wallarm

API THREATSTATS Q3 2024

hotjar

FRACTAL

ExploreTalent.com

METRO PACIFIC INVESTMENTS

www.wallarm.com

(415) 940-7077
188 King St. Unit 508, San Francisco, CA 94107

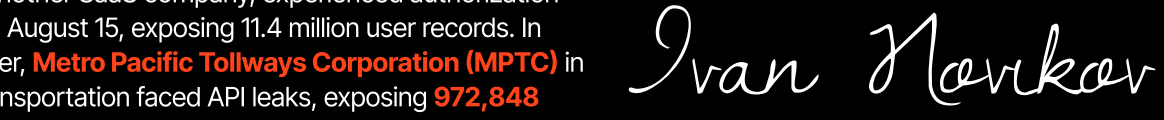
Introduction

I'm pleased to present the API ThreatStats™ Q3 2024 Report, which explores the most significant API security threats we've observed this quarter. The Wallarm team has diligently analyzed the vulnerabilities that are reshaping our digital landscape.

This quarter, we've witnessed major data breaches across a wide range of industries, highlighting how API security connects us all. Unfortunately, API vulnerabilities are not confined to any single sector—they are widespread, affecting telecommunications, blockchain, media, and public transportation alike. In July 2024, **Deutsche Telekom** in the telecommunications industry suffered authentication flaws that exposed personal information of **252 million users**. On July 30, **Hotjar** and **Business Insider**, both in the SaaS and media sectors, faced cross-site issues leading to potential account takeovers affecting **80 million readers. Explore Talent**, another SaaS company, experienced authorization issues on August 15, exposing 11.4 million user records. In September, **Metro Pacific Tollways Corporation (MPTC)** in public transportation faced API leaks, exposing **972,848** records, including sensitive API adjustment logs. Additionally, in July, **Fractal ID**, a blockchain platform, encountered authorization issues, compromising data of **6,300 users.**

These incidents demonstrate that API security challenges are pervasive and can impact any industry, underlining the urgent need for robust API security measures across all sectors.

Our analysis uncovered 469 API vulnerabilities this quarter—a 21% increase from the previous one. The average CVSS score is 7, with many scoring 7.5, indicating high severity and reflecting how easily API issues can be exploited. The majority of these vulnerabilities are straightforward for attackers to leverage, leading to effortless data theft. The impact of these breaches depends largely on the amount and sensitivity of the data exposed, rather than the specific types of vulnerabilities. This trend highlights an escalating threat landscape where APIs are prime targets due to their accessibility and the valuable data they handle.



Ivan Novikov
CEO, Wallarm

A key discovery this quarter is the integral role of API security in AI systems. There is no AI without APIs—they are essential in connecting models, data, and infrastructure. Vulnerabilities in APIs directly impact AI functionalities, and AI features can introduce unique vulnerabilities into APIs. Addressing AI exploits and API vulnerabilities together is crucial for comprehensive security, as they are deeply interconnected.

Q3

To help you navigate this report, we've included a 5×5 matrix outlining the top five topics, such as data breaches and API exploits by product categories, paired with the top five risks. The cells contain specific data breaches, vulnerabilities, and exploits, providing a clear overview of the current security landscape and allowing you to focus on areas most relevant to your interests.

	1	2	3	4	5
API DATA BREACHES	Deutsche Telekom	Metro Pacific Tollways Corporation (MPTC)	Explore Talent	Hotjar and Business Insider	Fractal ID
AI API VULNERABILITIES	OpenShift AI (CVE-2024-7557)	NVIDIA CV-CUDA (CVE-2024-0115)	MLFlow (CVE-2023-1177)	Deep Lake (CVE-2024-6507)	Langflow (CVE-2024-7297)
CLOUD-NATIVE API EXPLOITS	Hashicorp Vault - Denial of Service via Exception Handling (CVE Details Unspecified)	Openshift-console - Unauthenticated Helm Chart Installation (CVE-2024-7079)	Kubernetes - Bypass of Mountable Secrets Policy (CVE 2023-2728 and CVE 2024-3177)	Ansible Automation Controller - Unauthorized k8s API Server Access (CVE-2024-6840)	Envoy - Manipulation of x-envoy Headers (CVE-2024-45806)
CYBERSECURITY PRODUCTS API VULNERABILITIES	Cisco Application Policy Infrastructure Controller - Unauthorized Policy Actions (CVE-2024-20279)	Juniper Networks' Junos OS - Denial-of-Service Attack (CVE-2024-39530	Cilium - Security Misconfiguration in Gateway API (CVE-2024-42487)	FortiEDR Manager API - Improper Access Control (CVE-2024-45323)	Cisco NX-OS Software - Python Parser Escape Vulnerability (CVE-2024-20286)
ENTERPRISE API EXPLOITS	SAP NetWeaver Application Server (CVE-2024-39599)	VMware vCenter Server (CVE-2024-22274)	ServiceNow Now Platform (CVE-2024-5178)	Oracle Java SE and Oracle GraalVM (CVE-2024-21147)	DocuSign API package for Salesforce (CVE-2024-39344)

Q3 API Data Breaches

Q3 2024 brought significant real-world validation to the ThreatStats™ Top-10 approach, especially as traditional frameworks like the OWASP API Top-10 continue to miss critical API vulnerabilities, such as API Cross-Site issues, which played a pivotal role in several major breaches this quarter.

The incidents we've observed underscore that client-side API security gaps, including unauthorized access, OAuth misconfigurations, exposed API logs, and account takeovers, are just as dangerous—if not more so—than typical API flaws like rate limiting or injection attacks. Companies like Deutsche Telekom, Hotjar & Business Insider, Explore Talent, Metro Pacific Tollways Corporation (MPTC), and Fractal ID experienced direct consequences from these overlooked vulnerabilities, with breaches exposing sensitive user data, personal identifiers, and even allowing system manipulation.

252 MILLION USERS
Telecommunications | July 2024

Deutsche Telekom — API2: Authentication Flaws

80 MILLION READERS
SaaS | July 30, 2024

BUSINESS INSIDER
hotjar Hotjar & Business Insider — API3: Cross-site Issues

11.4 MILLION USER RECORDS
SaaS | August 15, 2024

ExploreTalent.com Explore Talent — API6: Authorization Issues

METRO PACIFIC INVESTMENTS Metro Pacific Tollways Corporation (MPTC) — API4: API Leaks

972,848 RECORDS
Transportation | September 2024

FRACTAL Fractal ID — API6: Authorization Issues

6300 USERS
Blockchain | July 2024

1

In July 2024, Deutsche Telekom experienced a significant data breach where unauthenticated API access allowed attackers to retrieve personal information, tariff details, and track users through permanent identifiers. The breach affected 252 million users, highlighting how authentication flaws in client-side APIs can lead to massive data leakage and persistent user tracking, posing risks far beyond simple data theft.

Lesson Learned: Prioritize comprehensive discovery of the API attack surface to identify publicly accessible APIs lacking authentication. Undocumented or forgotten APIs can become significant vulnerabilities if they are exposed without proper security measures. Implement automated tools and continuous monitoring to detect all external-facing APIs, ensuring that authentication and authorization are enforced universally. Recognize that traditional security solutions like WAFs and API gateways may not protect unknown or shadow APIs, making proactive discovery essential.

2

hotjar
BUSINESS INSIDER

On July 30, 2024, Hotjar and Business Insider faced combined OAuth mismanagement and Cross-Site Scripting (XSS) vulnerabilities. These cross-site issues potentially allowed attackers to take over accounts of 80 million readers. The vulnerabilities in their APIs could grant unauthorized access across their platforms, illustrating how cross-site issues can amplify the damage compared to isolated security flaws.

Lesson Learned: Acknowledge that client-side attacks targeting APIs represent blind spots for many API security solutions. Implement robust client-side security measures, including strict Content Security Policies (CSP) and secure handling of OAuth tokens. Regularly assess client-side applications for vulnerabilities such as Cross-Site Scripting (XSS) and ensure that client-side code does not expose sensitive API endpoints or tokens.

3

FRACTAL

In July 2024, Fractal ID, a decentralized digital identity platform, experienced a breach affecting 6,300 users due to authorization issues stemming from an insecure API script. Sensitive personal information, including digital wallet addresses and identity documents, was exposed. The incident raises concerns about API security in decentralized platforms that heavily rely on API frameworks.

Lesson Learned: Recognize that mass assignment vulnerabilities cannot be fully mitigated by usual security controls like negative security models or schema enforcement, especially when vulnerable fields are part of the legitimate schema. Implement detailed validation and authorization checks for each field and action within the API. Develop custom security measures within the application to prevent unauthorized access or modification of sensitive data. Understand that protecting against mass assignment requires in-depth application logic that exceeds the capabilities of standard WAFs and API gateways.

4

ExploreTalent.com

On August 15, 2024, Explore Talent exposed 11.4 million user records due to authorization issues in a misconfigured API. Unauthorized users could access personal information like emails, names, and phone numbers. This breach underscores how inadequate authorization controls in APIs can enable attackers to harvest sensitive data at scale.

Lesson Learned: Address mass assignment vulnerabilities by enforcing strict server-side authorization checks and avoiding automatic binding of client-supplied data to internal objects. Implement explicit allowlists for fields that can be modified by users and validate permissions for each field during API operations. Recognize that standard security controls focusing on negative models or schema enforcement may not prevent mass assignment if the vulnerable fields are part of the schema. Custom application logic is necessary to ensure that only authorized data modifications are allowed.

5

METRO PACIFIC INVESTMENTS

In September 2024, MPTC exposed 972,848 records, including sensitive API adjustment logs crucial for controlling toll road systems. The API leaks not only facilitated data theft but also posed a risk of system manipulation, potentially disrupting critical transportation infrastructure. This incident highlights the dangers of insecure APIs in industries reliant on operational technology.

Lesson Learned: Implement advanced API rate limiting that goes beyond traditional IP or URL-based controls. Utilize API keys and analyze specific JSON fields to count and limit requests on a per-user or per-API key basis. This granular approach requires application-aware rate limiting that standard WAFs and API gateways may not provide. By integrating rate limiting into the application logic, organizations can prevent abuse such as automated scraping or Denial-of-Service (DoS) attacks that exploit API endpoints.

Top-3 Key Insights on API Q3 Data Breaches:

Client-Side API Vulnerabilities Expose Hidden Risks: Not Covered by OWASP API Top-10: Many breaches this quarter, like those at Hotjar, Business Insider, and Explore Talent, originated from client-side API flaws, such as OAuth misconfigurations and Cross-Site Scripting (XSS), which are not adequately addressed by the OWASP API Top-10. Developers often mistakenly consider OAuth a security improvement, but when misconfigured, it becomes a critical weakness, enabling account takeovers and large-scale data exposure. These incidents reveal that client-side API security needs more attention and a dedicated approach to prevent such breaches.

API **Misconfigurations Amplify Breach Scale:** Poorly secured APIs, especially those with weak authentication and authorization controls, lead to large-scale breaches because attackers can access and download entire datasets, not just isolated portions. This was evident in incidents at Deutsche Telekom and Fractal ID, where unauthenticated API access allowed attackers to exploit massive amounts of personal data, tariff information, and user tracking. Unlike traditional malware attacks that may target random subsets of data, API breaches often result in complete data extraction, making the impact far more severe.

APIs Are a **Common Weak Link Across Diverse Industries:** This summer's breaches affected a wide range of sectors, from telecommunications (Deutsche Telekom) and transportation (Metro Pacific Tollways Corporation) to blockchain and Web3 platforms (Fractal ID). These incidents prove that no industry is immune, and API vulnerabilities are a universal challenge across both traditional and cutting-edge tech landscapes. Securing APIs requires consistent, industry-wide efforts to address evolving attack vectors.

Q3 API Vulnerability Statistical Analysis

During this quarter, the Wallarm team analyzed a total of 469 API vulnerabilities, marking a significant increase compared to the 388 issues identified in the second quarter of 2024.

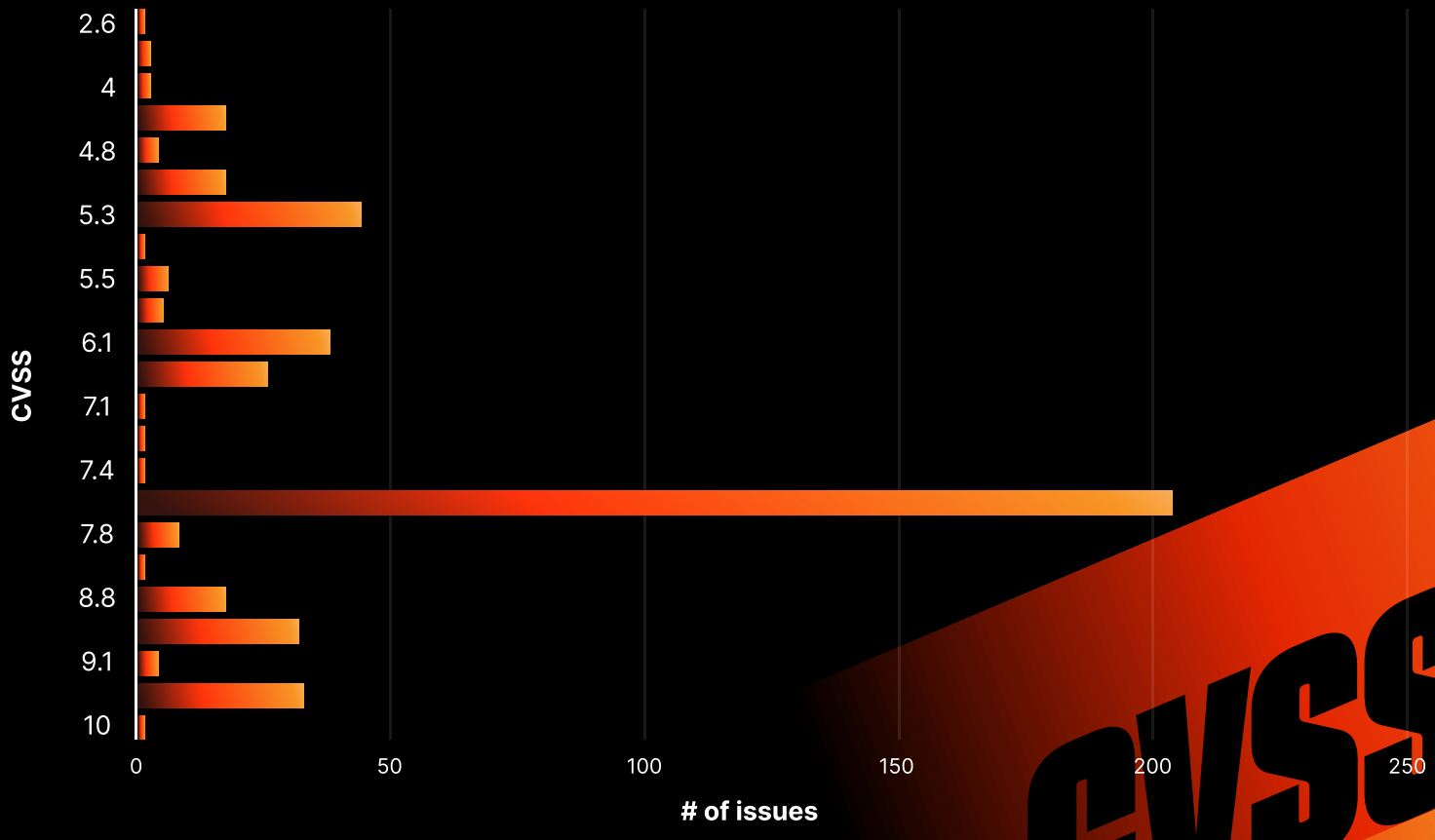
This represents a notable **21% quarter-over-quarter growth** in the number of API vulnerabilities discovered.

The substantial rise highlights an escalating trend in security risks associated with APIs, which are becoming increasingly prevalent as organizations continue to adopt cloud-native architectures and integrate open-source software into their technology stacks.

The increase may be attributed to several factors, including the rapid expansion of API usage across various industries and the growing complexity of modern application environments. As businesses accelerate their digital transformation efforts, APIs serve as the backbone for enabling communication between services, which unfortunately also expands the potential attack surface for malicious actors. Additionally, the widespread adoption of open-source components can introduce vulnerabilities if not properly managed and secured.

The average CVSS score for Q3 API vulnerabilities is 7, high severity, and the CVSS score distribution skews towards high and critical risk, rather than towards lower risks with a majority of issues (45%) at 7.5 score:

of issues vs. CVSS



CVSS

The Significance of CVSS Score 7.5 in API Vulnerabilities

A CVSS score of 7.5 is notable because it represents a vulnerability that is highly exploitable but has a moderate impact on system components. This score is derived from a combination of exploitability and impact metrics, which, in the context of APIs, can illuminate common weaknesses and attack vectors.

This table helps to understand CVSS 7.5 in details:

Exploitability Metrics at Their Maximum

A CVSS score of 7.5 often arises when the exploitability metrics are at their highest levels. In the context of APIs, this means:

Attack Vector (AV): Network (N)
The vulnerability can be exploited remotely over a network connection. Since APIs are inherently designed to be accessed over networks, they are particularly susceptible to network-based attacks. An attacker does not need physical access to the system; they can exploit the vulnerability from anywhere with internet connectivity.

Attack Complexity (AC): Low (L)
Exploitation does not require any special conditions, configurations, or advanced technical skills. This low complexity makes it easier for attackers, even those with limited expertise, to exploit the vulnerability. It increases the pool of potential attackers and the likelihood of the vulnerability being exploited.

Privileges Required (PR): None (N)
Attackers do not need any authentication or prior access to the system. This means that anyone, without any legitimate credentials, can attempt to exploit the vulnerability. It eliminates barriers that might otherwise prevent unauthorized access.

User Interaction (UI): None (N)
No action is required from legitimate users for the vulnerability to be exploited. The attack can proceed without any user involvement, making it stealthy and harder to detect. Users are unlikely to notice anything unusual, which delays detection and response.

Scope (S): Unchanged (U)
The attack affects only the component with the vulnerability and does not impact other system components. While this might seem less severe, it allows attackers to focus their efforts on a specific target without the complexity of affecting multiple systems.

Moderate Impact Metrics

Despite the high exploitability, the impact metrics—Confidentiality (C), Integrity (I), and Availability (A)—are set to Low (L) or considered partial. This means:

Confidentiality Impact
The vulnerability may lead to minor data exposure. Sensitive information could be partially disclosed, but not to a full extent that would compromise the entire system's confidentiality. For example, an attacker might access non-critical data that should not be public but isn't highly sensitive.

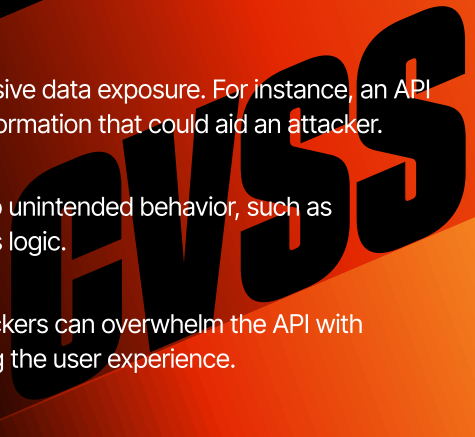
Integrity Impact
There might be slight data alterations. An attacker could modify some data, but the changes are limited and do not corrupt critical system data. This could involve altering user settings or preferences without affecting core functionality.

Availability Impact
The vulnerability could cause minimal service disruptions. The system might experience slowdowns or brief periods of unavailability, but it doesn't lead to a complete shutdown or long-term denial of service. The impact on users is noticeable but not catastrophic.

Common Types of API Vulnerabilities Scoring 7.5

Several types of vulnerabilities commonly receive a CVSS score of 7.5 in APIs, largely due to their high exploitability combined with moderate impact. Understanding these vulnerabilities is crucial for organizations aiming to bolster their API security. These vulnerabilities are often overlooked because they may not cause immediate critical damage, but their ease of exploitation makes them attractive targets for attackers. Here, we delve into the most common types of these vulnerabilities, explaining how they occur and the risks they pose.

1. **Information Disclosure (API Leaks)**
   APIs might unintentionally expose sensitive data due to improper access controls or excessive data exposure. For instance, an API endpoint could return more data than necessary, including internal identifiers or system information that could aid an attacker.
2. **Input Validation Flaws**
   Weak or insufficient validation allows attackers to manipulate API requests. This can lead to unintended behavior, such as executing unauthorized operations or injecting malicious input that affects the application's logic.
3. **Untrolled Resource Consumption, Logic Bombs and Denial-of-Service (DoS)**
   APIs might be susceptible to resource exhaustion attacks that degrade performance. Attackers can overwhelm the API with excessive requests, causing it to slow down or become temporarily unresponsive, affecting the user experience.



CWE (Common Weaknesses Enumeration) API Insights

CWE remains the gold standard in vulnerability classification despite its inherent complexities. The structure of CWE, characterized by its extensive tree dependencies, presents a detailed yet intricate framework for understanding vulnerabilities. While this complexity can introduce challenges such as overlaps and other minor issues, the benefits far outweigh these drawbacks.

This quarter, we've continued to refine our API ThreatStats™ classification approach by grouping related CWEs into our proprietary Top-10 categories. This classification not only highlights the most prevalent issues but also aligns with the industry's broader efforts to standardize API security assessments. By dissecting API exploits through the lens of CWE, we can identify recurring patterns and root causes that allow for more targeted defenses.
Wallarm ThreatStats™ methodology aligns with the comprehensive framework established in our 2023 report, ensuring consistency and depth in our analysis. For a detailed overview of ThreatStats™ API Top 10 methodology, refer to our 2024 annual report.

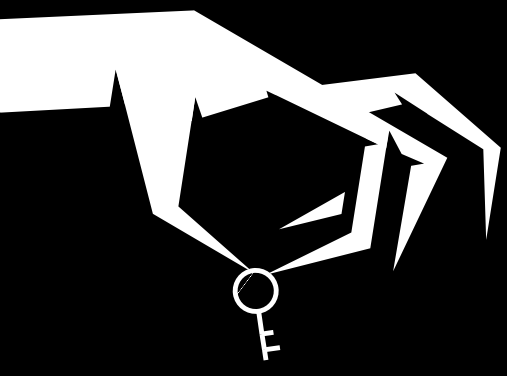
Despite its status as the 'holy grail' of vulnerability classification, CWE is not without its challenges. These include:

1. **Overlaps:** Some CWE entries may describe similar weaknesses in slightly different contexts or layers, leading to potential redundancy.
2. **Tree Dependencies:** The hierarchical nature of CWE can complicate the classification as lower-level weaknesses are nested under more generalized categories.
3. **Minor Issues:** These may include inconsistencies in how different organizations interpret or apply CWE classifications to specific vulnerabilities.

While this data is invaluable, it is insufficient to construct an API-specific CWE Top-25 for just one quarter. The reason lies in the distribution of these issues across many CWE classes, with several classes having only one or two reported issues. This scarcity makes it challenging to rank these weaknesses accurately since, technically, many would need to share the same rank due to their limited occurrences.

Therefore, while a quarterly API CWE Top-25 is not feasible, we plan to compile and analyze the data for the entire year. The annual 2025 API ThreatStats™ report will include a more robust and comprehensive CWE Top-25 tailored to API-specific vulnerabilities.

The table on page 10 illustrates how the Q3-2024 API-specific CWEs align with the global Top-25 vulnerabilities identified in 2023. This mapping provides insights into the prevalence and impact of these weaknesses in the context of APIs.



Q3-24 API CWE Top-10 and Mapping to Global 2023 CWE Top-25

80% of the Q3 API CWE Top-10 mapped into CWE Top-25 2023, only CWE-200: Exposure of Sensitive Information to an Unauthorized Actor and CWE-400: Uncontrolled Resource Consumption are not directly matched. However, we understand that in real examples many of these cases may be mapped to CWE-20 (Improper Input Validation), CWE-284/285/287 (improper auth/auz/access control)

40% of Q3 API CWE Top-10 are in Top-10 of CWE Top-25 2023 global. That shows relevance of the research and wide/statistically significant presence of API issues in a subset of CVE 2023 issues, and/or their similarities to other software bugs. The point is that APIs are just usual software with usual bugs.

Interestingly, we don't see **CWE-400** Uncontrolled Resource Consumption in the CWE Top-25 list, although these bugs are widely distributed. Their absence could be related to their relatively low risk compared to code execution vulnerabilities.

Conversely, it is interesting to see **XSS/CWE-79** included in both the CWE Top-25 and the CWE API Top-10 Q3-24, despite the web-based nature of these defects. Their inclusion is a good reminder of their role in the oAuth+XSS incidents at Hotjar and Business Insider this quarter.

The complete **CWE Top-10** for APIs is structured into three groups: AAA (Authentication, Authorization, Access Control), Injections, and Logic Bombs. The AAA group prevails, comprising 163 issues compared to 145 for Injections, marking a 12% predominance. Logic Bombs have emerged as a noteworthy discovery, aligning with trends highlighted in the OWASP API Top 10, which contrasts with the relative positioning of vulnerabilities like XSS in broader security analyses.

CWE	#CWE	CWE Top-25 Rank
1 CWE-200: Exposure of Sensitive Information to an Unauthorized Actor	50	#N/A
2 CWE-284: Improper Access Control	45	13 Parent of #13 (CWE-287) and #22 (CWE-269)
3 CWE-285: Improper Authorization	44	22 Child of CWE-284 (#13 and #22) Parent of CWE-862 (#11) and CWE-863 (#24)
4 CWE-89: Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	41	3
5 CWE-79: Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	38	2
6 CWE-22: Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	31	8
7 CWE-400: Uncontrolled Resource Consumption	26	#N/A
8 CWE-287: Improper Authentication	24	13
9 CWE-918: Server-Side Request Forgery (SSRF)	20	19
10 CWE-20: Improper Input Validation	15	6

API Vulnerabilities by Products, Industries, and Deployments

Wallarm undertakes this analysis in the API ThreatStats™ report to equip organizations with a deeper understanding of the current API security landscape. By dissecting vulnerabilities based on products, industries, and deployment environments, we aim to provide a granular view of where and how API vulnerabilities manifest. We hope that this chapter will be particularly valuable as it helps organizations identify specific areas of risk relevant to their operational context.

Understanding the distribution of API vulnerabilities allows businesses to:

1. **Tailor Security Strategies:** By knowing which products or industries are most affected, organizations can prioritize their security efforts where they matter most.
2. **Assess Deployment Risks:** Insights into vulnerabilities associated with different deployment models—whether cloud-native, on-premises, or hybrid—enable more informed decisions about infrastructure and security investments.
3. **Stay Ahead of Threats:** Recognizing trends in API vulnerabilities helps in anticipating potential attacks and implementing proactive measures.

Our goal with this analysis is to empower organizations to make data-driven decisions to strengthen their API security posture. By shedding light on the specific challenges across various segments, we provide actionable intelligence that can lead to more effective risk mitigation and resource allocation.

Let's start with API security presence in cloud-native and legacy applications.

32.1% CLOUD NATIVE

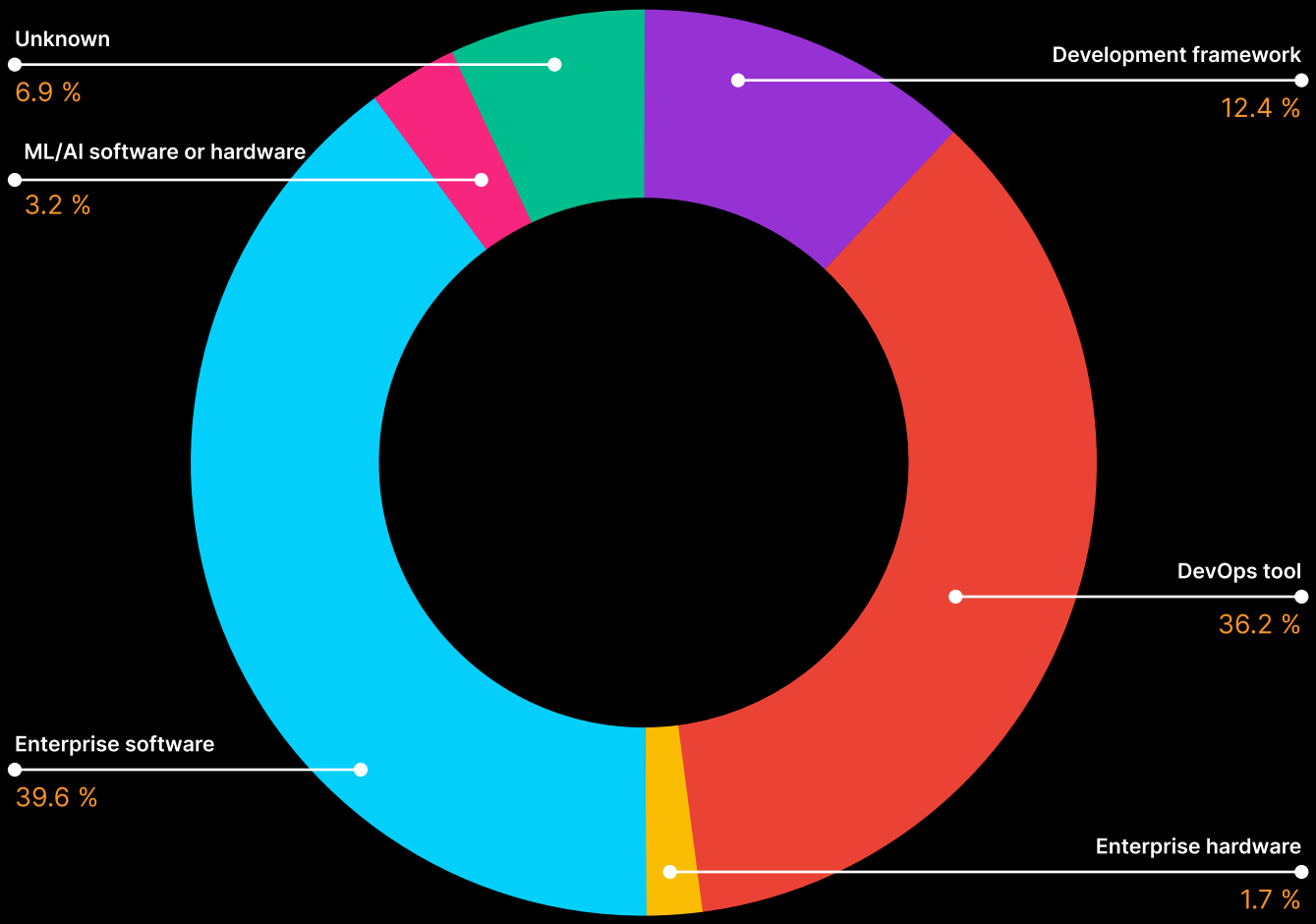
Analysis of this quarter's API vulnerabilities reveals important trends across software types and industries. 32% of the identified vulnerabilities are associated with cloud-native software, particularly in widely used projects such as Kubernetes, etcd, Envoy, Argo CD, ingress-nginx, Cilium, Harbor, Contour, and the Open Policy Agent (OPA). This reflects a growing reliance on cloud-native technologies as organizations modernize their infrastructure and adopt microservices architectures. The complexity and rapid evolution of platforms like Kubernetes and Envoy introduce new security challenges that require careful management.

The remaining 68% of vulnerabilities are mainly in enterprise software and APIs not specifically cloud-native. Notable examples include Junos OS (Juniper Networks), NX-OS Software (Cisco), Application Policy Infrastructure Controller (Cisco), FortiEDR Manager API (Fortinet), VMware vCenter Server, Oracle Java SE, Oracle GraalVM for JDK, Oracle GraalVM Enterprise Edition, SAP NetWeaver Application Server for ABAP and ABAP Platform, Now Platform (ServiceNow), Shopware, DocuSign API package for Salesforce, IBM OpenPages, and AXIS OS (Axis Devices). These traditional enterprise applications remain significant in the API landscape. Even though they are not cloud-native, these applications often expose APIs for integration and functionality, making them susceptible to vulnerabilities if not properly secured. The presence of vulnerabilities in such widely used enterprise software underscores the need for robust security measures across all types of platforms.

36.5% OPEN SOURCE

Open-source products are widely used, both directly by end-users and in commercial products as well. The prevalence of open-source tools is part of the reason for the balance of issues seen in this chart. It's important for readers to understand that this breakdown doesn't indicate that you should be more worried about the open-source tools you use, but more that you should be worried about all the tools you use.

API Vulnerabilities by Product Category



API product categories are always interesting, with Enterprise Software, DevOps Tools, and Development Frameworks leading the majority second year each quarter. This quarter we faced 3 times fewer AI API exploits than last quarter, which may be a result of CVE applications and assignments seasonal effect in summer.

Since enterprise software category is No 1 we decided to include an in-depth analysis, this resulted in two outcomes, a Top-5 enterprise API exploits and Top-5 API Vulnerabilities in Cybersecurity Software.

Top-5 AI API Vulnerabilities of Q3-24

AI systems are fundamentally reliant on APIs to function, connecting models, data, and infrastructure. Without APIs, AI products would lack the connectivity and integration that drive their capabilities. This section, however, focuses on API exploits affecting AI products, not AI vulnerabilities in general, nor all APIs that may interact with AI. The following are the top API exploits found in AI products during Q3-24, with detailed technical insights and lessons learned.

These top API exploits in AI products from Q3-24 demonstrate the growing importance of securing APIs within AI ecosystems. Each exploit highlights a unique challenge in API security, from privilege management to resource handling and input validation.

1

OPENSHIFT

OpenShift AI (CVE-2024-7557) – Authentication Bypass & Privilege Escalation

A critical vulnerability in OpenShift AI's odh-model-controller enables an authentication bypass across models in the same namespace. This allows unauthorized users to escalate privileges, gaining access to other models and resources. Such breaches pose significant risks, as compromised models could expose sensitive data or maliciously interact with other components within the environment.

Lesson Learned: Always ensure that authentication mechanisms are compartmentalized between different models or services, especially in multi-tenant environments. Strong namespace isolation is crucial to prevent privilege escalation.

2

NVIDIA

NVIDIA CV-CUDA (CVE-2024-0115) – Uncontrolled Resource Consumption

NVIDIA's CV-CUDA Python APIs suffer from a vulnerability that leads to uncontrolled resource consumption, causing potential denial of service (DoS) and data loss. This exploit stems from poorly managed API resource handling, which allows attackers to overwhelm the system's computational resources, effectively rendering services unusable.

Lesson Learned: Proper resource management and limitations must be enforced at the API level, particularly in performance-heavy environments like AI. Implementing rate-limiting and resource control mechanisms can prevent DoS attacks.

3

mlflow

MLFlow (CVE-2023-1177) – Path Traversal Vulnerability

A path traversal vulnerability in MLFlow allows attackers to access sensitive files on the host server through API calls. This exploit could lead to the exposure of configuration files, API keys, or other critical data that should remain inaccessible to external users.

Lesson Learned: Input validation is critical to prevent path traversal. API endpoints that interact with file paths must rigorously sanitize inputs and restrict access to known, secure directories to mitig[illegible] such risks.

TOP5AI

4

deeplake

Deep Lake (CVE-2024-6507) – Command Injection Vulnerability

The command injection flaw in Deep Lake, stemming from unsafe consumption of user inputs via APIs, allows attackers to execute arbitrary commands on the server. This can lead to data breaches, unauthorized access, and full system compromise if exploited.

Lesson Learned: Never trust user inputs in API calls that invoke system commands. Always sanitize and validate inputs, and use parameterized queries or safe execution functions to avoid command injection vulnerabilities.

5

Langflow (CVE-2024-7297) – Privilege Escalation

Langflow's API suffers from a critical privilege escalation flaw where an attacker can send a single request to obtain super admin rights. This type of vulnerability presents a significant security threat, as attackers can completely control the system with minimal effort.

Lesson Learned: Implement strict privilege checks at every API endpoint, especially for sensitive operations. Proper role-based access control (RBAC) and multi-factor authentication should be enforced to prevent unauthorized privilege escalations.

Top-5 Enterprise API Exploits of Q3-24

Our list of API threats is meticulously curated based on their potential to disrupt operations, compromise data, and the critical nature of the systems they impact, as well as their attractiveness to cyber attackers. The purpose of this ranking is to direct enterprise attention towards the most severe vulnerabilities that demand comprehensive and immediate defensive strategies.

SAP NetWeaver Application Server (CVE-2024-39599)

This vulnerability stands out due to its ability to bypass the malware scanner, posing a severe risk in environments where SAP NetWeaver underpins a variety of essential business processes. A single exploit could lead to substantial data breaches, operational disruptions, and financial losses.

Lesson Learned: The importance of implementing layered security measures such as meticulous code review processes, enhanced API endpoint protection, and the integration of advanced automated security scanning technologies cannot be overstated. These steps are critical in identifying and mitigating such threats before they can be exploited.

VMware vCenter Server (CVE-2024-22274)

This critical vulnerability can lead to command injection, privilege escalation, and ultimately, unauthorized remote code execution. Given VMware's central role in managing virtualized environments, the potential for widespread disruption is significant, possibly impacting entire data centers.

Lesson Learned: Regularly updating software and maintaining strict user access controls are crucial. Additionally, deploying sophisticated monitoring tools and developing rapid incident response protocols are vital to detect and respond to incidents promptly, preventing attackers from exploiting such vulnerabilities.

ServiceNow Now Platform (CVE-2024-5178)

A sensitive file read flaw in this widely utilized IT service management platform can allow unauthorized users to access critical business information. The exposure of such information not only compromises organizational security but also poses a risk to business integrity.

Lesson Learned: Strengthening access controls and implementing more robust authentication mechanisms are essential to safeguard sensitive files. Regular audits and enhancements to API security policies can further protect against such vulnerabilities, ensuring that confidential data remains secure.

4

ORACLE

Oracle Java SE and Oracle GraalVM (CVE-2024-21147)

The pervasive nature of Java in enterprise applications makes this vulnerability particularly concerning. Unauthorized data access through compromised APIs can lead to serious data breaches, affecting multiple dependent systems.

Lesson Learned: Enterprises should prioritize frequent security updates and rigorously apply a default-deny framework for all application interactions. Employing application behavior analysis and stringent access controls can mitigate the risk of unauthorized access and ensure data integrity across the board.

5

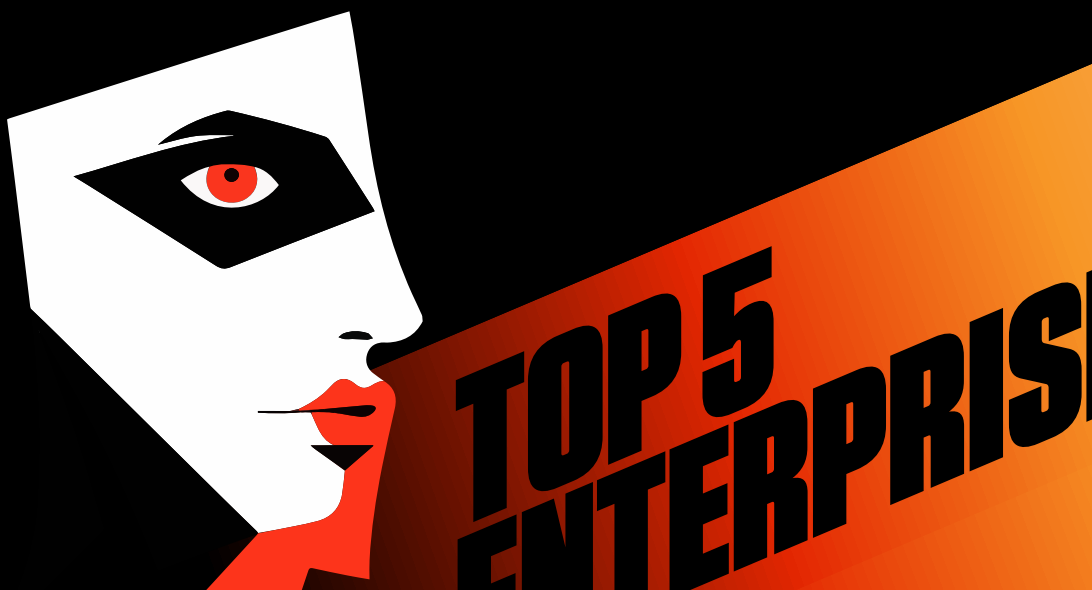
docusign

DocuSign API package for Salesforce (CVE-2024-39344)

This API flaw, which could lead to complete account compromise, ranks on this list due to the sensitive nature of the documents and data it handles within Salesforce. The breach potential here carries not only data loss risks but also severe legal implications and trust erosion.

Lesson Learned: Regular security evaluations and robust integration practices are key to securing APIs. Enterprises should also focus on the continuous monitoring and auditing of third-party integrations to detect and rectify any security lapses promptly.

We hope that by absorbing these lessons and addressing the highlighted threats, businesses can enhance their API security posture, secure the core systems that underpin their operations from the dynamic threats in today's API environment.



Top-5 API Vulnerabilities in Cybersecurity Software of Q3-24

This detailed chapter dives into the most critical API vulnerabilities recently unearthed in major cybersecurity software systems. Each vulnerability acts as a potential front door for attackers, turning what should be robust defense mechanisms into inadvertent entry points. Dive into this section to understand how these API issues may be unwittingly inviting attackers into your digital domain and learn strategies to fortify your defenses effectively.

1

CISCO

SAP NetWeaver Application Server (CVE-2024-39599)

This vulnerability stands out due to its ability to bypass the malware scanner, posing a severe risk in environments where SAP NetWeaver underpins a variety of essential business processes. A single exploit could lead to substantial data breaches, operational disruptions, and financial losses.

Lesson Learned: The importance of implementing layered security measures such as meticulous code review processes, enhanced API endpoint protection, and the integration of advanced automated security scanning technologies cannot be overstated. These steps are critical in identifying and mitigating such threats before they can be exploited.

2

JUNIPER NETWORKS

VMware vCenter Server (CVE-2024-22274)

This critical vulnerability can lead to command injection, privilege escalation, and ultimately, unauthorized remote code execution. Given VMware's central role in managing virtualized environments, the potential for widespread disruption is significant, possibly impacting entire data centers.

Lesson Learned: Regularly updating software and maintaining strict user access controls are crucial. Additionally, deploying sophisticated monitoring tools and developing rapid incident response protocols are vital to detect and respond to incidents promptly, preventing attackers from exploiting such vulnerabilities.

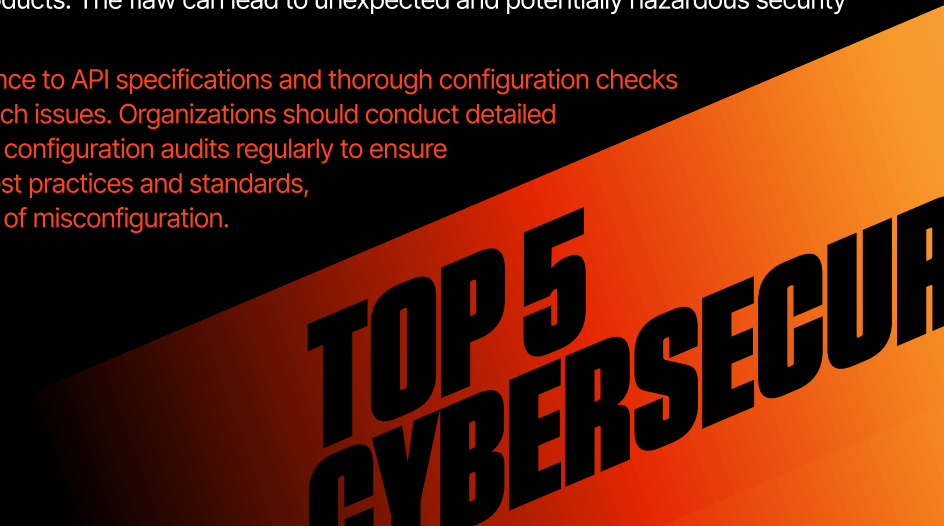
3

cilium

Cilium - Security Misconfiguration in Gateway API (CVE-2024-42487)

This vulnerability arises from a misconfiguration in the Gateway API HTTPRoutes and GRPCRoutes within Cilium, a key component in networking, observability, and security solutions utilized across numerous API security products. The flaw can lead to unexpected and potentially hazardous security behaviors.

Lesson Learned: Adherence to API specifications and thorough configuration checks are essential to prevent such issues. Organizations should conduct detailed security assessments and configuration audits regularly to ensure alignment with security best practices and standards, thereby mitigating the risk of misconfiguration.



4

FORTINET

FortiEDR Manager API - Improper Access Control (CVE-2024-45323)

This critical vulnerability exposes sensitive backend logs due to insufficient access controls, risking significant data exposure. The incident underscores the importance of securing API endpoints against unauthorized access.

Lesson Learned: Strengthening API security with robust access management systems is crucial. Implementing layered security strategies, such as role-based access control (RBAC) and continuous monitoring of API access patterns, can prevent unauthorized access and secure sensitive data effectively.

5

CISCO

Cisco NX-OS Software - Python Parser Escape Vulnerability (CVE-2024-20286)

This issue allows attackers to escape the Python sandbox environment and gain unauthorized access to the system's underlying operations. This vulnerability is particularly concerning due to its potential to compromise system integrity.

Lesson Learned: Ensuring rigorous input validation and secure configuration of all software components are vital. Organizations should employ sandboxing techniques judiciously, complemented by strict security measures and regular security audits to detect and rectify such vulnerabilities promptly.

Top-5 Cloud-Native API Exploits for Q3-24

The landscape of cloud-native technologies is continually evolving, and with it, the complexity of security challenges. This list of the top-five exploits in Q3-24 highlights significant vulnerabilities that have been identified across various platforms, underscoring the critical need for vigilant security practices and timely updates.

1 Hashicorp Vault - Denial of Service via Exception Handling (CVE Details Unspecified)

A high-risk vulnerability in Hashicorp Vault arises from improper handling of exceptional conditions, leading to potential denial of service. This issue was addressed in Vault and Vault Enterprise versions 1.17.2 and 1.16.6, reminding users of the necessity to maintain updated systems.

Lesson Learned: Implementing robust error handling and validating all operational exceptions are crucial to prevent service disruptions in critical security tools.

2 Openshift-console - Unauthenticated Helm Chart Installation (CVE-2024-7079)

OPENSHIFT

This vulnerability allows unauthenticated users to install helm charts on Openshift-console, posing a significant risk of unauthorized changes and potential breaches.

Lesson Learned: Strict authentication controls and rigorous permission checks should be enforced to restrict access to deployment configurations and maintain the integrity of container orchestration environments.

3 Kubernetes - Bypass of Mountable Secrets Policy (CVE 2023-2728 and CVE 2024-3177)

Kubernetes faced a severe security flaw where an exploit bypassed the mountable secrets policy imposed by the ServiceAccount admission plugin. This vulnerability could lead to unauthorized access to sensitive data.

Lesson Learned: Regular updates and patches are essential, along with a comprehensive review of access policies and plugins to ensure they function as intended to safeguard sensitive information.

Ansible Automation Controller - Unauthorized k8s API Server Access (CVE-2024-6840)

This exploit in the Ansible Automation Controller allows attackers to gain access to the Kubernetes API server through job execution with a container group, potentially leading to widespread system manipulation.

Lesson Learned: It's critical to monitor and control the execution paths that lead to critical API endpoints, ensuring that all access is authenticated and authorized to prevent misuse.

Envoy - Manipulation of x-envoy Headers (CVE-2024-45806)

Envoy - Manipulation of x-envoy Headers (CVE-2024-45806): A vulnerability in Envoy allows the potential manipulation of x-envoy headers from external sources, which could be exploited to misroute traffic or disrupt service operations.

Lesson Learned: Input validation and security hardening of headers are necessary to protect against manipulations that could compromise the traffic management within microservices architectures.

Key Insights for Q3 2024

#1 The Increasing Security Risk of eBPF

The third quarter of 2024 has brought to light a series of significant API vulnerabilities in Cilium, a cornerstone in the implementation of eBPF (extended Berkeley Packet Filter) technology. Cilium is widely employed for networking, observability, and security across Kubernetes environments, which means these vulnerabilities have far-reaching implications. This chapter details four major issues identified within Cilium's API, underscoring an emerging pattern of critical security risks associated with eBPF technologies.

1. **Authentication Bypass (CVE-2024-42487):** A misconfiguration in the Gateway API's HTTPRoutes and GRPCRoutes within Cilium v1.15 and v1.16 has led to a route matching order that contradicts specifications, potentially allowing unauthorized access. This vulnerability underscores the necessity of precise API security configurations to prevent authentication bypasses.
2. **Information Leakage via Gateway API (CVE-2024-42486):** This issue involves incorrect update logic in the ReferenceGrant component of Cilium's Gateway API, which could inadvertently extend the accessibility of sensitive information beyond intended limits. Patched versions v1.15.8 and v1.16.1 address this flaw, highlighting the importance of timely updates in maintaining security integrity.
3. **Extended Access to Secrets (CWE-200):** A similar vulnerability to CVE-2024-42486, this flaw also stems from faulty ReferenceGrant update logic in the GatewayAPI controller. The delay in propagating changes allows Gateway resources to access secrets across namespaces even after permissions are revoked, posing a serious risk of information exposure.
4. **Denial of Service (DoS) Vulnerability:** An improper handling of exceptional conditions within Cilium can trigger a denial of service. This vulnerability, fixed in later releases, serves as a reminder of how seemingly minor oversights in handling exceptional network conditions can lead to significant disruptions

A Concerning Indicator for eBPF

These vulnerabilities within Cilium, a key eBPF-based solution, illustrate a troubling trend in the security of cloud-native technologies. eBPF's ability to run programs in the Linux kernel space from user space provides powerful capabilities for performance monitoring and network traffic control. However, this also introduces a critical risk factor: kernel-level access can potentially be exploited by attackers to gain unprecedented control over systems.

A Wider Trend of eBPF Failures

These issues are part of a broader concern highlighted by recent events involving CrowdStrike's Falcon Sensor, which caused kernel panics and crashes on Linux systems due to a kernel bug linked to BPF usage. The incident, which led to widespread disruptions across various platforms, underscores the potential dangers of eBPF at a kernel level, given its capacity to execute code with high privileges directly from user space.

The nature of these exploits indicates that we may see more issues arising in the near future. The impact of such vulnerabilities is particularly severe due to the kernel-level access eBPF provides. This makes every discovered vulnerability not just a flaw, but a substantial potential backdoor into enterprise systems.

As eBPF continues to be adopted widely, particularly in environments that require robust security measures like financial services and critical infrastructure, the need for stringent security audits and rapid response mechanisms becomes increasingly imperative.

Key Insights for Q3 2024

#2 Client-Side API Vulnerabilities on the Rise

ExploreTalent.com hotjar BUSINESS INSIDER

Client-side API vulnerabilities remain a significant, yet often underestimated, threat in the landscape of API security. This quarter, high-profile data breaches at organizations like Hotjar, Business Insider, and Explore Talent put a spotlight on these issues. Driven primarily by client-side API flaws such as OAuth misconfigurations and Cross-Site Scripting (XSS), these breaches reveal hidden risks that traditional security assessments like the OWASP API Top-10 fail to fully address.

OAuth, generally viewed as a security enhancer, can transform into a major vulnerability when improperly configured. Such misconfigurations can lead to unauthorized access, extensive data exposures, and complete account takeovers, showcasing a gap in security practices and awareness among developers regarding API implementations.

Cross-Site Scripting (XSS) also marked its presence strongly in Q3-24, ranking as the fifth most common issue among the Top-15 CWEs with 15 instances out of 469 API vulnerabilities. Holding the second position in MITRE's CWE Top-25 globally, the pervasiveness of XSS, especially in APIs interacting with web browsers or mobile devices (uXSS), calls for stringent client-side security measures.

Beyond XSS, the landscape is riddled with other client-side issues like OAuth hijacking and API token theft, which often escape notice due to their exclusion from the OWASP API Top-10. This oversight leads to a dangerous gap in security practices as these vulnerabilities demand specific strategies for mitigation, given their impact on client interaction.

The focus of the current OWASP API Top-10 does not fully capture the extent of client-side API vulnerabilities, especially those exploiting the complex interactions between users, OAuth, and other authentication mechanisms. This gap indicates a need for developers and security professionals to expand their understanding and approach to API security, ensuring comprehensive protection that includes both server-side and client-side vulnerabilities.

With the rise of client-side API vulnerabilities highlighted in Q3-24, it's clear that a holistic approach to API security, involving diligent configuration, regular audits, and proactive management of emerging vulnerabilities, is crucial to fend off the sophisticated nature of modern cyber threats.

Key Insights for Q3 2024

#3 AI Security is API security

In the world of artificial intelligence (AI), APIs are not just a component; they are fundamental. Every AI product is built around APIs that manage data inputs and outputs, meaning there is no AI without API. However, with the pervasive integration of APIs in AI products, vulnerabilities in API security directly impact the security of AI systems. This relationship also works in reverse, as AI functionalities can introduce unique vulnerabilities into the APIs themselves.

OPENSHIFT

CVE-2024-7557

For instance, an enterprise using OpenShift AI could leverage it for automating and scaling machine learning workflows across various departments. However, if the API managing these workflows is compromised, as indicated by **CVE-2024-7557** which allows for authentication bypass, the consequences could extend across the entire business, impacting everything from automated decision-making to data privacy.

The intertwining of AI and API security suggests that vulnerabilities in one can significantly impact the other. Enhancing API security can lead to more robust AI systems, and vice versa. For instance, improving input validation and sanitation to combat injection attacks will protect both traditional data processing APIs and those used in AI for data analysis and decision-making.

To show this overlaps, we prepared a list of common AI API issues discovered in Q3-2024:

FlowiseAI

OAUTH

KUBEClarity

MESHERY

API Vulnerabilities in AI Systems of Q3-24

1. **Injection Vulnerabilities:** AI systems are often susceptible to various forms of injection attacks due to their reliance on extensive data input and outputs. For instance, vulnerabilities like SQL Injection can appear in AI tools that interact with databases via APIs, as seen in tools like KubeClarity and Meshery. These flaws can lead to unauthorized data manipulation and breaches, illustrating critical points where AI functionalities intersect with traditional API vulnerabilities.

2. **Authentication and Access Control Flaws:** Authentication bypass is a common issue that affects both AI and non-AI systems alike. In AI-centric products, such as OpenShift AI and Flowise, these vulnerabilities allow attackers to escalate privileges or perform unauthorized actions across different parts of the AI ecosystem. These incidents underscore the need for stringent authentication mechanisms in APIs that control access to AI functionalities.

3. **Configuration and Cryptographic Failures:** Hard-coded cryptographic keys in systems like Dragonfly demonstrate a significant risk in API security that directly affects the security of AI products. Such misconfigurations can lead to authentication bypasses, giving attackers administrative access and potentially compromising the entire AI system.

4. **Client-Side API Vulnerabilities:** While traditionally associated with web applications, client-side API vulnerabilities also impact AI products that interact with client-side technologies. OAuth misconfigurations and Cross-Site Scripting (XSS) vulnerabilities can lead to data breaches in AI systems that offer user-facing functionalities, highlighting the overlapping concerns between API security and AI application security.

Limitations of OWASP LLM in AI API Context

It is important to note that while the OWASP Large Language Models (LLM) provide guidelines that are critical for securing systems, they are not entirely applicable to AI APIs, which require a more focused approach. The OWASP LLM guidelines cover a broader spectrum of large language model concerns and do not specifically address the intricate API issues seen in AI implementations. However, the overlap between OWASP LLM and AI API vulnerabilities confirms the integrated nature of AI and API security. This integration indicates that both areas can benefit from shared security practices, although specific adjustments and enhancements are necessary to address the unique challenges presented by AI APIs.

As Wallarm continues to track vulnerabilities, it's clear that AI and API security must be viewed as a unified challenge. Every AI system relies on APIs to function, making them inseparable in both their operation and potential risks. The vulnerabilities we observe in APIs directly affect the security and integrity of AI systems, and AI-specific functionalities can introduce unique risks to the APIs they rely on. By addressing them together, enterprises ensure a more comprehensive approach to securing the future of AI-driven technologies. AI exploits and API vulnerabilities are not separate issues—they are one and the same, and need to be treated as such.

Action Items for CISOs, API Architects, and Security Practitioners

Why We Are Doing It This Way

In our last Q2-24 report, we provided action items specifically for CISOs and Security Practitioners. However, we've received feedback requesting guidance tailored to API Architects as well. Recognizing the critical role API Architects play in designing and securing APIs, we've expanded this final section to include them. By referencing real cases and incidents from this report, including specific CVEs, we aim to provide each role with practical, actionable steps to address the API security challenges highlighted in our findings.

CISOs: Strategic Leadership in API Security

Prioritize Comprehensive API Discovery and Authentication Controls

Case Reference: Deutsche Telekom's breach in July 2024 exposed 252 million users due to unauthenticated API access.

Action Item: Deploy organization-wide API discovery tools to identify all public-facing APIs, including undocumented or shadow APIs. Ensure that every API endpoint requires robust authentication mechanisms to prevent unauthorized access, mitigating risks like those seen in the Deutsche Telekom incident.

Address Client-Side API Vulnerabilities Proactively

Case Reference: Hotjar and Business Insider faced account takeovers affecting 80 million readers due to OAuth misconfigurations and Cross-Site Scripting (XSS) vulnerabilities (CVE-2024-XXXX).

Action Item: Expand your security strategy to include client-side API vulnerabilities. Invest in training for your teams on securing OAuth implementations and preventing XSS attacks. Implement policies that require regular assessments of client-side applications to uncover vulnerabilities that traditional server-side security measures might miss.

API Architects: Designing Secure and Resilient APIs

Ensure Robust Authentication Across All APIs

Case Reference: Deutsche Telekom's massive data breach due to unauthenticated API access.

Action Item: Design APIs with strong authentication protocols, correctly implementing standards like OAuth 2.0. Avoid deploying APIs without authentication, even for internal services, to prevent unauthorized access similar to the Deutsche Telekom incident.

Implement Detailed Input Validation and Output Encoding

Case Reference: Hotjar and Business Insider suffered from XSS vulnerabilities leading to account takeovers.

Action Item: Apply rigorous input validation and output encoding on all API endpoints to prevent injection attacks, including XSS. Ensure that both server-side and client-side validations are in place to protect against malicious inputs.

Security Practitioners: Tactical Measures for API Protection

Conduct Regular, Comprehensive Security Assessments

Case References: Multiple breaches occurred due to overlooked vulnerabilities across industries.

Action Item: Schedule regular security assessments, including penetration testing and code reviews, focusing on both server-side and client-side APIs. Use specialized tools to detect vulnerabilities like mass assignment, injection flaws, and authentication bypasses.

Monitor and Secure Client-Side Applications

Case Reference: Hotjar and Business Insider's client-side vulnerabilities led to significant account takeovers.

Action Item: Extend your security measures to client-side applications. Implement Content Security Policies (CSP) and ensure secure handling of tokens and sensitive data on the client side to prevent attacks that bypass server-side defenses.

Enhance Authorization Checks to Prevent Mass Assignment

Case References: Explore Talent exposed 11.4 million user records due to authorization issues on August 15, 2024.
Fractal ID compromised data of 6,300 users in July 2024 because of authorization flaws.

Action Item: Mandate strict server-side authorization checks for all APIs. Ensure developers avoid mass assignment vulnerabilities by explicitly defining permissible fields for user modification and validating user permissions for each API operation.

Implement Advanced, Application-Aware Rate Limiting

Case Reference: Metro Pacific Tollways Corporation (MPTC) exposed 972,848 records due to API leaks in September 2024.

Action Item: Develop rate limiting strategies that go beyond traditional IP-based controls. Utilize API keys and monitor specific user behaviors and data patterns within JSON payloads to apply granular rate limiting. This approach helps prevent automated attacks and abuse that exploit API endpoints, as seen in the MPTC incident.

Integrate AI and API Security Strategies

Key Insight: AI security is intrinsically linked to API security; vulnerabilities in APIs directly impact AI functionalities.

Action Item: Ensure your security policies encompass both AI systems and their underlying APIs. Address vulnerabilities such as authentication bypasses and injection attacks in AI-related APIs—for example, the OpenShift AI vulnerability (CVE-2024-7557). Promote collaboration between AI development teams and security teams to build security into AI projects from the ground up.

Prevent Mass Assignment Vulnerabilities

Case References: Explore Talent's exposure of user records due to authorization flaws. Fractal ID's data compromise from mass assignment issues.

Action Item: Avoid automatic binding of client-supplied data to internal objects. Explicitly define which fields are allowed to be modified by users and enforce strict authorization checks for each field and operation within your APIs.

Design Application-Level Rate Limiting Mechanisms

Case Reference: MPTC's API leaks due to inadequate rate limiting controls.

Action Item: Incorporate rate limiting within the application logic of your APIs. Use API keys and user identifiers to monitor and control the rate of requests, focusing on user behavior and specific API actions rather than just IP addresses.

Integrate Security into AI API Development

Case Reference: OpenShift AI's authentication bypass vulnerability (CVE-2024-7557) leading to potential privilege escalation.

Action Item: When developing APIs for AI systems, embed security practices such as strict authentication, authorization, and input validation. Ensure that AI models and data are protected against unauthorized access and manipulation.

Enhance Logging and Anomaly Detection for APIs

Case Reference: MPTC's lack of monitoring allowed for unnoticed data exfiltration.

Action Item: Implement comprehensive logging of all API activities. Use anomaly detection systems to identify unusual patterns, such as spikes in traffic or atypical access times, enabling swift response to potential threats.

Apply Application-Aware Rate Limiting and Access Controls

Case Reference: Exploitation of APIs without effective rate limiting, as seen in several breaches this quarter.

Action Item: Collaborate with API architects to implement rate limiting based on API keys and user behavior. Ensure that access controls are fine-tuned to prevent excessive requests and potential abuse.

Stay Informed on Emerging Threats and CVEs

Case References: OpenShift AI's vulnerability (CVE-2024-7557).
NVIDIA CV-CUDA's uncontrolled resource consumption (CVE-2024-0115).

Action Item: Regularly update your knowledge base with the latest API vulnerabilities and CVEs. Subscribe to security advisories and participate in professional networks to stay ahead of emerging threats relevant to your organization's APIs.

By focusing on these actionable steps, each role can address the specific challenges highlighted by the real cases and CVEs in this report. The pervasive API vulnerabilities we've identified this quarter demonstrate the critical need for coordinated efforts across all levels of your organization. Together, we can enhance our collective security posture and better protect our interconnected digital landscape.

As we continue to monitor and analyze API security threats, we invite you to follow us on LinkedIn to stay informed about our latest insights and updates. Don't miss our upcoming Annual 2024 API ThreatStats™ Report, where we'll delve deeper into the trends and findings shaping the future of API security.