RESEARCH REPORT

OWASP API Security Top 10™
A Critical View

RESEARCH REPORT

OWASP API Security Top 10™
A Critical View

Table of Contents

A Word

Letter from the CEO

As we unveil our research report, I want to take a moment to reflect on our mission: safeguarding the digital landscape from API vulnerabilities. With the surge in API attacks, our dedication to equipping you with the tools to stay ahead of evolving threats has never been more vital.

Our analysis highlights a critical point: **APIs now stand as the primary target for cyber attacks.** Although security leaders are looking to integrate API security programs, unfortunately, our research reveals that organizations are still at the early stages of effectively protecting their APIs. This underscores the importance of fostering stronger collaboration between development and security teams to enhance our collective API risk management efforts.

We also acknowledge the significant contributions of OWASP in setting application security standards. Their efforts have been foundational to our industry. However, leveraging our unique attack data, this report introduces an alternative perspective that both challenges and complements the OWASP top 10. But with all the attack data we've seen, we think there's room to shake things up a bit. In this report, we're proposing some new ideas that might challenge the OWASP top 10.

This report is packed with insights from thousands of API scans we analyzed in 2023. It's not just about the numbers; it's about what we can learn from them to make things better.

We believe that, with the help of our community, we can beat these threats and build a safer environment.

Regards,

Tzvika Shneider

CEO & Co-Founder

APIs Are The Number #1 Attack Surface

In the dynamic landscape of cybersecurity, the surge in API security vulnerabilities has positioned it as the primary attack surface. The severity of these threats is not confined to theoretical constructs; real-world incidents that underscore the critical need for a comprehensive understanding and fortification of API security.

Due to its escalating frequency of breaches and infamous nature, API vulnerabilities got a special seat at OWASP's risk table, releasing a special top 10 list dedicated to APIs, separately from the famous Top 10 vulnerability list we all know. However, the latest updates from OWASP fail to encapsulate the full spectrum of contemporary risks. This discrepancy between the evolving threat landscape and the OWASP Top 10 API vulnerabilities latest update (2023) prompts the necessity for a more current and field-aligned analysis. Pynt's proprietary research steps into this breach, bridging the gap between the traditional frameworks and the rapidly transforming API security environment.

Shifting our focus to the proactive approach of 'shifting left' in API security unveils a paradigm where vulnerabilities become not just identifiable but significantly easier and cost-effective to rectify. In this report, we dissect and contrast the Pynt API security vulnerabilities against the backdrop of the OWASP Top 10, shedding light on crucial differences and providing insights into the effectiveness of early detection and remediation.

Research Methodology

API security is not just about scanning; it requires an intimate understanding of the application behind the APIs. Traditional tools, often reliant on the OpenAPI specification, lack the necessary depth, especially for intricate vulnerabilities like BOLA in OWASP's Top 10. This report is compiled with attack data we collected during 2023, from thousands of applications. We used our own technology and leveraged the following methodologies and data:

We scanned **7000+** **VULNERABILITIES** from **900+** **APPLICATIONS**

Types of Vulnerabilities scanned BY POPULARITY

1. Security Misconfiguration
2. Missing Authentication
3. Injection
4. Broken Object Level Authorization Vulnerability
5. Broken Authentication
6. Mass assignment
7. Broken Function Level Authorization Vulnerability
8. Unrestricted Resource Consumption
9. Server Side Request Forgery
10. Local File Access

Data collection methodology

In 2023, we gathered data from a sample of Pynt runs. This data comprised a list of verified API Security findings detected by Pynt. While some findings were confirmed within the dataset, in other instances, we engaged with our users to validate the authenticity of the findings.

Data analysis methodology

We utilized our proprietary technology, which presents a unique approach: we meticulously learn an application's normal behavior by analyzing API traffic, then crafts and executes highly focused attack scenarios. This mirrors the tactics of sophisticated attackers, ensuring precise vulnerability detection with little to zero false positives. This approach is ideal for fast-paced development and CI/CD environments where accuracy and efficiency are critical.

Pynt's attack methodology

STEP #1 Pynt first analyzes API traffic to understand the structure of the APIs, the roles of each endpoint and parameter, and the relationship between different endpoints.

STEP #2 With the API model and the API's context in hand, we prepare attack scenarios and define the expected outcomes for each attack.

STEP #3 The analysis phase provides us with the knowledge and the context needed to execute various attack scenarios, including business logic attacks, effectively.

STEP #4 Attacks are executed, and the server's response is analyzed to determine if the API is vulnerable.

STEP #5 Based on the findings and the sensitivity analysis of endpoints (such as personally identifiable information handling), a risk score is assigned.

STEP #6 Our report displays the payload used in the attack, and our automatically generated penetration testing report also includes the server response.

Key Findings & Insights

The Top 3 Discrepancies

1 The Absence of Injection Attacks Is Notable

In the OWASP API Top 10 for 2023, "Injection" was notably absent despite its prevalence in attacks. OWASP justified this omission by arguing that this vulnerability isn't specific to APIs, opting instead to retain it in the general OWASP Top 10 list. However, other issues like authentication and authorization remained on both lists, potentially causing confusion for ordinary application security leaders regarding inclusion and exclusion criteria.

This decision is surprising given that Injections rank as the second most critical vulnerability according to the Pynt score, yet they are conspicuously absent from OWASP's Top 10 API vulnerabilities.

The OWASP Top 10 for APIs aims to highlight the most critical security risks to APIs. However, by not distinctly addressing vulnerabilities like injection attacks, which have been prevalent and damaging across various platforms, the list might give the impression that such threats are less relevant to APIs. This oversight can lead to a false sense of security among developers and security professionals who rely on the list as a comprehensive guide to API security.

As the API Threat Landscape continues to evolve, APIs are becoming the backbone of modern digital infrastructure, making them attractive targets for attackers. Injection flaws, such as SQL, NoSQL, Command Injection, etc., can be particularly devastating when exploited in APIs, leading to data breaches, unauthorized access, and other security incidents. The evolving nature of API attacks necessitates a broad coverage of potential vulnerabilities, including those traditionally associated with web applications.

2 Missing Authentication: Requires Special Attention

OWASP's approach combines missing authentication and broken authentication under the same roof, labeled as "broken authentication." However, these two issues differ significantly in terms of attack complexity and the level of research required to exploit them.

We have opted to separate authentication issues into two distinct categories. This decision stems from the prevalence of these issues and the varying likelihood of exploitation. Missing Authentication vulnerabilities are notably easier to identify and exploit compared to broken authentication.

According to the Pynt score, Missing Authentication is ranked as the number one vulnerability. Therefore, this differentiation is crucial for prioritizing security efforts effectively.

3 The Broken Umbrella: 'Stuff Is Broken' Is Not Enough

OWASP's approach to addressing certain vulnerabilities is to categorize them under a single umbrella. However, this approach may deviate from the original objective of the list: to assist application security professionals in effectively managing the most critical vulnerabilities. Umbrella terms like "Broken access" could encompass a wide array of vulnerability types, each possessing its own distinct characteristics and attributes.

To navigate this complexity, the AppSec professional must conduct thorough research, categorization, and understanding to determine which vulnerabilities fall under this umbrella. In the absence of specific guidelines detailing individual vulnerability risks, this task demands advanced expertise in Application Security, consequently heightening the risk associated with API security identification.

While Local File Inclusion may be ranked at #10 based on the Pynt score, failing to explicitly identify or flag it within a framework of generic descriptions may hinder professionals' ability to adequately address the inherent risks.

"The broken umbrella" method is a repeating element throughout OWASP top 10 and other listicles, which instead of helping can cause confusion and lack clarity on the next steps and action items security teams should take to avoid and prevent risks.

The Final Countdown:

Pynt Top 10 VS. OWASP Top 10 API Vulnerabilities

Scoring System Showdown: Pynt VS OWASP

pynt

- **Weakness Prevalence:** How widespread is this vulnerability in our sample?
- **Likelihood:** How likely is it to exploit this vulnerability and cause damage
- **Impact:** How impactful is the potential damage of this vulnerability?

OWASP

- **Exploitability:** how easy it is to exploit this vulnerability
- **Weakness Prevalence:** How widespread is this vulnerability in our sample?

  Compared to Pynt's score, OWASP popularity methodology is collecting data from bug bounty programs and vendors data. They then ask participants to fill a survey and combine the two sources results.
- **Weakness Detectability:** how reliably this issue can be detected by runtime monitoring solutions?
- **Technical Impact:** How impactful is the potential damage of this vulnerability?

The Final Countdown:

Pynt Top 10 VS. OWASP Top 10 API Vulnerabilities

The list below uncovers the top API vulnerabilities of Pynt VS OWASP, demonstrating both similarities and differences. The critical of all, is that according to Pynt's research methodology, two critical vulnerabilities are entirely missing from OWASP top 10 - Injections and Mass assignment.

#	OWASP 2023	PYNT Score 2023
API1	Broken Object Level Authorization (BOLA)	Missing Authentication
API2	Broken Authentication	Injection
API3	Broken Object Property Level Authorization	Broken Object Level Authorization (BOLA)
API4	Unrestricted Resource Consumption	Broken Authentication
API5	Broken Function Level Authorization	Security Misconfiguration
API6	Unrestricted Access to Sensitive Business Flows	Mass Assignment
API7	Server Side Request Forgery	Broken Function Level Authorization Vulnerability
API8	Security Misconfiguration	Unrestricted Resource Consumption
API9	Improper Inventory Management	Server Side Request Forgery
API10	Unsafe Consumption of APIs	Local File Inclusion

Missing Authentication:

This weakness describes a case where software does not perform validation of user identity before allowing access to any privileged application functionality.

Injection attack:

An injection attack refers to a type of security exploit where malicious code is inserted into an application or system, typically through input fields such as forms or URLs. The goal of an injection attack is to manipulate the application or system to execute unintended commands or access unauthorized data.

Local File Inclusion:

Local File Inclusion (LFI) is a malicious tactic where attackers manipulate a web application to access and execute files stored on the server or reveal their contents. These attacks can result in the exposure of sensitive data and, in extreme scenarios, facilitate cross-site scripting (XSS) or enable remote code execution.

API Breaches: Evolving Threats Demand Evolving Strategies

The increasing reliance on APIs as the backbone of modern digital infrastructure has elevated their attractiveness as targets for malicious attacks. In this research report we covered critical insights into the evolving landscape of API security threats and the effectiveness of current mitigation strategies. Our analysis suggests that there is a gap between the common security standards in the application security industry, and the risks herein.

We advocate for a proactive approach to API security, emphasizing early integration of security measures in the development lifecycle. By shifting left and prioritizing preventative measures, organizations can identify and mitigate vulnerabilities before they can be exploited.

The dynamic and ever-evolving landscape of API threats necessitates a proactive stance from Application Security leaders. It's crucial to not only adhere to OWASP standards but also to integrate them with other relevant security frameworks to ensure comprehensive coverage of all potential API vulnerabilities. This approach enables organizations to stay ahead of emerging threats and safeguard their digital assets effectively.

By adopting a comprehensive API security strategy that addresses emerging vulnerabilities and enhances visibility, organizations can effectively mitigate risks and protect their digital assets from malicious attacks.