noname

The 3 Phases of API Testing

The 3 Phases of API Testing

API security has become top of mind for CISOs ever since Gartner predicted API attacks would become the most-frequent attack vector by 2022. In the scramble that ensued to get protected, a number of best practices emerged to help organizations with their API security posture. One of the most notable being API testing. Though a leading tactic within the developer community, API testing has become somewhat of a nebulous concept as oftentimes, very little detail is provided about what that exactly entails. With that said, this document will outline the three phases of API testing and key considerations to successfully identify vulnerabilities.

1 Shift Left Testing Phase 1

Understanding the API

Before adequately assessing the state of API security, you need to understand its purpose, value to the business, and other factors that categorize the risks to the business for this API. You also need to note what data the API consumes and provides according to how your business classifies data. For example, an API that handles credit card and health data is subject to more regulations and potential risk versus one that provides public data like the current weather. Understanding the use-case for the API informs testing, especially for tricky items like business logic issues.

If you're going to perform active API testing, you'll also need some sort of documentation about what methods are exposed by the API and how to call them. Unfortunately, many APIs lack sufficient documentation, especially internal APIs, and often the tested API does not match the available documentation. This usually occurs when the API is deployed faster than the documentation is updated. Occasionally the API you're testing is a legacy or forgotten API and will have no documentation. Nevertheless, the documentation should provide the required data sent to the API and expected data in its responses. Having the expected sent and received data can help drive the optimal 'abuse-cases' that should be included in the testing efforts.

For the fortunate among us, the API will have Swagger/OpenAPI specifications as its documentation. Swagger/OpenAPI are a standard way to describe REST APIs, with OpenAPI being the most recent version. We'll use OpenAPI to mean both for simplicity. If you happen to have an API-aware security tool, it likely needs the OpenAPI file(s) to understand how to communicate with the API. While OpenAPI specs are great for accelerating testing efforts, they can hurt them when they are out of date, don't match the version of the API being tested, or are otherwise inaccurate. It is difficult to know if what you think the API does, e.g., the OpenAPI file, really matches how the API really works.

For the genuinely desperate security professional, there is always making friends with the quality team (QA/QE) and hoping that you can borrow their functional test suites and morph them to also be used as API testing tools.

2 Shift Left Testing Phase 2

Ensure you can interact with the API correctly

After understanding the API, you can start interacting with it. However, it's important not to jump straight to API security testing. Instead, make sure you can use the API as it was intended. This may seem counterintuitive, but it's essential to validate that your understanding of the API matches how the API works. For example, the 'normal' API calls allow you to validate the accuracy of the documentation or OpenAPI spec file. It is also crucially important, if you're manually API testing, to have examples of what routine requests and responses look like—knowing what normal looks like allows you to gauge the effects of attack-simulating tests against the APIs.

In cases where creating valid requests to an API is significantly difficult, there is a trick you can use. This trick only works for APIs with client tools designed to make API calls like kubectl for Kubernetes. By putting a local HTTP proxy like OWASP Zap or Burpsuite upstream of the client tool, you can capture the calls it makes to the API and get valid examples of how the API works.

3 Shift Left Testing Phase 3

Start sending attack traffic to the API

Now that you understand the API and know how to communicate with it in a correct fashion, it is time to start sending attack traffic to the API. This could be manually manipulating the requests to the API, inserting fuzzing strings into requests, or automated by an API security testing tool.

There are several questions you want to ask yourself about whatever type of API testing you decide to use for this phase:

- ✓ Do you have full coverage? Does the tool do more than the OWASP Top 10? Is that the OWASP Top 10 for applications or the OWASP API Top 10?
- ✓ How well does your choice 'know' API testing versus traditional web testing?
- ✓ What does the tool/DAST scanner miss? What is the false positive rate shown by real testing of _your_ APIs?
- ✓ Does the tool understand Swagger/ OpenAPI spec files? Do you have these for the APIs that are being tested? Are those spec files accurate and up to date?
- ✓ If you want to manually test APIs, do you have staff familiar with API testing? Do you have enough staff to test APIs for the number of APIs you have?

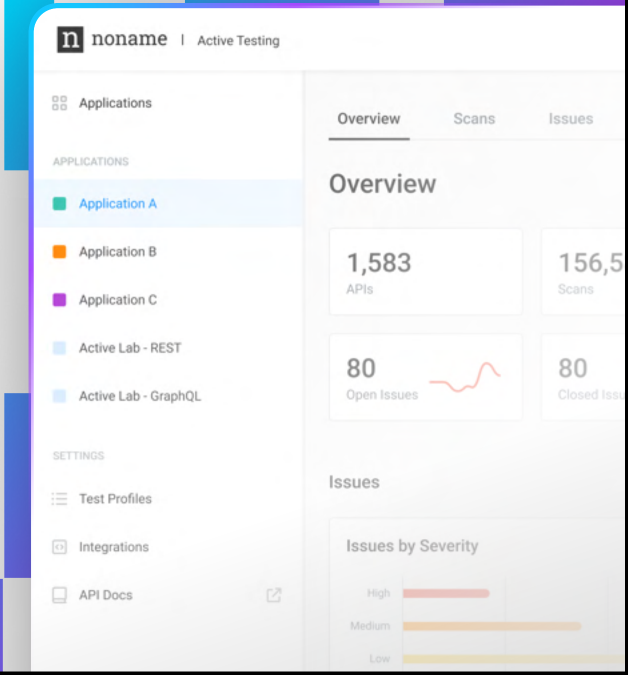
One option for API Testing

The Noname API Security Platform helps enterprises take a shift left security approach by addressing many of the issues listed above and many more. Here a just a few of the benefits you can expect:

- Inventory of all your APIs, including the data received and sent with autoclassification of sensitive data types.
- Define business-specific data types for the platform to find.
- Enable runtime/real-time security for your APIs by using anomaly detection and reacting to attack or abnormal traffic.
- Automatically active test APIs with an existing understanding of how the API communicates.
- Passively observe API traffic to:
  - Find misconfigurations and API security vulnerabilities
  - Alert on sensitive data in API traffic based on policies you write.
  - Understand how your APIs work and provide Swagger files representing what is really in network traffic.
  - Allow you to compare the Swagger spec it creates against the one from your dev teams/documentation.
  - Group and categorize APIs in a ton of different ways including userconfigurable ones.
  - Provide an assessment of an APIs security posture.

To find out more and see how your company can get started with Shift Left API Testing

Book a demo →



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