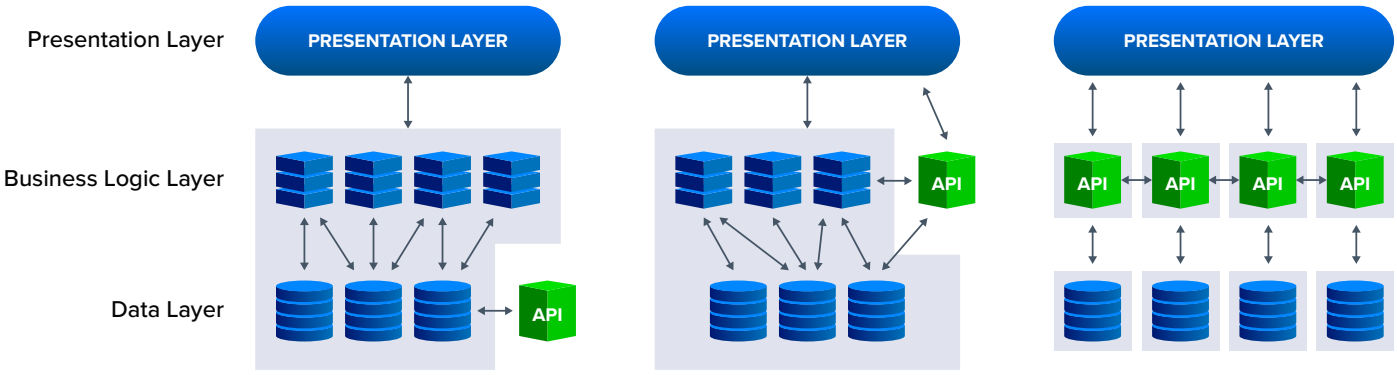
NTT Application Security

# Vantage Prevent

## Securing APIs with Vantage Prevent and Postman

Traditional security testing products are struggling to adapt to the unique design and architecture characteristics of modern Application Programming Interfaces (APIs). Over the last decade, APIs have evolved from connecting internal computer backend or B2B services where security concerns were minimal, to become critical services within the front-end Business Logic Layer (BLL) of web applications. This exponential growth has been accelerated in part by the popularity of API testing platforms such as Postman. In this paper, we will explore the current challenges facing API security testing and how organizations can concurrently establish both a robust API security strategy and a true DevSecOps process.

## Evolution of APIs Over Time




APIs enable the software applications that businesses rely on to interact with other applications and services. By connecting the various functions that allow applications to access data and external software components, operating systems and microservices, APIs optimize resources by improving efficiency and driving profitability. For the consumer, they are what bring login authentication to their banking app or connect maps to their fitness-tracking app. APIs are the "magic" that connects most modern applications together. Because of this, APIs constitute a critical service within the Business Logic Layer (BLL) of web applications, enable rich consumer experiences, and boost efficiencies and profitability across the enterprise.

> **"By 2025, less than 50% of APIs will be managed, as explosive growth in APIs surpasses the capabilities of API management tools." [1]**

Today's enterprises recognize the value that the modern web application service architecture of the BLL and inherited APIs bring to their innovation and product solutions. Understanding that APIs act as pathways that deliver business value within this newly evolved BLL, enterprises have accelerated their use of APIs, both those developed in-house and by 3rd parties. This adoption is by no means slowing down. According to Gartner's "Predicts 2022: APIs Demand Improved Security and Management,"

---

[1] **Gartner**, Predicts 2022: APIs Demand Improved Security and Management **6 December 2021**

## Unique Operational Concerns for Secure APIs

APIs also represent a tantalizing target for hackers because they enable access to sensitive software functions and data. Therefore, it is reasonable to conclude that while APIs are transforming modern application development, modern requirements must be considered so that they are responsibly built, delivered, and tested - especially for security.

Now broadly consider the combinatorial effects of Agile methods and CI/CD tools on API security. A group of API requests, such as Postman Collections, enable fast deployment of functional and customizable access to the BLL. Agility fuels the rate of pushing developmental iterations (CI). Continuous delivery (CD) automation affects the scale of exposure. So, at the intersection of contemporary methods, technology, and API security, Agile + CI/CD can automatically push to production and dramatically amplify a simple security vulnerability.

## State of DevSecOps

A new DevSecOps culture and toolchain is emerging within the realm of the API economy. Gone are the days of asking basic questions like "How do I test my APIs for security?" Organizations must be able to identify API vulnerabilities in seconds and minutes, not days and weeks. However, most traditional security products attempt to solve the API testing puzzle by retrofitting legacy technologies and continuing outdated approaches. Unfortunately, traditional API security testing products are neither modern nor API native and are not capable of adequately solving today's API security testing challenges; especially for the security of those valuable API services embedded in the BLL.

**Much like locking all the doors in a house but leaving the windows open, if an API is not well designed and tested for security, all the data that passes through it is at risk. The moment APIs are exposed externally to the internet, they create windows of attack vectors in the Business Logic Layer. Easy targets for hackers to extract data.**

# Current API Security Testing Challenges

## Problems with Documentation

Discussions about API security have traditionally focused on specification documentation. Most security testing products rely on ingesting API specification documentation. Incomplete documentation of endpoints is commonplace, and in some instances, specifications are completely absent. Although some API frameworks and API testing platforms like Postman can auto generate documentation, those documents nonetheless still require an inventory check to ensure accuracy.

## The Rise of Shadow APIs

Further complicating the documentation matter, different released versions of API specifications may not accurately reflect the version of the API in production. Such an issue may give rise to "Shadow APIs". A Shadow API is an undocumented API or a not fully deprecated endpoint. Sometimes operating as hidden developer utilities (backdoors) that bypass API gateways, or simply a long-lived orphan endpoint that slipped through the cracks; shadow APIs can remain undetected for long periods of time before security teams manually discover and inspect them. Ultimately, security testing dependent on documentation alone can become an impossible task to manage.

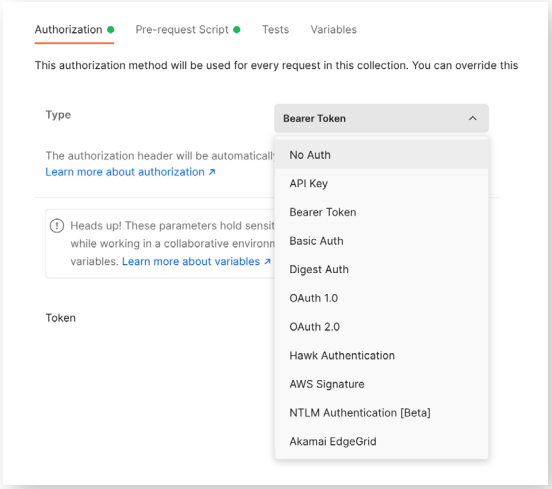
## Problems with Authentication and Authorization

Another challenge facing organizations is the adoption of a clear strategy for the Authentication and Authorization of APIs. Authentication is the verification of the identity of the user. Whereas Authorization involves determining what resources a user have read/write access. Both of these can be used by attackers for malicious means and therefore must be validated when testing for operational vulnerabilities. For example, there are several protocols developers leverage for APIs that require no password such as: Bearer tokens, API keys, and OAuth. The confusion, however, arises from the nuanced differences between authentication and authorization. For example, Bear tokens represent a form of authentication, while API keys are a form of authorization, and yet refresh tokens for OAuth2 flow blend elements of both authentication and authorization.  Not addressing the authentication and authorization concerns of APIs can produce catastrophic results as it may allow an attacker to impersonate an admin-level user and perform any actions they desire.

## Security Expertise and Manual API Testing

The architecture in which APIs are built is unique. Therefore, the vulnerabilities inherent within APIs are also unique. This makes testing for those vulnerabilities challenging as they require specialized tools and expertise. In effect, a "specialization of a specialization." So, overcoming the unique API specific security and authentication challenges specified above requires security expertise that is in short supply across many organizations. Even for those organizations that have in-house security expertise available to them, relying on those security teams to manually test APIs remains a time-consuming process and is therefore difficult to scale. For developers and QA engineers whose focus is on developing quality functionality, waiting for days or weeks to get security results is both counter-productive and can result in a loss of productivity and profitability for the organization as a whole.

*Figure 1: Authentication Methods Supported within Postman*




## Achieve DevSecOps with Vantage Prevent and Postman in 3 Easy Steps

Security testing needs to be fast and accurate. It needs to be completed by engineers in the earliest stages of the development process. For developers, security testing should be easy to use and integrate into their toolchain.

Built on our revolutionary Intelligence-Direct DAST (ID-DAST) technology, Vantage Prevent intelligently manages session state, is fast and accurate, and is ideally suited for automation rich environments. Additionally, the Vantage Prevent API Security Tester Postman Integration provides a native testing experience with Postman tooling.

**STEP 1**

The initial step on the path to DevSecOps is to create API collections in the Postman App that strategically align with business value and function. Once this step has been completed, your organization can then prioritize security testing of those high value Business Logic Workflow Collections utilizing Vantage Prevent natively within Postman. If credentials are included within the functional workflow, Vantage Prevent will capture, honor, and manage those credentials with its patented intelligent session state management technology.

**STEP 2**

The next step is expanding this DevSecOps strategy by implementing Vantage Prevent throughout the software development lifecycle. This means API security testing in the earliest functional stages of development in Postman, as well as part of QA + Security test suites and pre-production stages with Vantage Prevent CLI. If undeclared shadow APIs are embedded and exercised as part of a website, rest assured that they will be discovered and tested for vulnerabilities via HTTP Archive (HAR) file.

**STEP 3**

The final step to achieve true DevSecOps by integrating and scaling Vantage Prevent security testing into an automated CI/CD pipeline. By passing the CLI option "-fail-on-severity [critical | high | medium | low]", Vantage Prevent functions as a security gate that prevents vulnerabilities from getting released/deployed to production. [See Figure 2 below]

*Figure 2: Vantage Prevent CLI*


## No specs, no creds, no problem!

API collections are the next generation of workflows that link application business logic and data access layers to deliver business value to consumers. This critical link must be tested for security with a fundamentally new technology and native experience that empowers organizations to overcome the API specific obstacles of modern application development.

With Vantage Prevent and Postman, organizations can establish a true DevSecOps strategy and process that answers the API security hardships of documentation, shadow APIs, credential management, manual testing, and expertise.