SALT

Understanding API Attacks:

Why are they different and how can you stop them?

E-BOOK

Understanding API Attacks:
Why are they different and how can you stop them?

APIs fuel today's digital economy. As such, they have become very lucrative for attackers eager to exploit any flaws they can find to exfiltrate valuable data. Organizations looking to deliver on their business innovation initiatives and delight their customers can no longer treat API security as a luxury – protecting these highways linking critical data and services is a clear necessity.

API attacks have been making headlines in recent years – from the large-scale data breach suffered by Australia's second-largest telecommunications company, Optus, in September 2022, to the series of API breaches affecting high-profile brands such as Peloton, Experian, and John Deere in 2021. These and several other incidents bear out that the Gartner® prediction that "by 2022, API abuses will move from an infrequent to the most-frequent attack vector, resulting in data breaches for enterprise web applications" was very accurate indeed.

But what exactly are API attacks? And why have they become so prevalent? Is there a way to reliably protect against them? Building an understanding of these issues will prepare us to build the necessary defenses.

What are API attacks?

Simply put, an API attack is the hostile usage of an API using API endpoints to access and exploit data, taking advantage of code flaws or business logic vulnerabilities to force behavior that wasn't intended during the API's development process.

As the myriad of API attacks and breaches that have taken place in recent years have made abundantly clear, these attacks can have heavy financial and reputational consequences for the targeted organizations, bring services to a halt, slow down business innovation and hinder business growth.

Although smaller, less-known companies may believe that the old "security through obscurity" method may be enough to protect them, the truth is no company – big or small – is safe from today's API attacks. No matter how unknown your organization, how uninteresting you think your data is or how safe you think your APIs are, bad actors may find lucrative vulnerabilities to exploit. Protecting your APIs against present and future threats is the only way to ensure your business can continue to grow.

Why have APIs become a priority target?

The explosion in API usage

The reliance on APIs has exploded in recent years, massively expanding the attack surface available to bad actors – and they are taking full advantage of it.

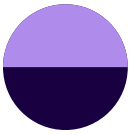
The State of API Security Report Q1 2023 released by Salt Labs in March 2023 shows that 94% of respondents have experienced security problems in production APIs over the past year, with 17% having experienced an API-related breach. Another eye-opening finding from Salt customer data is that attackers have upped their pace with a 400% increase in unique attackers within a six-month period.

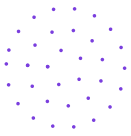
The high value of data

Due to the large volumes and nature of the data they expose, including personally identifiable information (PII), APIs have become the most lucrative attack vector available to attackers looking to access, expose or sell the valuable data they uncover. Unfortunately, exploiting these high-reward API vulnerabilities doesn't necessarily take a lot of skill or resources. Prodding for flaws in APIs is more about having the time and patience to look for flaws that can be manipulated and exploited using a trial-and-error approach.

API sprawl and poor governance

The issue of securing APIs has become even more complex due to the often-uncontrolled proliferation of APIs within an organization, usually referred to as API sprawl. In most of today's organizations, multiple development teams are building different APIs in different ways without a centralized security strategy or governance framework in place. Adding insult to injury, the widespread use of open-source, no-code, or third-party APIs makes secure API integration and visibility even more difficult.





According to the State of API Security Q1 2023 report, 37% of organizations update their APIs at least weekly while 48% of survey respondents admit to only updating their API documentation less than twice a year. This means that documentation is clearly failing to keep up with the ever-changing and continuously expanding nature of APIs. As a result, the number of unknown or shadow APIs that are not noted in documentation rises, adding significant data exposure risks to an organization.

Out-of-date APIs that organizations often assume have been disabled – known as zombie APIs – present an additional type of risk, with 43% of State of API Security report respondents rating them as a high concern. Simply put, you can't protect what you can't see. As most organizations are unable to see all their APIs, API sprawl represents a massive opportunity for attackers looking to exploit API vulnerabilities.

How and why have API attacks changed?

From pattern-based attacks to business logic exploits

While the number of APIs increases and the attack surface expands, the types of threats faced by APIs have also been changing. Whereas in the past transaction-based attacks, such as SQL injection and other code execution methods, were the most prevalent tactics, today's attacks often aim to exploit the underlying application and business logic behind each API.

Although transaction-based attacks are still happening and traditional tools, such as API gateways and web application firewalls (WAFs), can play an important role in blocking them, business logic-based attacks have become the most pressing type of threat.

Why reconnaissance is a game-changer

In application logic attacks, hackers use reconnaissance techniques over time, poking and prodding the APIs, looking for vulnerabilities in business logic or other flaws, such as gaps in authentication or authorization controls. During reconnaissance, attackers seek areas to explore, such as gaining unauthorized access to data or functionality within the API, or weaknesses in the API to launch pinpoint, low-traffic, application denial-of-service (DoS) attacks. Because every API endpoint is different, every attack is unique, making all API vulnerabilities essentially zero-day vulnerabilities. Traditional tools, using their proxy architecture and rules and signatures, cannot detect this kind of behavior or stop these kinds of attacks. Only by gaining full context, over time, of the business function and logic behind each API can these attacks be stopped.

These reconnaissance activities are quite drawn out. Bad actors may take weeks or even months to look for data, poking for flaws or finding subtle ways to disrupt the supply chain, or exfiltrate data from an API. The traffic patterns often stay below established rate limits set to block volumetric attacks, and their activities go undetected by traditional security devices.

The limitations of testing

Along with the rise in DevOps, companies are increasingly moving to DevSecOps, looking to "shift left" and spot security flaws before deploying code, with testing, code scanning, and other techniques. This approach has significant limits when it comes to APIs.

First, the rapid rate of development makes it very hard for testing to keep up. API sprawl complicates the process, with development processes and teams being distributed and disparate. No company has a single pipeline to secure.

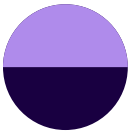
Second, the nature of APIs limits the effectiveness of pre-prod testing. To protect APIs effectively, organizations must be able to find logic-based vulnerabilities and understand the functional purpose of each of their API endpoints. They also need to understand the behavioral characteristics of each parameter and element in use by those endpoints. A single API endpoint can have thousands of possible permutations of business and application logic that would need to be vetted and exercised to understand if there's any potential for malicious behavior.

If an organization were to try to test for every single business logic vulnerability in every API endpoint, development cycles would come to a standstill. Today's organizations have no choice but to accept that some API vulnerabilities will most likely make it into production, which means that APIs can be secured only through a robust API security strategy that includes threat detection at runtime protection along with capturing exploit insights that help teams harden APIs over time, along with pre-prod testing of APIs.

What are the most common types of API attacks?

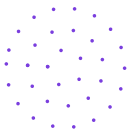
Today's most common API attacks can be split into four categories:

Lack of visibility and governance



In this type of attack, the attackers take advantage of APIs that are either completely unknown to an organization – shadow or zombie APIs – or APIs whose security posture is not visible, such as unmanaged or third-party APIs.

Abuse and misuse of APIs



To execute this type of attack, a bad actor will use an API exactly as designed but leverage the results in an unintended and malicious manner by exploiting

design or development flaws that allow for malicious outcomes, such as data exfiltration.

Business logic flaw exploitation

In a business logic-based attack, hackers will use reconnaissance techniques over time to poke for vulnerabilities in each API's unique business logic. During the reconnaissance phase, which can last days, weeks, or even months, attackers seek areas to explore, such as gaining unauthorized access to data or functionality within the API.

Stolen credentials and social engineering

This type of threat manifests when a bad actor uses social engineering techniques to access privileged API keys. This allows them to steal credentials and use the API as if they were a legitimate, authenticated user or admin. According to the State of API Security Report Q1 2023, in the last year, 78% of attacks came from seemingly legitimate users who have maliciously achieved the proper authentication.

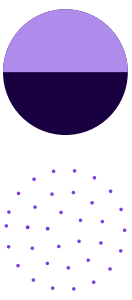
The OWASP API Security Top 10

To help the API security industry gain a deeper understanding of the most common API attacks, the Open Web Application Security Project (OWASP) released its first-ever API Security Top 10 list of vulnerabilities in 2019. The list has been updated in June 2023 and includes the following ten API attack types:

API1:2023 Broken Object Level Authorization

Broken object level authorization (BOLA) represents around 40% of API attacks and is the most common API threat.

As APIs frequently expose endpoints that handle object IDs, this creates a large potential attack surface. Object level authorization is an access control method that is typically implemented at the code level to verify a user's ability to access a certain object. Modern applications use a variety of intricate and pervasive authorization and access control systems. Developers frequently neglect to apply these checks before accessing an object, even when an application includes a robust infrastructure for authorization checks. Attackers can easily exploit API endpoints that are vulnerable to broken object level authorization by manipulating the ID of an object that is sent within an API request. BOLA authorization flaws can lead to data exfiltration as well as unauthorized viewing, modification, or destruction of data.



Ultimately, BOLA can lead to a full account takeover (ATO).

API2:2023 Broken Authentication

Attackers can easily target authentication processes, especially if they are fully exposed or accessible to the public. The second most frequent vulnerability reported by OWASP is broken user authentication, which enables attackers to utilize credential stuffing, stolen authentication tokens, and brute-force attacks to obtain unauthorized access to apps. Attackers are then able to control users' accounts, gain unlawful access to other users' data, and conduct unauthorized transactions. Technological issues, such as inadequate password complexity, missing account lockout criteria, overly long rotation times for passwords and certificates, or the usage of API keys as the only authentication method, can result in faulty authentication in APIs.

Attackers who can successfully take advantage of weaknesses in authentication procedures may be able to access another user's data without authorization and carry out illicit transactions using that user's account.

API3:2023 Broken Object Property Level Authorization

Broken Object Property Level Authorization merges attacks that happen by gaining unauthorized access to sensitive information by way of Excessive Data Exposure (previously listed as number 3 in the 2019 OWASP API Security Top 10) or Mass Assignment (previously in sixth place in the 2019 list). Both techniques are based on API endpoint manipulation to gain access to sensitive data.

The main reason for introducing this new threat on the list is that, even if an API can enforce sufficient object-level authorization security measures, this might still not be enough to protect it. More specific authorization that covers the objects and their characteristics is often required. The varying access levels within an API object must also be considered, as an API object often has both a public property and a private one.

API4:2023 Unrestricted Resource Consumption

The Unrestricted Resource Consumption vulnerability has replaced the previous number 4 in the OWASP API Security Top 10, Lack of Resources and Rate Limiting.

However, while the name changed, this vulnerability remains the same overall. Resources like the network, CPU, memory, and storage are used up by API calls. The user's input and the endpoint's business logic have a significant impact on the number of resources needed to fulfill a request. The size or quantity of resources that a client or user may request may not necessarily be

constrained by APIs. This not only has the potential to negatively affect API server performance and cause Denial of Service (DoS), but it also makes APIs that support authentication and data retrieval vulnerable to brute-force and enumeration assaults, including token and credential cracking.

API5:2023 Broken Function Level Authorization (BFLA)

Authorization issues are frequently the result of incorrectly configured or poorly implemented authorization. Because contemporary systems often might have a wide variety of roles, groups, and user hierarchies, including sub-users and users with multiple roles, implementing suitable authorization procedures is a challenging undertaking. Distributed application architectures and cloud-native design make this even more difficult. In this respect, BOLA and broken function level authorization (BFLA) are quite similar, except BFLA targets API functions rather than the objects that APIs interact with.

By exploiting BFLA vulnerabilities, attackers can access restricted resources, hijack another user's account, create or delete accounts, or escalate privileges to acquire administrative access by taking advantage of broken function level authorization flaws.

API6:2023 Unrestricted Access to Sensitive Business Flows

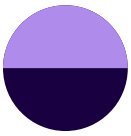
This vulnerability, which has replaced Mass Assignment as number 6 on the OWASP API Security Top 10 list, occurs when an API exposes a business flow, such as buying a ticket or posting a comment, without compensating for how the functionality could cause harm if used excessively through automation.

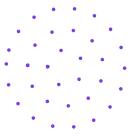
To exploit this vulnerability, a bad actor will need to understand the business model backed by the API in question, find sensitive business flow and automate access to these flows in a way that can harm the targeted business. A successful attack can mean that the attacker can buy a product flow, spam a system by creating a comment/post flow, or reserve all the slots available for a given service. preventing legitimate users from using the system.

API7:2023 Server Side Request Forgery

This threat has replaced Mass Assignment as number 6 on the OWASP API Security Top 10 list.

Server Side Request Forgery (SSRF) can occur when a user-controlled URL is passed over an API and is honored and processed by the back-end server. The risk for the environment materializes if the back-end server tries to connect to the user-supplied URL, which opens the door for SSRF.





As a result of a successful SSRF attack, attackers can gain access to internal network resources within a web-based environment, compromising security mechanisms within the web service.

API8:2023 Security Misconfiguration

The security misconfiguration threat represents a catch-all for a variety of security setup errors that frequently have a detrimental influence on API security as a whole and unintentionally expose vulnerabilities. Insecure default configurations, incomplete or ad-hoc configurations, open cloud storage, incorrect HTTP headers, unneeded HTTP methods, excessively permissive Cross-Origin resource sharing (CORS), and verbose error messages are a few instances of security misconfigurations.

During their reconnaissance phase, attackers can use security flaws to learn about the application and API components. Specific faults, such as stack trace problems, can reveal private user information and system specifics that can help an attacker locate exploitable technologies, such as out-of-date or improperly configured web and application servers.

API9:2023 Improper Inventory Management

Improper Inventory Management has replaced Improper Assets Management as number 9 in the OWASP API Security Top 10 and, while the name has been changed, the threat remains the same.

Understanding potential exposure and risk depends on maintaining an accurate, complete, and up-to-date API inventory. Older API versions that need to be decommissioned are difficult to find because of unexpected holes in the API attack surface caused by an outdated or incomplete inventory. Similar to how faulty documentation makes it difficult to find vulnerabilities that need to be fixed, it also exposes hazards like unknown disclosure of sensitive data.

API10:2023 Unsafe Consumption of APIs

Unsafe Consumption of APIs has come to replace Insufficient Logging and Monitoring as number 10 in the OWASP API Security Top 10 and it contains a mix of two common API issues: the consumption of API data itself, which was largely addressed in the Injection category of the 2019 list section but now includes attacks that are not explicitly injection-related, such as de-serialization issues and some types of desync attacks; and integrations, which can include any third-party service or functionality embedded into the API implementation or in their supporting back-end services.

Although not exclusive to this category, API-based supply-chain attacks serve as a good example of the danger it represents.

Unfortunately, the State of API Security Report Q1 2023 indicated that only 54% of respondents consider the OWASP API Security Top 10 an area of focus, even though 62% of attempted attacks against organizations leverage at least one of those methods.

The MITRE Attack Framework and how it applies to API security

The MITRE ATT&CK Framework is a vital resource of open-source knowledge for the security industry, with CISOs and cybersecurity experts relying on it to expand their knowledge about different attack tactics and procedures (TTPs). Although it contains relevant insights on several specific platforms and environments, currently, the framework doesn't contain a specific API security matrix. However, security leaders can still leverage the MITRE ATT&CK Framework to identify and defend themselves against an increasing number of API threats as attackers frequently use many of the MITRE-outlined TTPs in their API attack campaigns.

In a recent white paper, Salt Security experts have looked at the MITRE ATT&CK Enterprise Matrix and identified tactics in this matrix that are also being applied in API attack campaigns, specifically when it comes to three common API security threats:

- Broken object level authorization (BOLA)
- Stolen credentials
- Leaky public APIs

For each of these threats, Salt experts have mapped a typical attack lifecycle to the TTPs found in the Enterprise Matrix and outlined the steps that bad actors can take in each scenario from reconnaissance and phishing to evasion and data exfiltration or abuse.

Hopefully, API security threats will be added to an API-specific matrix within MITRE ATT&CK Framework in the future. But, in the meantime, this analysis can provide valuable insights into how to defend against these attacks and develop more effective incident response plans by leveraging the existing security framework.

How can you stop today's API attacks?

APIs require a different approach to security that goes beyond traditional, signature and rule-based tools, which can only look at transactions in isolation.

In a research note released in 2022, Gartner® acknowledged the need for dedicated solutions by adding API Security as a distinct pillar in its updated Security Reference Architecture. With this new architecture, Gartner® recognizes that, to protect APIs effectively, organizations need to rely on more than traditional tools, such as WAFs – even intelligent WAFs – API gateways and CDNs.

To secure APIs in today's complex landscape, organizations need a solution specifically created to address three key unique API security considerations:

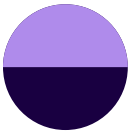
Visibility and governance

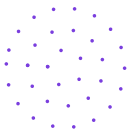
As we've discussed, having complete visibility over the expanding API attack surface is essential to effectively protect APIs, but API sprawl, constantly changing APIs, and testing limitations make this particularly challenging.

To build an effective API strategy, organizations need to be able to see all the APIs in their environment, including shadow and zombie APIs. API discovery must be automatic and continuous to keep up with the constant release of new and updated APIs and must cover all customer-facing, partner-facing, and internal APIs.

However, visibility is not enough. Today's organizations also need to understand each API, its functionality, and the business logic behind it to be able to accurately assess the risk it poses and determine if it exposes any sensitive data.

Attack prevention through cloud-scale big data and mature AI





To spot today's subtle, business logic-based attack methods, organizations need an API security solution that combines big data, AI, and ML to capture large volumes of API traffic, create a baseline of normal activity and look for any deviations in behavior.

This continuous and automated analysis of API traffic provides the only path to understanding normal behavior for each unique API and gaining the

context required to pinpoint attackers in the reconnaissance phase. The reconnaissance needed to find flaws to exploit and launch attacks like these takes a lot of time, which means that a complete API security solution must also be able to analyze API traffic continuously over time.

Eliminating future security gaps

Agile development practices and tight release development cycles can cause development teams to miss security gaps. With limited testing times and capabilities meaning that some vulnerabilities will most likely be deployed into production, runtime protection is certainly critical to prevent the exploitation of those vulnerabilities by bad actors. But relying solely on runtime protection leaves organizations in a situation where they are constantly trying to catch up to attackers.

That's why an API security solution must be able to not only stop attackers but also learn from their activity as they probe and manipulate an API. These learnings can then provide valuable insights into the vulnerabilities that are unique to that API and help development teams prioritize and eliminate gaps quickly and proactively, improving the API security practices employed during development.

The Gartner® take on API security

In the latest version of its "API Security: What You Need to Do to Protect Your APIs" report, Gartner® reaffirms the importance of API discovery saying that "you cannot secure what you cannot find or categorize".

The leading analyst firm advises organizations to use a combination of API management, WAFs, and identity infrastructure to effectively protect their APIs, reinforcing the idea that an effective API security strategy doesn't translate into a single platform but is a combination of tools and practices.

Gartner® also recognizes that organizations need to adopt a continuous approach to API security that covers the entire API lifecycle, from design to post-production, to create a robust and reusable set of API security policies.

What does the future hold for API Security?

As API usage continues to grow and security threats continue to change, it is now clear that API security is an ever-evolving discipline, which will most certainly continue to evolve and expand.

Recognizing that "API security is a growing concern as API traffic grows," Gartner® has predicted in its "Gartner® 2022 Innovation Insight for API Protection" that, by 2025, the explosive growth in APIs will surpass the capabilities of API management tools, meaning that less than 50% of enterprise APIs will be managed by then. The analyst firm also believes that, by 2026, 40% of organizations will select their web application and API protection vendors based on their advanced API protection capabilities – a number that has risen from just 15% in 2021.

With industry experts and analysts becoming growingly aware of the importance of API security and APIs becoming the number one attack vector, Salt Security experts have identified some key trends that are likely to shape the API security landscape in the next few years:

The key role of ML and AI in threat detection

Artificial intelligence (AI) and machine learning (ML) have become top of mind for many industries recently, and for good reason: their immense potential is only matched by the speed at which AI and ML technology is evolving. They have become essential to the field of API security, as only through mature AI and ML models is it possible to analyze large volumes of API traffic, baseline typical behavior and detect even the most subtle anomalies.

Although AI technology will most certainly take threat detection capabilities to new heights, organizations need to be aware that the same technology is also available to bad actors, who are and will continue to use it to develop new and even more sophisticated attack tactics. Security vendors and the organizations they serve will be even more pressed to use technological advances in the field of AI to protect against API attacks before attackers can successfully use them against them.

Increased focus on API governance and documentation

The risks of API sprawl have become clear to organizations of all shapes and sizes in recent years, but the problem is far from resolved. In fact, with API usage continuing to grow, the speed of API development continuing to increase, and the proliferation of open-source and no-code APIs coming to stay, visibility and control issues will only tend to increase, and organizations will have no choice but to focus their efforts on API governance and documentation in order to mitigate the risks posed by shadow and zombie APIs that can expose sensitive data.

In the next few years, we will be likely to see organizations putting API governance plans in place to strengthen their security programs.

AI likely to reshape DevOps and cybersecurity

Although this is unlikely to happen in the immediate future, AI capabilities will most likely evolve to allow reliable and fully automated code creation at some point in the future. This would no doubt reshape DevOps cycles entirely, with less reliance on human intervention, putting into question current processes and functional roles, and forcing the cybersecurity industry to adapt and evolve in unprecedented ways.

Why is Salt uniquely positioned to protect APIs?

Reliance on APIs is continuing to grow as they become increasingly imperative to business success and innovation. Simultaneously, APIs are getting more difficult to protect as current tools and processes can't keep up with new threats and an ever-expanding attack surface.

Here's why the Salt Security API Protection Platform is in a unique position to help today's organizations secure their APIs effectively:

Better API discovery and improved security posture

Salt can automatically inventory all your APIs, including shadow and zombie APIs, and highlight all instances where APIs expose sensitive data. Smart aggregation or APIs and continuous discovery ensure APIs stay protected even as an environment evolves and changes with agile DevOps practices. Salt proactively identifies vulnerabilities in your APIs even before they serve production traffic. The platform also uses attackers like pen testers, capturing their minor successes to provide insights for dev teams while stopping attackers before they reach their objective.

Stopping attacks with advanced threat protection

Combining cloud-scale big data with time-tested AI and ML algorithms, the Salt platform can pinpoint and stop threats to APIs by identifying attackers early, during their reconnaissance phase, and prevent them from advancing. With continuous and automated analysis of huge volumes of API traffic over time, Salt correlates activities back to a single entity, sends a consolidated alert to avoid alert fatigue, and blocks the attacker rather than transactions.

Why are the Salt platform's AI and ML models so different from others in the industry? In short, models are only as good as their training data and, with limited exposure, they raise lots of false positives. In fact, Salt holds the industry's only patent for blocking API attacks and has had customers in production for 5+ years, in thousands of environments. Through this exposure, our models have been trained based on tens of trillions of API calls and millions of malicious attacks. This enables Salt to detect low and slow attacks and produce high-fidelity alerts, with very low false positive rates (which is particularly critical in large, production-scale environments). Additionally, because our architecture is carefully crafted to balance privacy and cloud-scale analysis, when one company in Salt's customer base gets attacked, that data set trains models that all other customers benefit from, creating a very strong network effect.

Hardening APIs with insightful remediation guidance

Augment pre-prod scanning with remediation insights learned in runtime to help security and dev teams improve the security posture of their APIs. Salt gives your security and dev teams continuous, real-world insights that pre-production scanning or testing can't. Salt Insights include the unique context of how your organization's APIs are being used and how they are being misused in production environments. Such tangible and relevant insights help your developers build their security awareness and improve development best practices to minimize future API vulnerabilities.

At a time when API security is becoming a business priority, organizations must go beyond traditional security practices and tools to build a modern security strategy that addresses security at every stage of the API lifecycle and provides comprehensive protection while fostering cross-team collaboration.

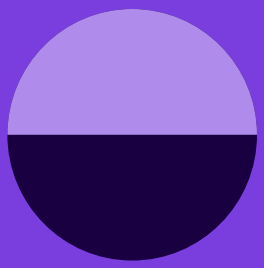
SALT

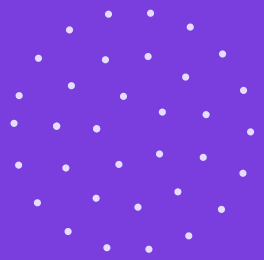
Salt Security protects the APIs that form the core of every modern application. Its patented API Protection Platform is the only API security solution that combines the power of cloud-scale big data and time-tested ML/AI to detect and prevent API attacks. By correlating activities across millions of APIs and users over time, Salt delivers deep context with real-time analysis and continuous insights for API discovery, attack prevention, and shift-left practices. Deployed in minutes and seamlessly integrated within existing systems, the Salt platform gives customers immediate value and protection, so they can innovate with confidence and accelerate their digital transformation initiatives.

Request a demo today!

info@salt.security

www.salt.security





SALT

Securing your Innovation.