Security met Rapyd's needs with comprehensive visibility and runtime protection that uses machine learning to create a baseline of the traffic for every API, with automated anomaly detection and remediation.

Read full customer story

> “Now we can assess our risk in the most scientifically true way possible and control our destiny.
>
> – Nir Rothenberg
> CISO, Rapyd

# Runtime protection of Akamai API Security

The ability to identify and thwart API attacks as they are happening should be an integral part of your compliance and risk assessment program. You can think of it as your last line of defense if other security controls fall short.

Akamai API Security's runtime protection module includes all of the features described in the previous section. Its primary function is to detect and block API attacks in real time. Automated machine learning–based monitoring is used to conduct traffic analysis and provide contextual insights into data leakage, data tampering, data policy violations, suspicious behavior, and API security attacks. Runtime protection detects anomalies and potential threats in your API traffic, and facilitates remediation based on preselected incident response policies.

The runtime protection integrates with WAFs, API gateways, ITSMS, SIEMs, and other workflow tools to deliver a holistic defense against attacks. You can choose to fully automate threat remediation or require different levels of manual intervention for greater visibility and control. The Akamai API Security solution also has native integration with the Akamai platform that allows us to block attacker IPs directly at the edge.

## Issue generation

Using machine learning, Akamai builds a model for each API. This baseline of normal behavior is then used to detect API business logic attacks such as Broken Object Level Authorization (BOLAs), where an individual gains access to data they should not have access to. Akamai will generate an issue in real time whenever

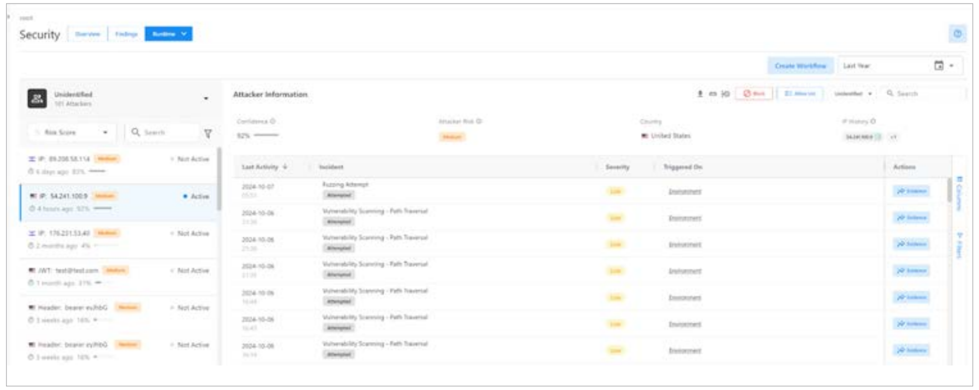
API traffic deviates from normal behavior. An issue is much like an alert and is generated whenever anomalous API behavior is detected or when a misconfiguration is found. As issues are generated, alerts can be sent automatically to a SIEM such as Splunk or QRadar. Alerts can also be sent automatically to a ticketing system such as ServiceNow or Jira.

## Issue details

Every issue generated by Akamai API Security's runtime protection module includes severity, status, a mapping to the OWASP API Top 10 — and attacker details where applicable.

Issue detail pages include a description of the issue and its potential impact to your organization, and provide remediation recommendations. Akamai API Security also allows organizations to see what types of actions attackers took over a specific period of time, with a historical record of each attack, and the ability to take action against malicious actors.

### Example: Visibility into attackers' actions



IP: 89.208.58.114 Medium Not Active
6 days ago 83%

IP: 54.241.100.9 Medium Active
4 hours ago 92%

IP: 176.231.53.40 Medium Not Active
2 months ago 4%

JWT: test@test.com Medium Not Active
1 month ago 31%

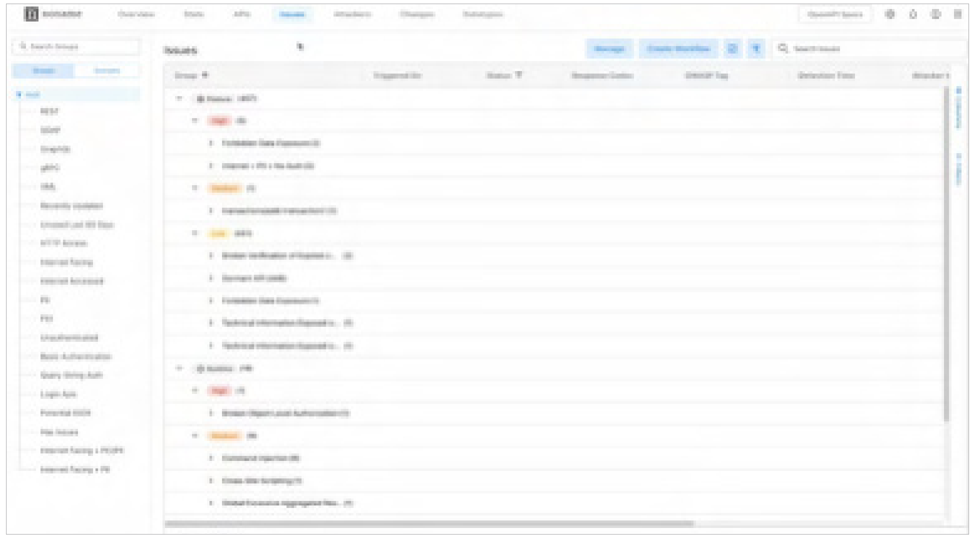
Header: bearer eyJhbG Medium Not Active
3 weeks ago 16%

Header: bearer eyJhbG Medium Not Active
3 weeks ago 16%

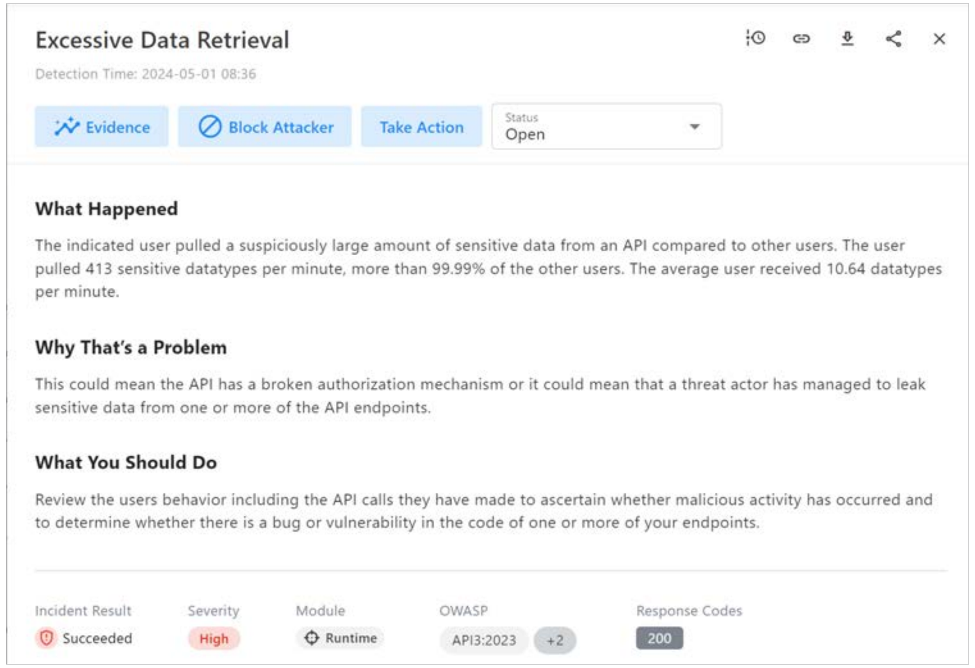
| Last Activity | Incident | Severity | Triggered On | Actions |
|---|---|---|---|---|
| 2024-10-07 | Fuzzing Attempt<br>Attempted | Low | Environment | Evidence |
| 2024-10-06 | Vulnerability Scanning - Path Traversal<br>Attempted | Low | Environment | Evidence |
| 2024-10-06 | Vulnerability Scanning - Path Traversal<br>Attempted | Low | Environment | Evidence |
| 2024-10-06 | Vulnerability Scanning - Path Traversal<br>Attempted | Low | Environment | Evidence |
| 2024-10-06 | Vulnerability Scanning - Path Traversal<br>Attempted | Low | Environment | Evidence |
| 2024-10-06 | Vulnerability Scanning - Path Traversal<br>Attempted | Low | Environment | Evidence |
| 2024-10-06 | Vulnerability Scanning - Path Traversal<br>Attempted | Low | Environment | Evidence |

Every issue includes evidence. Evidence is the attacker session details that led up to the issue being generated, and a copy of the API request and response — both the headers and the body — to aid in triaging and remediating the issue quickly. With intuitive dashboards, filtering functions, alerts, and reporting capabilities, the Akamai API Security solution's runtime protection module can help organizations determine what happened, why it happened, and what exactly needs to be done.

## Example: Reporting on API issues with evidence



## Example: Insights on excessive data retrieval



Excessive Data Retrieval

Detection Time: 2024-05-01 08:36

Status: Open

**What Happened**

The indicated user pulled a suspiciously large amount of sensitive data from an API compared to other users. The user pulled 413 sensitive datatypes per minute, more than 99.99% of the other users. The average user received 10.64 datatypes per minute.

**Why That's a Problem**

This could mean the API has a broken authorization mechanism or it could mean that a threat actor has managed to leak sensitive data from one or more of the API endpoints.

**What You Should Do**

Review the users behavior including the API calls they have made to ascertain whether malicious activity has occurred and to determine whether there is a bug or vulnerability in the code of one or more of your endpoints.

| Incident Result | Severity | Module | OWASP | Response Codes |
|---|---|---|---|---|
| Succeeded | High | Runtime | API3:2023 +2 | 200 |

## Policy actions

Akamai API Security provides the ability to take a semiautomated policy action for every issue generated. Actions may include opening a ticket, sending information to a SIEM, or sending a webhook to a third-party system. They may also include blocking an attacker. The types of actions available are determined by the types of integrations configured into the Akamai platform.

The solution includes numerous predefined policies out of the box for detecting API attacks and API misconfigurations. Akamai API Security also includes 20+ preconfigured data types to help you create the data policies you need to detect and take action when sensitive data types are traversing your APIs.

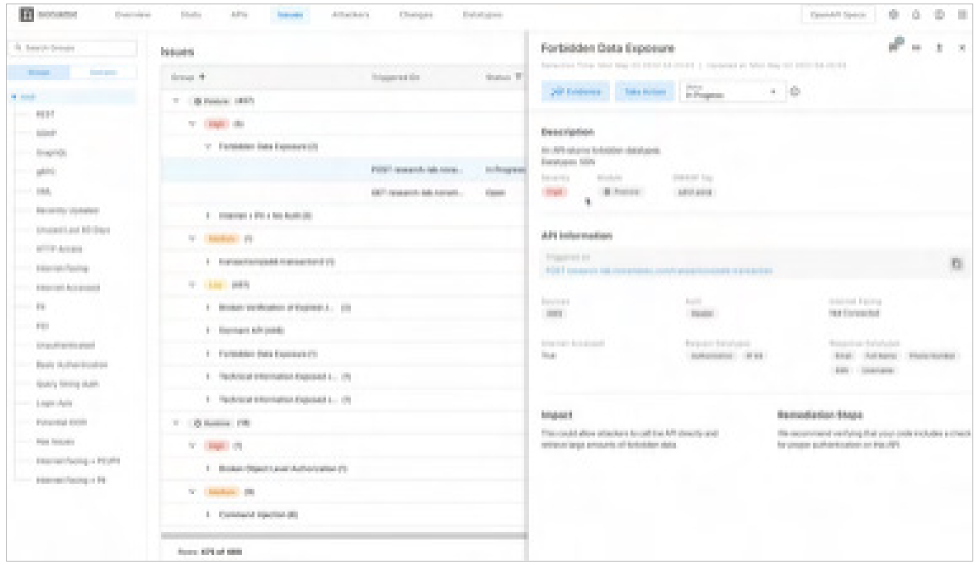
In summary, the Akamai API Security solution's runtime protection module includes real-time detection and prevention of API attacks along with continuous detection of API misconfigurations while including many popular workflow integrations to simplify operations and remediation.

## Anatomy of an API security incident

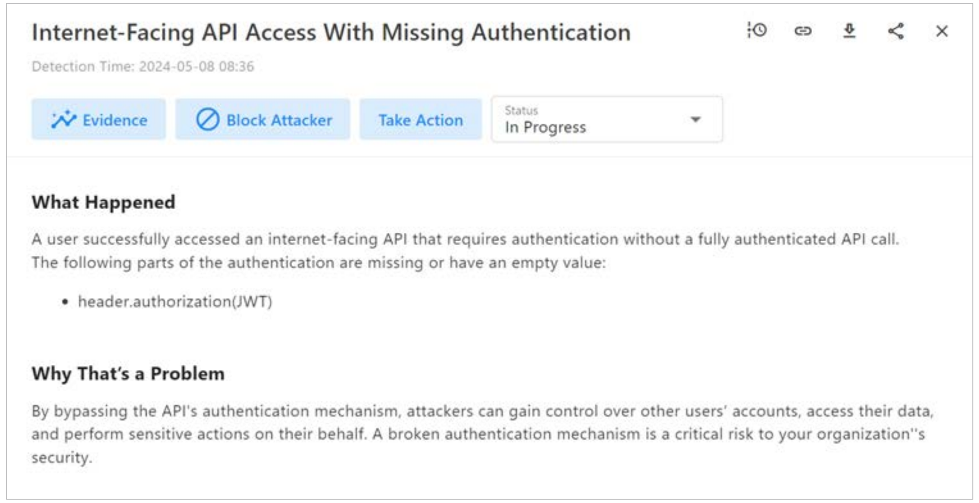
Let’s look closer at an example of forbidden data exposure. This example illustrates a posture issue internal to an API. The Akamai platform is contextually aware of the data types and values associated with every API.

In the figure below, forbidden data is being exposed by an API. The Akamai platform detected the type of data being transmitted, in this instance a Social Security number (SSN), and understood that the SSN data type had previously been tagged as forbidden. Akamai can also detect misconfigurations external to the API such as APIs that are internet accessible but are not registered with an API gateway.

## Example: Insights on forbidden data exposure



## Example: Identifying APIs with missing authentication



Internet-Facing API Access With Missing Authentication

Detection Time: 2024-05-08 08:36

Status
In Progress

**What Happened**

A user successfully accessed an internet-facing API that requires authentication without a fully authenticated API call. The following parts of the authentication are missing or have an empty value:

- header.authorization(JWT)

**Why That’s a Problem**

By bypassing the API's authentication mechanism, attackers can gain control over other users' accounts, access their data, and perform sensitive actions on their behalf. A broken authentication mechanism is a critical risk to your organization''s security.

# Next steps for attaining effective API runtime protection

Every time a customer, partner, or vendor engages with your organization digitally, there's an API behind the scenes facilitating a rapid exchange of (often sensitive) data. Implementing key API runtime protection capabilities — for example, API monitoring to defend against misconfiguration and exploitation, and API attack prevention — can help you protect your organization against a fast-rising attack vector.

Learn how to evaluate API security vendors to ensure they offer critical runtime protection capabilities.

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

Akamai