API Security Automation Development Cost

Whitepaper: API Security Automation Development Cost

APISec^{TM} - API Security Platform

https://apisec.ai

Intesar Shannan Mohammed, CTO

Abstract

This whitepaper aims to help organizations understand the true in-house development cost of API Security Automation. The scope of this is limited to just the development of security tests. What it doesn't cover is the entire program cost that includes running tests, prioritizing/reporting issues, validation/closing issues, maintenance costs, tools license costs, and test infrastructure costs.

Target Audience

CISO, CTO, AppSec, Engineering Leadership

API Assets

Public APIs and Private APIs powering Mobile, Web, & IoT applications.

API Proliferation

The rise of APIs

1. 83% of all web traffic is API calls - Akamai[1]

2. 40% surface area for attack is exposed as APIs rather than the UI for web-enabled applications - Gartner[2]

What's Fueling the Growth?

3. More and more companies are adopting the API-First approach for faster innovation and rapidly getting the products and services out of the door.

4. Many new successful opportunities have been created by aggregating APIs from different sources and targeting new customer segments. E.g. Uber[3] relied on third-party mapping, payment, and communication APIs to build their services fast.

5. Modern mobile and web application backends are all API-based. The entire application is exposed as an API and is visible to the internet.

Why is there a Security Gap?

6. Most developers don’t consider mobile/web backend APIs as a possible attack vector

7. Missing security tools. Legacy web security revolved around SQL/Command injections and browser weaknesses like (XSS, Cookies, CORS, Sessions, etc.)

8. API top exploits are around access-control flaws like business-logic, role-configuration, etc.

9. API attacks are more automated and bot orchestrated.

10. API accidental data exposure can send private data to an non-entitled client.

[1] "Akamai State of the Internet Security Report: Retailers Most ...." 27 Feb. 2019, http://www.akamai.com/us/en/about/news/press/2019-press/state-of-the-internet-security-retail-attacks-and-api-traffic.jsp. Accessed 9 Apr. 2020.

[2] "OWASP API Security Top 10 Official Definition." https://www.cybersecuriti.ai/wp-content/uploads/2020/04/Whitepaper_-1-API-Risk-Mitigation-Strategy.pdf. Accessed 9 Apr. 2020.

[3] "The Importance of APIs for Business - GlowTouch." https://www.glowtouch.com/importance-apis-business/. Accessed 10 Apr. 2020.

Key Considerations for API Security Testing

1. Security Standard

The first step is to get your security standard right. This could be the easiest part. Most AppSec teams can start with the OWASP API Security Top 10 list[4].

2. Coverage / Permutations

This is the hardest part. Getting your test permutations right requires intimate and up to date knowledge of the product. A typical small API with 100 endpoints requires 20X validations i.e. 2000 validations to properly cover OWASP API Top 10 categories. Out of which 60% are business-logic, and role-configuration validation and the rest are injections. Getting it not right means AppSec still risks breaches. This step can overwhelm any size AppSec team.

3. Data

Another challenging step is to get your test data right. This requires access to sample requests, responses, and validations, etc. The scope of the work requires the AppSec team to work with the developers who have built the APIs.

4. Remediation

Even though the development team is responsible for fixing vulnerabilities, the entire process of prioritizing vulnerabilities and chasing developers to fix it can easily take up the majority of AppSec time.

[4] "OWASP API Security Project." https://owasp.org/www-project-api-security/. Accessed 9 Apr. 2020.

Testing Methods

1. Manual Audits

Some companies take a periodic manual audit approach against mobile and web applications, where the interface is designed for users to navigate using clicks and data entry methods.

When it comes to APIs, there is no user interface to navigate. Instead these approaches restrict the audits to a few checkpoints.

2. Vulnerability Assessment[5]

This approach comes with a lot of limitations, as vulnerability scanning software only looks at your system based on past common vulnerabilities. So if you're conducting a vulnerability assessment, it's imperative that the software is up to date. However, this makes the vulnerability assessment software only as effective as the maintenance performed by the software vendor. The software itself isn't resistant to a breach and has the potential of coming with software engineering flaws.

3. Pentesting

Penetration tests go beyond security audits and vulnerability assessments by trying to breach applications just like a hacker. It uses both open source and commercial software. Most efforts focus on Injection attacks related to SQL, Command, etc.

4. Security Automation

This is the most comprehensive way of securing and fixing API flaws, writing your API security tests using CURL, Postman, and Python-based tools. Make sure the coverage is comprehensive against the OWASP API Security Top 10 list and integrated with CI/CD pipeline (DevSecOps). This approach will help you detect issues as new code is written and can block the check-ins that break the security validations.

[5] "Security Audits and Penetration Testing | Springboard Blog." 10 Jul. 2018, https://www.springboard.com/blog/security-audits-and-penetration-testing/. Accessed 9 Apr. 2020.

Automation COST

Assumptions:

1. Security resources cost on average $75 per hour.
2. Writing 6 tests a day or 1 test per 1.3 hour.

API Size	**Required Tests (20X)**	**Development Time @ 6 tests / day**	**Development Cost @ $75 / hr**
Small API 100 endpoints	2,000	2,600 hrs	$200,000
Medium API 250 endpoints	5,000	6,600 hrs	$500,000
Large API 500 endpoints	10,000	13,000 Hrs	1,000,000

Conclusions:

1. On average, it costs $100 per test.
2. Security Automation if done right provides the most comprehensive coverage
3. DevSecOps is the right approach to detect issues early in the development cycle
4. Other methods leave huge security gaps and increase breach risk

References:

https://www.springboard.com/blog/security-audits-and-penetration-testing/

Reviewers:

Faizel Lakhani, CEO APISec.ai