API Penetration Testing Methodology

Our API security testing, leveraging manual and automated testing methods, is designed to cover both traditional and modern APIs across various architectures. The methodology ensures comprehensive coverage including OWASP API Top 10 risks and many additional checks. Below is the high level methodology used for a comprehensive security review of APIs -

Understanding API Business Logic & Use Cases / Integrations

As a foundational step, we perform a detailed analysis of the API's business logic, use cases, and integration points. This includes reviewing how APIs interact with internal and external systems, data flows between services, and the functional role of each API endpoint. Gaining this context is critical to understanding the intended behaviour of the APIs, which in turn enables accurate identification of security risks that may arise from misuse, abuse, or misconfigurations.

Threat Modelling

Before initiating the actual assessment, a structured threat modelling exercise is conducted to map out potential attack vectors across the API landscape. This includes identifying key assets, trust boundaries, entry and exit points, data sensitivity levels, third-party integrations and user roles. The model considers Role-Based Access Control (RBAC) configurations to evaluate whether access permissions are properly enforced across different privilege levels. The objective is to anticipate exploitation paths and guide the assessment toward both technical vulnerabilities and business logic flaws that are specific to the application's architecture and usage patterns.

Security Control and Test Cases

1. **Authentication & Authorization Weaknesses -** This category focuses on verifying that authentication, authorization and session are implemented correctly. It includes checks like API key and token validation mechanisms, JWT validation and integrity, improper token expiration/invalidation, Role-based access flaws etc.
2. **Excessive Data Exposure & Mass Assignment** - This area assesses whether APIs are exposing more data than necessary or allowing unauthorized updates to object properties. The APIs might have over-exposed fields in responses, inclusion of internal data like internal IDs, flags, or configurations, mass assignment via PUT/PATCH requests to update restricted properties etc.
3. **Security Misconfigurations & Improper Inventory Management -** This category includes detection of misconfigurations and poorly maintained API inventory via exposure of development/debug endpoints, unlisted/undocumented endpoints accessible via fuzzing,

Copyright © Blueinfy

multiple versions (v1, v2, etc.) of the same API accessible, misconfigured HTTP headers, or verbose error messages etc.

4. **Injection & Parsing Vulnerabilities -** This category targets injection risks arising from user inputs or untrusted data covering all input validation risks like SQL, NoSQL, and command injections, header, XML, or JSON injection, deserialization and expression language injection, GraphQL-specific injection flaws, un-sanitized data reflected in response etc.
5. **Rate Limiting & Resource Exhaustion** - This focuses on the lack of proper controls to prevent abuse or DoS-like behaviour arising due to lack of rate limiting on critical endpoints or inefficient backend processing for large payloads or nested requests.

The list provided is not limited to but represents core test cases, but the high level methodology varies based on the logic/business use case of the services being tested.

Vulnerability Assessment

On the basis of resource attributes and control categories, a thorough vulnerability assessment is performed using human intelligence with the help of supportive semi-automatic and automatic tools. This detects the vulnerabilities residing in the applications, thus giving the actionable item list from application security standpoint.

Mitigation Strategies

Based on the identified vulnerabilities, weaknesses, and overall risk posture—along with the system architecture and industry best practices—a comprehensive mitigation plan is formulated. This includes a set of actionable, prioritized recommendations tailored to APIs, outlining the security measures required to effectively harden the environment.

Actionable Report with Zero False Positives

A key deliverable of the assessment is a highly actionable, well-structured report designed to drive immediate remediation. The report is curated to maintain zero false positives and includes the following critical components: -

- Executive Summary
- Description of Discovered Vulnerabilities
- Risk Rating (curated after business impact assessment and industry security standards like CVSS/CWE/CVE)
- Evidence of Vulnerabilities (screenshots, HTTP traffic, vulnerable parameter, exploit vector, tool results, reproduction steps etc.)
- Exploit Evidence of Vulnerabilities (if required)
- Mitigation Strategies and Defence Approaches (catered to help Developers)
- Report Readout and Guidance

Tools

Copyright © Blueinfy

Blueinfy uses its own tools along with open source tools and products during the assessment process. Blueinfy has its own tools and utilities for performing manual penetration testing. Some of these tools are available at https://www.blueinfy.com/tools.html

Copyright © Blueinfy