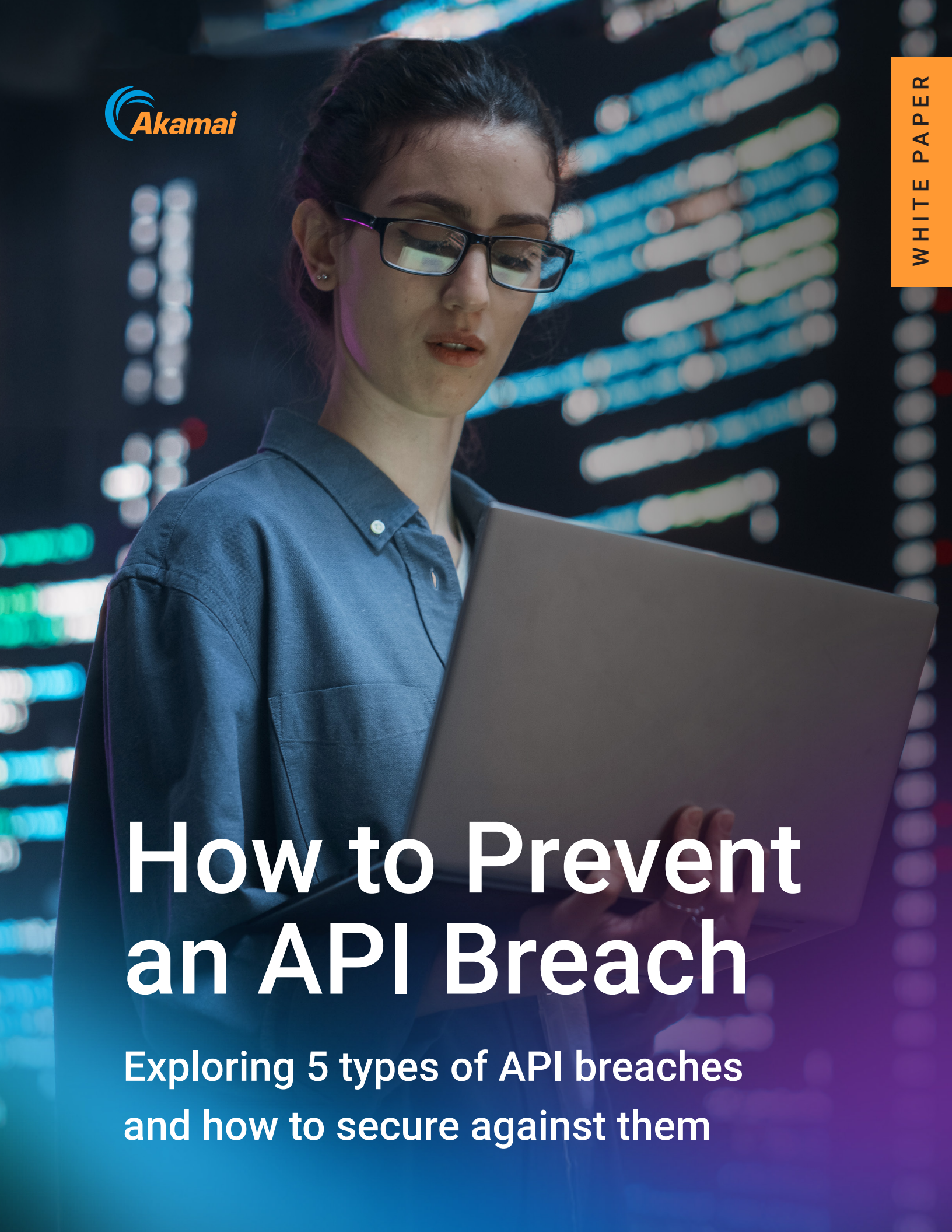

Akamai

WHITE PAPER

How to Prevent an API Breach

Exploring 5 types of API breaches and how to secure against them

In this report

Introduction

APIs connect your business by exchanging data with partners, suppliers, and customers. And yet, API security remains less than comprehensive in most organizations. In fact, vulnerable APIs have become a targeted weakness for companies in recent years, with attackers abusing them to access sensitive data, sell it to other threat actors, or publish it for the world to see. In 2024, global brands across consumer telecom, enterprise computing, and virtual collaboration saw API breaches release huge amounts of customer and other sensitive data — inflicting hefty financial and reputational costs.

What is an API breach?

Simply put, an API breach is any intentional misuse or abuse of an API, often to gain access to sensitive data. Types of API breaches can be subdivided according to various criteria. To identify risks and avoid breaches in production operations, it's useful to consider the following scheme, which breaks down risks into five categories:

1. **Known vulnerabilities**
   - Attackers exploit known vulnerabilities that haven't been patched.
2. **Shadow, rogue, zombie, and deprecated APIs**
   - Unmanaged and forgotten APIs can leave operations vulnerable.
3. **External exposures**
   - Credentials, keys, and other exposures may exist outside your control.
4. **Misconfigurations and operator errors**
   - Security misconfigurations in infrastructure and services can create entry points for exploitation by threat actors.
5. **Undiscovered vulnerabilities and bugs**
   - Threat actors seek to identify bugs and vulnerabilities that made it into the production environment despite your best efforts.

This ebook explains where the security failures occur in each of these five types of API breaches and how to prevent them. This ebook also aims to help you zero in on specific weaknesses in your API security program to maximize API security and minimize risk.

Breach type: Known vulnerabilities

API breaches that take advantage of known vulnerabilities (that haven't been patched) are perhaps the most common. If cybercriminals want to get your data, a common first step is for them to check whether your organization has left any back doors open.

In January 2024, an attacker compromised a widely used project management tool by exploiting an API endpoint lacking authentication controls. After breaching the API, the threat actor gained unauthorized access to information on millions of users and months later leaked over 21 GB of data — including email addresses and board memberships — on the internet.

Authentication and authorization issues are among the most common API problems. The OWASP Top 10 API Security Risks provides education on the 10 most critical API vulnerabilities that organizations must protect against, including broken authentication.

In addition to securing APIs from the types of risk included in the OWASP Top 10, organizations should protect API code against the full list of Common Vulnerabilities and Exposures (CVEs) created by the U.S. National Cybersecurity Federally Funded Research and Development Center (FFRDC), operated by MITRE. You might recall the well-publicized Apache Log4j 2 vulnerability (CVE-2021-44228), also referred to as "Log4Shell." Because of a bug in the Log4j library — a popular open source logging library for the Java programming language — attackers could remotely execute arbitrary code to gain system access. Malicious actors routinely probe enterprise systems for known vulnerabilities like this one.

In the United States, the Cybersecurity and Infrastructure Security Agency (CISA) maintains a catalog of known CVEs. Other countries may maintain similar catalogs.

The OWASP Top 10 API Security Risks list was built in 2019 and updated in 2023. While useful, it can't keep up with the speed of change in the attack surface. In 2024 alone, more than 24,000 new CVEs have been added to CISA's catalog, more than 500 of which are API related (as of mid-August 2024).

Fully protecting your organization from known vulnerabilities requires a two-pronged effort:

1. Make sure your development and testing processes are robust enough to avoid introducing known vulnerabilities into production.

2. Patch new vulnerabilities as quickly as possible after they are identified.

Many organizations struggle with both of these steps. On top of this, they use APIs and code from third-party sources that can introduce a separate set of vulnerabilities. In 2022, a team of researchers discovered critical API flaws that affected several manufacturers across the automotive industry. These flaws could have exposed sensitive customer data and even a vehicle's location, allowing a car to be unlocked, started, or disabled via a compromised remote management system.

How to prevent them

One well-known way to protect your organization against API breaches due to known vulnerabilities is to quickly update software and systems when security patches are released. It's also essential to ensure that your development and testing processes are comprehensive and rooted in API security best practices. This includes:

- **Securing your software supply chain:** Ensure that any libraries, open source software (OSS), and other third-party code you use are secure.
- **Implementing shift-left security testing:** Move tasks related to API security and software testing earlier in the development process. This can help you uncover vulnerabilities such as coding errors and misconfigurations made by developer teams under pressure to rapidly release software or updates.
- **Leveraging API security posture management:** This combines API discovery with sensitive data identification and vulnerability detection, ensuring that remediation efforts focus on the most critical APIs first.

How Akamai API Security helps you

Akamai API Security enables your teams to reduce known vulnerabilities for every new build, without sacrificing speed. API Security is a purpose-built API security testing solution that provides comprehensive coverage of API-specific vulnerabilities. Active testing helps bake API security testing into every phase of development.

- **Find and test every API** based on an understanding of the application's business logic.
- **Shift left** with integrations into the entire software development lifecycle. Teams get dynamic API visibility across multiple states and environments throughout the CI/CD process.
- **Empower developers** with best-in-class usability, including simple setup and automation, in-line test results, and contextual guidance for fixing identified issues.

In addition, API Security's posture management provides a comprehensive view of traffic, code, and configurations to assess your API security posture. API Security looks at the widest possible set of sources to detect vulnerabilities, including log files, replays of historical traffic, configuration files, and much more. It also detects all vulnerabilities in the OWASP Top 10 API Security Risks (for more on posture management, see the "Misconfigurations and operator errors" section).

Breach type: Shadow, rogue, zombie, and deprecated APIs

You can't protect what you can't see, and in many companies, a large percentage of APIs are unmanaged, making shadow, rogue, zombie, and deprecated APIs (see sidebar on next page) targets that are unseen or unaccounted for in your API estate. In addition, attackers often hunt for API variants they can exploit by looking at an organization's exposed APIs and then fuzzing or modifying values to find old versions.

This is what happened to a large Australian telecom company that accidentally exposed more than 11.2 million customer records, including names, addresses, birth dates, and some government-issued ID numbers. The attack took advantage of an API used for testing that had somehow become accessible to the open internet. Because this rogue API lacked authentication checks, an attacker was able to request and receive millions of records.

Most organizations operate using a variety of legacy and new APIs. It's unfortunately all too common to find alongside them rogue, zombie, and shadow APIs that expose the business to a range of cybersecurity risks and operational difficulties.

These unseen APIs have a variety of sources:

- **Commercial APIs:** Some commercial software packages include APIs to connect with other applications and external data sources. These may be activated without anyone noticing (a problem that can be addressed by thorough API discovery).
- **Old API versions:** In many cases, an older version of an API — possibly with weaker security or a known vulnerability — may never be removed. An old version may need to coexist with a new version for some time while software is updated, but when process failures prevent the old API from being shut down, it turns into a zombie API.
- **Shortcuts and process failures:** Shadow APIs result from failing to inform the right people. For example, a line of business team may create APIs to address specific needs without informing the IT or security teams, or a developer may not follow procedure.
- **Inherited APIs:** APIs that have been inherited as part of mergers or acquisitions are also frequently overlooked and become shadow APIs.
- **Reactivated code:** In some cases, old versions of APIs can be accidentally reactivated.

How to prevent them

A manual API audit to document all inputs that must be accurately inventoried can take several hours, especially considering the time it takes to assess and act on every API you find. This is not a realistic task for already overworked security teams. To protect your business against exploitations of rogue, zombie, and shadow APIs, you need automated API discovery capable of identifying all APIs in use — of every type. It's critical to locate and inventory every API across your operations and discover APIs and API domains that are not managed by an API gateway.

How Akamai API Security helps you

API Security leverages a broad collection of integration sources to ingest API data, such as raw traffic, logging, and much more. Data derived from these sources enables API Security to identify APIs, their misconfigurations, vulnerabilities, and API abuse. Our discovery tools detect all vulnerabilities in the OWASP Top 10 API Security Risks.

Additional discovery capabilities allow you to:

- Locate and inventory all of your APIs, regardless of configuration or type, including RESTful, GraphQL, SOAP, XML-RPC, JSON-RPC, and gRPC
- Uncover dormant, legacy, and zombie APIs
- Identify forgotten, neglected, or otherwise unknown shadow domains
- Maintain API inventories and ensure API documentation accuracy

High-risk unmanaged APIs that attackers seek

Shadow APIs (aka "undocumented APIs") exist and operate outside the official monitored channels of an organization. They may be created by well-meaning developers to accelerate their work, or they could be a remnant from previous software versions.

Rogue APIs are unauthorized or malicious APIs that pose a security risk to a system or network.

Zombie APIs include any API that is left running even after being replaced by new versions or other APIs entirely.

Deprecated APIs are APIs that are no longer recommended for use because of changes in the APIs. While deprecated classes, methods, and fields are still implemented, they may be removed in future implementations, so you should not use them in new code.

Breach type: External exposures

Outside API vulnerabilities are typically the result of poor practices or procedural errors, such as API key and credential leakage, API code and schema exposure, loose documentation, and repo vulnerabilities. The ability to discover potential attack vectors outside the boundaries of your operations has become imperative. In the past year, a number of high-profile breaches have resulted from the accidental exposure of API keys or other credentials from outside sources. For example, hackers used a phishing campaign to gain unauthorized access to 130 of Dropbox's source code repositories. This allowed them to access API keys improperly stored on GitHub. This type of exposure has become so common that GitHub has taken steps to block leaks of API keys and other secrets from occurring, but other public repositories may still be vulnerable.

In another well-publicized example of external exposure, researchers uncovered more than 3,000 mobile apps that exposed Twitter API keys to the public. This type of mistake is surprisingly common because developers often embed API keys in application code during development for convenience. If they fail to remove those embedded keys before a public release, that becomes a potential source of key exposure.

How to prevent them

Reducing or eliminating these types of external exposures requires a two-pronged attack:

- Tighten up procedures to identify and eliminate sources of exposure like leaked keys and credentials, improper use of repositories, etc.
- Regularly scan the external attack surface to detect and remediate vulnerabilities.

To protect yourself against the widest range of API threats, you need both inside-out discovery (as described in the "Breaches from rogue APIs" section) and outside-in discovery, which can identify exposures and reduce your external attack surface.

How Akamai API Security helps you

API Security helps you stay ahead of attackers by simulating the reconnaissance techniques that hackers use and enabling you to rapidly find and fix issues. With outside-in discovery, API Security automatically scans your external attack surface at regular intervals to find vulnerabilities before attackers do, enabling you to:

- **Find public vulnerabilities:** Rapidly find and fix critical issues like API key and credential leakages, code exposure, misconfigurations, repo vulnerabilities, and more.
- **Discover domains and subdomains related to your company:** Leverage data gathered from various sources, including internet registrars, certificate registrars, and open sources.
- **Incorporate real attack methods:** Simulate an attacker performing outside reconnaissance to collect information by executing limited queries to company domains or subdomains.

Breach type: Misconfigurations and operator errors

Many cyberattackers gain entry by exploiting misconfiguration of the servers, networks, API gateways, and firewalls that broker and protect API traffic. A study from IBM Security X-Force found that two-thirds of cloud breaches are tied to misconfigured APIs. Security misconfigurations can be caused by insecure default configurations, cloud storage without access control (surprisingly common), and incomplete or ad hoc configurations. As your digital footprint expands, your operations may expand to more locations, including multiple public cloud availability zones or public clouds such as AWS, Microsoft Azure, and Google Cloud. These environments often operate under different security controls, making it complex and difficult to ensure that security is correctly configured everywhere.

How to prevent them

One of the best ways to protect against security misconfigurations on the infrastructure side is to avoid manual configuration of servers, network devices, gateways, and firewalls as much as possible. If your company's admin teams routinely configure infrastructure and application security controls manually — or "tweak" them regularly — the chance of introducing configuration vulnerabilities increases.

Automation is your best friend when it comes to security. Some companies are embracing the idea of immutable infrastructure as a way to avoid manual mistakes.

Even if you've done everything you can to ensure that your infrastructure, services, and APIs are bulletproof, you still need API posture management. Posture management gives you the tools to manage, monitor, and maintain the security of your APIs throughout the API lifecycle.

How API Security helps you

API Security's posture management module analyzes API calls and infrastructure to identify misconfigurations. These misconfigurations are typically Amazon S3 bucket problems, sensitive data on unauthenticated APIs, and different Kubernetes access-based misconfigurations.

The posture management module provides a comprehensive view of traffic, code, and configurations, offering a view of the entire attack surface across APIs and web applications, including all forms of sensitive data moving through your APIs, such as personally identifiable information. It also helps you confirm that your API management tool is using strong protocols and ciphers to avoid weak encryption that could expose this sensitive data. Additionally, APIs should not accept expired JSON Web Tokens, as doing so would allow unauthorized access and increase security risks. The module also helps prevent misconfigurations, such as application load balancers listening on insecure ports without redirection. All these measures collectively strengthen the security posture of APIs, ensuring a more resilient defense against potential threats.

Breach type: Undiscovered vulnerabilities

As with most breach types, cybercriminals scanning your infrastructure routinely look for CVEs, the OWASP API Security Top 10, and other common misconfigurations, as well as rogue, zombie, and shadow APIs. They also probe your exposed APIs for new vulnerabilities they can exploit in libraries, open source code, and other types of public code, as well as in coding errors, bugs, and misconfigurations in your API estate. These vulnerabilities allow cybercriminals to manipulate API calls and insert fuzzing strings into requests. As a result, the techniques cybercriminals use are constantly evolving.

How to prevent them

An important part of prevention is ensuring that your code is as free from bugs and vulnerabilities as possible (see the "Known vulnerabilities" section). However, you should still assume that threat actors will find bugs or gain access to keys or credentials that allow them to exploit APIs.

API runtime protection is designed to identify hackers exploiting any vulnerability — known or unknown. It's the only way to protect your API estate against previously unidentified bugs and misconfigurations that slip into production, and it's the best protection against credentials and keys that have been compromised.

Runtime protection identifies unusual patterns and anomalies in API use and data access so that ongoing attacks that might slip under the radar can be identified and remediated before thousands or millions of data records are extracted.

API runtime protection helps you identify and block malicious API requests, including:

- Attacks pulling large volumes of sensitive data from an API
- Broken Object Level Authorization (BOLA) attacks

An API runtime protection solution can detect:

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

In addition, runtime protection logs API traffic, monitors sensitive data access, detects threats, and blocks or remediates attack vectors.

How API Security helps you

Think of runtime protection as your last line of defense when other prevention measures fall short. The primary function of runtime protection is to detect and block API attacks in real time. Autonomous machine learning (ML)–based monitoring is used to conduct real-time traffic analysis and provide contextual insights into data leakage, data tampering, data policy violations, suspicious behavior, and API security attacks. API Security detects anomalies and potential threats in your API traffic and facilitates remediation based on preselected incident response policies.

Using ML, API Security builds a behavior model for each API. This baseline of normal behavior is then used to detect API business logic attacks. Every issue generated by runtime protection includes severity, status, a mapping to the OWASP API Security Top 10, and attacker details where applicable. Issues also include evidence like the attacker's session details and a copy of the API request and response to aid in triaging and remediating the issue.

API Security's runtime protection offers real-time detection and prevention of API attacks along with continuous detection of API misconfigurations, in addition to many popular workflow integrations that simplify operations and remediation.

Perhaps the best news for your team is that API Security 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.

5 breach types, 5 prevention principles

Now that you better understand how APIs are used by cybercriminals, you can focus on preventing them. Here are the five prevention tools and strategic perspectives you need to use in combination:

1. Shift-left API security

- Shift-left API security means extensively testing APIs in development so that you aren't exposing vulnerabilities in your production environment where probing cybercriminals can find them

2. Inside-out discovery

- Identify all APIs across your entire operation

3. Outside-in discovery

- Identify and eliminate sources of exposure — such as leaked keys and credentials and improper repo use — and regularly scan the external attack surface to detect and remediate vulnerabilities

4. Comprehensive posture management

- Always put your best foot forward when it comes to API security by avoiding misconfigurations and vulnerabilities

5. Runtime protection

- Detect anomalous API activity and protect against all possible threats, including previously unidentified vulnerabilities and bugs

Request a demo

Experience just how easy it is to identify and remedy misconfigurations in your APIs and protect yourself from malicious API attacks by seeing Akamai API Security in action. Learn firsthand why leading enterprises choose our API security solution.

Get a demo

Akamai Security protects the applications that drive your business at every point of interaction, without compromising performance or customer experience. By leveraging the scale of our global platform and its visibility to threats, we partner with you to prevent, detect, and mitigate threats, so you can build brand trust and deliver on your vision. Learn more about Akamai's cloud computing, security, and content delivery solutions at akamai.com and akamai.com/blog, or follow Akamai Technologies on X, formerly known as Twitter, and LinkedIn. Published 11/24.