API Testing Methodology

Jyoti Raval

harness

Agenda

- Introduction
- What is API?
- Why API testing?
- How API testing works?
- Harness API testing strategy
- Automation & Scaling

About Me

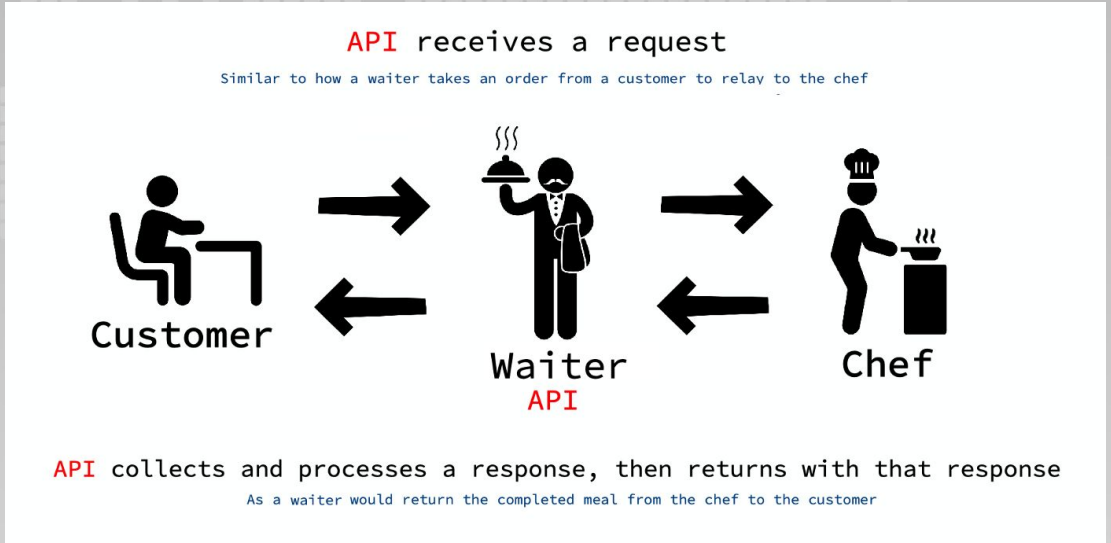
- I am Jyoti Raval, working as Staff Product Security Engineer with 

- Responsible for researching on new security trends and help secure product end-to-end
- Application security enthusiast
- Presented at BlackHat,DefCon,HITB,NullCon,InfosecGirls and OWASP.
- Author of tool - Phishing Simulation Assessment & Managing Pentest - MPT
- OWASP Pune chapter leader and goes by jenyraval on github

What is an API?

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other. Each time you use an app like Facebook, send an instant message, or check the weather on your phone, you're using an API.

Example:When you use an application on your mobile phone, the application connects to the Internet and sends data to a server. The server then retrieves that data, interprets it, performs the necessary actions and sends it back to your phone. The application then interprets that data and presents you with the information you wanted in a readable way. This is what an API is - all of this happens via API.



What is the origin?

APIs continue to grow in number, serving as a fundamental part of modern software development across industries.This increase in API usage is due in part to the standards that organisations have developed to encourage API adoption. Specifications like OpenAPI/swagger help define the files required to describe APIs. These files then help organisations identify the necessary documentation, integrations, and testing tools to effectively manage their APIs.

Why API testing?

APIs are the heart of many applications, providing developers with powerful interfaces to the services an organisation has to offer. Ensuring that APIs are conformant to published specifications and resilient to bad and potentially malicious input is critical to an organisation’s overall security.

Why now?

Salt Labs, the research division of Salt Security, released its Q1 2022 State of API Security report indicating that API attacks more than doubled the overall growth in API traffic in the last 12 months.

According to the report, 95% of organisations running production APIs have experienced an API security incident in the last 12 months.

However, most organisations are unprepared to handle these challenges, with over a third (34%) having no API security strategy.

Similarly, traditional security measures continue to fail, giving organisations a false sense of security.

How API testing works?

Security testing helps ensure that basic security requirements have been met, including the conditions of user access, encryption, and authentication concerns. The idea behind API scanning is to craft inputs to coax bugs and undefined behaviour out of an API, essentially mimicking the actions and attack vectors of would-be hackers.

High level Steps:

- Define the APIs that needs to be tested
- Testers provide information on inputs and outputs of the API, using a variety of specification formats including OpenAPI v2 / v3, Postman Collections, and HAR files.
- API security tests use this information to construct fuzzed input tailored to the input the API expects.
- The output of API security testing is a report of any vulnerabilities or bugs found while fuzzing the API. This could include findings such as SQL and OS command injections, authorisation/authentication bypasses, path traversal issues, and OWASP Top 10 API vulnerabilities including broken auth, security misconfiguration, and data exposure.

Harness API testing strategy

Scoping & Understanding APIs and Specifications:

1.Defining the Scope

2. Understand the APIs and business use

3. API as a contract — first, check the spec!

Mapping Attacks:

4. Mind-map the attacks

Automated + Manual Test - OWASP API Top 10:

5. Run an automated tool

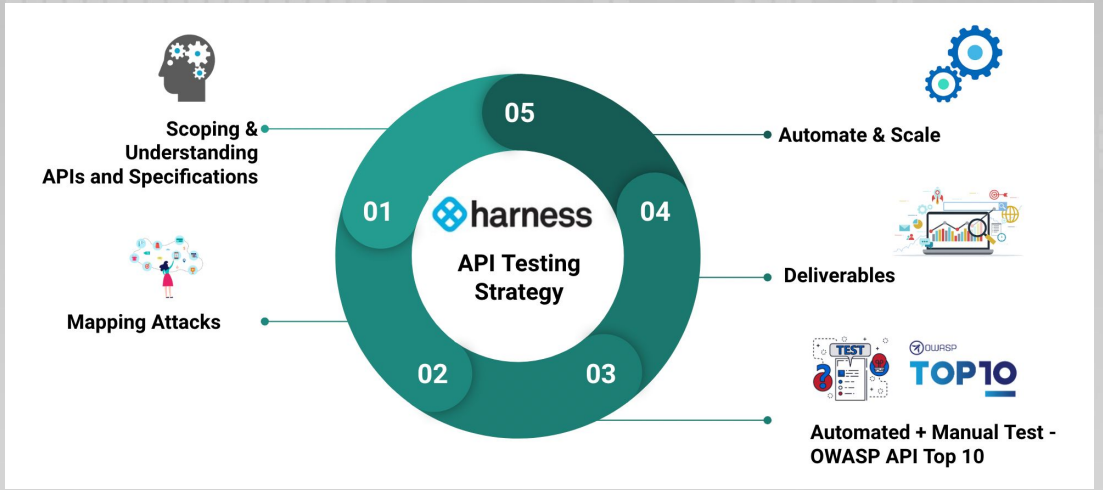
6. OWASP API top 10

7. Test business logic flaws

Deliverables:

8. Report

9. Automate and scale



Harness API testing strategy

1. Defining the scope:

It is important to define first, what all areas we would be covering as a part of testing.

2. Understand the APIs and business use:

Understanding APIs and their business use is of paramount importance in order to understand the amount of impact security loopholes would have on business. Also helps us building tailor made attacks on APIs. It is important to think like an attacker.

APIs expand the capabilities and functionalities business can offer, without putting a ton of resources behind integrations. To get the most out of APIs, they use APIs in the following ways:

- Integrate with third-party APIs
- Build APIs for internal use
- Build APIs and expose APIs for external use

Understand among which of the category above current testing is falling unto.

Harness API testing strategy

3. API as a contract — first, check the spec!

An API is essentially a contract between the client and the server or between two applications. Before any implementation test can begin, it is important to make sure that the contract is correct. That can be done first by inspecting the spec (or the service contract itself, for example a Swagger interface or OpenAPI reference) and making sure that endpoints are correctly named, that resources and their types correctly reflect the object model, that there is no missing functionality or duplicate functionality, and that relationships between resources are reflected in the API correctly.

4. Mind-map the attacks:

Now that you know why APIs is designed for and what is underlying technology and specification it is so much more easier to mind map the attacks.

All the low hanging fruits would be covered by automated tools generally. What you really want to focus on here is how business logic flows can be covered and how to focus on business specific flows and try and find the scenario which is generally very difficult for any security tool to find.

You can also focus on chained attacks here. Taking what tools reported as baseline and associating it with other findings to create bigger impacting attack scenario could be a focus here.

Harness API testing strategy

5. Run an automated tool:

Here you can run the tool of your choice for which you have done POC thoroughly.

Though there are various tools, we started with BurpSuite Pro, Traceable AI which you can use as well to find security issues.

Additionally open source tools like Postman and/or SoapUI can be used as well.

Burp has extensive set of plug-ins which can be used to focus on various attacks. for eg: OpenAPI Parser

6. OWASP API top 10

7. Test business logic flows

Business logic vulnerabilities are flaws in the design and implementation of an API that allow an attacker to elicit unintended behaviour. This potentially enables attackers to manipulate legitimate functionality to achieve a malicious goal.

For example, Look at how certain permissions are given to resources, how licensing is being validated, how cost to customer is calculated if manipulation or flaw in above scenario exists then it's a CIA breach and could cost a revenue loss to business.

Harness API testing strategy

8.Report

9.Automate and Scale:

Why automate?

Considering the amount of rapid development happening around the year at any fast growing organisation, it is important for security team to cope up with the speed at which it is being developed.

Automation would be the way for us to scale.

How to automate?

There could be various ways to automate API testing and testing the change during each release.

It is not possible to test all the APIs during each of the release cycle. We will have to pick a frequency at which we would do complete API testing and rest of the releases we would be picking up just delta.

Putting a API testing in our DevOps pipeline making it DevSecOps.

My coordinates - LinkedIn | Twitter (@JenyRaval)

Thank you!