WHITE PAPER

API SECURITY TESTING

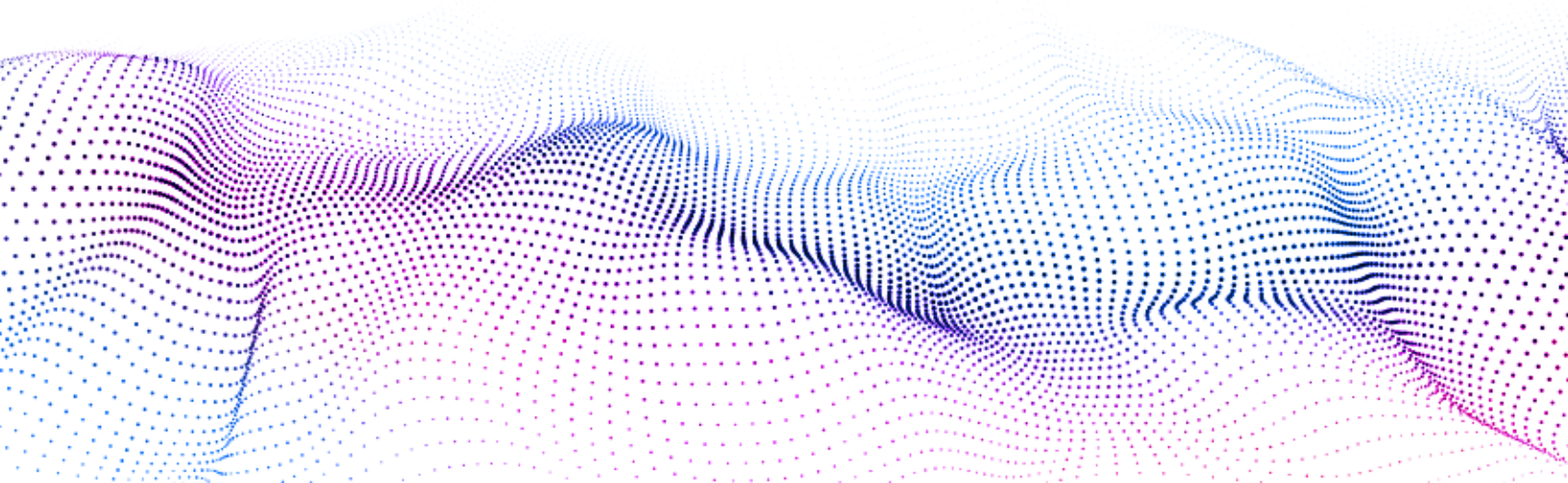
A smarter approach to address one of today's fastest-growing threats

Synack

Executive Summary

Modern software application architectures increasingly rely on application programming interfaces (APIs) to ensure that their components can communicate with one another. This development pattern is a hallmark of cloud computing, and is coming to dominate the way that software is built today. As a result, the number of APIs is skyrocketing and API-based vulnerabilities are becoming more and more common. API-related breaches are also becoming more costly and damaging.

A secure-by-design approach is needed to solve this problem. Testing for security flaws in APIs should be done continuously, starting early in the development lifecycle if possible. Because the types of potential vulnerabilities in APIs are so varied and diverse, it's essential to apply ingenuity and creativity—as well as thoroughness and rigor—within the API security testing process. For this reason, a human-led approach harnessing the expertise of diverse global security researchers and ethical hackers is the one that's best suited for the task.



The rise of APIs and the API economy

Today's software applications are designed in ways that are very different than they were even a few short years ago. Applications are much more likely to be loosely coupled collections of discrete components and functions, and much less likely to be monolithic units. As a result, communication — among the parts of an individual application and between applications — is now the lifeblood of computer systems. In modern software design, all the parts of an application are constantly "talking" to one another to ensure that the system as a whole can run. They do this "talking" via application programming interfaces (APIs).

APIs are interfaces that enable software programs to transmit data to other software programs, typically in response to requests (called API calls) that are formatted in accordance with the individual API's requirements. Integrating applications via APIs makes it possible for one piece of software to make use of the capabilities of another. In today's increasingly interconnected digital world, these types of system-to-system communications are becoming more and more common.

As increasing numbers of business processes are digitized, organizations need to expose their data and services to the outside world — to serve their customers, collaborate with partners and enable a myriad of application integrations. APIs are the primary mechanism through which they achieve this interconnectivity. As a result, the number of publicly-exposed APIs is growing exponentially.

The move to the cloud has further increased reliance on APIs. Cloud migration was well underway before 2020, of course, but the past few years' events, including the large-scale support for remote work and rapid growth in markets for digital products and services, have accelerated its progress. Cloud applications — particularly those, called cloud-native, that were purpose-built to run in cloud environments — are by definition comprised of interconnected sets of independent services. All of these services communicate via APIs.

In general, application architectures are being designed to make greater use of APIs. Modern architectures are typically microservices-based, meaning that they're made up of many loosely coupled, independently deployable small services. Besides being well suited to run in the cloud, this architecture pattern makes it easier to maintain and update the application (making the business more agile) and possible to scale service usage (improving cost efficiency). Plus, regulatory mandates are forcing API adoption in industries like financial services and healthcare.

As increasing numbers of business processes are digitized, organizations need to expose their data and services to the outside world ... as a result, the number of publicly-exposed APIs is growing exponentially.

These days, APIs are just about everywhere. They connect and power mobile applications, Internet of Things (IoT) sensors, cloud-based services, internal applications, partner platforms and more.

Researchers estimate that more than **83% of web traffic** now flows through APIs.[1] Some predict that there will be **more than 1.7 billion active APIs by 2030.**[2]

This rapid proliferation of APIs has been dubbed 'the API economy.' The sheer number of APIs continues to increase as organizations leverage them to improve efficiency, optimize resource use and create new opportunities for generating revenue within the digital ecosystem.

However, as the popularity of APIs surges and organizations become increasingly reliant upon them, API is raising new security concerns. In particular, the openness of APIs — along with the fact that they're capable of exposing sensitive and business-critical data — is significantly expanding the attack surface.

According to Gartner, 90% of web applications now have a larger attack surface exposed via APIs than through the user interface.[3] Gartner has also stated that API abuse is now the most frequently-exploited attack vector in data breaches involving enterprise web applications. Risk analysts estimate that U.S. companies will face combined annual losses from API-related compromises totaling between $12 and $23 million.[4]

90%

Web apps with exposed APIs

$12–$23M

Annual losses from API-related compromises

Securing APIs is a core aspect of web application security, but it isn't always easy. It requires a different approach than traditional network security, and isn't identical to the management and testing needed for general web application security. Its importance isn't always obvious to business leaders, in large part because the proliferation of APIs has occurred so quickly.

1. Akamai, State of the Internet / Security Report, Volume 5, Issue 2: Retail Attacks and API Traffic.
2. f5, Office of the CTO Report, Continuous API Sprawl: Challenges and Opportunities in an API-Driven Economy.
3. Gartner Research, Gartner's API Strategy Maturity Model, October 2019.
4. Imperva, Quantifying the Costs of API Insecurity: Report.

Major recent API-related data breaches

Because APIs are omnipresent, just about any organization, large or small, can suffer an API breach. To exemplify the risk, here are a few API-related data breaches that made headlines of late.

Experian

An incorrectly implemented authentication mechanism on a partner's website made it possible for anyone to look up the credit scores of millions of Americans just by entering their name and mailing address. Security researchers were also able to obtain risk factors within a person's credit history that explained the reason for their scores. Experian says that it quickly fixed the unprotected endpoint, but it's unknown how many people's financial information might have been accessed by bad actors.[5]

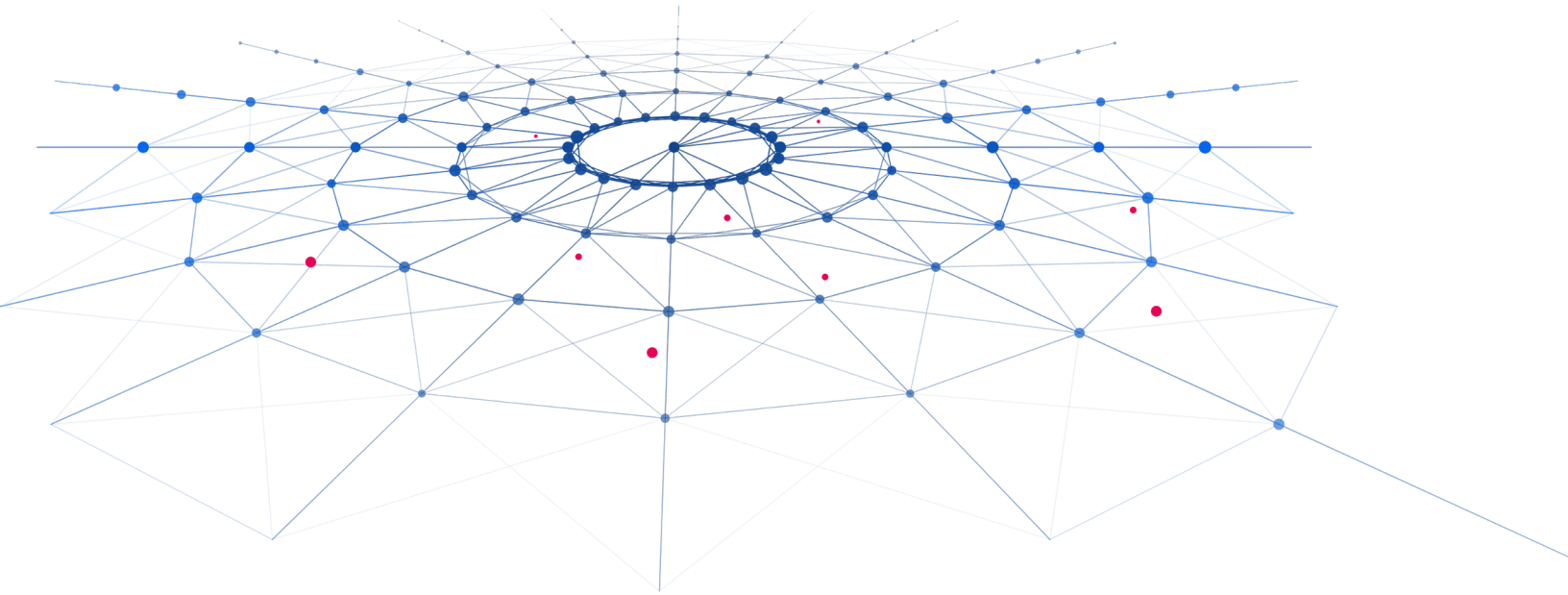
Peloton

A similar vulnerability in the at-home fitness brand's subscription-based app allowed unauthorized parties to request user account data from Peloton's servers. Security researchers were able to access Peloton users' age, gender, weight, city of residence and workout history without the users' permission. Peloton repaired the vulnerability soon after its details were publicized.[6]

LinkedIn

Records from nearly all of the professional social networking site's users — over one billion records in total — were obtained by attackers who later offered them for sale on an underground hacking forum. The perpetrator leveraged an improperly secured API to extract the information, all of which users had been voluntarily published onto LinkedIn.[7]

5. Krebs on Security, "Experian API Exposed Credit Scores of Most Americans," April 2021.
6. TechCrunch, "Peloton's leaky API let anyone grab riders' private account data," May 2021.
7. ThreatPost, "LinkedIn's 1.2B Data-Scrape Victims Already Being Targeted by Attackers," July 2021.



API security: offensive, defensive and everything in between

APIs are proliferating so rapidly because their use provides both technical and business benefits. When unchecked, however, API sprawl can create significant management and security challenges.

Securing APIs demands collaboration between security and development teams. It's necessary to take a multi-layered approach that involves discovery, cataloging and inventory tracking (because you can't protect what you don't know about). Web application firewall (WAF) rules can also be set to block requests from known-bad IP addresses, which can prevent some bot and distributed denial-of-service (DDos) attacks (in which APIs are flooded with an excess of requests in an attempt to slow down or halt service).

These strategies cannot protect against all malicious requests that exploit built-in vulnerabilities, however. Instead, it's essential to take a secure-by-design approach, adopting what Gartner has described as a "continuous approach to API security across the API development and delivery cycle, designing security directly into APIs."[8]

To achieve this, it's imperative to prioritize secure coding practices. Despite the "shift left" that's been discussed for years now, many development teams are still pushing vulnerable code, often because they hold to a "push first, patch second" mentality. According to recent research, 49% of development organizations regularly push vulnerable code simply because they believe the risks of doing so are low.[9]

8. Gartner Research, API Security: What You Need to Do to Protect Your APIs, August 2019.
9. Enterprise Strategy Group, Survey Report: Modern Application Development Security.

Instead, it's critical to test for security flaws in web applications (including in APIs) earlier in the development process. This approach can improve security while lowering costs and speeding time to production. Early testing also enables developers to make informed decisions about which functionalities to include, what the risks are of doing so, and how best to avoid those risks.

An API security testing strategy needs to be built upon a foundational understanding of the differences between web application testing and API security testing. Classical web application security has mainly concerned itself with threats like injection attacks, cross-site scripting and buffer overflows, but API breaches typically occur through authorization and authentication issues, which lead to unintended access to business logic or data.

API gateways and WAFs do provide capabilities for managing traffic, authentication and authorization, but gaps remain. According to one recent survey, 83% of security and development professionals believe that their existing API gateways and WAFs aren't very effective in preventing API attacks.[10]

Web application testing also isn't enough. This type of testing typically covers only the API calls made by the application, but APIs may have a great deal of additional functionality beyond this. Plus, web application testing doesn't cover headless APIs – those without an accompanying web application and graphical user interface (GUI). And automated solutions like scanners only test for a small subset of the potential security issues and vulnerabilities that can impact APIs. Hence, a testing approach that's more creative, far-reaching and thorough is called for.

Human expertise can make the API security testing process into something that's far more nuanced and complete than automated solutions can. For this reason, penetration testing performed by security researchers with a deep understanding of API logic and functionality should be a critical element of your API delivery cycle. This is essential if you want to maintain a strong API security posture on an ongoing basis.

Top API security flaws you should be testing for

- **Broken Object Level Identification:** According to the Open Web Application Security Project (OWASP), this is the most common API threat, targeted in about 40% of API attacks today.[11] Attackers are able to access data that they shouldn't have access to by sending an object identifier to the API, which will then return the requested data.
- **Broken User Authentication:** If authentication isn't set up properly, attackers can impersonate legitimate API users, enabling them to access confidential data. Broken user authentication makes it possible for attackers to use stolen authentication tokens, credential stuffing and/or brute force attacks to gain unauthorized access to applications and data.
- **Excessive Data Exposure:** Many APIs are built to err on the side of exposing too much data, counting on their users to filter the data properly. If the client is expected to filter out sensitive data, it could be read while in transit, exploited by a malicious API client or revealed by accident.
- **Security Misconfigurations:** If mistakes are made when setting up the API, HTTP headers might be misconfigured, HTTP methods might be incorrect or error messages containing sensitive information might be returned. Such error messages can provide attackers with information that will be useful to them in planning exploits that will target additional parts of the system.
- **Injection:** In injection attacks, attackers send specialized commands to the API that trick it into revealing data or performing some other unexpected action. The vulnerability is the result of a lack of validation of user input; instead, protection mechanisms should be in place to ensure that malicious commands cannot be executed by the API.

10. Salt, State of API Security, 2022.
11. OWASP API Security Project, API Security Top 10, 2019 version.

Hardening your API attack surface with pentesting

Many different types of APIs are in widespread use today. Securing them all requires a diverse array of access control rules and configurations. To find vulnerabilities amidst this complexity, researchers need to adopt a true adversarial perspective. This means testing APIs methodically, one endpoint and method at a time, while relying on API-specific background knowledge to accurately interpret the results. No matter how many certifications a practitioner has, or how many tools they know how to use, there will always be exploits that they haven't considered.

This is why diversity — of background and thought — is so important in API security testing. Bringing together a breadth of perspectives is essential to supply the right balance of rigor and creativity. Hence, a community-based approach, in which sizable groups of ethical hackers band together to pool their expertise, is a particularly good fit for testing for vulnerabilities in APIs.

Elite researchers with proven expertise will have the right background to catch the vulnerabilities that are most likely to be exploited by a real-world attacker. They're also least likely to find false positives. And they can explain their findings thoroughly, with the right level of technical detail to ensure that vulnerabilities can be remediated quickly and accurately.

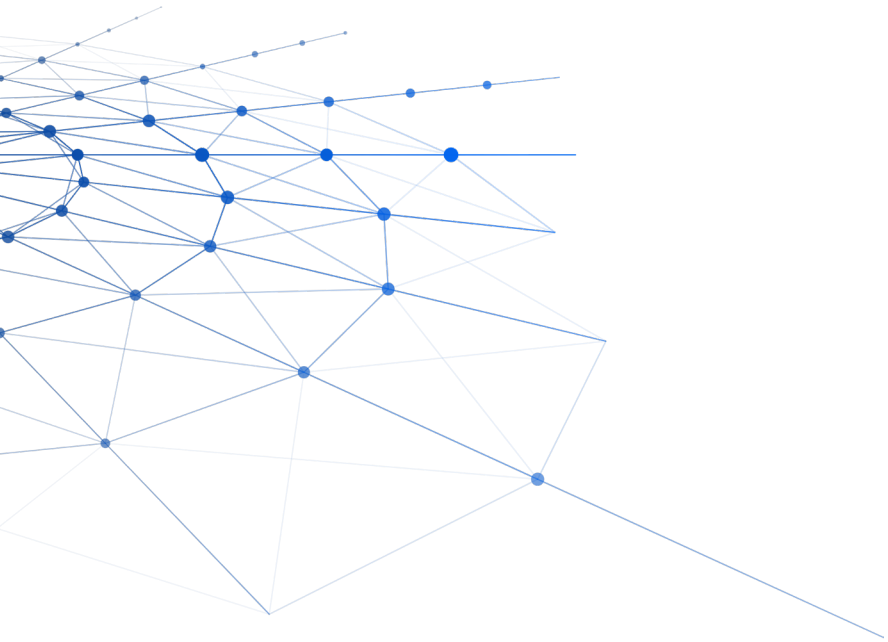
Actionable results

Things that should be included in all API security test findings reports

- Proof-of-concept describing the vulnerability and its exploit path
- Instructions on how to patch/remediate the vulnerability
- Plain-language descriptions that are easily understood by business stakeholders and auditors alike

Extras that are nice to have

- The ability to efficiently address findings within a single, centralized client portal, including the ability to verify patches
- Integrations with your existing vulnerability management process or DevOps toolchain for rapid remediation



How Synack can help

Synack's premier security testing platform provides continuous, on-demand access to a diverse global community of ethical hackers. Over 1,500 researchers from around the world are currently on-platform, bringing the varied backgrounds and skill sets needed for truly effective API security testing.

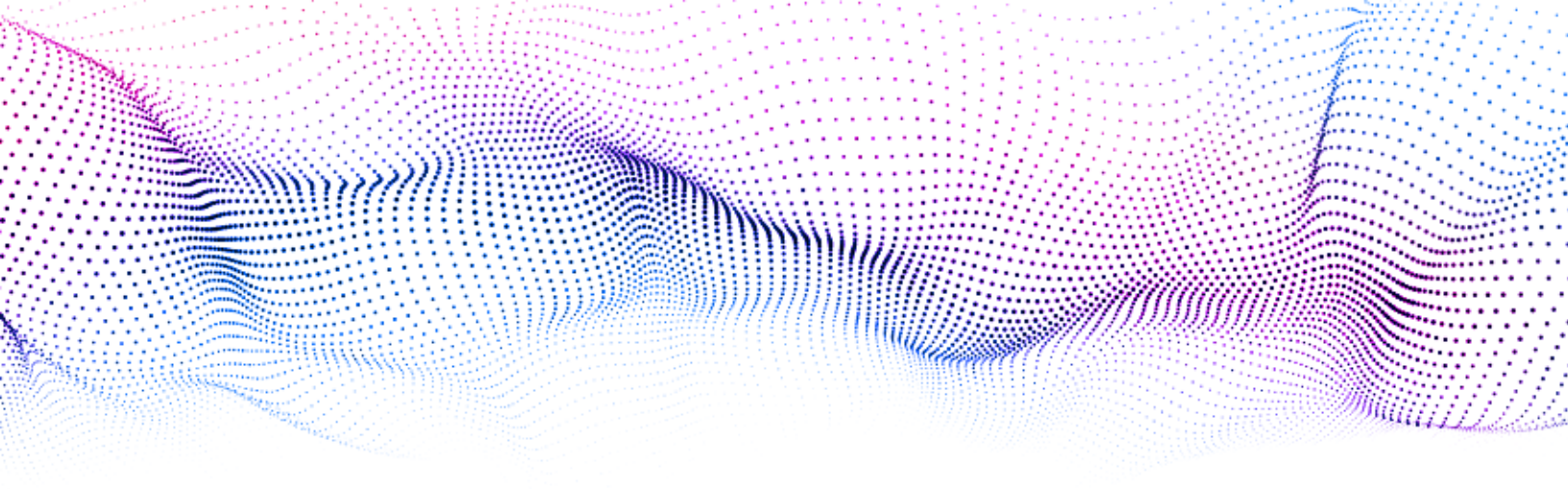
We take great care to ensure that our security testing community is comprised of elite talent you can trust. Applicants go through a rigorous five-step vetting process when they seek to become part of the Synack Red Team community. Each successful candidate brings a different employment history, set of reconnaissance skills, and certification and educational background to the task of API security testing. They're familiar with the OWASP Top 10 and other industry standard frameworks. And their abilities go far beyond what automated solutions like API scanners, firewall solutions and traffic monitoring can achieve.

In a world where cybersecurity talent is scarce and hiring managers struggle to find the advanced skills that are relevant to their organization's unique environment, we're able to bridge the gaps.

Results are reported within a platform where they're easy to access and understand. This simplifies remediation. It's also perfectly suited to demonstrate compliance to auditors and other stakeholders.

Our testing is available continuously and on-demand.

This means that we can retest almost instantly, whenever your APIs are updated. This makes it possible to balance proactive and reactive approaches to vulnerability management to achieve a stronger overall security posture.



Next Steps

As APIs continue to proliferate, the number of API-related vulnerabilities in today's organizational attack surfaces and the number of skilled security researchers available to test them have not grown in parallel. Synack can give your security team on-demand access to an ethical hacker community with verified API skills to test and propose remediation for your API attack surface.

Interested in learning more? Contact a member of our team to set up a demo today.