wallarm

Q3-2023 Wallarm API ThreatStats™ Report

Top 10 API Threats Surprises, Critical API Leaks, And Increasing Risks of OAuth/JWT

Executive Summary

Navigating the rapidly evolving realm of cybersecurity is crucial for the protection of critical digital resources. **The Wallarm API ThreatStats™ Report for Q3'2023** underscores a surge in threats centered around APIs, necessitating urgent and concentrated efforts from business leaders and cybersecurity professionals. The API ThreatStats™ report offers a detailed examination into the ever changing threats targeting APIs, and uncovers critical vulnerabilities. It serves as an essential guide through this complex cyber landscape and provides expert insights and recommendations.

The Q3'2023 report also introduces a revamped "Top 10 API Security Threats" compilation, a real-time data-driven list covering the **239 vulnerabilities** that emerged in the quarter. What distinguishes Wallarm's methodology from traditional frameworks like OWASP, is our unique approach and classification that specifically targets pressing vulnerabilities that are critical in today's modern API ecosystem. Recognizing these threats and their implications enables organizations to take immediate and proactive steps to strengthen their defenses and safeguard critical assets.

Persistent dangers, notably injections and the growing issue of API data leaks take center stage in this quarter's report. Evidence of these risks is found in the recent serious data breaches suffered by leading firms including Netflix, VMware, and SAP. This underscores the essential role of API leak prevention within corporate security planning. It is evident that protecting against API leaks needs to be a central focus of any corporate security strategy, not an afterthought.

Key Recommendations

Limitations of Existing Frameworks and Benchmarks

While established frameworks like the OWASP API Security Top 10 provide an invaluable guideline and framework, it's important to recognize their limitations. They may not comprehensively address the expansive security needs of today's diverse API portfolio and infrastructure, and is not sufficient to prioritize security efforts. Wallarm's real-time data-driven threat list, which identifies severe threats and critical vulnerabilities not covered by static frameworks should be an integral part of an overall security strategy.

Prioritizing AAA Principles

The foundation of robust API security lies in the core principles of Authorization, Authentication, Access Control (referred to as AAA) and are fundamental pieces for robust API security. It's essential to acknowledge these core principles are also subject to security issues and flaws that can have severe ramifications. The Q3'2023 report witnessed a surge in vulnerabilities related to AAA within technology stacks that are used by various technology companies. This underscores the necessity for organizations to regularly commit and update their entire supply-stack to effectively mitigate potential risks.

Incorporating API Leak Protection Measures

API leaks have emerged as a significant threat, yet they are often overlooked. It is crucial to incorporate API leak protection measures into a security strategy program. Despite not being covered in the OWASP API Security guidelines, the Q3'2023 report highlights a multitude of incidents traced back to leaked credentials (including by 3rd parties) leading to security breaches. It is paramount to implement an automatic discovery system of leaked API keys and secrets, enforcing controls and measures to block their use, and protect against any subsequent attacks.

Don't Underestimate Traditional Vulnerabilities

Don't be misled by those who say traditional app-level vulnerabilities are no longer a threat for APIs. In the Q3'2023 report, Injections are #1 issue in the Top 10 API Security Risks, based on the sheer number of vulnerabilities. The fact that some incidents are not related to modern concerns like BOLA but to traditional legacy threats like CSRF, SSRF, or XSS doesn't diminish the severity of the problem and the prompt resolution of the issue.

Prepare for Reducing SLA Risks

Be prepared for the potential hazards and risks related to reduced Service Level Agreements (SLAs) as your APIs are susceptible to misuse and/or Distributed Denial-of-Service (DDoS) attacks. Certain API issues, including Logic Bombs and insecure API resource consumptions, can make it easier for attackers to disrupt service availability.

Part 1. Top-10 API Threats in Q3-2023

Risk Taxonomy

Navigating the complex labyrinth of API security can be daunting. The foundation of strong security is rooted in understanding the environment within which you are operating.

As a benchmark, many have turned to OWASP API Security Top 10 to identify API-related threats and determine a framework for prioritizing their API Security program efforts. Although this resource is valuable for understanding API related threats, we recognized the importance for more in-depth real-time analysis from the 239 API vulnerabilities documented in the Q3'2023 report.

The results: While some of the risks we identified align with the OWASP API Security list, others fall outside of its scope, and extend beyond its boundaries. This highlights the peril of relying solely on static frameworks and potentially overlooking crucial current key trends.

The Top 10 API Security Risks in Q3'2023

Rank #	Issues	Class Description
1	59	**Injections:** Encompasses a wide variety of attack vectors like SQL, XML, and Command injections.
2	37	**Authentication Flaws:** Issues where identity verification fails.
3	30	**Cross-site Issues:** Includes CSRF, XSS and other threats targeted across different sites.
4	26	**API Leaks:** Leaking sensitive information such as API Keys, JWT tokens, etc.
5	23	**Broken Access Control:** Access governance loopholes that may lead to unauthorized data exposure.
6	19	**Authorization Issues:** Lapses in resource access controls post-authentication.
7	15	**Insecure Resource Consumptions:** Server exhaustion and service disruptions.
8	13	**Weak Secrets and Cryptography:** Issues like hardcoded secrets or weak encryption algorithms.
9	9	**Sessions and Password Management:** Inadequate session handling and poor password management schemes.
10	7	**SSRF:** Server-Side Request Forgery attacks, distinct from injections.

By mapping our Q3'2023 data onto OWASP's framework, we offer a granular, API-focused lens through which to evaluate your organization's security posture. Moreover, the table includes a column that provides a detailed explanation for the observed shifts or splits in categories, grounded in the API specific data from Q3. This perspective enriches our understanding, enabling more effective planning to strengthen API security.

Triple-A of API Security: Authentication, Authorization and Access Control

Within API security, the Triple-A model, encompassing Authentication, Authorization, and Access Control forms the cornerstone. However, it's important to recognize that each element within the Triple-A model has distinct characteristics and functions in protecting an API ecosystem. Notably, vulnerabilities associated with Authentication, Authorization, and Access Control collectively constitute 33% (79 out of 239) of the API security issues reported in Q3 2023.

Authentication	Authorization	Access Control
Your first line of defense, your gatekeeper. It verifies if you are who you claim to be, usually via credentials like username and password. It answers the question, "Who are you?"	The backstage manager deciding what levels of the system you have the keys to. It operates on the principle of least privilege, answering the question, "What can you do?"	The systematic enforcer that combines the decisions of Authentication and Authorization and ensures they are carried out consistently. It dictates, "Here is where you can go, and here's what you can touch."
→ By dividing this from other aspects, we can zoom in on vulnerabilities that specifically exploit authentication loopholes. That's why it deserves its own spotlight and a dedicated incident category.	→ Separating this from authentication allows us to identify when the system grants more permissions than necessary, thus exposing itself to risk.	→ Access Control takes it a step further by also considering contextual factors like IP address, time of access, and even the device used. The reason we isolate Access Control as its unique category is that it allows us to identify flaws in the actual enforcement mechanism, separate from decision-making errors in Authentication or Authorization.

Usually authorization and access controls are merged into one class. So why this taxonomy? By dividing these into three separate categories, we can identify the weak links in each, which enables targeted strategies for remediation.

Injections and SSRF - the most dangerous threats still

Once the reigning champions of the security vulnerability world, Injections have been dethroned in the recent OWASP API Security Top 10 list for 2023. Does that mean they've become less dangerous? Absolutely not. Their fall from grace in the latest OWASP rankings has caused ripples of concern among experts, us at Wallarm included. Now, what we see is #1 threat related to API-related issues in Q3 are actually Injections.

Switching gears to SSRF—Server-Side Request Forgery. Here's a term that Wallarm was already buzzing about a year before OWASP caught up. We saw the writing on the wall—that SSRF was too specific and too nasty to be bundled under the broad umbrella of Injections. And voila, in 2021, OWASP Top 10 gave SSRF its own exclusive seat at the table. Why? Because SSRF is a unique beast. It tricks the server into making unintended requests, becoming a gateway (for the attacker) to internal resources that are otherwise shielded from the external world. A sneaky double agent, in a manner of speaking.

Cross-site attacks are not for APIs? They are #3 in the Top-10!

Cross-site attacks—let's tackle this elephant in the room. While many think cross-site issues like Cross-Site Request Forgery (CSRF) and Cross-Site Scripting (XSS) are old-school web application vulnerabilities, our data shows they're very much alive and kicking in the API world.

This quarter, we've seen a range of high-risk cases, each scoring CVSS 8+ and proving that cross-site attacks are still a force to be reckoned with. For instance, XWiki, Rockwell, and Dolibarr' have all been victims of attacks such as CSRF or XSS via their APIs. Now, these aren't minor entities, they are well established platforms in the industry.

However, here's the critical point: not all cross-site attacks are compatible for APIs. It's akin to trying to inset square pegs into round holes; not all will fit and make it through. For instance, traditional cookie-based CSRF attacks often don't apply because many APIs use token-based authentication. Nonetheless, attackers are creative. They adapt and find new avenues, as shown by CSRF issues in Fastify due to reused OAuth2 state, or CSRF vulnerabilities in the Jenkins Pipeline and Assembla Auth plugins.

Then there are the lesser-known but equally menacing types like Cross-site WebSocket hijacking, demonstrated by ZOHO NCM and Movim.

The key takeaway? Cross-site vulnerabilities are adaptable and tenacious. They evolve and find a way to fit into the API landscape even when you think they shouldn't. And the risks are very real.

API Leaks - #4 in the Top 10 API threats in Q3'2023

API leaks are relatively new, but they've skyrocketed to the number four position in our Top 10 API Security issues this quarter. Noteworthy is that API leaks aren't even on OWASP's radar—yet. Why are they a concern? Their threat lies in their potential for unrestrained disclosure of sensitive data, often through the most negligent methods.

Take The King's Temple Church, for instance. They literally left their Stripe API key out in the open in a public code repository. Even giants like Netflix and VMware have had lapses; Netflix Dispatch exposed JWT Secret Keys in error messages, and VMware Tanzu disclosed sensitive information vulnerabilities.

It's not just about leaving keys and secrets out in the open. Information disclosure vulnerabilities are manifold. SAP Commerce, for example, had a similar issue with its OCC API, and KubePi could leak any user's password hash.

The issue is often one of carelessness. Consider MediaWiki: a simple log showed usernames that should have been hidden. Similarly, the goauthentik platform was vulnerable to a username enumeration attack. These are not sophisticated, nation-state hacks; these are oversights and design flaws that create open doors for attackers. Occasionally, it's not even about security negligence; it's a scenario of debugging mishaps. Apache Superset, for instance, left stack traces enabled by default, providing a treasure trove of information for potential attackers.

Sessions and password management issues - #9

Sessions and password management issues may not dominate our list, sitting at only 9th place, but don't mistake their lower rank for insignificance. These issues are about maintaining the sanctity of user sessions and ensuring that passwords are managed securely. When these controls go south, it's like letting someone else hold the keys to your digital kingdom.

Take FortiOS, for example. Even after deleting an API admin, an existing websocket connection persisted. That's akin to firing an employee but forgetting to revoke their building access. Or look at Graylog, where user sessions remained active even after logout. This is a textbook example of 'Session Management Gone Wrong'—you've left the front door open even when you think it's shut and locked.

Password woes persist too, such as the case with Ruijie Networks, where a vulnerability allowed arbitrary password modification. Imagine walking into a house and being able to change the locks at will, without the owner even knowing. That's essentially what's happening here. And then we have Kirby, which failed to expire sessions after a password change, leaving the door wide open for anyone with an old key.

We've also seen examples like Jenkins, where non-constant time token comparison made it easier for attackers to hijack accounts. In a similar vein, ZKTeco BioTime had a password reset vulnerability, making it child's play for an attacker to gain control over the system.

In essence, sessions and password management issues are the cracks in the foundation. They may not seem as glamorous or dire as injections or API leaks, but they're the chinks in the armor that can bring down the fortress. Despite being lower on the ladder, they need to be treated with the same level of concern and vigilance.

Logic bombs and insecure API resource consumptions

When delving into API security, it's easy to overlook issues that might not make the Top 10 list, but can still have detrimental impacts on both your system's integrity and your business operations. One such class is the matter of resource exhaustion, better known as Denial-of-Service (DoS) attacks. The issues we've seen this quarter throw light on just how severe these can be, especially in an era increasingly reliant on APIs.

Let's review a few examples here. Starting with "Improper Rate Limiting in Strapi's admin screen," this flaw can give malicious actors a window to overwhelm the API by sending numerous requests in quick succession. Strapi being an often used headless CMS, the risk here is massive, affecting a range of businesses relying on it for content management.

The next set of issues involve gRPC, a high-performance, universal remote procedure call framework. "Connection Confusion in gRPC" and "gRPC Reachable Assertion issue" both expose the API to unwanted entries, and if not managed, can lead to resource consumption issues.

Similarly, vulnerability in Feather (API and real-time application framework), "Socket handler allows abusing implicit toString" could allow attackers to manipulate string conversion to initiate a DoS attack. Gitlab's "Projects API pagination can be skipped," takes it a notch higher by not just affecting service availability but also potentially causing data inconsistency.

Moving to the Kubernetes ecosystem, both "K3s apiserver port" and "RKE2 supervisor port" exhibit vulnerabilities to unauthenticated remote DoS attacks. They could cause the API server to crash, affecting all services reliant on it, a nightmare scenario for microservices-based architectures.

We also have issues in the logging systems like "Envoy - gRPC access log crash caused by the listener draining," where the simple act of logging can lead to a system crash.

Resource exhaustion attacks like "Graphql-js - Resource exhaustion exploit when parsing queries" show that even modern, well-designed frameworks like GraphQL are not immune to these kinds of attacks.

The last on this list, "BIG-IQ iControl SOAP Denial of Service," shows that older technologies like SOAP can still be used to initiate a DoS attack, serving as a wake-up call to those who think legacy systems are not worth protecting.

The primary business risk with all these vulnerabilities is unavailability. When your APIs are down, so is your business.

Weak credentials and cryptography - the most persistent API threat

The eighth spot on our list is occupied by weak credentials and cryptography issues—a category that may not make headlines, but still consistently manages to raise eyebrows. It's the sort of security faux pas that feels like a 'Security 101' lesson gone wrong. Whether it's hard-coded credentials or easily decipherable encryption keys, these flaws point to a more fundamental issue: neglect in the basics of security.

We've seen instances like Folio's Data Export Spring Module, which used hard-coded system user credentials—a classic, yet alarming example of how not to manage sensitive data. And then there's CasaOS and its weak JWT secrets, giving attackers a much easier time at breaking in.

Another example is Control ID IDSecure's use of a hard-coded cryptographic key, or ConnectedIO's hard-coded username and password pair. These might sound like small infractions, but they can result in significant security breaches. Add to that Dataprobe's iBoot PDU and its use of hard-coded credentials, as well as IRM's multiple instances of the same issue in various endpoints and libraries.

Skyhigh Secure Web Gateway's storage of sensitive information in clear text should also be a cause for concern. The absence of encryption or obfuscation is akin to leaving your safe open, with the combination written on a post-it note stuck to the door.

Then there are cases like Peppermint, where a hardcoded secret allowed any user to encrypt and decrypt data through a session cookie. This could give anyone the keys to the vault, so to speak.

The point is, weak credentials and poor cryptography practices are not not surprising because we've seen them time and again. But their consistent appearance on the list only underscores the need to root them out and bolster our security measures.

Part 2. Special Chapter on API Leaks: The Silent Alarm Bells

APIs are no longer just connectors; they're the valves that control the flow of data in an organization. Any leak, minor or major, can result in significant setbacks, from compliance failures to catastrophic data breaches. These aren't theoretical risks; they're happening now, warranting immediate attention and action. For an industry that's supposed to be ramping up security measures, we seem to be springing a lot of leaks.

In Q3'2023, the number of API leaks surged, exposing various types of sensitive information. From API keys to secret tokens, the damage inflicted varies in magnitude but not in its alarming nature. See the breakdown (page 12).

What makes this surge in API leaks concerning is that it reveals a systemic issue. Why?

- Tech stacks are getting more complicated – securing both legacy and modern APIs, supporting more authentication/authorization methods, enabling more tooling diversity used by different teams, and covering more environments – which leads to mistakes and accidental leakage.
- Engineering teams are on ever-tightening schedules, which means shipping faster with less QA coverage.
- Software supply chains are getting longer and more complicated, which means these leaks can occur anywhere – by your in-house teams, by your partners, by the open-source code being used, or even by your customers.

It's about time we give API leaks the attention they deserve. While they might be relatively new, they are malicious and widespread, permeating a range of industry sectors including cloud services to video conferencing.

Issue	What's Leaked	Leak Source
King's Temple Church website	Stripe API Key	Public Code Repository
Exception message	Key	API
Rotem CRM	Information	Uri Interface
KubePi	Password Hash	API
Strapi	User Information	API with Prefix Fields
SAP Commerce	Various	OCC API
Netflix Dispatch	JWT Secret Key	Server Response
Archer platform	Sensitive info	API
VMware Tanzu	Various	API
LMS by Masteriyo	Various	API
Control ID IDSecure	Sensitive Info	API
AWS SDK v2	Email Content	SES v1 API
Argo CD	Cluster Secret	Cluster Details Page
Leyka	Sensitive Info	API
Ivanti Endpoint Manager	File	API
GLPI	Sensitive Fields	API
MediaWiki	Usernames	CheckUserLog
Royal Elementor Addons	MailChimp API Key	API
Nomad Search	CSI Plugins Info	API
Essential Addons for Elementor	MailChimp API Key	API
Zoom	Sensitive Info	API
Datasette 1.0	DB and Table Names	API
goauthentik	Usernames	API
FULL - Customer	Sensitive Info	Health Check API
Apache Superset	Stack Traces	API
Spring for GraphQL	Data & Identity	Wrong Session

Part 3. The Biggest API Security Findings of the Quarter: OAuth, SSO, and JWT can be weak

The vulnerabilities we saw this quarter in big-name products like Sentry, Consul, Red Hat, Netflix, and WordPress are unsettling, particularly because they're platforms trusted to keep things secure. Those are excellent products but no products are security-perfect.

Starting with **Sentry**, a great real-time error tracking tool that developers across industries rely on, the "Incorrect credential validation on OAuth token requests" exposes projects to unauthorized access.

HashiCorp's Consul, often used for service discovery and configuration, had a hole in its JWT Auth. The "Mismatched Service Identity and JWT Providers" vulnerability could allow attackers to mimic legitimate services, leading to unauthorized data manipulation or leakage. In terms of impact, it's akin to an imposter walking into a secure facility using a fake ID. Once inside, they have the same access to sensitive information or systems as the person they're impersonating.

Red Hat Single Sign-On, a core component for many enterprise-level applications, suffered from "Cross-Site Scripting." A successful exploit could let an attacker hijack user accounts or redirect users to malicious websites. Given the scale at which Red Hat is deployed in corporate environments, this can lead to widespread data breaches or even ransomware attacks.

Netflix's "Server response includes the JWT Secret Key used for signing JWT tokens in error message" issue is a slap in the face, considering Netflix's reputation for robust cloud architecture. Any platform using Netflix's Dispatch could be compromised, potentially giving attackers control over what is often the backbone of many organizations' incident management. The damage could range from data breaches to hampering real-time incident responses during a crisis.

WordPress, the CMS giant, also joins the list with its "OAuth Single Sign On – SSO (OAuth Client) Plugin Broken Authentication." This is especially troubling given WordPress's widespread adoption. From blogs and corporate sites to online stores, a wide range of websites could be exposed to unauthorized access, leading to data theft or even direct financial loss.

Even the most advanced and reputable tech organizations are not immune to flaws in authentication and authorization mechanisms. These are foundational pillars for API security, yet their vulnerabilities can surface in even the most robust systems. This serves as a reminder that, regardless of an organization's stature in the tech world, continual vigilance and refinement of authentication and authorization processes are paramount.

Unveiling Key Insights from Wallarm's Q3'2023 ThreatStats™ Report

The New Top 10: Dynamic List Of API Threats Based On The Data

We've curated a cutting-edge taxonomy that captures all 239 API security threats from this quarter. We're highlighting areas where OWASP may have overlooked and diving deep into critical issues like "Injections: The Old King" and "API Leaks: The New Kid on the Block."

The Leaky Bucket: API Spills Everyone Should Worry About

Forget small leaks; we're talking major breaches. Major players like Netflix, VMware, and SAP experienced significant data exposures , from leaked JWT Secret Keys to sensitive user information. This section is a wake-up call to include protection against API (and other) leaks into your product security program.

The Security Paradox: When Protectors Become Predators

Technologies like OAuth, JWT, and SSO— designed to shield you—are sometimes "wolves in sheep's clothing" and can become vulnerabilities if not implemented correctly. Trusted names like Red Hat, Consul, Sentry, and even Netflix encountered security challenges that are hard to ignore.

Implementing Next Steps from the Insights of the Q3'2023 ThreatStats™ Report

Make API Security Assessment a Priority

- Immediately prioritize a review of your organization's API security in light of the Wallarm Q3'2023 ThreatStats findings.
- Assess how these findings may impact your specific API infrastructure and applications.

Implement Targeted Countermeasures:

- Develop and implement targeted countermeasures to address the specific threats and vulnerabilities identified in the report.
- Ensure these countermeasures are tailored to your API landscape.

Review and Update Policies:

- Review your organization's security policies and update them to address the specific challenges identified in the report.

Prioritize API Security and Adopt an Agile Approach:

- Highlight the importance of API security within your organization, emphasizing that it's a top level concern.
- Embrace an agile approach to security that allows you to adapt rapidly to evolving API threats.

Regularly Test and Validate:

- Engage in regular penetration testing and audits to validate the effectiveness of your API security measures.
- Ensure that your security controls remain robust over time.

Identify Vulnerabilities and Weaknesses:

- Conduct an in-depth analysis to identify vulnerabilities and weaknesses within your API systems.
- Focus on areas highlighted in the report, such as API injections and leaks.

Educate Your Team and Stay Informed and Adaptive:

- Provide training and awareness programs to your teams, emphasizing the importance of API security and the specific threats outlined in the report.
- Stay informed about emerging API threats and maintain an adaptive security strategy that can swiftly respond to changing conditions.

Engage with Industry Peers:

- Connect with industry peers and participate in knowledge-sharing forums to exchange insights and best practices related to API security.

Monitor for API Leaks:

- Establish a system for monitoring and detecting API leaks, including leaked credentials and sensitive data.
- Put measures in place to promptly respond to and mitigate any leaks.

Collaborate with Wallarm

- Explore collaboration with Wallarm to leverage their expertise in API security.
- Consider how Wallarm's integrated platform solution can augment your security posture based on the report's insights.

By following these actionable steps, you can effectively respond to the insights provided in the Wallarm Q3'2023 ThreatStats report, to fortify your API security, and protect your digital assets from the evolving threat landscape.

Stay proactive, stay protected, stay ahead with Wallarm.

Follow us on LinkedIn to stay in touch with the latest API security threats discoveries and risk analysis.

www.linkedin.com/company/wallarm

Wallarm Research Team
lab.wallarm.com

www.wallarm.com

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