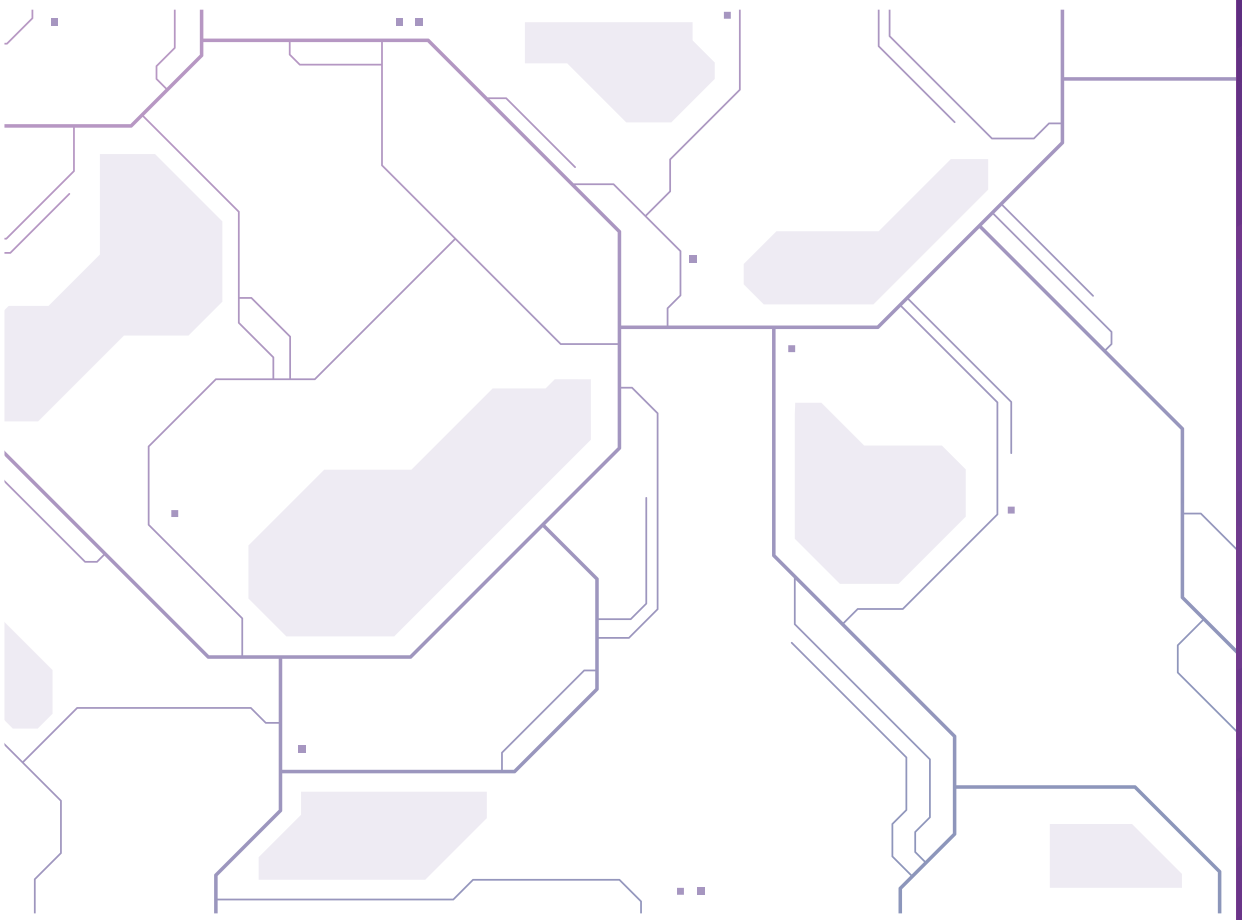
42Crunch API Security vs Behavioral Analytics

Why using **Machine Learning** Alone Falls Short for API Security and Why **'Secure by Design'** is Essential.



Executive Summary

As organizations race to deliver innovative digital services, the rapid development of APIs, super-charged by AI automation and vibe-programming, risks overwhelming security teams who are tasked with protecting the business from API security breaches.

Vendors such as Traceable.ai, Salt Security, Noname, and Cequence have responded with machine learning (ML) and anomaly detection technologies that promise to identify abnormal behavior in API traffic. Behavioral analysis can be valuable for detecting long-term anomalies, such as repeated fraud or advanced persistent threats (APTs), but these tools are reactive by nature, and that's their fatal flaw in the API space. Behavioral analytics systems are notoriously difficult to install and maintain, require time to establish a baseline of "normal" behavior, time to retrain after API changes, and often suffer from generating false positives and blind spots due the dynamic nature of APIs.

Meanwhile, attackers don't wait. Reconnaissance and exploitation unfold within minutes or hours, not weeks, rendering behavioral analytics too slow to prevent real-time threats.

This whitepaper introduces a "Secure by Design" approach, widely backed by global cybersecurity agencies. This approach establishes an accurate baseline of secure API behavior based on design intent, to deliver a more complete and accurate real-time protection for APIs that scales and adapts to changes with the speed of DevOps.

Table of Contents

The Limits of Reactive API Security

Machine learning and anomaly detection promise to uncover threats based on traffic patterns over time. But this strategy faces critical challenges for securing dynamic APIs:

Time to Train

ML systems need large volumes of traffic to learn what “normal” looks like, a process that can take weeks, during which attacks may go undetected. Frequent API changes break models and require constant retraining, while each API’s unique data structure and security requirements prevents reliable reuse of training data across APIs. As a result, anomaly detection often fails to keep up with fast-moving API environments.

Most API Attacks Are Targeted, Not Behavioral

Most real-world API vulnerabilities and breaches don’t resemble slow-moving fraud or long-term misuse of static applications. Instead, attackers use targeted probing to uncover flaws described in the OWASP API Security Top 10. These attacks are executed rapidly by threat actors equipped with automation tools, malicious bots, and increasingly, AI-driven techniques. They exploit vulnerabilities in API data structures and access controls in real time, long before anomaly detection systems have time to react.

After-the-fact Blocking

Reactive API security solutions typically work by analyzing behavior at the user or session level. They might detect that a user is abusing an API and eventually raise an alert, but only after malicious requests have already succeeded. This approach fails to protect against transaction-level threats, where a single API call can exfiltrate sensitive data or trigger unintended business logic. Most API attacks exploit specific endpoints or payload structures, not anomalous behavioural patterns over time. Without real-time enforcement, these individual attacks slip through perimeter defenses.

Why API Security Must Start with Development

The only accurate source of truth for API behavior is the OpenAPI contract, authored by the developers and audited as part of a hardening process to ensure the contract is precise, complete and follows API security best practices. The contract defines the expected inputs, outputs, supported paths, operations, authentication schemes, and security requirements. It represents the true design intent of the API, directly from the source.

Some anomaly detection tools have added support for uploading OpenAPI files in an attempt to improve traffic analysis and baseline accuracy. But without a rigorous and enforceable hardening process to ensure all OpenAPI contracts represent security best practices, as well as the developer's design intent, inconsistencies and low quality contracts will expose gaps in the runtime protection.

Security Aligned with API Changes

As a best practice, API contracts should be kept in sync with implementation to ensure accurate, up-to-date documentation and a consistent experience for API consumers. Because the OpenAPI contract is updated whenever the API changes, it provides a reliable and evolving blueprint for both functionality and security, unlike ML baselines, which are broken by each change and must be re-learned.

A Secure by Design approach reinforces this discipline by leveraging the OpenAPI contract as the foundation for security. This means that every time an API is updated, the documentation, security policies, and runtime behavior remain fully aligned, ensuring precision, reducing drift, and maintaining trust for both internal teams and external partners.

Promoted by Cybersecurity Agencies

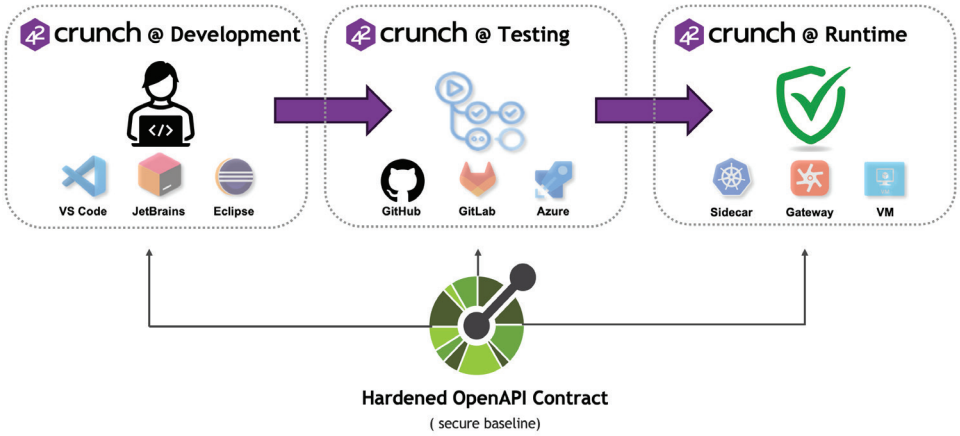
Government and cybersecurity agencies worldwide now advocate for Secure by Design as the standard for software development, including APIs, and as the most effective way to remove critical vulnerabilities.

- **United States (CISA):** The Cybersecurity and Infrastructure Security Agency's Secure by Design guidelines promote embedding security into development practices, and code hardening during the design phase.
- **United Kingdom (NCSC):** The UK's National Cyber Security Centre, in collaboration with the Government Digital Service, has integrated Secure by Design principles into its cybersecurity framework. These principles mandate that security be embedded throughout the digital service lifecycle.
- **Australia (ACSC):** The Australian Cyber Security Centre includes secure software development in its Essential Eight, urging secure design practices and early-stage vulnerability prevention.
- **Singapore (CSA):** The Cyber Security Agency of Singapore promotes secure development practices in its National Cybersecurity Strategy, emphasizing "security by design" in its guidance on software and systems development.

These agencies concur: initiating security measures at runtime is too late.

How 42Crunch Enables Secure by Design

42Crunch delivers a practical, developer-centric path to Secure by Design, built on the principle that the most reliable and precise definition of API behavior and security comes directly from the developers who design the API. Unlike machine learning or anomaly detection, which rely on retrospective guesswork and incomplete traffic samples, the 42Crunch approach ensures that API security is defined **intentionally**, not **inferred**, using trusted standards and best practices for vulnerability **prevention** as outlined in the OWASP API Security Top 10 guidelines.

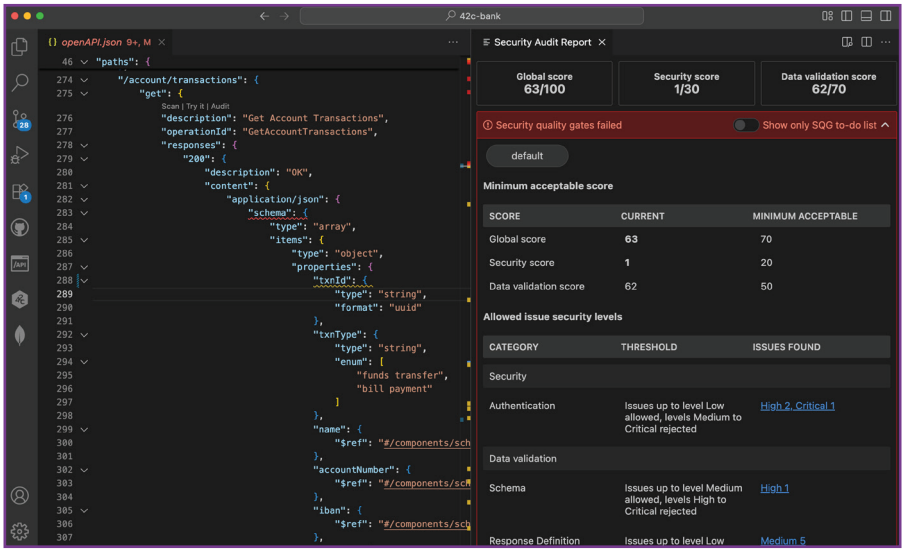


Developer-First Security Enablement

42Crunch has pioneered developer engagement in API security by delivering tools built by developers, for developers. By integrating into popular IDEs like **VSCode**, **JetBrains**, and **Eclipse**, developers receive real-time feedback and guidance directly in their workflow, empowering them to adopt secure design and coding practices without context switching to unfamiliar tools or platforms.

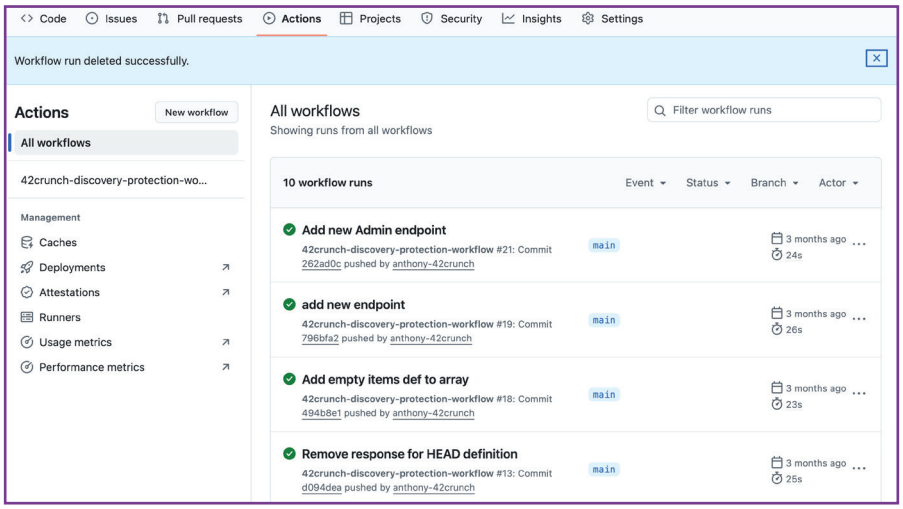
Through the **42Crunch API Scan** capability, developers can validate that their API code aligns with the defined contract, and that essential security controls, like authentication, authorization, and data validation, are implemented correctly. This enables teams to catch and remove vulnerabilities and anomalous behaviors early, before code is submitted to CI/CD pipelines, and well before they reach staging or production.

Tests are meaningful and provide full API coverage because they're based on the API design, not on unreliable or incomplete runtime traffic. This also enables security testing to start at the earliest stages of development, making shift-left a practical reality.



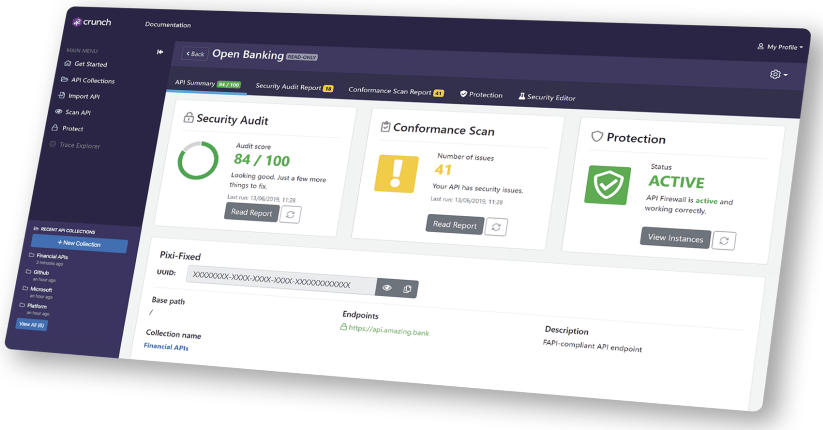
Security Integrated with CI/CD

42Crunch supports **pull-request security enforcement,** giving AppSec teams confidence that security policies and development standards are followed consistently across teams. Every change to an API triggers automated checks to verify contract conformance and detect missing or misconfigured security controls, providing fast, actionable feedback where it's most effective: early in the development lifecycle.



Runtime Contract Enforcement

In production, 42Crunch enforces the OpenAPI contract at runtime, validating that every API request conforms to the defined schema, parameters, and behaviors. This approach ensures for every API transaction only legitimate and expected traffic is accepted, and malformed, malicious or unexpected traffic is blocked in real time.



Cost Benefits of Shifting Left

Addressing security vulnerabilities during design or development is far more cost-effective than post-deployment fixes. Reducing vulnerabilities before production also lowers organizational risk by limiting opportunities for attacks. IBM's 2024 Cost of a Data Breach Report cites the average breach cost at $4.88 million globally, an increase of 10% from 2023. With APIs as a growing target, preventing vulnerabilities minimizes the risk of costly incidents and post-release remediation.

Developers Are Most Effective While in Context

Security issues identified during a pull request or while coding can be fixed immediately and accurately. Waiting for anomaly detection tools to raise alerts weeks later increases risk, as developers may have moved on to other tasks or may not recall the implementation details. The best time and place to provide vulnerability feedback is in the developer's IDE as they work on the API.

Late Fixes Introduce Regressions

Implementing fixes after deployment is risky. Even small changes can unintentionally impact unrelated parts of the system. The result? Breakages, customer issues, emergency patches, and mounting technical debt.

Customer Results: Proven Solutions at Scale

Organizations across industries are transforming how they build and secure APIs with 42Crunch. From financial services and automotive to healthcare and insurance, some of the world's most demanding and regulated environments now rely on 42Crunch to adopt secure-by-design practices at scale.

These organizations are embedding API security directly into development, enabling developers to design, test, and deploy secure APIs without disrupting workflows, while AppSec teams enforce policy through automated checks in CI/CD and at runtime.

One global telecommunications leader, with over 6,000 developers worldwide, uses 42Crunch to manage and secure more than 20,000 APIs across its enterprise. By integrating security into the development lifecycle and automating API testing, the company ensures built-in protection, consistent governance, and continuous monitoring at scale.

Teams using 42Crunch consistently report a dramatic reduction in the number of vulnerabilities reaching production, faster mean-time-to-remediation, and improved alignment between development, security, and compliance efforts, proving that secure-by-design isn't just possible, it's already working.

Conclusion

API security is complex, and no single solution can cover it all. Machine learning and anomaly detection play a valuable role in identifying long-term patterns, fraud, and advanced persistent threats, offering operational teams insights into behavioral trends.

But most API attacks aren't slow or predictable, they're fast, targeted, and exploit flaws tied to specific API logic or data structures, as outlined in the OWASP API Top 10. In fast-paced, large-scale API environments, relying on traffic patterns alone adds noise, delays response, and undermines confidence in automated defenses.

That's why Secure by Design is essential. By defining security during API design, in the OpenAPI contract, and verifying it throughout development, organizations can prevent vulnerabilities before deployment and reduce risk at scale.

42Crunch enables this shift. Security is defined by developers, verified in CI/CD, and enforced at runtime, ensuring precision, alignment, and protection across the API lifecycle.

Define it. Enforce it. Scale it. With 42Crunch.

About 42Crunch

42Crunch enables a standardized approach to API security that automates the enforcement of security compliance across distributed development and security ecosystems. Our API security testing and runtime threat protection services are used by Fortune 500 enterprises, mid-sized firms and more than 1.8 million developers worldwide.

The 42Crunch API security platform empowers developers to build security from the IDE into the API pipeline and gives application security teams full governance from the CI/CD across the entire API lifecycle. This seamless DevSecOps approach to API security reduces governance costs and accelerates the delivery of secure APIs.

Contact Details

Email: **sales@42crunch.com**
Website: **42crunch.com**