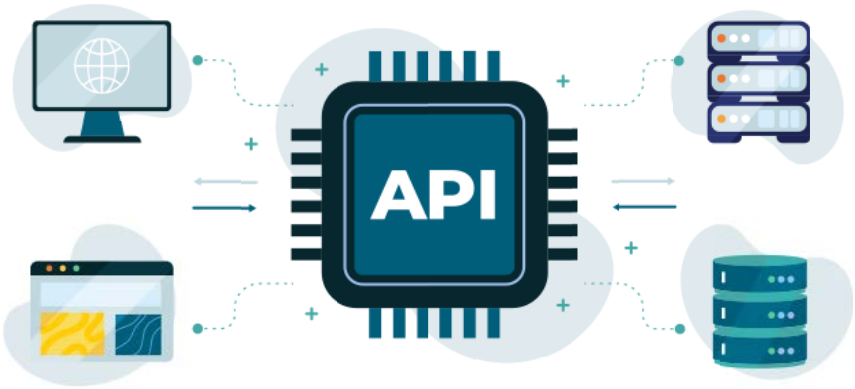
APIsec


# API Security Testing Buyer's Guide

## A Practitioner's Companion to Shift Left API Security




# 1. What is API Security? and Why Now

Application programming interfaces (APIs) allow software components to communicate with one another using set definitions and protocols. APIs have long been a cornerstone for developers to connect users, applications, systems, and data. Salesforce, eBay, and Amazon are widely credited with launching modern APIs in the early 00s as a means to fuel commerce on the web. In the two decades since, migration to cloud computing and the explosion of mobile devices has increased API usage exponentially. Now, with the rise in connected devices, APIs provide the foundation for our connected world.

Organizations of every size in every industry are reliant on APIs to fuel innovation and digital transformation. The increasing number of APIs and the connectivity they provide make APIs a leading target for attackers. API attacks have become so pervasive that API security and the vulnerabilities that lead to security breaches are now a focal point for security leaders, analyst groups, and cybersecurity vendors — even OWASP, the project that maintains the Top 10 web application vulnerabilities, now maintains API-specific tracking in their OWASP API Top 10.

## Traditional Application Security is not Enough

Traditional Application Security (AppSec) tools and methodologies are not enough to satisfy modern enterprise security needs, particularly when it comes to APIs. APIs have unique capabilities and behaviors giving rise to unique vulnerabilities and exploitations that most application security and testing solutions do not account for, like the OWASP API Top 10.

API security is a layered, complex proposition requiring strategy and solutions across three core pillars: API Governance, API Security Testing/Posture Management, and API Runtime Security.

# The 3 Core Pillars of API Security (Definitions)

## Governance

API Governance which encompasses discovery/ visibility and standards for documentation. Most organizations do not know how many APIs they have (discovery), and what the function and behavior for each API is (documentation). Because of this, most organizations today lack the visibility necessary to understand which APIs are mission critical from a security perspective (can communicate sensitive data) and aren't able to layer the appropriate security controls against those APIs.

The API Governance pillar seeks to find and classify every API in your environment to eliminate blind spots and establish best practices and security guard rails around future API development.

## Security Testing/ Posture Management

API Posture Management refers to the ability to identify and remediate risks associated with new and existing APIs. API Security Testing leverages continuous testing during the software development lifecycle to proactively identify and remediate security/ design vulnerabilities or simple misconfigurations that exist within internal and external APIs before applications are pushed to production.

Think of API Posture Management as the goal and API Security Testing as way organizations can accomplish that goal.

## Runtime Security

API Runtime Security refers to active protection for your production APIs from malicious threats and anomalous behavior. This is done by identifying and blocking risky API requests.

This API Runtime Security pillar often requires an inline solution or an integration with a cloud provider or API gateway for enforcement.

# How to Prioritize API Security by Impact vs. Effort

API security is not a sequential journey; each API security pillar is important for businesses to explore, but the effort, impact, and complexity for each is vastly different. It's helpful to understand the overall impact each pillar has on your API security as well as the effort required to realize value from the initiative.

## API Governance (Medium Impact, Medium Effort)

*"You can't secure what you can't see.*

It's easy for imaginations to run wild with "what if" scenarios with APIs that are essentially invisible to the organization. API security vendors address this issue by integrating with cloud providers, API gateways, WAFs, and load balancers to create a centralized inventory of known APIs. Some vendors also employ traffic mirroring (sometimes referred to as a virtual network TAP or vTAP) to find other trafficked APIs that are properly managed or deployed.

API security vendors offer different degrees of API Discovery — from providing you a basic count of APIs to a detailed classification of APIs and what types of data is getting requested. The visibility gained from comprehensive discovery/documentation is valuable, but on its own provides little to no improvement in the security of those APIs. "Then what?" Are the newly discovered APIs safe and secure? If not, is there a clear course of action to remediate the issue?

Yes, there is value in shining a light into the unknown of your API inventory and layering in security guidelines/guardrails in front of future API development will improve API security and increase visibility. But it is just one pillar in API Security, often the first and alarmingly often the sole focus of many organizations.

Alone, API Governance can be a heavy lift offering little in the way of addressing existing, gaping API security vulnerabilities.

# API Security Testing / Posture Management (High Impact, Low Effort)

*You can't secure what you can't see, but don't wait to secure what you CAN see.*

API Security Testing and Posture Management usually is inclusive to both APIs currently deployed in production and APIs in the CI/CD pipeline. APIs are now being deployed at a rate faster than they can be secured, which is why it is critical to identify design flaws, vulnerabilities, and configuration issues before they reach production.

There are many ways that API Security Testing/Posture Management can be implemented. One of the trends in the API security space is to adopt "Shift Left" verbiage. What this really conveys is the importance and value of ensuring testing and quality evaluations throughout the development cycle before production. Shift Left is already common practice in many areas of software development, but most organizations struggle to implement Shift Left practices for their APIs.

Shift Left API Security Testing can take on many different forms, including pre-packaged basic security tests that can be deployed to APIs in the CI/CD pipeline or those already in production. An example includes EthicalCheck, powered by APISec, that provides free and instant API penetration testing to any Open API or Postman URL.

Other forms of API Posture Management include comparing the APIs actual behavior to the expected behavior. This is sometimes referred to as Schema Validation.

Some API security vendors also offer customized testing that analyzes each API and creates specific tests based on API functionality. These vendors typically offer automated testing that can run continuously in both a pre-production and live environment to further improve API security posture by ensuring APIs remain secure, even in dynamic cloud native applications and environments.

Running API Security Testing against what you know today (current and future API builds) offers the lowest effort, highest impact solution among the three pillars giving you peace of mind that what is being pushed to production is free of exploitable vulnerabilities.

## API Runtime Security
(High Impact, High Effort)

API Runtime Security (monitoring) is another core pillar of API security. After all, nobody wants malicious attackers exploiting our APIs.

While API Runtime Security is easy to understand, it is often the most difficult to implement. True runtime security must sit inline with your API traffic so that it can block bad API requests. Most security organizations do not want to add more inline security solutions as it increases complexity and degrades performance. Additionally, security organizations are wary that inline solutions that can block traffic could be susceptible to false positives. As traffic patterns, API behaviors, and cloud environments are extremely dynamic, most security teams opt for alerting instead of blocking.

In order to avoid many of these concerns, most API security vendors have developed API Runtime Security solutions that function as a sidecar to existing inline solutions, like an API gateway or a WAF (Web Application Firewall). This “out-of-band” method is able to observe malicious or anomalous activity and relay enforcement commands back to the API gateway or WAF.

Deploying and maintaining an effective API Runtime Security solution is very complex. It requires deep integrations with multiple cloud environments and existing security infrastructure. Most enterprises who use API Runtime Security only use it in specific cloud environments because enterprise-wide deployments are not practical or even possible with the solutions on the market today.


# 2. Practical Guide to API Security Testing

*"You can design an API you think is ultra-secure, but if you don't test it, then a cybercriminal somewhere is going to do it for you."*

~ Corey Ball, Author, Hacking APIs

Here's a great resource to help you build and execute a comprehensive API testing strategy that covers functional testing, performance testing, and security testing. Here we'll dive into what is arguably the most critical phase of the testing process, API security testing.

API security testing falls into two categories: **manual** and **automated**

Manual security testing is still the industry norm, despite the severe limitations that prevent it from being a complete, one-size-fits-all solution to API security such as:

- Challenges testing all permutations of each API endpoint
- Dependency on the skill level of the developer or penetration tester running the test
- Difficulty in implementing manual tests at scale

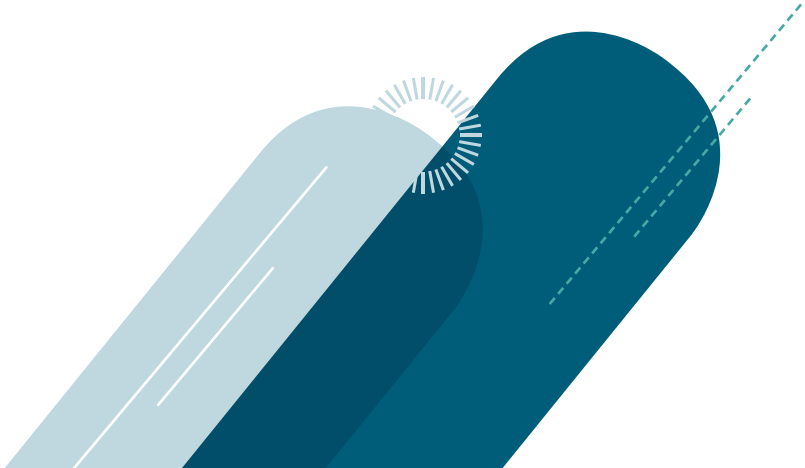
Automated security testing provides a comprehensive toolset to continuously check APIs for vulnerabilities at scale while eliminating human error.

The most common types of API security testing include:

- Penetration testing: uncovers loopholes that hackers can exploit to compromise the system's integrity, especially for known vulnerabilities using widely accepted industry guidelines set out by OWASP.
- Vulnerability scans: analyze security loopholes and business logic flaws that are often overlooked, despite being more susceptible to security vulnerabilities and the main target for hackers nowadays.

# Top 5 Best Practices for API Security Testing

1. **Integrate with existing API management tools/ gateways** to speed and simplify API ingestion into your testing solution.
2. **Implement API security testing as early as possible** to ensure stronger security and minimize development impact at the final stages of deployment.
3. **Ensure comprehensive attack playbooks** that cover all of the OWASP top 10 plus new and emerging security categories.
4. **Employ automation** to ingest and analyze APIs then compile and run custom attack playbooks at scale.
5. **Integrate with existing CI/CD tools** to generate tickets with detailed coverage reports that allow dev teams to resolve potential vulnerabilities quickly.



# Architectural Considerations for API Security Testing

When evaluating API Security Testing vendors, it's critical to evaluate their architecture requirements and dependencies. API Security Testing can either come as a component of an API security platform or as an independent testing solution. Below we will highlight the pros and cons of each.

## Pros and Cons of API Security Platforms

**PROS**

### 1. Broad API Security Feature Sets

API security platforms offer a broad set of features, including API Discovery, API Runtime Security, and API Posture Management/Security Testing. Organizations that are investing heavily across each of the aspects of API security may get more value from this multi-faceted approach.

**CONS**

### 1. Complicated Deployments

One of the challenges with platform-based API security solutions is that many of the features are difficult to deploy and often only able to provide value into the specific environments where they have been able to be implemented. For example, deploying an API security platform in an AWS instance delivers the discovery, runtime, and testing in that specific environment. Those features are not necessarily available across other Azure, Google Cloud, or on-premises environments until the platform has been deployed there as well. Given the complexity, rigidity, and dependency on multiple other security solutions (e.g. API gateways, load balancers, WAFs), it is often difficult for certain business units and teams to get access to the features they need.

### 2. Limited API Security Testing Options

In many ways, API Security Testing is a natural extension of traditional security testing, though specific to APIs. As more security teams began to demand Shift Left API solutions, many platform vendors bolted on basic API pen testing solutions to check the API security testing box. The depth of some of the API Security Testing solutions range from vaporware to wrapping an open source solution into their platform, like OWASP ZAP. However, most of the API security vendors are beginning to invest more resources into build out these solutions.

## Pros and Cons of Standalone API Security Testing Tool

**PROS**

### 1. Deep API Security Testing Feature Set

On the other hand, standalone API Security Testing tools specialize in just API tests and testing. Out of the box they offer far more tests, testing options, and even customized tests.

### 2. Faster Time to Value

Because standalone API Security Testing tools are cloud and platform agnostic, they can very quickly and easily be deployed across any cloud or on-premises environments. Even with the Open API or Postman URL you can get test results immediately using a tool like EthicalCheck, powered by APISec.

**CONS**

### 1. Singular Focus

API Security Testing tools have deep specialization in testing — they do not offer API Discovery or API Runtime Security features. If you are looking for a broad set of features across all API security categories, an API security platform may be worth evaluating. If your focus is on testing, you'll get better results and faster value from a standalone tool.

# API Security Testing Buyer's Guide Checklist

## Speak to a customer who is an actual user of the platform

Before committing to any API Security Testing vendor, insist on speaking with at least one customer who is a user of the specific platform. Ask questions about how difficult it was to get started, what kind of tests were run, and the value they receive from the platform.

Speaking with a practitioner is critical as, often, the most vocal advocates for security solutions are those that champion the idea of the product, but have little to no experience with the product itself. The best way to cut through the noise is to speak to those with first hand experience using the product.

In addition, given the context of API security testing solutions mentioned above, evaluate the following attributes for your organization and ensure that the vendors you're evaluating can deliver on your priorities.

Assign each attribute below a score based on the following criteria:

| HIGH | MEDIUM | LOW |
|---|---|---|
| Currently unmet and significant need that will drive value | Nice to have, but not something you need right away | Not a priority based on your current needs or handled well by existing solution |

## Use this checklist to understand your priorities and needs for application services.

| API Security Testing Solution Attribute | HIGH | MED | LOW |
|---|---|---|---|
| Provide consistent API security testing across cloud environments | | | |
| Detailed reporting about discovered vulnerabilities and how to fix them | | | |
| Out-of-the-box testing for common misconfigurations and vulnerabilities | | | |
| Custom testing for advanced use cases and complex environments | | | |
| Accelerate new API deployments with faster API security testing | | | |
| Continuous API security testing | | | |
| Flexible pricing based on usage | | | |
| Highly accurate with minimal false positives | | | |
| Automated processes | | | |
| Self-service for developer audience | | | |
| Integrations with CI/CD toolsets | | | |
| Faster time to value (implementation methodology) | | | |

### SEE

Want to dig deeper into the power of APIsec?

Let us pull together a custom demo to show you APIsec on your APIs.

Demo

### LEARN

Ready to become a certified API Security Expert?

Join the free APISECU course written & taught by Corey Ball.

Register

### DO

Eager to get hands on with API security testing?

Dive in and test your own APIs with our free testing service.

Free Pen Test

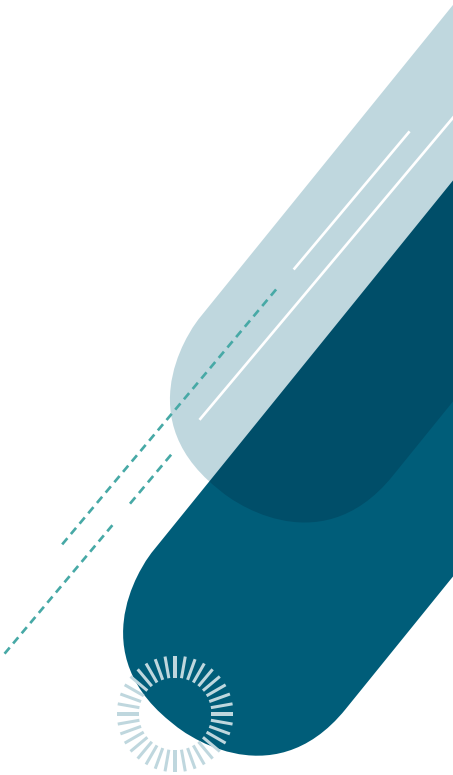
## Start Today

See the power of APIsec against your APIs to take the first step toward protecting sensitive customer data by securing your critical API infrastructure.


**Contact Sales**

sales@apisec.ai

+1 415.236.0601

APIsec

www.apisec.ai