noname

Shift Left API Security Testing for Healthcare Institutions

Application programming interfaces, or APIs, enable healthcare institutions to seamlessly share data throughout the continuum of care. When you think about the proliferation of data across healthcare systems, electronic health records (EHRs), and internal applications, APIs provide the necessary interoperability for data to be shared. Similar to how retailers create personalized experiences by integrating location, inventory, and payment data, healthcare institutions also create contextual experiences for patients and staff by leveraging APIs.

The impact of APIs can be seen across facets of healthcare, including pharmacies, insurance providers, payers, hospitals, laboratories, local clinics, and home devices. With the seamless connection between datasets, institutions are able to enhance the patient experience and streamline care. How exactly? APIs reduce the workload and redundant operations performed by developers by allowing them to access data stored elsewhere. This enables them to quickly execute on tasks and take on more important projects of greater impact.

APIs empower patients to easily obtain and share data, as well as consume and shift between different digital services. APIs also allow healthcare providers to connect and engage with patients in totally new ways. Over the years, providers have introduced new web-based and mobile platforms to communicate with patients, answer questions, and share data. For instance, APIs can route data about medications, doctor visits, and clinical trials via a provider's mobile application.

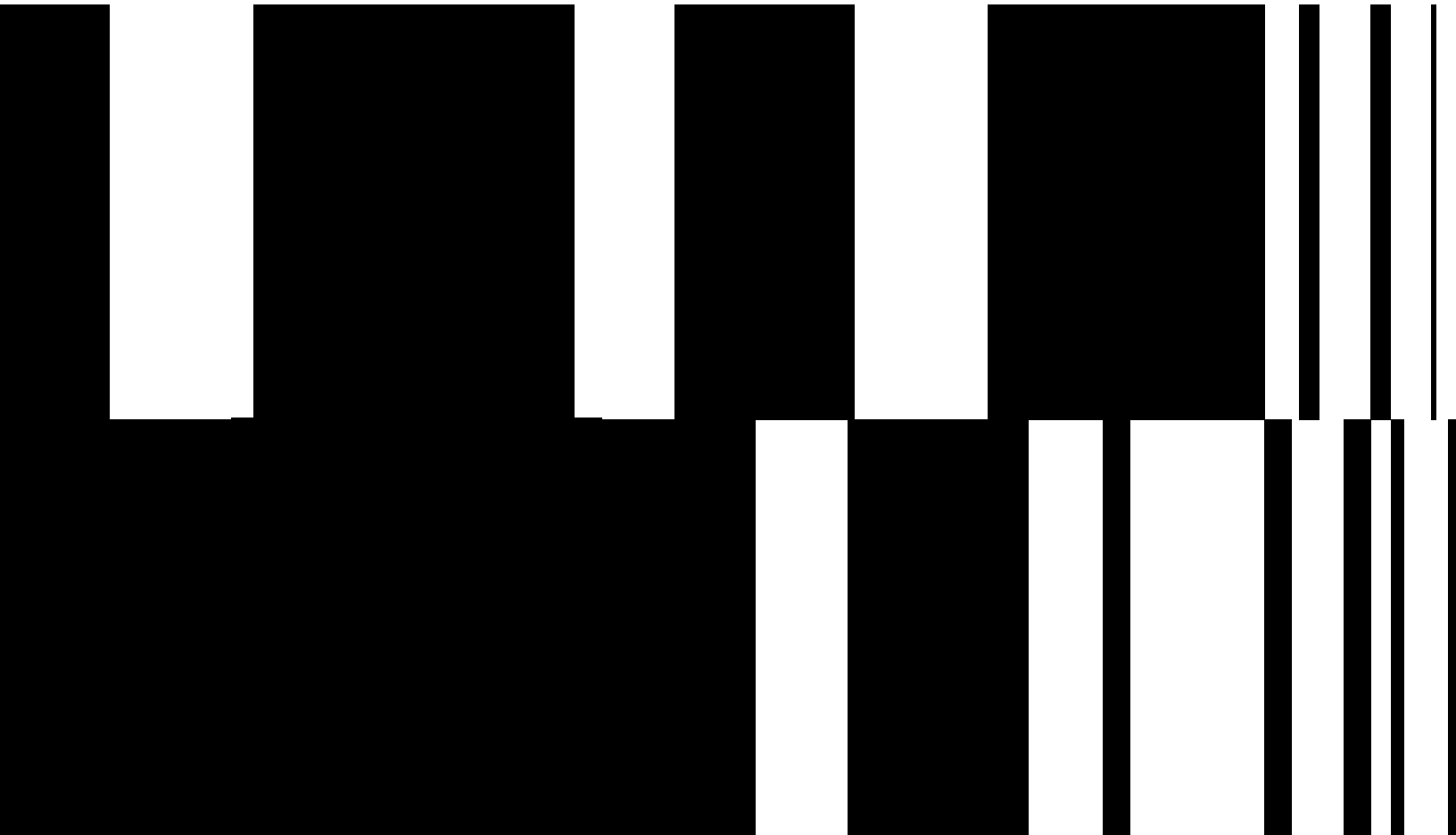
One key to this level of interoperability within the healthcare ecosystem is the Fast Healthcare Interoperability Resource (FHIR) data standard. In a nutshell, FHIR is a method for presenting and sharing data between healthcare clinicians and institutions in a standard way. This is regardless of how their EHR system may represent or store the data. FHIR is arguably the foundation for how data flows throughout the care ecosystem.

Despite the myriad of benefits APIs present to healthcare organizations, APIs also expand the attack surface. And with the number of APIs skyrocketing, institutions are facing increasing challenges when it comes to security. The reality is, existing security tools can’t address this on their own. This calls for new API security solutions and best practices to identify and remediate vulnerabilities before they’re discovered by malicious actors. One of the most notable best practices involves API security testing.

As documented by the OWASP API Security Top 10, many of the most prevalent vulnerabilities and design flaws can be addressed with API security testing. The problem is, oftentimes, either there aren't enough security personnel who know how to test APIs, the number of APIs are growing faster than the security team can keep up with, or the existing security tools lack adequate coverage. Any one of these three scenarios can spell disaster for your environment. Additionally, another overlooked aspect that could also weaken your API security posture even if the aforementioned issues are addressed is the timing of API testing.

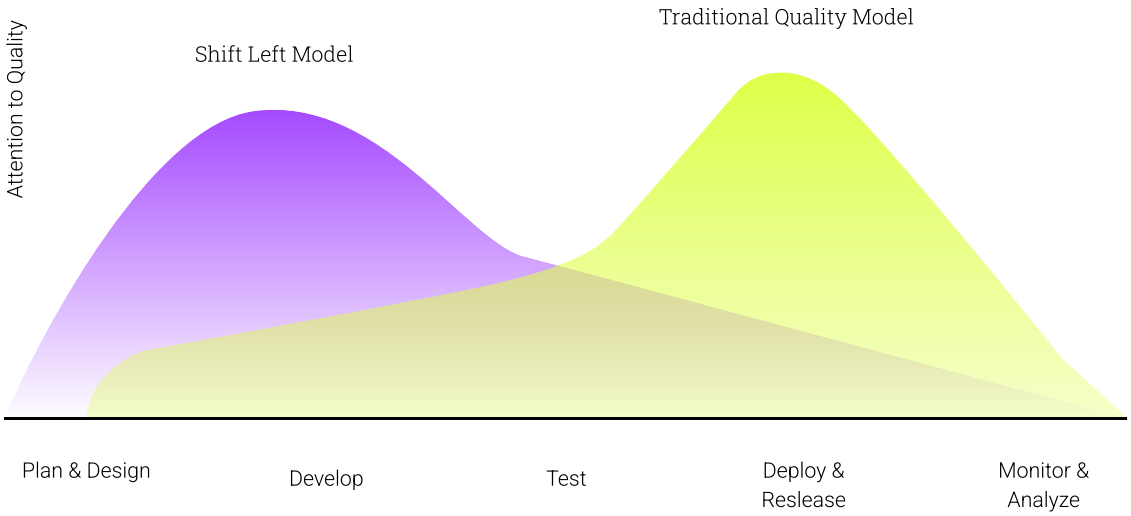
When it comes to API development, it’s not just a matter of testing for security gaps but also when you test your APIs. The traditional model places testing closer to the deployment phase. Although a vital step, testing only during this time is insufficient and can lead to serious vulnerabilities. How exactly? By consolidating testing into one phase of the software development lifecycle (SDLC), you create a bottleneck in the process as there is a never-ending supply of code to test.

To alleviate this bottleneck and expedite the process, corners may get cut and certain steps in this evaluation process may be incomplete or mistakenly bypassed altogether. As a result, code quality suffers and your API attack surface widens. And remember, this is the one phase in the process for testing - so anything missed here won’t be caught until it's too late.



What is Shift Left Testing?

Shift Left is an approach of moving a variety of tasks earlier in the development process. This means that tasks traditionally done at a later stage of the operation should instead be performed in earlier stages–particularly those related to API security and software testing.



With security and testing baked into each step of the API development or DevOps process, a shift left approach ensures developers will be monitoring for vulnerabilities throughout the lifecycle. Shift left principles enable security teams to increase developer autonomy by providing support, expertise, and tooling while still delivering the required level of oversight. Developers can release more secure code at scale, build API security into the design, and make fixes early in the development process instead of scrambling to fix them later. Code testers are able to evaluate features as they are created and help ensure higher quality.

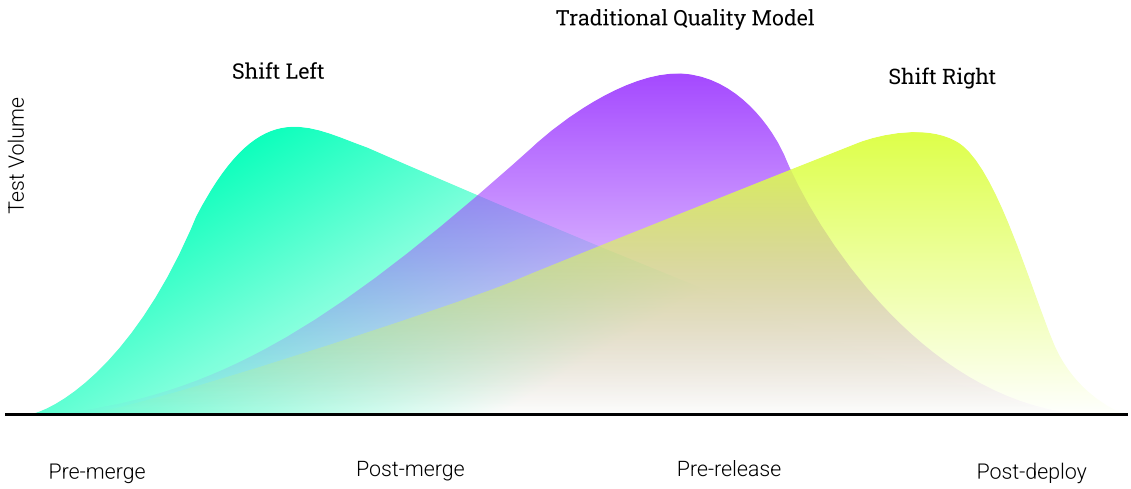
The shift left testing process is continuous, running parallel with development, and involves continuous communication between the clients, developers, and testers. The shift left testing process typically involves several steps:

1. Studying client requirements, application behavior, and end-user expectations
2. Developing tests for unit, integration, and functionality
3. Executing tests via end-to-end automation

The practice also helps minimize defects along the way by encouraging both Test-Driven Development (TDD) and Behavior-Driven Development (BDD).

Shift Left vs Shift Right: Why Shift Left Testing?

As we just established, a shift left security approach moves testing to the left on the timeline, so the team performs tests earlier and more often in the life cycle. In contrast, a shift right approach considers testing in production with real users to be more useful.



The goals of shift right testing include:

- Verifying backend stability
- Establishing software usability via actual user preference such as Blue Green deployments or A/B testing
- Identifying real production issues when end users experience them

Shift right testing ensures real world software stability and performance by testing in production environments, with scenarios that aren't possible in the development environment, and improves user experience by collecting feedback and reviews from application users. Though this approach can improve some aspects of performance, it can leave you vulnerable since you won't be able to catch misconfigurations and design flaws sooner.

Shift left testing saves time by identifying bugs earlier in the life cycle. More rigorous detection of errors and bottlenecks in advance enables testers to improve initial designs versus trying to remediate errors post production – which can be very time consuming. The shift left approach also ensures quality by allowing the development team to bake automation tools like API testing or unit testing right into the building process. Ultimately, apps, microservices, and APIs are more sufficiently protected. Waiting to test during production means the team is always playing catch up and facing inherently greater risk with potentially misconfigured APIs.

How to Implement Shift Left Testing

There are some basic things to keep in mind when implementing shift left security and testing.

Define goals

Since shift left demands organizational and cultural change, management should first define goals for the process to ensure any new tool or process introduced into the development cycle will work for the team's existing development and testing methodologies.

Understand the supply chain

Know how and where your agency develops apps and software before architecting a comprehensive shift left security program. The security risk posture of the supply chain is largely dependent on the security proficiency of others in the chain. This also helps your developers identify small steps where testing might be placed earlier in the life cycle.

Automate security processes with security automation tools

Use continuous integration (CI) tools, issue tracking tools, and test automation tools to help teams establish and automate security practices during all stages of the life cycle.

Train development teams in coding securely and visible culture

Do not neglect the human aspects of risk during the move to shift left security. Constant visibility into application security should be part of the culture.

Value of Shift Left API Testing

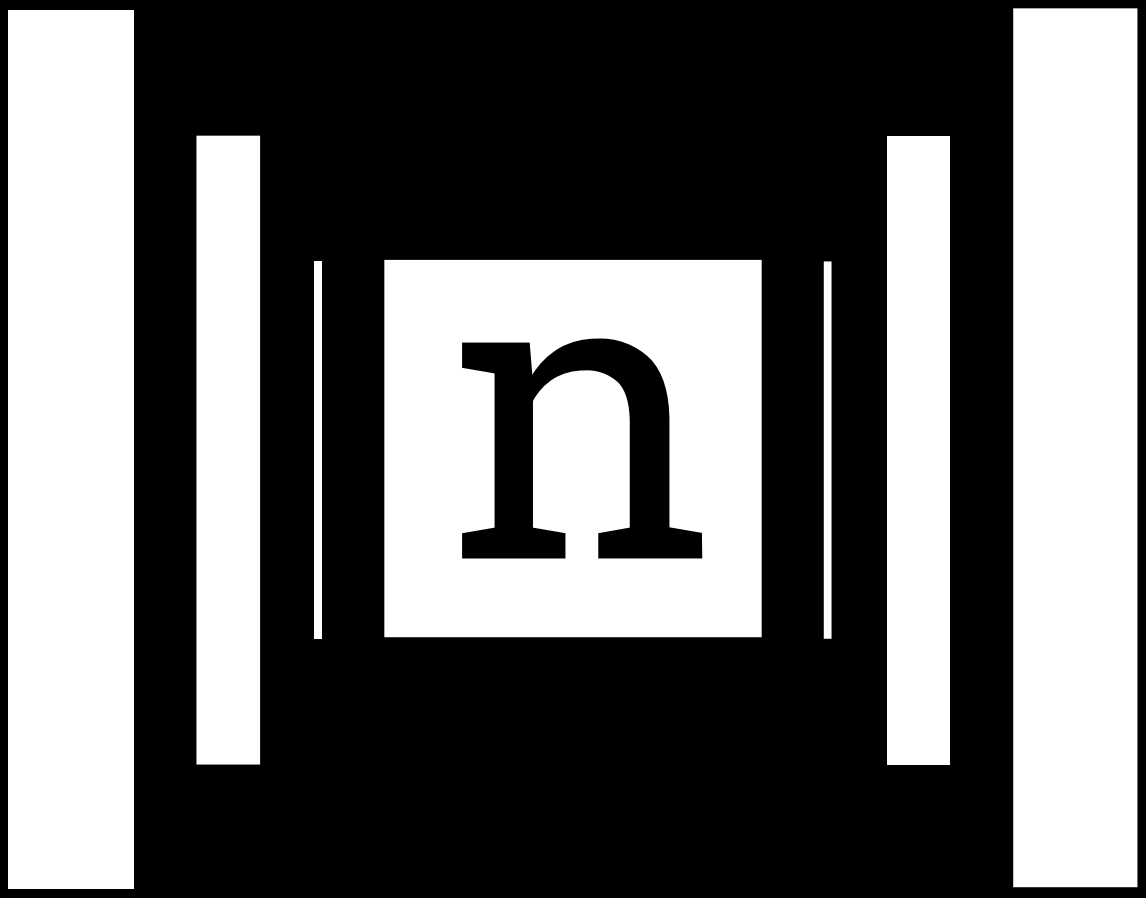
Shifting testing earlier in the development process offers a myriad of benefits for developers. These benefits can be summarized in two distinct outcomes: fixing vulnerabilities before production and innovating faster.

Address vulnerabilities before production

As discussed earlier in this document, testing APIs early and often shrinks the API attack surface and reduces the risk of successful attacks in production. Now you can minimize the potential for dataleaks and manipulation of government APIs without any modifications to production infrastructure. By finding and fixing issues earlier, not only can you remediate faster, but you also can lower remediation costs by up to 100x. This includes improving compliance, and avoiding regulatory fines and reputational damage from incidents.

Innovate Faster

By testing APIs early, you're also able to improve security while increasing velocity. A shift left approach empowers you to eliminate the bottlenecks identified earlier. You can increase your agency's confidence in APIs with continuous testing and reduce redundant pentesting and other third-party security testing costs. Ultimately, you are able to deliver secure code without having to become a security expert.



About Noname Security

Noname Security is the only company taking a complete, proactive approach to API Security. Noname works with 20% of the Fortune 500 and covers the entire API security scope across four pillars — Discovery, Posture Management, Runtime Security, and API Security Testing. Noname Security is privately held, remote-first with headquarters in Silicon Valley, California, and offices in London.

nonamesecurity.com info@nonamesecurity.com +1 (415) 993-7371