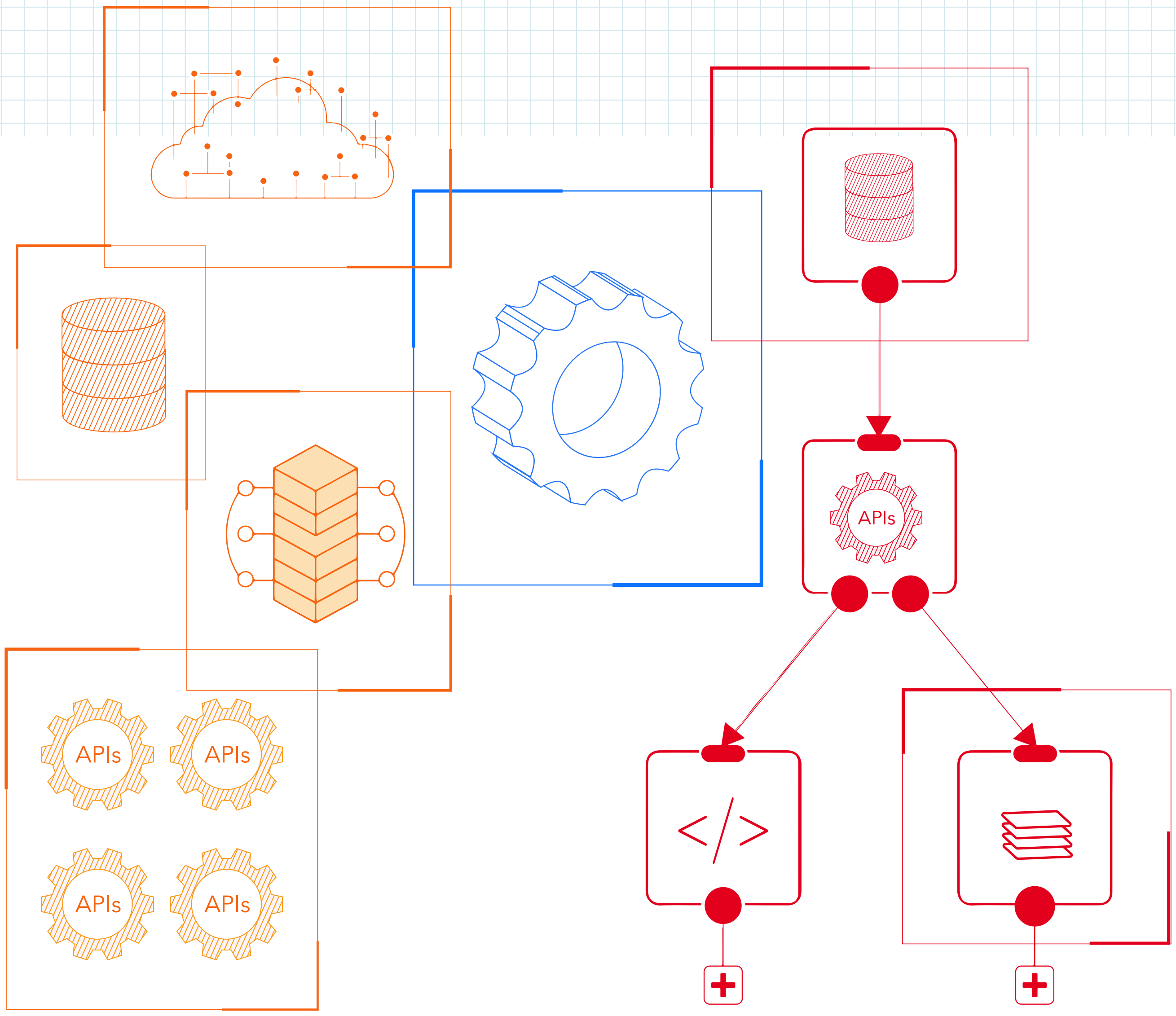
The 4 Pillars of API Security

Implementing a strategy for API security in 2025



Sponsored by datatheorem

Table of contents

Acknowledgments

Sponsored by **Data Theorem.**

Special thanks to: Richard Smith, Doug Dooley, Jesse Hafterson, Ivan (Guillaume) Druet and Corey Ball.

Published by apidays.
May 2025

Written and researched by Mark Boyd.
Design by Rebeca Vittorazo.

Foreword

Today API security is more important than ever and tomorrow it will only get worse. I began writing Hacking APIs in December of 2019 and since then we have watched the APIs take center stage not just for enabling the data-driven economy, but also for becoming one of the leading criminal attack vectors.

As a penetration tester, I've worked with APIs in banks, healthcare, technology, government, gaming, retail, and in agriculture. I've had first-hand experience with production internet-facing APIs that would have allowed adversaries to empty bank accounts, distribute malware, and compromise physical security, literally unlocking the front door and the back.

What has taken some time for enterprise security programs to realize is that APIs are a pipeline that often bypasses the layers of defense that organizations have built to protect their data and because of this they require more attention than most other attack vectors.

API usage is at an all-time high and dev teams deploy new APIs at an alarming rate. Vulnerability management programs aim scanners that are ineffective APIs and end up with a dangerous set of comforting false-negative results. APIs become unsupported and forgotten with every merger, acquisition, layoff, and shift in business objectives. Rogue APIs are growing in the shadows, creating an invisible expanding attack surface.

All of this has created the perfect storm that will, unfortunately, keep major API breaches in the headlines for many years to come.

Despite the obstacles set against API security, I remain hopeful. Over the past few years, more API-focused security books have released (Black Hat GraphQL, Defending APIs, and API Security for White Hat Hackers), educational resources like APIsec University and API Masters have helped democratize API security, and research papers like this one continue to help organizations secure their APIs.

The transition from ignorance to awareness gives me the confidence that we can rise to meet the API security challenges ahead.

Even as we see the focus of every industry turns towards large language models, model context protocols, and artificial intelligence, we again pull back the mask and find that APIs are the key technology at the center of it all.

The four-pillar approach outlined in this research provides a strategic and practical approach to API security. Discovery helps organizations shine light on their rogue APIs. Posture Management ensures that security controls are applied consistently across the API environment. Contextual testing helps vulnerability management programs move beyond misleading false-negatives, identify genuine security gaps, and remediate the high-impact weaknesses that attackers actually exploit. Finally, Runtime monitoring provides continuous visibility into API traffic, helping detect and block the attacks that might bypass the other layers of defense.

I encourage you to dive into the research presented in The 4 Pillars of API Security and I challenge you to try the 30-minute strategy to help your organization secure this vital attack vector.

Corey J. Ball

Founder and CEO @ hAPI Labs | Author of Hacking APIs | Founder of APIsec University

API Security in 2025

As 2025 reaches the halfway mark, we are seeing a fundamental shift in the approach needed for API Security.

Over the last couple of years, with the rapid rise of generative AI and the explosive growth in the number of APIs being managed by enterprises and businesses of all sizes, the API security sector has been living through its wild west moment.

In 2025, all businesses, from traditional, highly regulated monolithic enterprises to flexible, modular, ready-to-pivot startups and everything in-between are digital businesses. And digital businesses use APIs to build products, integrate systems, automate workflows, and deliver services. In recent years, as APIs expanded to become a core technological asset for all businesses and organisations, it became clear that there were limitations in the current toolset — in particular, the feature set of API management solutions — in providing the specialised subject matter expertise needed to manage API security. API security vendors entered the market en masse to offer the type of detailed API security governance and capabilities needed today.

At the same time, as an emerging technology, generative AI was leveraged by a new class of security-focused startups that believed AI could manage API security at scale, and directly in production: seeking to leverage AI to assess security threats and risks in runtime and promising to respond to challenges as they arose.

Now, almost two years into this hype of promises, the shiny object syndrome of AI and the flurry of activity seeking to capitalize on new opportunities rather than "get left behind", API security issues have been revealed as being as challenging and stubborn to resolve as ever. Implementing API Security strategies now requires a rethink and recommitment to a suite of practices that build robust, fortified digital businesses. API security needs to refocus on being a suite of practices aimed at preventing malicious attacks, protecting data, and bolstering defences against vulnerabilities.

In 2025, to deliver a true API security strategy, API providers and all types of digital businesses that make use of APIs need to focus on four key pillars: discovery, contextual testing, posture management, and runtime monitoring. There are no easy fixes for maintaining full visibility across API security, but there is an approach that can be adopted in just 30 minutes per week to implement a successful security strategy.

This white paper discusses the current drivers and needs for API security, and how businesses and organisations of all sizes can address risks methodically using the 4 Pillars Strategy. We also take a pragmatic look at where AI truly fits into the security landscape for API-first businesses in 2025.

The need for API Security in 2025

Key drivers impacting digital businesses

There are **5 key drivers** that demonstrate today's need for a thoughtful, well-defended API security strategy.

API usage is at record levels and growing every year

Today's businesses and enterprises manage between 500 and 1000+ APIs, with the number of API endpoints having doubled in the past year. Business analysts estimate that almost all businesses now manage some APIs, whether that be APIs they create for internal or partner use, consumption of services and integration via APIs, or for making APIs externally available to third party users.

A new focus on ecosystems

Being digital means that you don't necessarily need to make use of a complete, well-defined product to run your business. Instead of making use of software-as-a-service offerings, businesses may prefer to integrate directly to bring external function data and services directly into their own systems. In other cases, a digital business might want to build their product or service using your data or functionality directly in their product. Drawing on platform business models, organisations can offer APIs to assist partners and third party providers to co-create or collaborate when building their offerings. This creates a constellation of external apps and services that draw on core functionalities and data sets provided via APIs — an ecosystem business model. But when participating in an ecosystem, as either a provider or consumer (usually both) means a greater surface attack area that is partly out of your control.

More complex infrastructure

Almost three-quarters (71%) of all businesses consume APIs made available by an external provider. Around 60% of businesses use cloud computing, with 80% of those indicating their operations are spread across multiple cloud providers. With this complex web of infrastructure and APi consumption, digital businesses face multiple potential vulnerabilities. This is also increased by the number of programming frameworks used within applications built by a business. In addition, with AI technologies being adopted, digital businesses are consuming large language models and other AI components that increase possible vectors of risk.

Ongoing exposure risks

It is an axiom of security that a provider needs to ensure consistent capability to block all attacks and eliminate vulnerabilities, while a hacker only needs to succeed once to gain entry. New patterns constantly emerge for malicious hackers looking for entry points. New vulnerabilities arise in infrastructure and programming language frameworks. Digital businesses need to quickly assess the potential relevance of these new threats to their own infrastructure configurations and take deterrent action, as necessary.

Use of GenAI for malicious uses

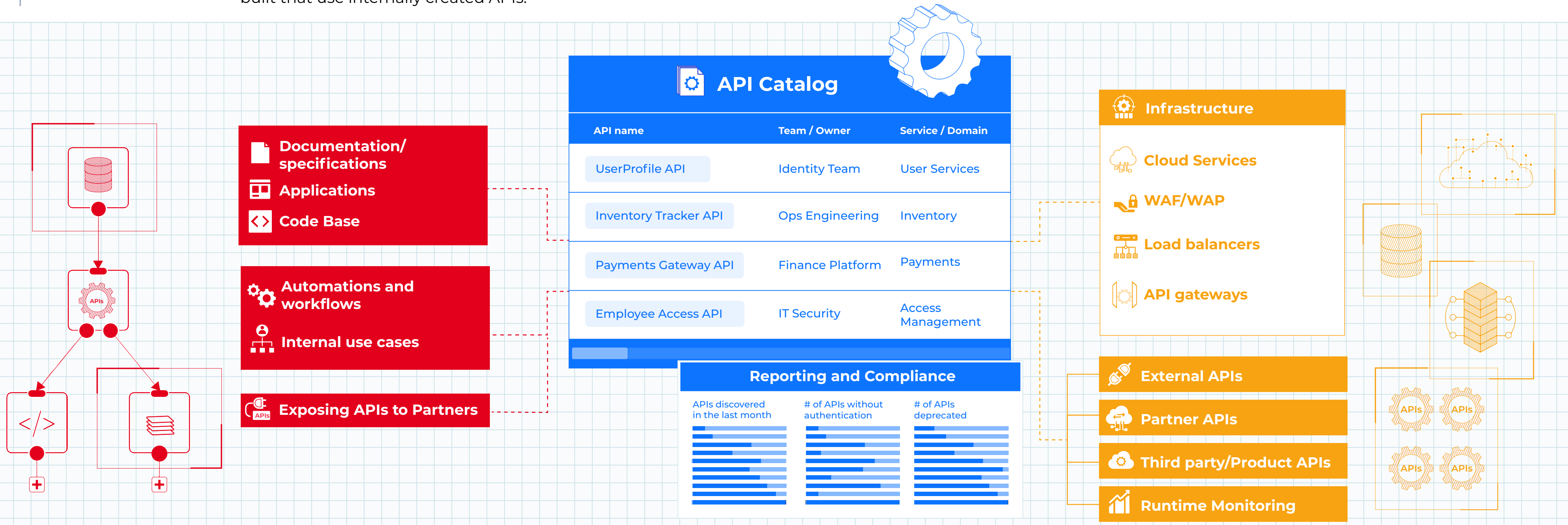
Related to this ongoing risk of exposure that needs to be monitored, GenAI presents an exponential threat: malicious attackers can more quickly repeat multiple attempts to gain access to systems, cycling through strategies and adapting approaches based on initial responses to API calls.

Driver	Strategic Pillar	Rationale
High API usage	Discovery	Maintain global overview of all APIs being deployed by the organisation
Ecosystem focus	Posture management & ASPM	Implement API best practices to reduce risks during consumption
Complex infra	Testing in context	Undertake testing across the infrastructure and API consumption surface area
Ongoing risks	Observability & Runtime	Monitor that APIs are being used as intended without new threats emerging
Use of GenAI	Testing in context Observability & Runtime	Regularly test infra to ensure GenAI usage does not introduce new risks Monitor for anomalies or abuses from malicious GenAI tools

API Security in 2025

API Security in 2025 involves ensuring there is oversight and management of infrastructure and defense when consuming third party APIs, as well as management of APIs provided by the business/organisation and app security oversight for digital products, services and workflows built that use internally created APIs.

datatheorem



API name	Team / Owner	Service / Domain
UserProfile API	Identity Team	User Services
Inventory Tracker API	Ops Engineering	Inventory
Payments Gateway API	Finance Platform	Payments
Employee Access API	IT Security	Access Management

Pillar 1: Discovery

A full API inventory is required that needs to be actively maintained and updated.

This includes

- Ensuring that scans for shadow APIs are conducted regularly and new undocumented APIs are identified and catalogued
- Zombie APIs are identified and retired All endpoints are regularly reviewed and unused endpoints are deprecated
- APIs that provide unused functionalities are regularly reviewed and additional operational capabilities are switched off when no longer in use
- APIs that make use of personally identifiable information or other data that is subject to specific regulatory oversight are identified for regular monitoring

Pillar 2: Posture Management & ASPM

Maintaining a robust API security posture particularly when evaluated with an ASPM (Application Security Posture Management) framework is essential when designing, deploying and managing APIs.

This includes

- Mapping infrastructure risks and understanding the surface attack area
- Cataloguing consumption of third party APIs and subscribing to alerts that inform of any breaches or downtime from these sources
- Assessing use of any AI technologies that have been integrated into your infrastructure or products
- Regular security testing including SAST, DAST, BOLA, IDOR, and SSRF testing

Pillar 3: Testing with Context

Continuous API Penetration (Pen) testing should be conducted regularly, on both pre-production and production APIs, especially deployed in public cloud environments where API changes on a weekly basis from many of the popular cloud platforms: AWS, Azure, GCP, etc. The contextual environment of cloud services in which your digital business operates plays a significant role in the need for testing and validation.

This includes

- Ensuring appropriate authentication and authorization is in place for all APIs
- API keys are not published anywhere
- Demonstration videos and developer resources do not inadvertently expose API keys
- TLS-level encryption is in place

Pillar 4: Observability & Runtime

As part of both your observability and API management tooling, it is important to continue to monitor security threats, alert on violations, and block malicious attack traffic. WAF rules are a simple way to deal with OWASP Top 10 API issues and known bad actors verified with threat intel.

This includes

- Ensuring rate limits are maintained and that unseasonal spikes in usage are alerted and investigated
- Alerts and thresholds are in place for anomalies and strange variations in standard API usage patterns
- Any abuse of APIs is identified and addressed quickly

The Four Pillars

An API Security Strategy for every digital business

AI will not save you

In 2025, digital businesses are waking up from the magical thinking that AI will enable instant, runtime observability of the complete API portfolio and be able to manage risks and eliminate threats in real time.

A strategic vision

Instead, digital leaders are embracing a 4 Pillars approach that provides complete security strategic management across operations: from infrastructure usage to API design and cataloguing to how APIs are exposed by internal developers in applications and to external users via API products and client applications.

The 4 pillars are:

Pillar 1: Discovery

A full API inventory is required that needs to be actively maintained and updated.

Pillar 2: Posture Management & ASPM

Maintaining a robust API security posture particularly when evaluated with an ASPM (Application Security Posture Management) framework is essential when designing, deploying and managing APIs.

Pillar 3: Testing with Context

Continuous API Penetration (Pen) testing should be conducted regularly, on both pre-production and production APIs, especially deployed in public cloud environments where API changes on a weekly basis from many of the popular cloud platforms: AWS, Azure, GCP, etc. The contextual environment of cloud services in which your digital business operates plays a significant role in the need for testing and validation.

Pillar 4: Observability & Runtime

As part of both your observability and API management tooling, it is important to continue to monitor security threats, alert on violations, and block malicious attack traffic. WAF rules are a simple way to deal with OWASP Top 10 API issues and known bad actors verified with threat intel.

30 mins

Maintaining API security strategic practices across these 4 pillars can be done in 30 minutes each week.

The benefits of proactively managing API security also needs to be regularly communicated: tracking business metrics that demonstrate the cost and operational savings from proactive API security ensure this 30 minute a week investment is preserved and funded as a core operational task.

Pillar 1: Discovery

A full API inventory is required that needs to be actively maintained and updated.

⚠ The risk

As your developers build out digital products and services and automate workflows, the value of APIs becomes evident. Developers can quickly create new features using composable, modular APIs that meet specific use cases. External-facing APIs might be created strategically, with senior architects or business leads involved in deciding on what APIs should be built to integrate systems with high value customers or partners, or to be externally exposed to an ecosystem of potential users. But deciding to build internal APIs may come down to a team or single engineer decision, based on their immediate need to get a feature shipped or to reduce time to market.

Over time, the catalogue of APIs being managed by an organisation becomes more complex:

Shadow APIs emerge that were built by a single engineer or developer team to meet an immediate need, but these have never been fully documented and are not visible in internal catalogs

---

Zombie APIs clutter your portfolio: APIs that were once built for specific needs but no longer have active users.

Unused endpoints and dormant operational functionalities remain available. Originally envisaged as being useful for API users (internally or externally), over time, some API endpoints and functionalities have not found a market-fit or true use case need, and are not used but are still available as an opportunity for malicious actors to leverage to enter your systems.

How to address this risk

While APIs create great value for digital businesses, they do also increase the number of attack vectors you must manage.

Each API and endpoint represents an opportunity for external actors to try and gain access to your systems. Each API and endpoint represents a risk for accidental data leakage if not appropriately secured.

API discovery is the process of scanning and ensuring you have full visibility of all APIs, endpoints, and operational functionalities across your API inventory or portfolio.

Leveraging Discovery as a Pillar of API Security: Strategic actions you can take

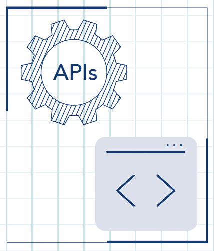
- ✓ Analyse network traffic to identify if there are API calls coming from undocumented APIs. This can be done by analysing logs coming off application load balancers and data from your gateway infrastructure to identify if there is any traffic that is aligned with known API calls. This may surface undocumented APIs.
- ✓ Monitor which APIs are being used and which endpoints are being called on a regular basis to identify if there are unused APIs and endpoints that can be deprecated.
- ✓ Track what type of API operational calls are being made to each API and retire operational capabilities for API endpoints where they are not being used.
- ✓ Scan code repositories for API specifications and references.
- ✓ Implement app security reverse engineering practices such as decompiling production apps, to review whether all API endpoints in your apps are aligned with expected calls and infrastructure. This may surface APIs that are not visible via network traffic monitoring.
- ✓ Track what APIs make use of personally identifiable information or other data that is subject to specific regulatory oversight are identified for regular monitoring. Deprecate these APIs or endpoints if they are not in use.

Surfacing Shadow APIs

Data Theorem has been able to help a broad range of enterprise clients to identify undocumented APIs. In many cases, these were APIs that were quickly created to provide some specific functionality, or to glue an automated workflow process together for internal use and the developer team didn't see a need to explain to others what the internal API was doing, as they were in a rush to complete their work. Data Theorem's work to surface all of the APIs in a client's inventory to enable better security monitoring is a standard part of their approach.

However, a truly hidden set of shadow APIs was identified with one client. The client infrastructure was solely on Google Cloud Platform and had built a number of enterprise applications using internal APIs, and had exposed similar APIs to partners and externally. By reverse engineering the enterprise applications using AppSec practices including decompiling code, Data Theorem discovered their enterprise client had a consumer-facing app that made use of several lambda APIs operating on a small AWS cloud instance. To keep up with a feature set need, a developer team had decided the best solution for a quick deployment would be to build a couple of APIs to leverage a serverless architecture approach to quickly compute some data and return a result. The developer team used a credit card to instantly purchase sufficient serverless cloud infrastructure to perform this small serverless function, creating a security gap for the organisation that now had APIs without sufficient guardrails deployed on a non-approved cloud provider's platform and operating outside the identified organisational-wide API inventory.

Pillar 2: Posture Management & ASPM



Maintaining a robust API security posture particularly when evaluated with an ASPM (Application Security Posture Management) framework is essential when designing, deploying and managing APIs.

⚠ The risk

Pentesting is a common approach to managing security, but most often occurs on the applications and APIs built by a digital business. As digital infrastructure complexity increases, we now all work in a web of distributed architecture: making use of multiple cloud provider infrastructure, drawing on open source tools that require access to open source library components, making use of programming language frameworks and SDKs directly in our applications, and consuming external APIs in our value chains.

Our pentesting approach needs to be adapted to reflect this complexity. Instead of security testing solely on the products we release, we need now to also focus on pentesting and security analysis on the infrastructure we make use of as well (as shown in the diagram on page X).

In addition, with the current AI gold rush and deployment, many GenAI startups and companies are working quickly to expose AI functionalities via API as quickly as possible. That often means that rigorous security vetting has not occurred and is shifted onto the customer or API users responsibility. This means that proper due diligence, security reviews, and exposure mapping must be conducted on any infrastructure that is making use of GenAIs in your digital supply chain. Because of these known risks in the rush to deployment, there are early signs that hackers are focused on this AI infrastructure as a potential entrypoint.

This year, the majority of large data breaches and security threats have emerged from large language models where bulk registers of API keys have been published openly.

How to address this risk

Understanding the complete infrastructure and contextual landscape in which you operate is essential to understanding your potential security surface attack area and all possible risk vectors.

With this mapping, you are able to conduct security and pentesting across your contextualised environment at all points of risk and vulnerability.

Security testing with contextualization is the process of ensuring that you are conducting sufficient security monitoring across all of your operations and not just on the products and services that you generate with APIs.

Leveraging Posture Management & ASPM as a Pillar of API Security:

Strategic actions you can take

- ✓ Map your infrastructure to identify and understand the surface attack area so that you can monitor accordingly.
- ✓ Conduct appropriate due diligence on the external APIs that you consume: do they leak data? Do the APIs you plan to use have encryption in place? Catalogue the external APIs that you use and where possible ensure a process is in place to review and approve APIs before they can be used across your business.
- ✓ If you make use of SDKs and open source libraries, make sure that IP addresses do not appear on any threat intel watchlist.
- ✓ If you are a sufficiently large enterprise, you may wish to investigate using a threat intel watch list feed.
- ✓ Make use of agentic AI and large language models carefully. Assess use of any
- ✓ AI technologies that have been integrated into your infrastructure or products.
- ✓ Ensure appropriate security vetting has been conducted before using in production scenarios. Ensure the GenAI APIs do not become a threat vector in themselves.
- ✓ Conduct regular testing including SAST, DAST, BOLA, IDOR, and SSRF testing.

A deep dive on the types of testing you can perform

SAST: Static Application Security Testing is an automated process to analyze the source code of your applications as they are developed.

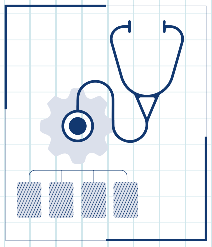
DAST: Dynamic application security testing is a complementary testing activity aimed at detecting potential security vulnerabilities when an application is running.

BOLA: Broken Object Level Authorization testing is an essential part of API security predominantly because this is the most common type of API security vulnerability experienced. It is considered the top risk in the Open Web's Application Security Project (OWASP) list of top 10 API threats and occurs when there are insufficient safeguards to determine whether someone should be able to make the API call and access the digital services via API that they are requesting.

IDOR: Insecure Direct Object Reference testing tests whether users are prevented from accessing resources that they should not have access to. While BOLA focuses on overall access and authentication, IDOR tests whether authenticated users are restricted from accessing resources that they should not be able to within their permission access levels.

SSRF: Server-side request forgery testing is undertaken to ensure that data cannot be accessed or is not vulnerable to being leaked, from where it is stored in a system to external access.

Pillar 3: Testing with Context



Continuous API Penetration (Pen) testing should be conducted regularly, on both pre-production and production APIs, especially deployed in public cloud environments where API changes on a weekly basis from many of the popular cloud platforms: AWS, Azure, GCP, etc. The contextual environment of cloud services in which your digital business operates plays a significant role in the need for testing and validation.

⚠ The risk

Globally, compliance is a key issue when participating in digital ecosystems. Across Europe, Latin America, and other regions, when working in highly regulated digital ecosystems including open banking, digital health, insurance, energy and utilities, and supply chain management, businesses need to demonstrate their capacity to meet stringent cybersecurity standards and protocols, as well as adhere to data protection regulations and share confidential commercial data with government authorities.

But compliance isn't always regulatory. Across the United States, for example, a strong security posture is an essential requirement when onboarding collaboration partners, accessing investment, or fostering a trustworthy reputation with end customers.

Posture management often involves implementing API design and deployment best practices. While these best practices might relate to the efficient management and observability of value generation of APIs, they also embed strong security hygiene and governance.

Chief among these best practices is authentication. Authentication and authorization risks account for the majority of vulnerabilities and security breaches from APIs. This can include APIs that do not have adequate levels of authentication, as well as unsecured API endpoints and functionalities that allow users to access resources for which they are not authorized.

While authentication provides multiple benefits for security, it does come with one weakness: API keys. The very mechanism that enables security to be enhanced can also be a new risk vector if not properly protected. A range of developer resources aimed at supporting API users can be an area where API keys are inadvertently shared. For example, some design and testing services like Postman offer platforms where API providers can share recordings. Other services including GitHub and GitLab might be used to share code snippets or example applications. Even video tutorials and demo presentation recordings could include examples where API keys are shown. This can also be a particular concern in enterprises that use contracted developer teams, as more external teams are provided with testing and training resources to assist them work independently, with API keys recorded and shared widely in these contracted environments.

How to address this risk

In 2025, Basic AUTH is no longer sufficient for securing APIs. API providers need to shift to OAuth2 as a minimum best practice for authentication over the course of this year. Unfortunately today, hackers are truly adept at navigating around Basic AUTH restrictions and can attack APIs if this is the defined authentication mechanism. Shifting v to OAuth2 does take some time but can be achieved by making regular, consistent progress towards updating your APIs. For personally identifiable information or authentication in highly regulated industries like banking, moving towards JWT as a minimum is recommended.

On a similar note, encryption is also a necessity. Requiring APIs to make use of transport security layer 1.2 or above (that is, TLS 1.2 or greater) is similarly essential in 2025.

Ensuring API keys are not published is an essential part of posture management.

Posture management involves API deployment best practices aimed at putting authentication and authorization in place, and managing access and publication of API keys is restricted.

Leveraging Testing with Context as a Pillar of API Security: Strategic actions you can take

- ✓ Put adequate authentication and authorization in place for all APIs and regularly check that all of your APIs require some level of authentication to access
- ✓ Create a plan to shift to OAuth2 as a minimum level of authentication for all of your APIs
- ✓ Regularly check that API keys are not published anywhere in developer resources, code snippets or or code stored in online repositories
- ✓ Review developer resources including demonstration videos, recordings, testing tools and so on to ensure that API keys are not being inadvertently exposed
- ✓ Make sure TLS-level encryption at minimum of level 1.2 is in place for all APIs.

How does authentication drop off?

Surprisingly, studies show that the majority of APIs are not adequately authenticated and that processes to restrict API access to authorized users are not often in place.

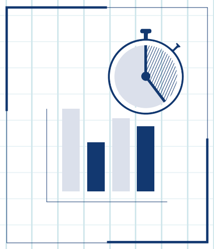
But how does that occur when authentication is readily accepted as a foundational element of good API design and deployment?

Authentication processes are workflows can be challenging to implement and are often a separate process to test and deploy from other API testing processes that might focus on the functionality and design of an API and whether API calls are returning the correct information, are paginated appropriately, make sense for end users, and manage data and service functionality in a way that aids the end user to perform their tasks.

When these API reviews are conducted, AP{I authentication may be switched off to ensure the API works as designed and expected.

The challenge is that API authentication processes are then forgotten to be added when other API reviews are completed. Instead, the rush to ship and deploy is made without adequately securing APIs to require authentication and authorization for use. Many recent data breaches have been the result of APIs lacking assumed security that had previously been in place.

Pillar 4: Observability & Runtime



As part of both your observability and API management tooling, it is important to continue to monitor security threats, alert on violations, and block malicious attack traffic. WAF rules are a simple way to deal with OWASP Top 10 API issues and known bad actors verified with threat intel.

The risk

As your APIs are exposed to public audiences — either to partners or third parties — directly as Product APIs, or within the applications you build — either internal enterprise apps or for public audiences — you need to ensure that they are not creating new security risks and data leakage vulnerabilities.

This requires ongoing monitoring and observability to ensure that new attempts to break into your digital business via your APIs is not occurring, and that brute force attacks are not being coordinated in a way to wear down your API infrastructure to create breach zones.

Now, with the ready availability of GenAI tooling, the risk of militarized attack forces on your APIs is also increased. Malicious actors can set up GenAI to continually attempt to breach your API infrastructure by repeatedly attempting access, cycling through responses or slightly altering attempts until a breach opportunity is found.

Attempts to address runtime monitoring using AI has been unsuccessful. At this stage in 2025, it is still far too early in the technology maturity journey to rely on AI for this assistance. This is due to two main reasons: First, API security requires a commitment to all four pillars of strategic action, not just to runtime monitoring. And secondly, AI currently creates far too much noise to be a successful tool. The rate of false positives from AI-enabled alerting creates more workload for security specialists without any return on identifying true risks and breaches.

How to address this risk

Using security observability tooling to continually identify potential anomalies and abuses, and having alert thresholds set up to alert when this occurs, is an essential component of API security.

Runtime monitoring needs to be seen as an element of API security, that is, as one of the four pillars, and not the primary mechanism to ensure API security oversight. Runtime monitoring provides an ongoing opportunity to manage security observability as hackers and vulnerabilities emerge but should be a last defence after efforts are in place to act on the other three API Security pillars.

Leveraging Observability & Runtime as a Pillar of API Security:

Strategic actions you can take

- ✓ Define rate limits and seasonal fluctuations, and establish alerts to track and report unexpected spikes in usage.
- ✓ Set alerts and thresholds for anomalies and strange variations in standard API usage patterns.
- ✓ Accept an initial period of false positives when any threshold level is defined and tweak metrics accordingly to help identify true anomalies.
- ✓ Allocate resources to respond to alerts of anomalies and abuses so that they can be addressed quickly.

Reporting and compliance

Part of the Return on Investment from sufficient security management is the number of events (such as data breaches or malicious attacks) that DO NOT occur. Therefore it is vital for business decision-makers to receive regular KPI reports informing them of the savings generated from robust security management.

API security tooling with dashboard features allows your team to maintain oversight of the security and compliance risks that are being addressed by your focus on the 4 Pillars.

Metrics reported to business could include:

- **Number of new APIs discovered each month**
- **Number of APIs and/or operations and/or endpoints deprecated/ made inaccessible**
- **Number of APIs identified without authentication in place**
- **Number of APIs without authentication that were fixed**
- **PII Leaks found**
- **PII Leaks fixed**

Conclusion

One of the key challenges for API security over the past couple of years has been the overpromise of being able to manage security threats in runtime.

With the emergence of AI, speculation mounted that this new technology could be leveraged to provide a real-time monitoring platform on top of an organisation's API infrastructure and applications to analyze all threats, vulnerabilities, attempted breaches and other abuses and immediately respond. This became known as a "block and stop" approach, that startup API security tooling vendors promised they could deliver with AI.

Two years into this experiment and it has become clear that there are no quick fixes to API Security.

The truth is that API infrastructure is complex: cloud providers and other infrastructure, managing API consumption and production, working with a range of API consumer audiences, maintaining applications, services and workflows that draw on APIs... The API security attack surface is widespread and requires constant vigilance.

API Security requires a strategic approach based on 4 Pillars:

Pillar 1: Discovery

Pillar 2: Posture Management & ASPM

Pillar 3: Testing with Context

Pillar 4: Observability & Runtime

Setting aside **30 minutes each week** to work through strategies aimed at fortifying each of these pillars makes API security work achievable when building your robust, compliant, and reliable digital business.

API Security Checklist

- [ ] Get Authentication Re-enabled on Leaky APIs
- [ ] Remove PII from Unauthenticated APIs
- [ ] Remove API Keys (not IDs) from Git repositories and apps
- [ ] Takedown Zombie APIs
- [ ] Assess online tutorials, demos, recordings and presentations to ensure API keys are not published
- [ ] Get rid of BASIC Auth and favor OAuth2
- [ ] Enabled TLS 1.2 or later
- [ ] Define Rate Limiting
- [ ] Disable unnecessary HTTP methods
- [ ] Conduct BOLA, IDOR and SSRF assessments regularly