SALT

State of API Security Q1 2023

SALT
LABS

Executive Summary

The State of API Security in Q1 2023

The State of API Security Report from Salt Labs is the industry's first on API security risks, challenges, and strategies. The fifth edition of this pioneering research offers security, DevOps, and risk management teams a deeper perspective into the dozens of factors that impact API security. It also provides insights on building strategies to reduce the growing API attack surface.

As with previous editions, this report incorporates survey results and empirical data from the Salt SaaS platform hosting our customers' API metadata. This year, we have also included some thought-provoking API vulnerability research from Salt Labs that illustrates how some of the API security concerns highlighted by survey respondents can manifest in real-world scenarios.

The most eye-opening finding from the Q1 2023 report was that **94% of respondents have experienced security problems in production APIs** over the past year, with **17% having experienced an API-related breach**. Another important finding from Salt customer data is that attackers have upped their pace with a **400% increase in unique attackers over the same time period six months ago**. There is little doubt that API security must become a key focus for security professionals in 2023.

Attackers are also finding new and unexpected ways to target their efforts. In the past, organizations believed that proper authentication to interact with an API was enough deterrent to send attackers elsewhere. Salt Labs data shows that **78% of attacks come from seemingly legitimate users, but are actually attackers who have maliciously achieved the proper authentication**. Additionally, attackers have been targeting internal APIs, with 8% of attacks perpetrated against these supposedly well protected assets.

Survey respondents also indicate that API security has become a major business issue for their organizations. **59% have experienced application rollout delays due to security issues identified in APIs**. Application rollout issues inevitably cause business disruption, which raises alarms at all levels. And, with API security breaches becoming so newsworthy, it's no surprise that **48% of survey respondents say that API security is now a C-level discussion**.

Respondents identified vulnerabilities as one of the top security issues they had experienced with production APIs. While 41% stated they had experienced such API vulnerabilities, Salt Labs' research suggests that this number is severely underestimated. **In 90% of investigations, Salt Labs identifies API security vulnerabilities, 50% of which should be considered critical**.

Despite all of the API challenges, API security practices are still maturing. **Survey respondents are largely relying on traditional approaches to API security such as WAFs, API gateways, and analyzing log files, but only 23% find these methods very effective**. It's therefore not surprising that only 12% of respondents believe their API security programs are advanced, while 30% say they are non-existent or in planning.

Documentation continues to be a challenge for organizations as only 19% of respondents were very confident that they have a complete API inventory. One challenge in this area is the frequency of API updates – **37% of organizations update their APIs at least weekly**. Similarly, only 18% are very confident they understand which APIs expose PII data.

APIs are at the core of every modern application, and attackers continue their efforts at unprecedented rates. Survey responses and Salt customer data overwhelmingly demonstrate that the time is now for organizations to get serious about securing their APIs.

Research Methodology

To understand the state of API security today, Salt Labs – the API threat research arm of Salt Security – initiated and compiled this API security industry report. Our in-depth research combines survey responses and empirical data sfrom Salt Security customers. The findings reflect the input of nearly 400 security, DevOps, and app development professionals across companies big and small, in a variety of industries across the globe (page 19). Salt Labs also pulls aggregated and anonymized data from the SaaS component of the Salt Security API Protection Platform – this empirical data gives more context to the survey response findings.

SALT

Table of Contents

API attacks are on the rise

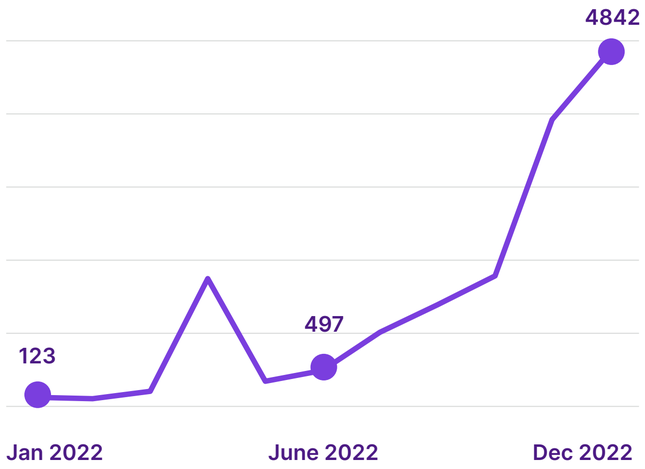
Attackers are more relentless than ever and are starting to target internal and authenticated APIs

Bad actors are tenacious and are continuing to find new and unexpected ways to attack. In the past, organizations believed that proper authentication to interact with an API was enough of a deterrent to send attackers elsewhere. Salt Labs data shows that **78% of attacks come from seemingly legitimate users, but are actually attackers who have maliciously achieved the proper authentication**.

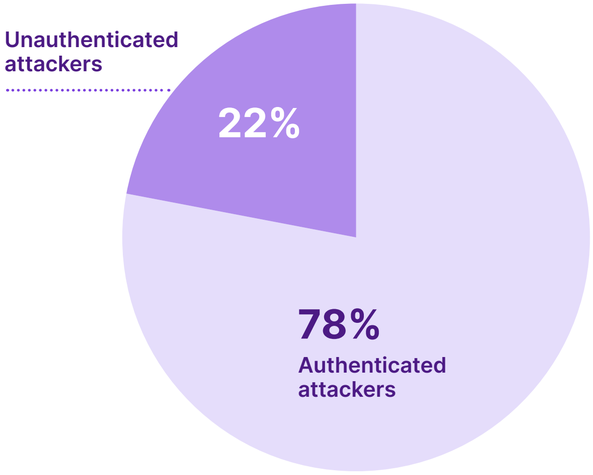
Internal-facing applications have also historically been deemed “safe” and, as such, security teams weren’t particularly concerned about requiring robust security because these applications weren’t facing the outside world. However, Salt Labs data shows that **8% of attack attempts are perpetrated against internal-facing APIs**, which are typically left entirely unprotected.

Also interesting is the significant rise in attackers targeting our customer base. **The end of last year saw a major spike, with 4,845 attackers operating in December alone – a 400% increase from just a few months prior**. This report marks the first time we have shared this level of information, but given this increase, we felt it necessary to advise the industry that attackers targeting APIs are incredibly active.

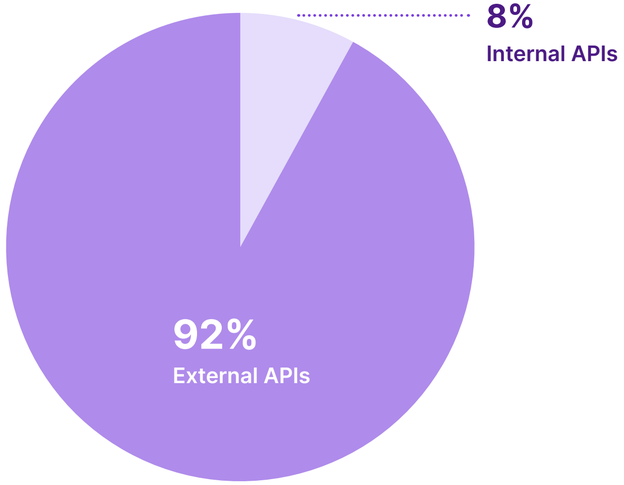
Salt customer data: Unique attackers targeting customer APIs during 2022



Salt customer data: Attack attempts from authenticated vs. unauthenticated attackers



Salt customer data: Attack attempts against internal and external facing API endpoints



API security has emerged as a significant business issue, not just a security problem

59% have delayed an application rollout over API security issues, and the C Suite is getting involved in the discussion

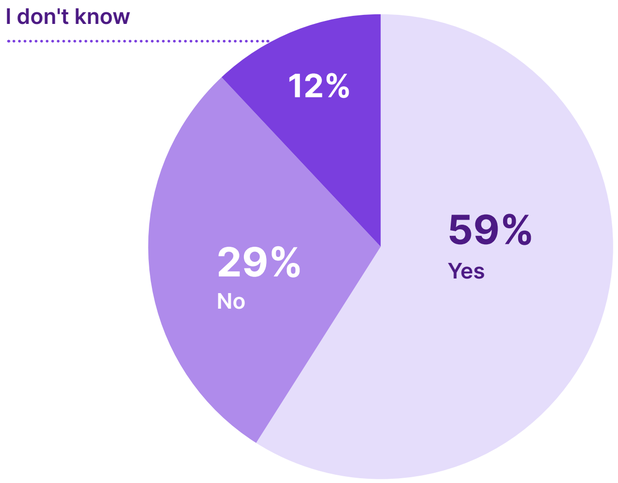
Every application owner's worst nightmare is a delayed rollout or the rollback of a new application. Survey respondents told us that API security concerns have led to this very result far too frequently. An unfortunate **59% have experienced application rollout delays resulting from security issues identified in APIs**.

This high percentage illustrates the sad fact that no amount of testing and security-minded code development can address today's API security challenges. Developers cannot anticipate every possible business logic gap in their APIs, and pre-prod API testing tools similarly cannot identify these gaps.

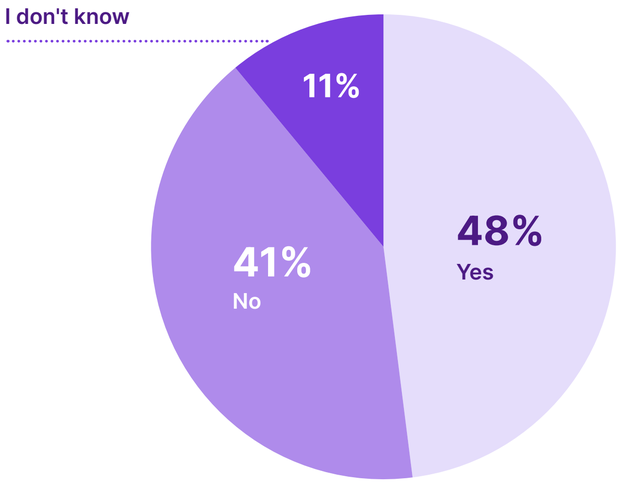
Application rollout delays cause business disruption, which raises alarms at all levels. Add the fact that API security breaches are making headlines so often and it's no surprise that this topic is now being discussed in executive and board meetings. In fact, **48% of survey respondents say that API security has become a C-level discussion** over the past year.

It is interesting to note that **the C Suite is paying closer attention in highly regulated industries such as technology (59%), financial services (56%), and energy/utilities (55%)**. When factoring in company size, executives are most involved in the API security discussion at companies in the 5001-10,000 employee range (71%) and least involved in companies of 10,000 employees or more (34%).

Have you ever slowed the rollout of a new application into production because of API security concerns?



Has the security of your APIs become a C-level discussion at your organization?



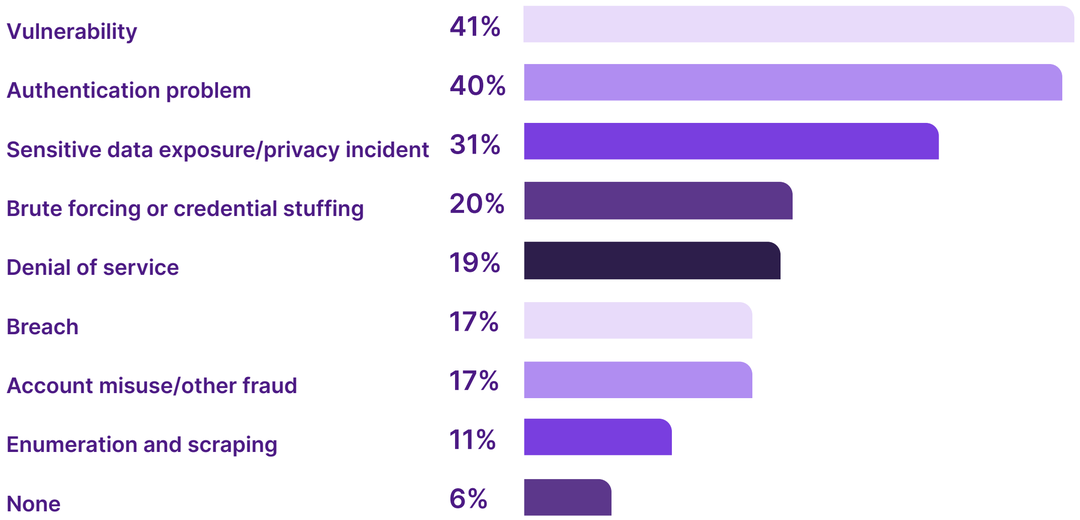
Respondents are experiencing significant API security challenges

94% of respondents have experienced security problems in production APIs, with 17% having experienced a breach

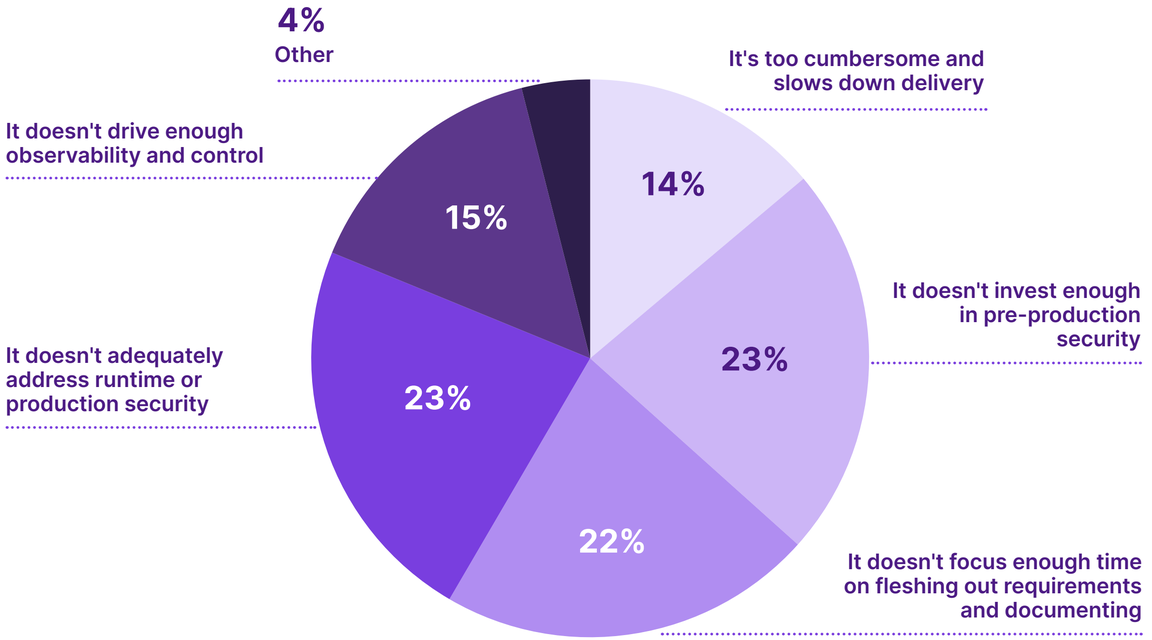
API security problems are a real concern for survey respondents. **94% had some security issue with their production APIs over the past year**, with vulnerabilities topping the list at 41%, followed closely by authentication problems at 40%. Of more concern, 31% had experienced a sensitive data exposure or privacy incident and **17% had experienced a security breach**; such events have significant costs and reputational damage associated with them.

Given the prevalence of security events, it's no surprise that respondents lack confidence in the security aspects of their API programs. Nearly half of respondents cited security gaps as their top concern, with 23% each citing inadequate runtime or production security and insufficient investment in pre-production security.

In the past 12 months, what security problems have you found in production APIs?



What is your biggest concern about your company's existing API program?



The OWASP API Security Top 10 – a critical starting point

66% of attack attempts leverage one or more of the OWASP API Security Top 10 methods, but only 54% of respondents focus on this industry standard

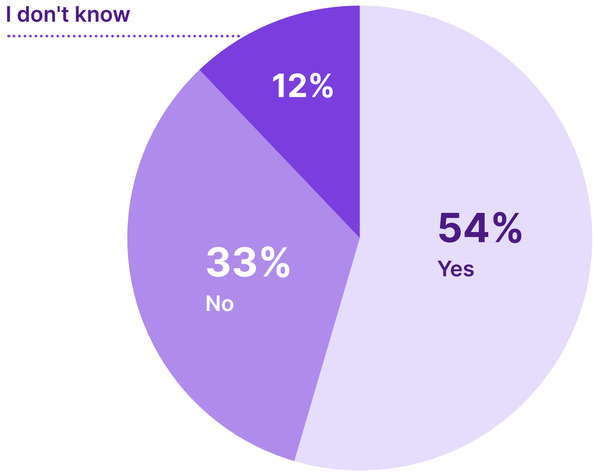
The OWASP API Security Top 10 list is an industry standard in the API space, but **it's a focus area for security programs at only 54% of respondents' organizations**. This low percentage is disheartening, since Salt customer data shows that **66% of all attack attempts leverage at least one of these 10 security vulnerabilities**. Typically, bad actors use combinations of these 10 attacks to propagate more sophisticated attacks. With such a large percentage of attacks taking advantage of these most common and well-documented security flaws, organizations cannot afford to overlook this fundamental principle in API security.

This list was introduced in 2019 and is being refreshed for 2023, so we eagerly await the next set of survey results, and we hope to see this update increase both awareness and focus.

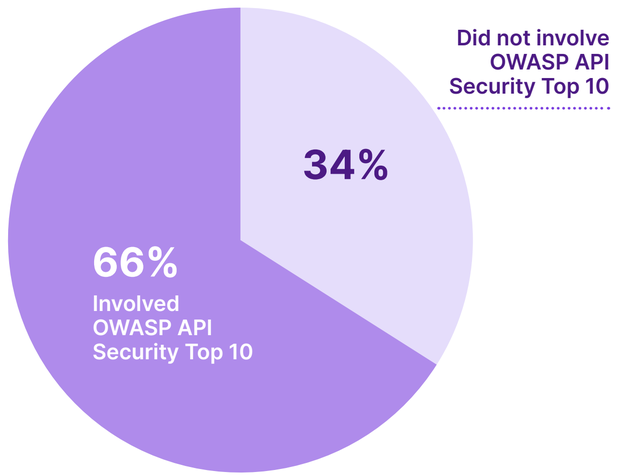
When mapping attempted attacks to the OWASP API Security Top 10, we saw a lot of #8 Simple Injection Attacks (29%) – which represents a carryover from the standard OWASP list. The next most common attacks were #7 Security Misconfiguration (23%) and #4 Lack of Resources and Rate Limiting (20%). Lack of resource and rate limiting is an API issue that requires that attack activity be investigated at the user level vs. the aggregate level, a nuance traditional tools like WAFs simply can't distinguish.

We also uncovered more sophisticated, drawn-out attacks like Broken User Authentication (9%) and Broken Object Level Authorization (7%). These attacks take advantage of business logic gaps, and the resulting exploitation potential is quite high because these attacks simply cannot be detected by traditional tools.

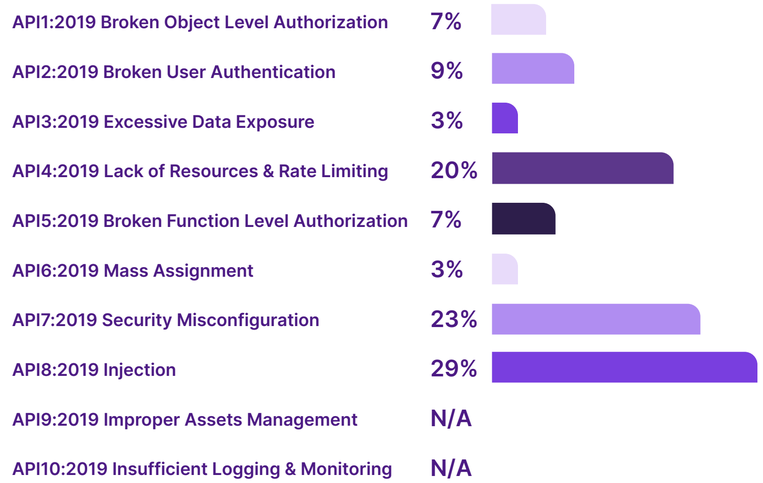
Has your security team highlighted the OWASP API Security Top 10 threats as a focus area for your security program?



Salt customer data: Attack attempts leveraging the OWASP API Security Top 10 vs. other attack types



Salt customer data: Attack attempts that map to the OWASP API Security Top 10

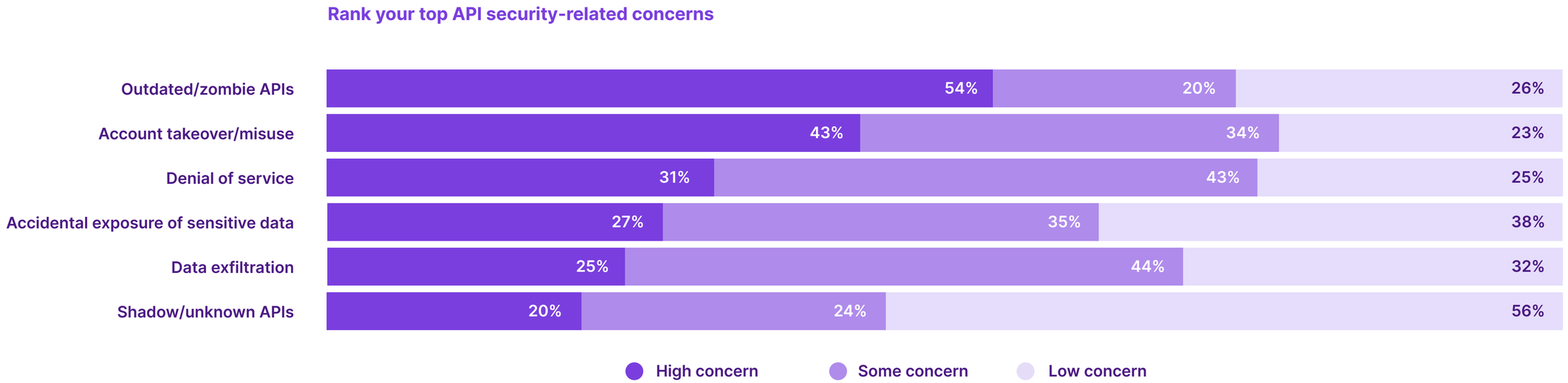


“Zombie” APIs top the list of API security concerns

Fears over account takeover/misuse rank second, with 43% rating it a “high concern”

With significant API security issues happening regularly to survey respondents, it stands to reason that they have real concerns about their API security programs. **Outdated/zombie APIs top their concerns, with 54% indicating that this risk is of high concern.** Given significant documentation challenges at organizations (Page 12), it’s highly likely most environments are running APIs that are not documented, so even though the lowest percentage (20%) cited shadow APIs as a top concern, the risk in this area is likely higher than many respondents realize.

Account takeover is also keeping security professionals up at night, with 43% stating it is a high concern. See the Salt Labs “in the wild” use cases (Page 15) for a deeper understanding of why this concern is so well founded.



Most API security strategies remain immature

Only 12% of respondents consider their API security programs to be advanced, and 30% admit they're non-existent or just in the planning stage

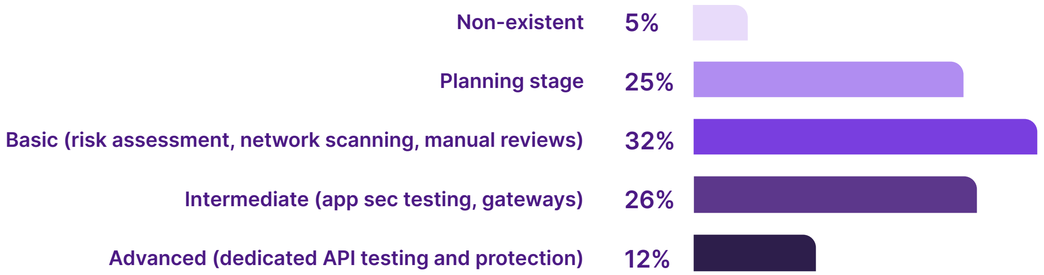
With reliance on APIs at an all-time high and critical business outcomes relying upon them, it is even more imperative that organizations build and implement a strong API security strategy. Unfortunately, **only 12% of respondents' organizations have what they consider to be advanced API security strategies** that include dedicated API testing and runtime protection. This number is up from 10% in Q3 2022, so security teams are making progress in this arena. Another 26% of respondents believe their API security strategy is intermediate, using application security testing and API gateways.

On the opposite side of the spectrum, **30% of respondents – all of whom have APIs running in production – admit they have no current API strategy**, with 5% saying such programs are non-existent and 25% saying they're in planning.

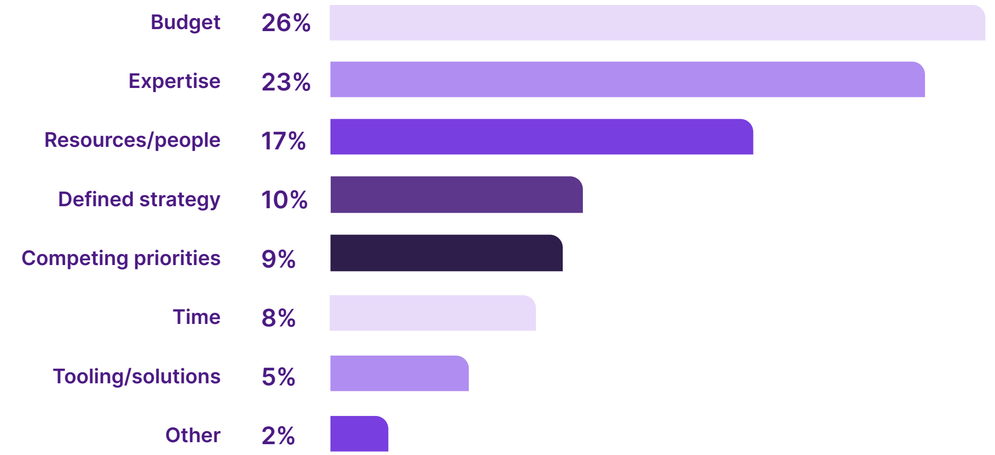
What's getting in the way of adopting such strategies? The same three obstacles have topped the list survey after survey – budget (26%), expertise (23%), and people resources (17%). These three have topped the list in every report save one – in Q3 2021 – when the second-highest obstacle cited was competing priorities.

Fortunately, today's API security solutions do not require much expertise or investment in people to manage them, and the return on investment is quite high.

How would you describe the security strategy for your API development program?



What is the biggest obstacle keeping you from implementing an optimal API security strategy?



What are security teams looking for in an API security solution?

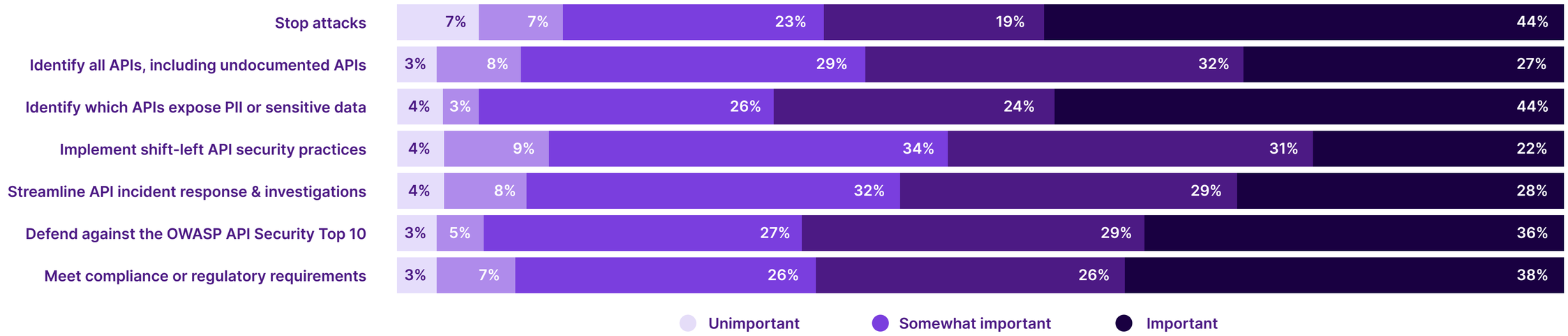
Respondents say they most value the ability to stop API attacks (44%) and identify APIs that expose PII (44%)

API security is taking center stage for many organizations, but what exactly are they looking for? The capabilities that respondents identified as **most valuable were the ability to identify which APIs expose PII or sensitive data (44%), stop attacks (44%), and meet compliance or regulatory requirements (38%)**.

Respondents considered the **ability to implement shift-left API security practices as their lowest valued attribute, with only 22% citing it as highly important**.

It is also interesting that respondents seem not to value the ability to identify all APIs or streamline API incident response and investigations. These findings may reinforce that organizations are not aware of how many shadow APIs they actually have and may not be very far down the path of operationalizing API security.

How do you rate the value of each of these attributes of an API security platform?



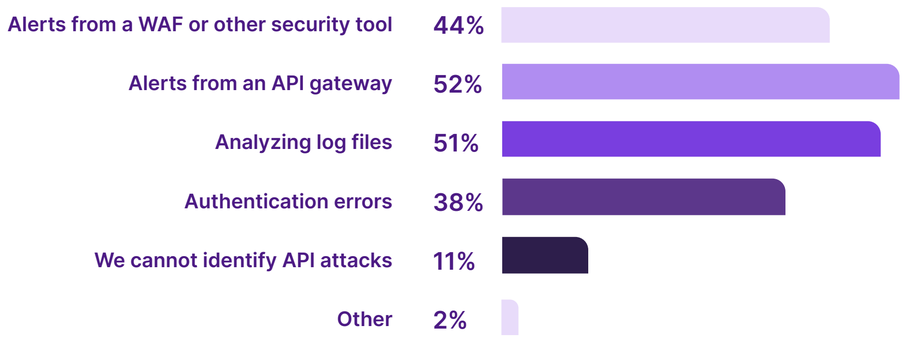
Traditional approaches to API security are falling short

Only 23% of respondents believe their existing security approaches are very effective at preventing API attacks

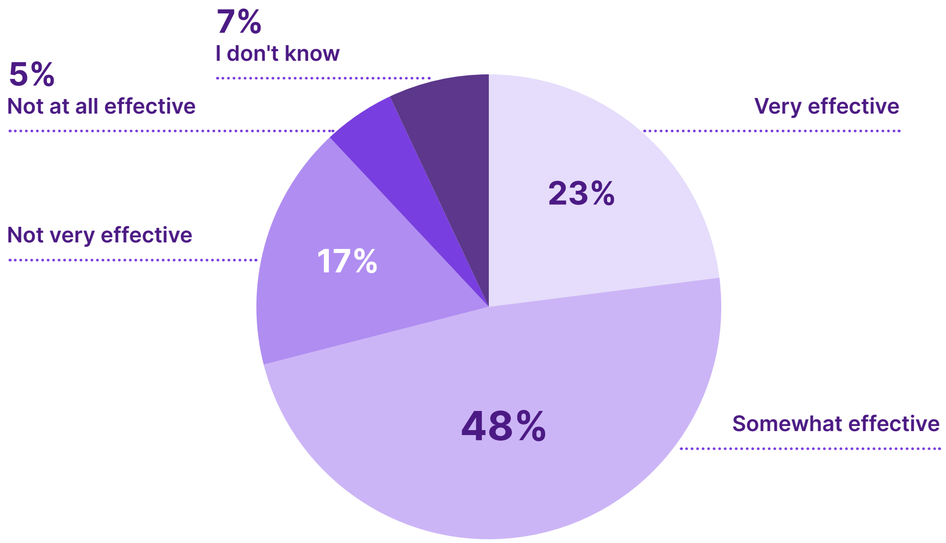
As in previous surveys, this quarter's respondents indicated that they primarily rely on traditional tools and processes to secure their APIs. However, they don't believe these methods are particularly effective, with **77% of respondents saying their existing tools aren't very effective in preventing API attacks**.

While it is true that traditional approaches to API security provide some coarse application protection, they cannot spot – much less defend against – today's business logic-based API attacks. API gateways (52% of respondents) employ traditional protections such as authentication, authorization, encryption, and rate-limiting (on a coarse rather than per-user basis). Analyzing log files (51% of respondents) to identify API attacks is tedious, reactive, and highly ineffective – attackers will be long gone with valuable data by the time a security analyst can parse log files. WAF alerts (44% of respondents) are known to be ineffective since WAFs use proxy architectures to apply signatures that detect only well-known attacks such as cross-site scripting (XSS), SQL injection (SQLi), and JSON injection. WAFs can't stitch together the data needed to spot today's API attacks.

How do you identify an attack or attacker targeting your APIs?



How effective are your existing security tools in preventing API attacks?



APIs are changing constantly and documentation is failing to keep up

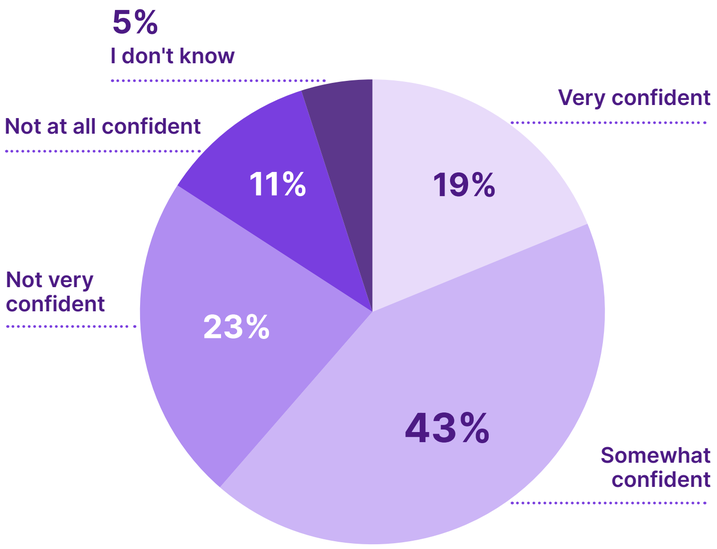
37% update their APIs at least weekly, but 48% update their documentation less than twice a year

Having a comprehensive view of your API attack surface is widely agreed to be the first step to protecting APIs. Unfortunately, respondents tell us that their **confidence in a complete and accurate API inventory is low, with only 19% saying they feel very confident**. 34% are either not at all or not very confident. And 43% fall somewhere in between, saying that they are somewhat confident in their inventory.

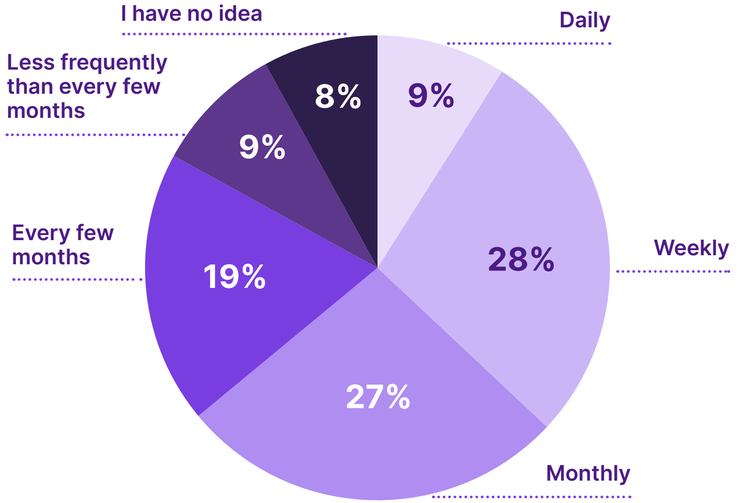
Why? APIs are constantly changing, making them nearly impossible to document well. **37% of organizations update their APIs at least weekly**, up from 32% in Q3 2022. And **9% update their primary APIs on a daily basis**.

Even if these APIs were initially documented, the frequency of documentation updates does not keep up with the frequency of API changes. **OAS and Swagger files are updated at least weekly in only 12% of organizations**. 20% update documentation with no regular cadence, and 23% update it approximately every six months. These gaps reinforce the shortcomings of relying on shift-left practices for securing APIs.

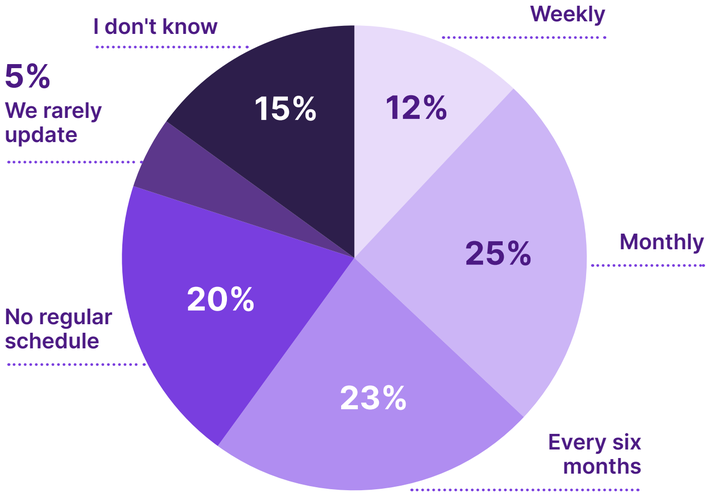
How confident are you that your API inventory is complete?



On average, how often are your primary APIs updated?



How frequently do you update your OAS or Swagger files?



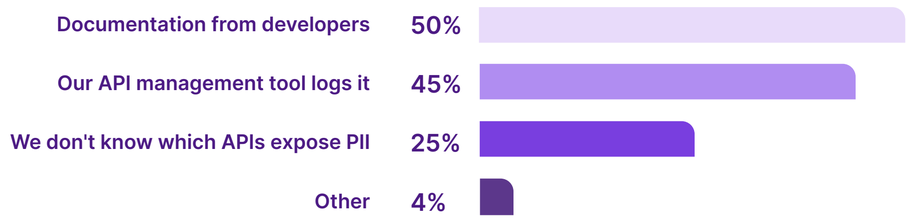
Security teams have a difficult time understanding which APIs expose PII

Only 18% are very confident they understand which APIs expose PII data

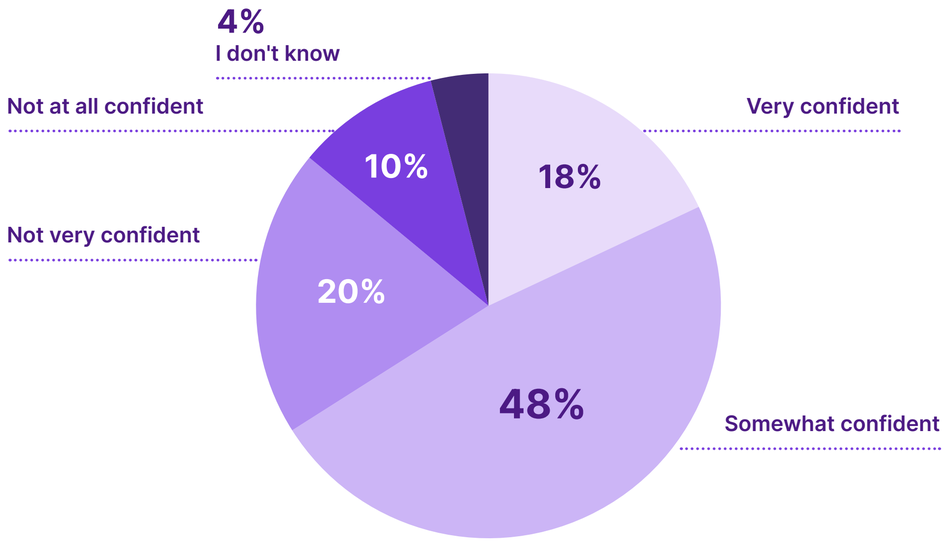
Respondents are less than confident in their ability to recognize what sensitive or personal identifiable information (PII) is exposed within their APIs. Only **18% say they are very confident that their API inventories provide enough detail about their APIs and the sensitive data within**. On the other hand, 30% admit that they lack confidence in this area. Respondents have maintained a similar level of concern about improper PII documentation throughout all of the State of API Security reports, having ranged between 20% and 30% over the past two years.

Unfortunately, the lack of confidence that respondents have in their organizations' ability to properly document PII within their APIs is to be expected. **The tools they are relying on to discover the sensitive data within their APIs include logs from their API management tools (38%) and developer documentation (41%)**, which we already understand are dangerously lacking. It is therefore also not surprising that 31% have experienced sensitive data exposure (Page 6).

How do you know which APIs expose sensitive data or PII?



How confident are you that your API inventory provides enough detail about your APIs, including exposure of sensitive data or PII?



APIs continue on their explosive growth trajectory

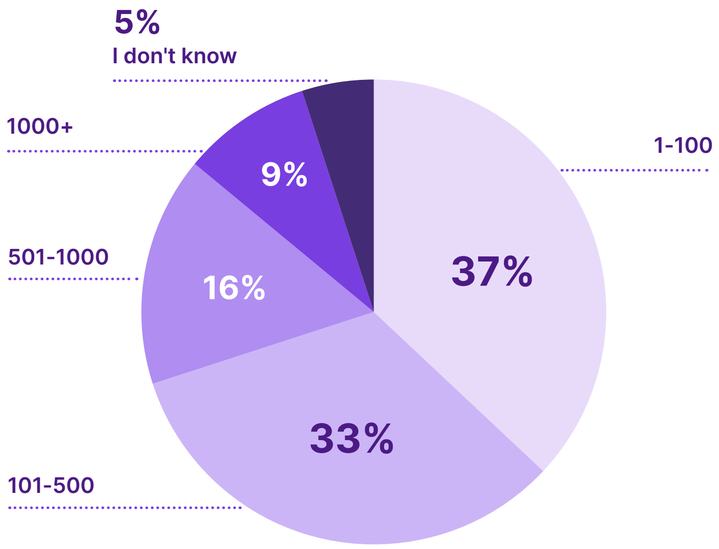
59% of respondents now manage more than 100 APIs, and 27% have more than doubled their API count over the past year

APIs fuel today's digital economy and enable organizations to deliver the services that their customers expect. So it's not surprising that survey respondents tell us they are experiencing dramatic growth in the APIs they manage.

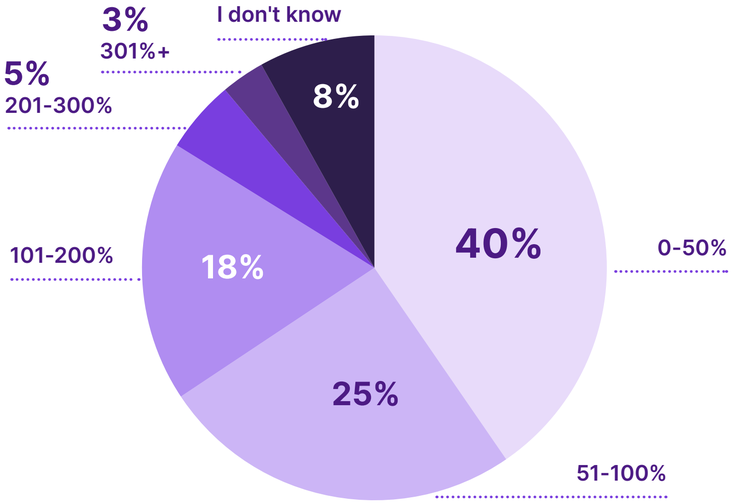
59% of respondents now manage more than 100 APIs, and 25% manage more than 500. This number is only growing, with **27% saying they have more than doubled their API count over the past year**. Another quarter said their API numbers increased by 51-100%.

Along with outright API number growth, the number of requests sent to respondents' APIs each month has also grown. The number of respondents citing the smallest bracket of API requests (0-10 million requests per month) is down to 32%, compared to 37% six months ago. On the other end of the scale, **APIs processing more than 500 million requests grew from 11% six months ago to 16% today**.

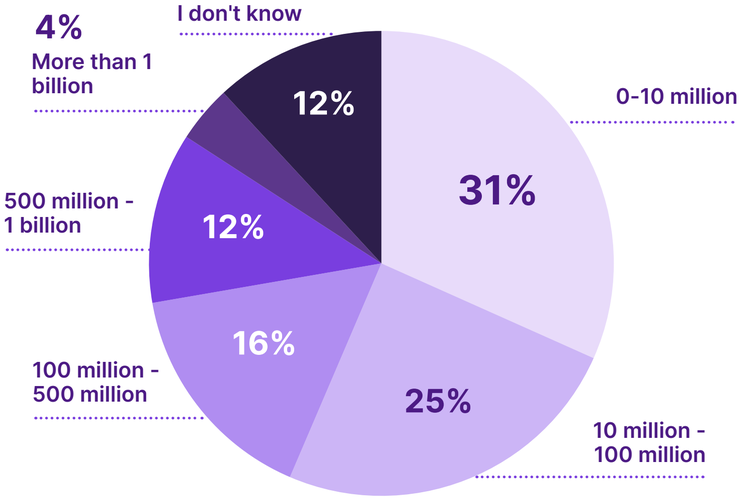
How many APIs does your organization develop, deliver, and/or integrate?



By how much has the number of APIs increased over the past 12 months?



How many requests are sent to your applications' APIs each month?



Salt Labs research: Vulnerabilities discovered in the wild

Our researchers uncover API security vulnerabilities in 90% of our investigations, and 50% of them should be considered critical

Salt Labs is the research division of Salt Security, and as such our mission is to constantly identify and surface API vulnerabilities in major online websites and services. Our researchers are continuously probing these services – old and new, big and small, across all geographical regions and business sectors. We opt to publish a subset of these important findings as part of our efforts to educate the industry about API security.

In this section of the report, we wanted to augment the survey and empirical data to showcase some vulnerabilities that the Salt Labs team has recently discovered. While these particular vulnerabilities have been disclosed to the companies involved and the issues have been resolved, we have chosen to anonymize the companies and applications – the focus should be on the nature of the security gap, not on a particular company who had that gap, because our research shows that when one service has a flaw like one of these, many others do as well.

It is interesting to note that 41% of survey respondents stated that they had identified a vulnerability in their production APIs. This number has fluctuated between 39% and 55% since we began conducting this survey, but Salt Labs research indicates this number is substantially higher.

One unique and important point to consider when dealing with API security is that, as opposed to many other fields in security and offensive research, success rates (cases in which we found a significant API security issue) are very high. **Our research team uncovers API security vulnerabilities in 90% of the services we inspect, and 50% of those vulnerabilities are considered critical**. If we can find these security gaps, you can bet attackers will too.

These findings provide yet another very strong indication that API security is one of the most vital security disciplines today and that every organization employing a web service should make a concerted effort to invest time and resources into securing their APIs.

Found in the wild: a critical BOLA at a household brand

In this case, we inspected a very popular U.S.-based lifestyle service. This service is being used by millions of people, often on a daily basis. The service follows a pretty similar design pattern as the rest of the services in its business category. This design is completely web service-oriented, and it provides a communication channel between end-consumers, service providers, and the company itself.

Finding an API security issue in this service was quite challenging, as it was apparent that the service was well-designed and had passed an effective security validation. However, given the nature of rapid API development and how they're released (i.e., the CI/CD process, with APIs publishing in minutes), keeping pace with the security validation process for these new API endpoints is very challenging. So as researchers looking for gaps, we seek out these kinds of endpoints.

Our assumption was that at least one of them might lead us to find a substantial and serious security risk – and, we were right. After inspecting some of the JavaScript code delivered as part of the service interaction, we found several endpoints that did not seem to have a specific purpose and were not part of the main business logic flows.

One of these endpoints was actually a GraphQL endpoint (while the rest of the published services were exclusively REST-based). Digging a bit deeper into this endpoint, we quickly realized that the "Descriptive GraphQL Errors" feature was enabled, which allowed us to quickly and easily map the entire functionality of this endpoint.

After we gained a solid understanding of the endpoint's functionality, we started looking for security issues, and we were able to spot a strong potential for a BOLA. We were able to confirm that a BOLA attack would succeed, which illuminated the significant business impact this vulnerability could have had.

An attacker exploiting this vulnerability would have been able to extract the entire user database. The data set included many PII data components, such as full name, address, email address, phone number, partial credit card numbers, shipping addresses, and much more. It also included internal service information such as the roles and permissions defined for each account. The data set contained both customers as well as internal company employees.

A malicious actor finding these gaps, rather than us, would have caused substantial damage to this service, its customers and employees, and the company's reputation. Fortunately, we discovered it first, notified the company, and worked with the team to quickly resolve the issue and confirm no exploit had taken place.

Found in the wild: Broken authentication to open crypto wallets

The volatile crypto market has been a hotbed of innovation and opportunities. With such meteoric growth (and now compression) over the past years, a completely new market emerged, paving the way for hundreds of new online crypto services, from online exchanges, marketplaces, online wallets, and many more.

The state of the crypto landscape today reveals a rapidly changing market, tapping new “cutting-edge” technologies while dealing with massive amounts of digital (and physical) currencies. The rapid pace of development creates an explosive situation in which the chance of finding security issues is high. Many of the security issues could lead to tremendous losses in scales which we have never witnessed before. So Salt Labs decided to look deeper into some of the most popular crypto services found on the Internet today.

One of these services handles several billion U.S. dollars in digital currency and serves millions of global users. One area ripe for exploration was its complex login functionality. Login functionality is the first step in a user’s interaction with a service. Upon successful login, users are identified and gain access to their data, currencies, digital assets, and other service elements. As a result, the login process is one of the most crucial places to protect from a security perspective.

In this specific case, this application’s login functionality included the popular option to “Login via external account.” This functionality is often implemented using several de facto standards including OAuth and OIDC. When inspecting the OIDC functionality implemented by this service, we were quickly able to spot something very strange.

In a “normal” OpenID Connect (OIDC) flow, a user should not be asked to pass username or credentials to the web service. Rather, the user is first directed to a trusted third party (Microsoft, Google, Facebook, and so on), which validates the user’s identity, followed by an out-of-band query by the service itself to confirm the identity. Only once the user’s identity has been verified will the user be permitted to log in and interact with the service.

While inspecting the OIDC flow in this crypto application, we noticed that the process was indeed following the guidelines. However, for some reason, the steps included the users transmitting their email directly to the crypto service. Our biggest question was “What would happen if a user tried to manipulate this login process by following the external login process while sending a different username to the crypto service?” This manipulation would create a conflict, and we were interested to see how the crypto service would handle this conflict.

As it turned out, taking this step revealed a very serious security issue. The crypto service respected the forged email sent by the user rather than the real user’s identity. In the most simple terms, an attacker logs in to the crypto service using a legit external account. However, instead of sending his or her own username to the crypto service, the hacker sent the email of a victim. The crypto service honored the victim's email address, and the attacker was now logged into the victim’s account, despite having no knowledge of the victim’s password or credentials. Once logged in, the attacker could then perform any action on behalf of the victim, including selling/buying currency, transferring currency, and gaining access to all of the victim’s PII information stored on the crypto service.

This type of authentication failure scenario could have potentially led to billions of dollars worth of damage to the service's users and shows how important it is to properly protect your authentication endpoints and to ensure you have a deep understanding of any new technology (in this case OIDC) you incorporate into your online service.

Recommendations and conclusions

Implications for API security

The results from the Q1 2023 State of API Security survey are clear. Respondents overwhelmingly told us that reliance on APIs is continuing to grow as APIs become ever more imperative to their organizations' success. At the same time, APIs are getting harder to protect as current tools and processes can't keep pace with new attack trends. Organizations must move from traditional security practices and last-generation tools to a modern security strategy that addresses security at every stage of the API lifecycle and provides a broad range of protections that foster collaboration across teams. Here are some tips to consider as you build a more robust and manageable API security program:

Define a robust API security strategy
WAFs and API gateways leave significant gaps when defending against API attacks, so companies need to define and execute an API security strategy that covers the complete API lifecycle and addresses cross-functional responsibilities. A comprehensive program must include API design analysis and drift analysis, automatic and continuous discovery, augmented runtime protections, a feedback loop for developers to use runtime insights to harden APIs, training for SecOps teams to understand and triage API security incidents, and a clear model for shared responsibility across functional groups.

Assess your current level of risk
Validate current API designs against API security best practices, checking whether authentication and authorization controls are in place throughout the sequence of API calls for a given business function, for example. Launch simulated attacks based on the OWASP API Security Top 10 list to understand the gaps in protection from WAFs and API gateways. Emulate the tactics of well-known API security incidents of 2022 to see whether similar business logic flaws exist in your APIs.

Enable frictionless API security across all your application environments
With APIs being the foundation of all application development today, you can't afford to leave some of your environments unprotected. You must be able to apply API discovery and runtime protection on prem and in the cloud and on legacy apps, as well as your container and Kubernetes deployments. How you connect the API security tooling into your environments is also crucial – avoid inline deployments, agents, or the need to instrument code to keep your API security platform from being blamed for any application impact.

Focus on robust runtime security
No one will ever write perfectly secure code, so runtime protection provides immediate and continuous risk reduction. Since every API is unique, bad actors must perform extensive reconnaissance to identify vulnerabilities or gaps in business logic they can exploit. Attackers know how to probe your systems with subtlety, to avoid tripping coarse security protections such as rate limiting on WAFs. To see these nefarious but quiet activities, an API security platform must be able to capture millions of data points over a long period of time, since API attacks can take weeks and months to unfold. Then, the platform must tap AI and ML to process all that data in near real time, so it can discern the recon activities of a bad actor and correlate them into a single attacker profile to avoid alerting on each bad action. Such robust analysis requires cloud-scale big data and mature AI algorithms – it cannot be achieved with on-prem API security and immature AI and ML.

Don't over-rotate on shift-left tactics
Shift-left and secure build pipeline approaches have their merits. But most API security gaps can't be detected as part of pre-prod API testing – they can be detected only in runtime. Look for an API security platform that complements pipeline testing and OAS analysis with robust runtime protection. Shift-left tactics take much longer to deliver value, ultimately offer limited value since they can identify only a fraction of API security gaps, and leave your security teams dependent on developers to work through a backlog of security fixes. Get your APIs protected today with runtime security – then you can make hardening APIs over time a realistic goal.

About the data

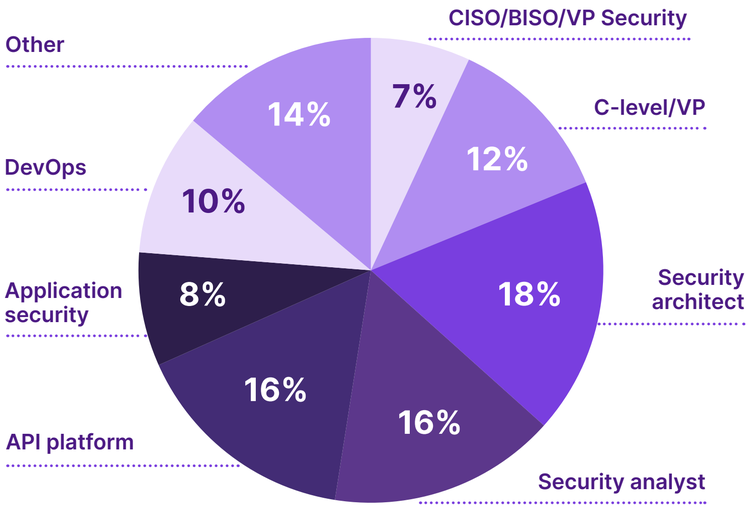
Insights from nearly 400 security professionals and API developers, plus analysis of real-world API attack attempts

These report findings combine live Salt customer data and the survey responses of 378 respondents. The survey respondents were fairly evenly distributed across a broad range of job responsibilities, industries, and company sizes. Nearly half (48%) hold roles in security, 19% are executive-level security or IT leaders, and 26% sit within the platform or DevOps teams. Technology and financial services companies – widely viewed as being at the forefront of API use – comprise 48% of respondents. Companies large and small were evenly represented.

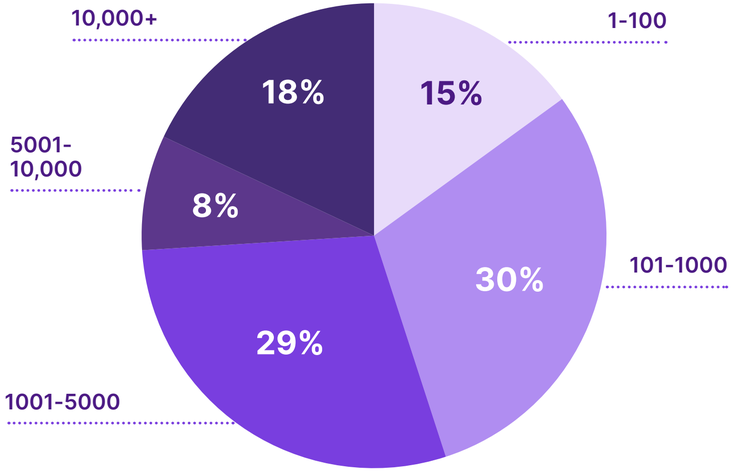
The report also includes real-world API attack attempt data from the Salt Security API Protection Platform. This empirical customer data is anonymized, aggregated, and then analyzed by Salt API security researchers to identify critical trends that can help educate the broader security industry.

Finally, the "in the wild" vulnerability research comes from our in-house research arm. Salt Labs, the industry's only dedicated API research team, undertakes projects to more deeply understand the evolution of API attacks to improve the Salt platform detection models and educate the companies involved and the industry as a whole.

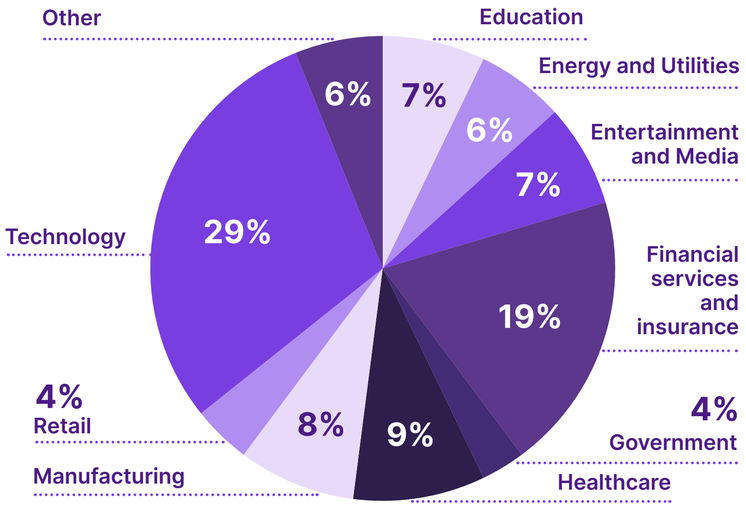
What area best represents your functional role?



Size of company (employee count)

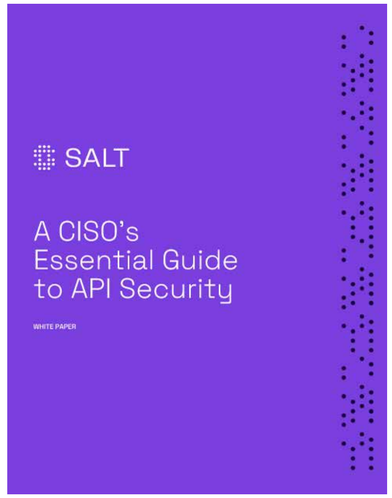


Industry

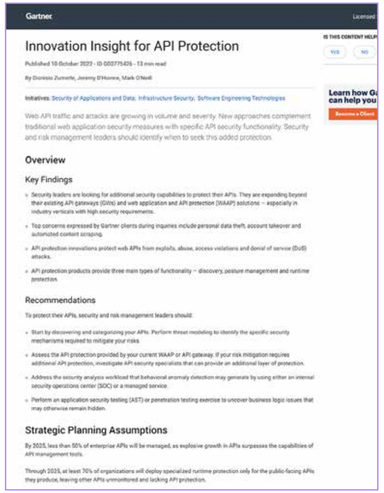


Additional resources

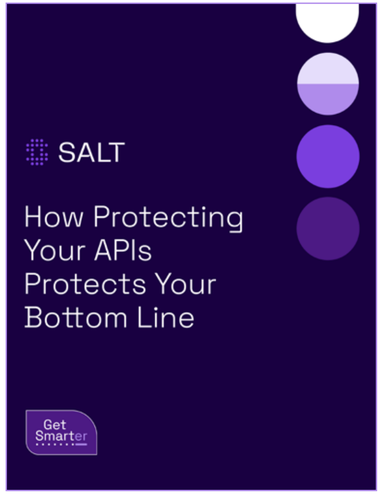
These key assets will help you get even smarter about API security.



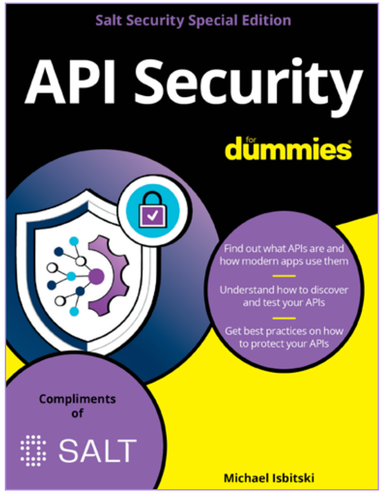
[A CISO's Essential Guide to API Security]



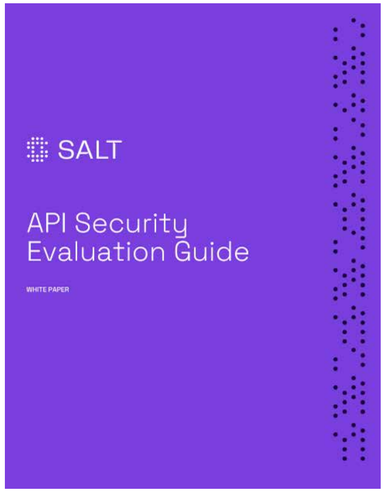
[Gartner® Innovation Insight for API Protection]



[The Business Value of API Security]



[API Security for Dummies]



[API Security Evaluation Guide]

About Salt Security

Salt protects the APIs that form the core of every modern application.

The Salt Security API Protection Platform secures your APIs across the full API lifecycle. The Salt platform collects a copy of API traffic across your entire application landscape and uses big data, machine learning (ML), and artificial intelligence (AI) to discover all your APIs and their exposed data, stop attacks, and eliminate vulnerabilities at their source. The Salt platform:

Discovers all APIs and exposed data – Automatically inventory all your APIs, including shadow and zombie APIs, and highlight all instances where your APIs expose sensitive data. Continuous discovery ensures your APIs stay protected even as your environment evolves and changes with agile DevOps practices.

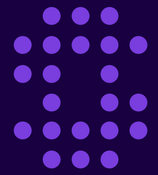
Stops API attackers – Pinpoint and stop threats to your APIs by identifying attackers early, during their reconnaissance phase, and prevent them from advancing. The Salt platform correlates activities back to a single entity, sends a consolidated alert to avoid alert fatigue, and blocks the attacker rather than transactions.

Improves your API security posture – 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.

SALT LABS

About Salt Labs

Salt Labs identifies API threats and vulnerabilities in customer deployments and in the wild. Our in-depth API threat research reports document the steps of an exploit, including the processes and tooling, to reveal an attacker's approach, the details of an exploit, the risk to the business, and the steps an organization can follow to avoid falling victim to a similar attack. We also apply our research findings to improve the ML and AI algorithms at the heart of our API security platform, so that all our customers benefit from our ongoing research. Our industry reports, such as this State of API Security Report, tap empirical and survey data to educate the market on API security trends.



SALT

Securing Your Innovation