SALT SECURITY

State of API Security

Q1 2021

State of API Security

APIs are everywhere, enabling the applications and services we increasingly depend on and take for granted. In our personal lives, APIs fuel our shopping, banking, healthcare, socializing, and entertainment. In our professional lives, APIs increase our productivity, driving services such as video conferencing, file sharing, and project management. APIs are also the lifeblood of organizations, found at the core of the modern digital platforms underlying digital transformation, driving revenue, and enabling rapid innovation.

APIs have been around for decades but have changed dramatically, especially in recent years.

We've seen an explosion in the use of APIs, underpinning modern web, mobile, and cloud-native application design. APIs help enable rapid delivery as seen with DevOps practices, allowing developers to quickly develop, build, and release new functionality. As use cases have evolved for modern web and mobile applications, more sensitive data is being sent over APIs to enable services. APIs are also driving ever more critical services in the applications they enable.

As the number and functionality of APIs has grown, so has their attraction to attackers. If Willie Sutton robbed banks because "that's where the money is," today he'd hack APIs, because that's where the money is now. As APIs have increased in both business value and risk factor, API security has emerged as a key priority for today's organizations.

To understand the state of API security today, Salt Security has compiled the industry's first API security report. Our pioneering research combines survey responses and empirical data from Salt Security customers. The survey data comes from the responses of nearly 200 security, application, and DevOps professionals across companies big and small, from a variety of industries. The Salt Security customer data comes from anonymized, aggregated data in the SaaS platform portion of our customers' deployments.

Many findings are quite unsettling. The vast majority of organizations are experiencing API security problems, few have the tools needed to cope, and most have had to delay innovation as a result. We encourage you to benchmark yourself against the data in this report and use these findings to guide your own organization's approach to improving API security.

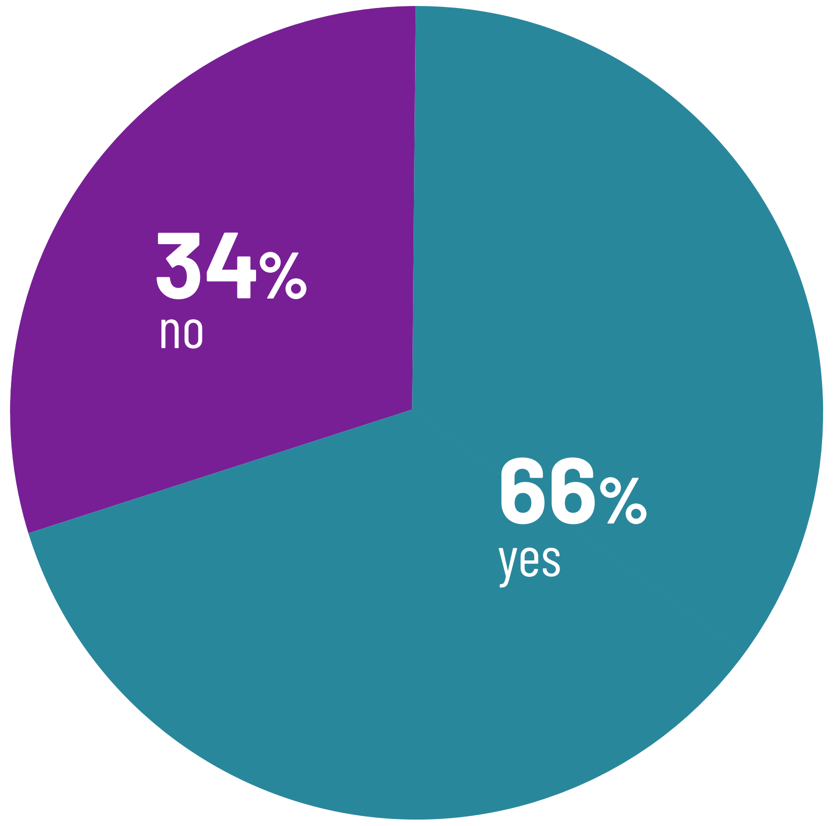
API security concerns are inhibiting business innovation

Two thirds of organizations have delayed rolling out new applications over API security concerns

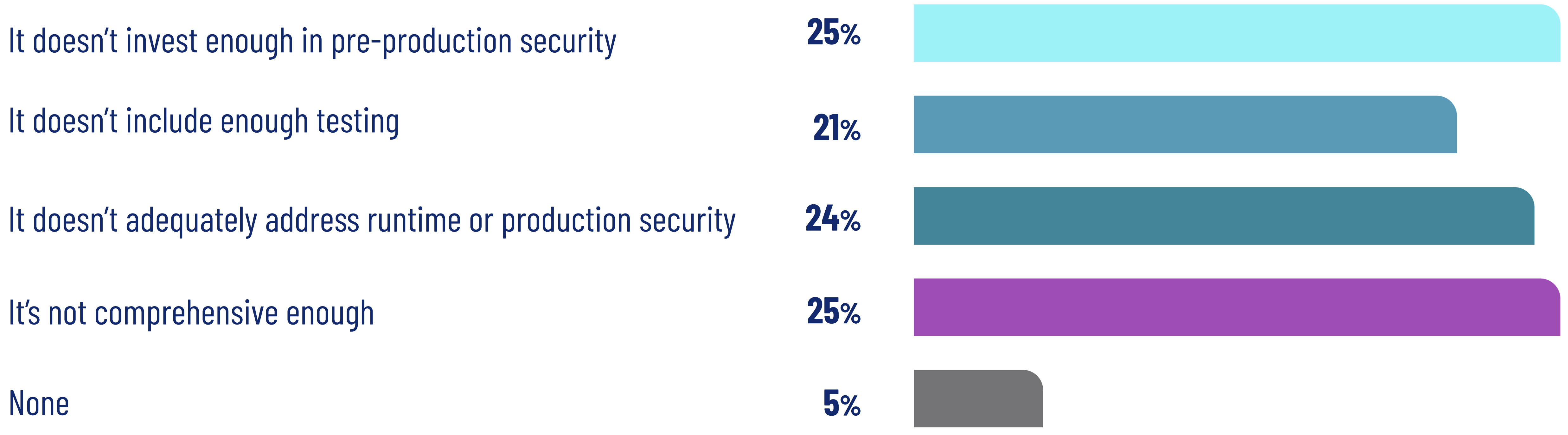
Application development and integration is the heartbeat of business innovation – delivering differentiated services, opening new revenue streams, and ensuring customer satisfaction. Given that APIs form the core of most modern apps, API security has a clear impact on application delivery. A full 66% of survey respondents stated that they have delayed the deployment of a new application because of API security concerns. When businesses cannot meet the demands of continuous application delivery, they hamper their digital transformation and DevOps initiatives and they cede ground to their competition.

Nearly half of respondents also cite security as the top concern about their API program, across both pre-production and runtime security. Organizations need a full lifecycle approach to API security to have the confidence to deploy new API-based applications at the speed that business demands.

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



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



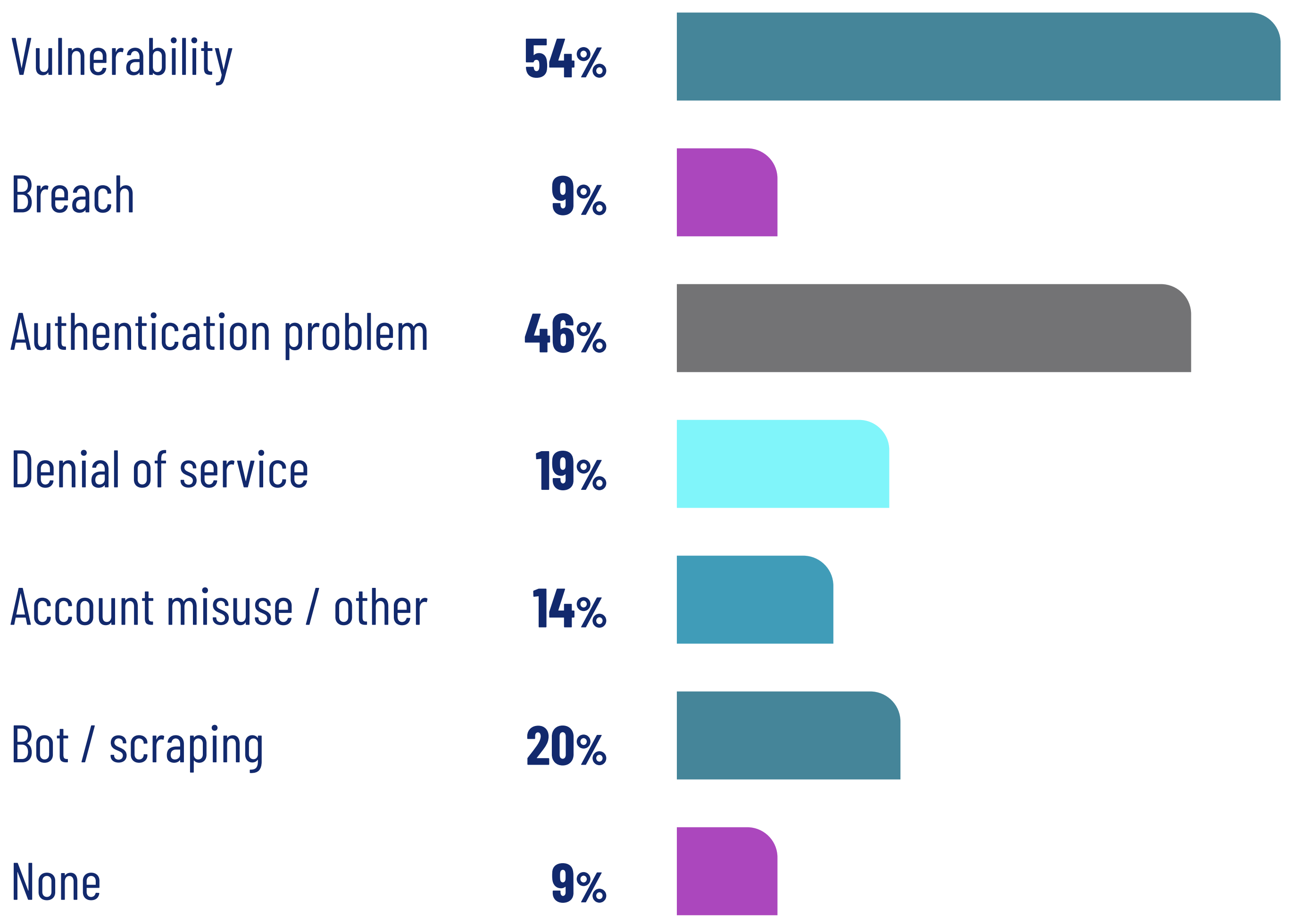
91% of respondents experienced an API security incident last year

Vulnerabilities and authentication issues top the list

APIs continue to spread across data center and cloud environments and increasingly expose sensitive data, making APIs a prime target for attackers. Only 9% of respondents didn't suffer an API security incident over the past 12 months. More than half of respondents found a vulnerability in production APIs, meaning that pre-production security efforts, while crucial, cannot provide the full answer. These vulnerabilities remain until an attacker discovers and exploits them, which can result in data exfiltration, account misuse, or service downtime. Nearly half experienced authentication problems as well.

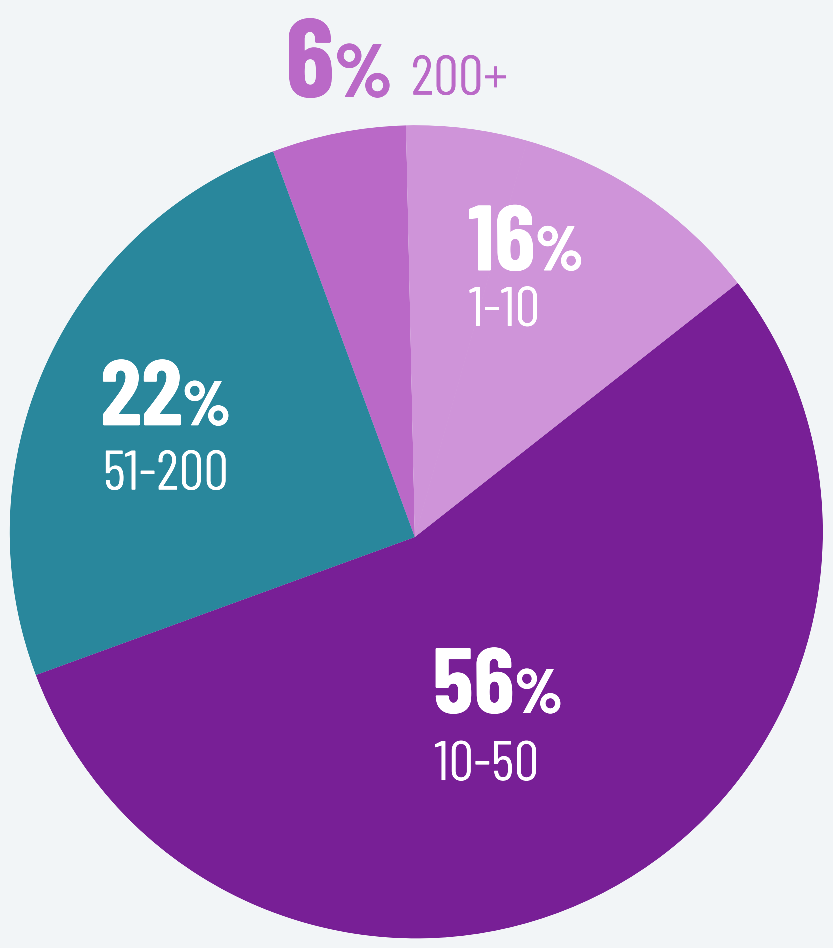
Looking at the Salt Security customer data, no customer experienced zero attacks any month last year, and 84% suffered at least 10 attacks per month. Slightly more than half experienced 10 to 50 attacks in a month, and an unfortunate 6% suffered more than 200 attacks every month. Fortunately, the Salt platform thwarted these attackers each time. Given the increased rate of attacks targeting APIs, it's not hard to understand why API security concerns have slowed application rollouts (page 2).

In the past 12 months, what security problems have you found in production APIs? (pick all that apply)



Salt customer data

Average number of API attacks per month per customer



API traffic is growing, but malicious API traffic is growing faster

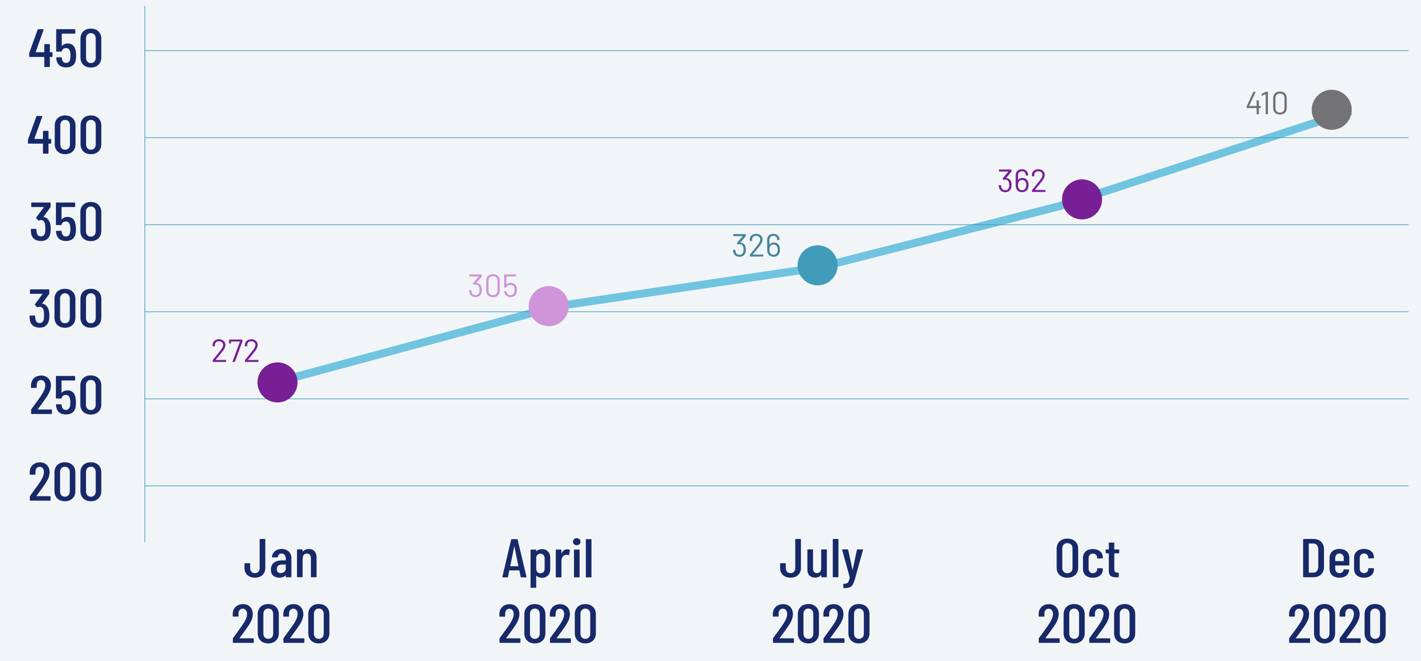
Salt Security customers' monthly volume of API calls grew 51%, while the percentage of malicious traffic grew 211%

Through a combination of new APIs, new API endpoints, and new functions in existing APIs, the per-customer average monthly API call volume for Salt customers increased over the past 12 months, from 272 million calls per month at the start of 2020 to 410 million at the end of the year. The Salt Security platform, after baselining all APIs, detects changes in API traffic. It applies its AI engine to distinguish "safe" changes, such as a simple user error or a back-end API modification where all users' patterns change, from "bad actor" changes – the probing consistent with attacker reconnaissance.

In the same time period where our platform measured growth in API call volume at 51%, it measured growth in malicious traffic at 211%. At the start of the year, 0.45% of all our customers' API traffic was malicious, and by year end, that percentage had climbed to 1.40%. Note that our customers all have WAFs and API gateways deployed, so this malicious traffic got past those devices. Such findings are consistent with broader industry research showing APIs rising to be the dominant application attack vector.

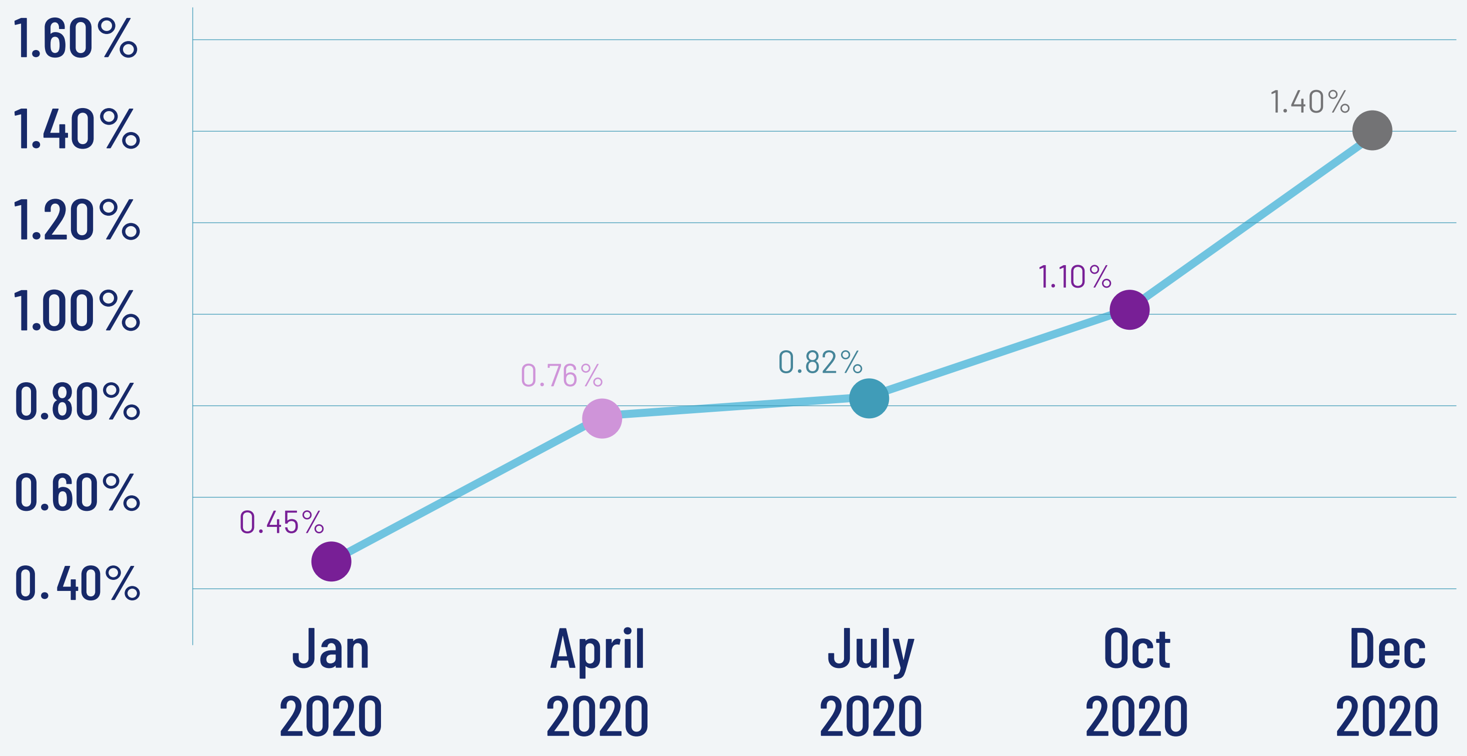
Salt customer data

Monthly API calls, in millions (average across all customers, last 12 months)



Salt customer data

Malicious traffic as a percent of overall traffic (average across all customers, last 12 months)



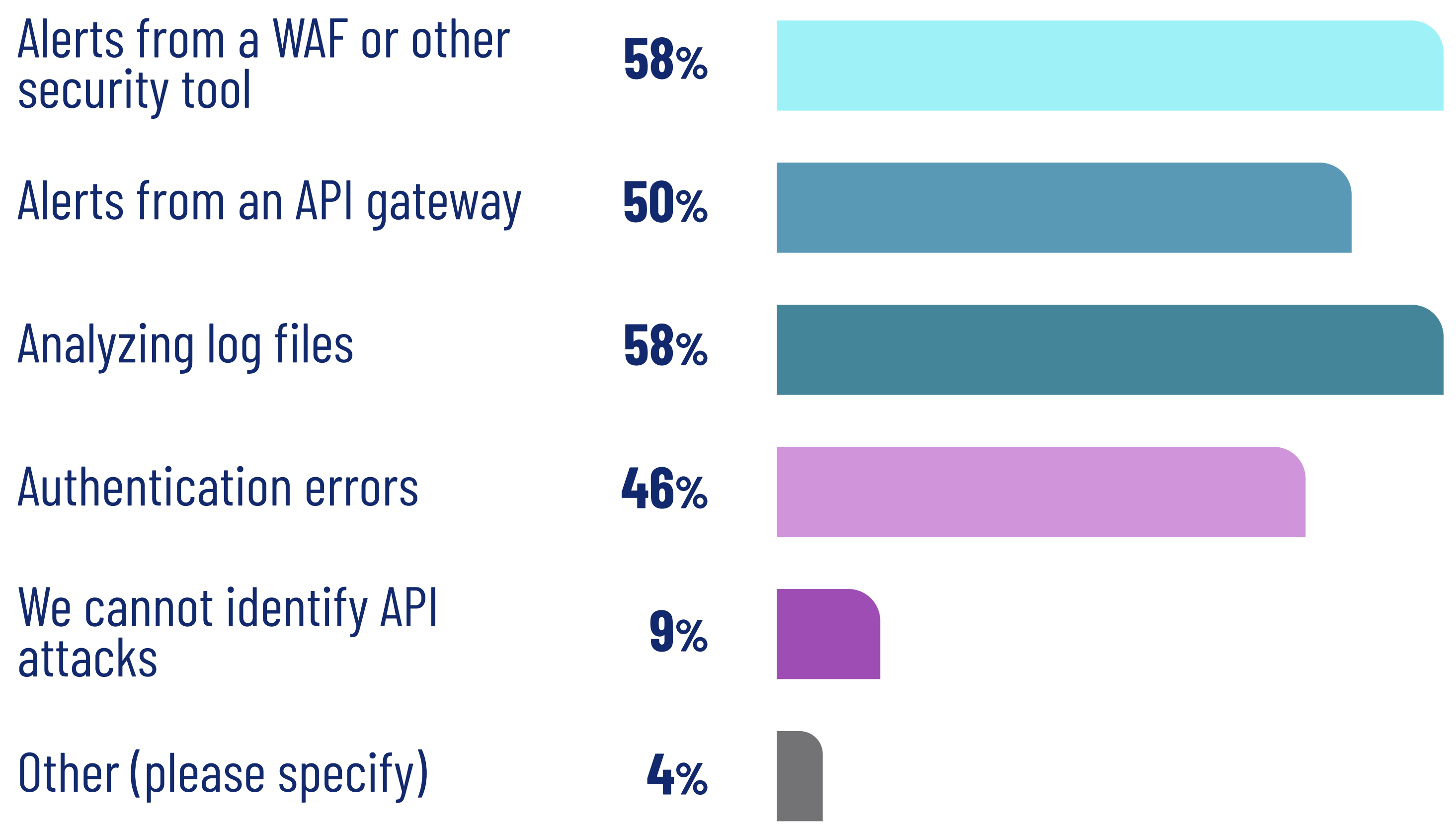
WAFs and API Gateways cannot stop API attacks

100% of Salt Security customers have WAFs and API gateways, and 100% of Salt Security customers have API attacks that get past those tools

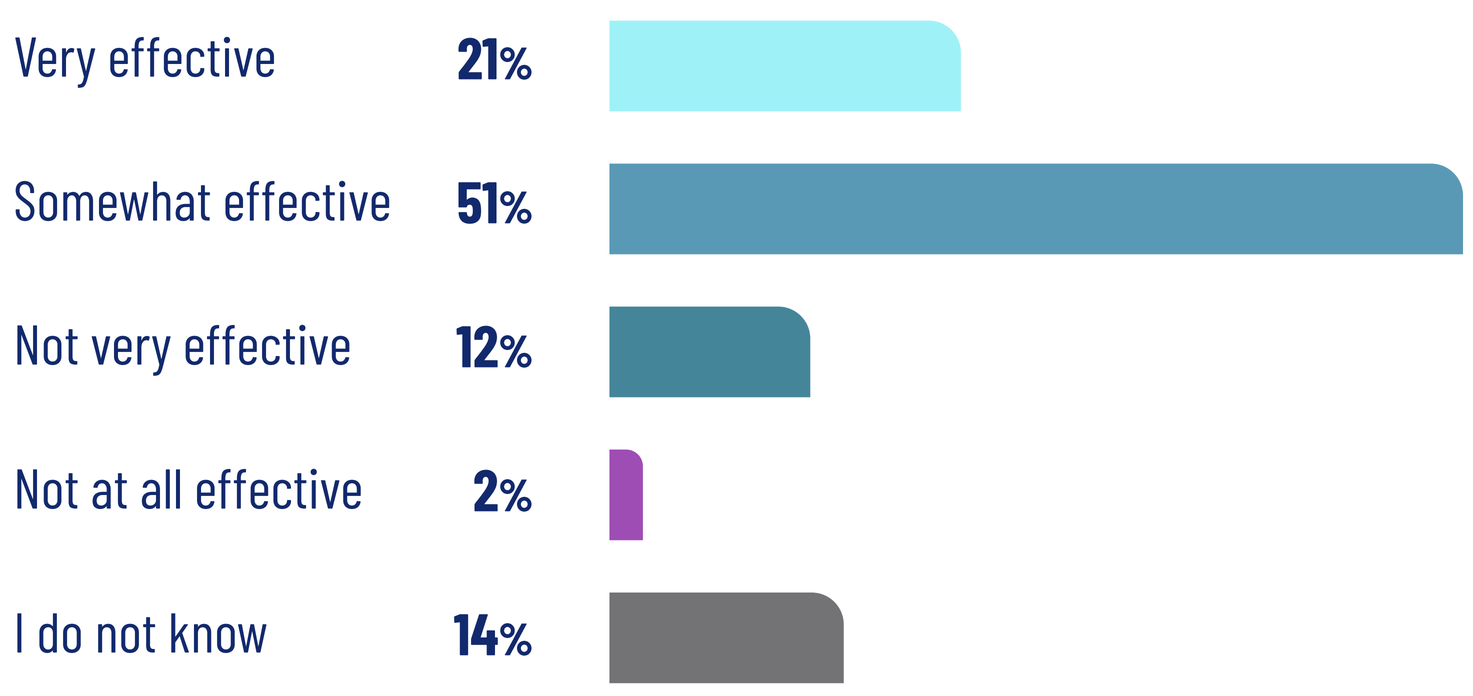
Every Salt Security customer is seeing attacks that get past their WAFs and API gateways, but more than half of the survey respondents cite using alerts from WAFs or API gateways to identify API attacks. Clearly these techniques leave organizations unprotected. Given that 96% of API exploits happen against authenticated APIs, clearly the protection techniques common in WAFs and API gateways (TLS, rate limiting, and IP allow/block lists for example) are insufficient.

Nearly 60% say they analyze log files to identify an attack - an after-the-fact approach to protecting APIs and one that cannot scale. Nearly a tenth of respondents admit they have no mechanism to identify API attacks, and 79% confess their existing approaches to API security are, at best, only somewhat effective. The challenge is that the dominant approaches of WAFs and API gateways miss 90% of the threats highlighted in the OWASP API Security Top 10 list of threats.

How do you identify an attack or attacker targeting your APIs? (pick all that apply)

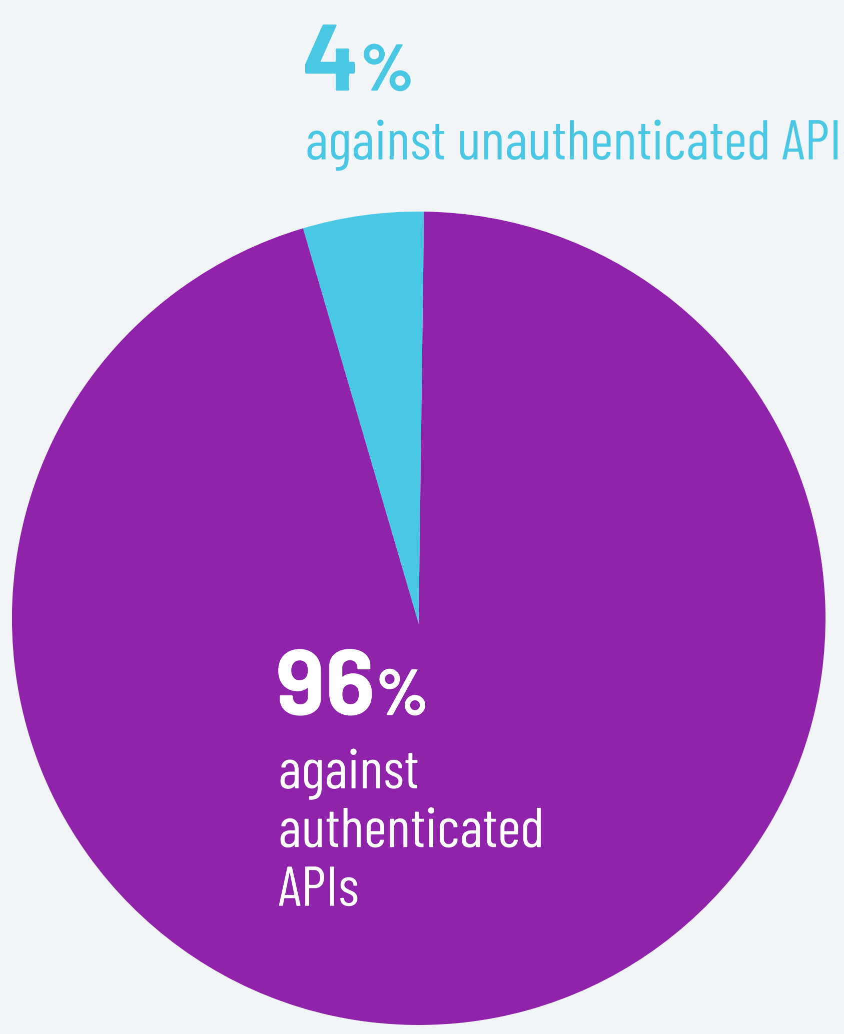


How effective are your existing security tools in preventing API attacks?



Salt customer data

% API exploits againt authenticated vs. unauthenticated APIs

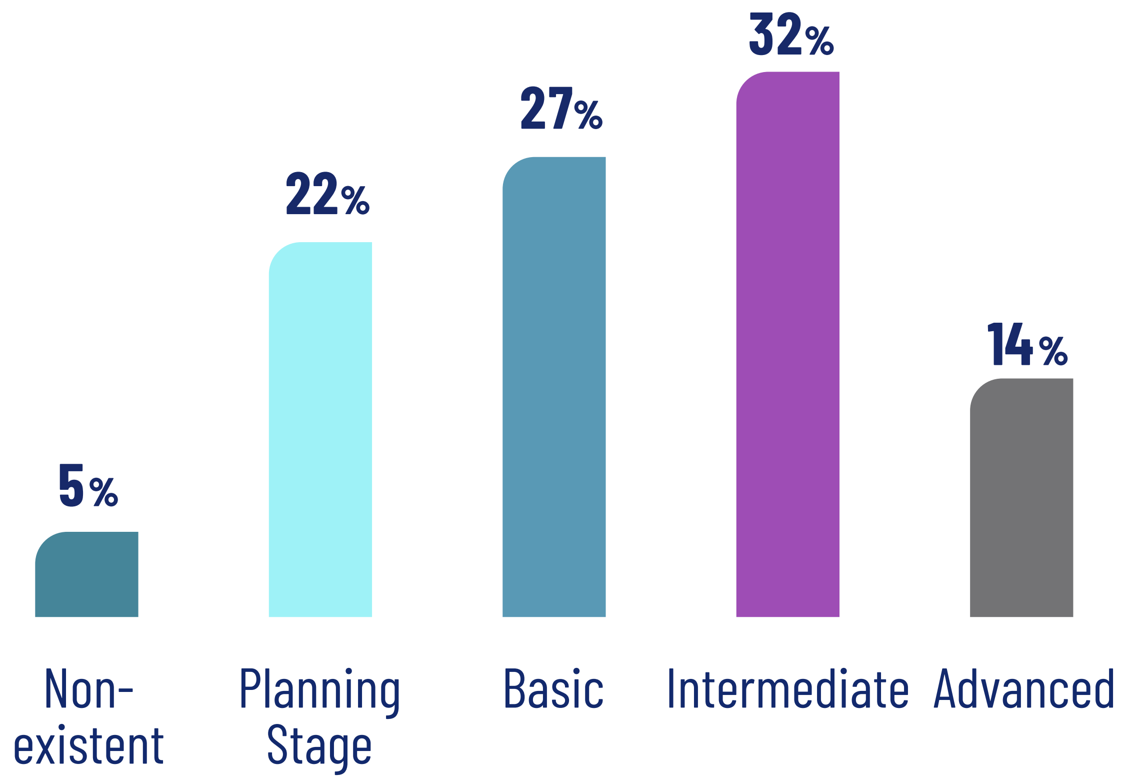


More than a quarter of organizations running production APIs have no API security strategy

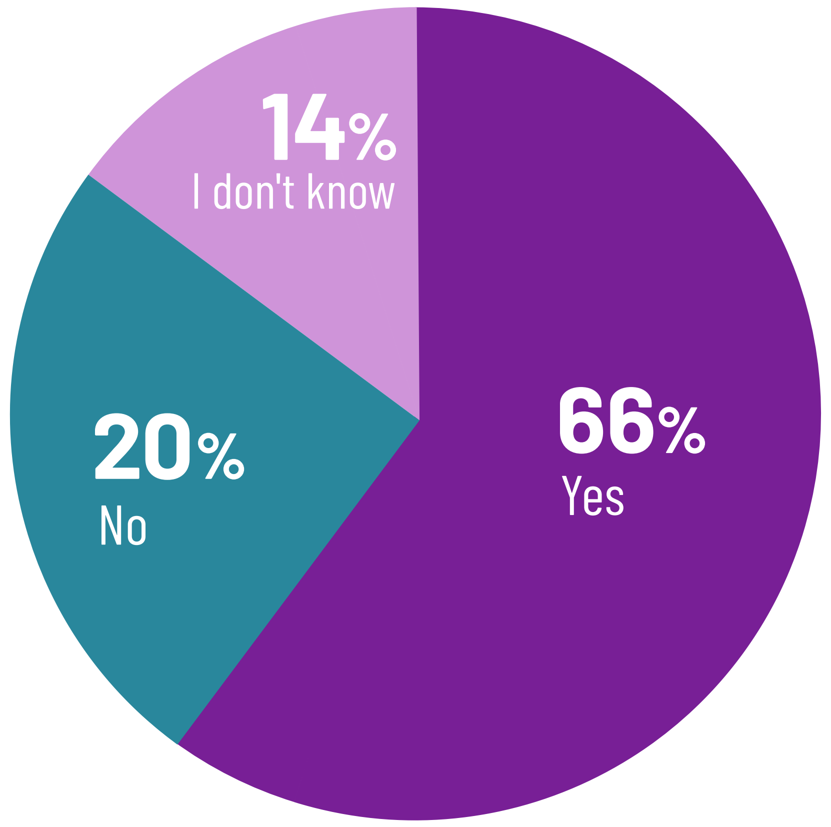
Another 27% have only a basic strategy for API security

With the rise of DevOps and rapid development, security is increasingly forced to play catch up. At the same time, new application architectures have rendered many traditional security approaches obsolete. Gartner predicts, "By 2022, API abuses will move from an infrequent to the most-frequent attack vector, resulting in data breaches for enterprise web applications. Already APIs have become the entry point of choice for attackers looking for valuable data to steal from enterprises." The bad news is that such a high percentage of respondents running production APIs lack any kind of API security strategy. The good news is that two thirds of respondents say their security teams have a focus on the OWASP API Security Top 10 threats. The conundrum is how so many organizations haven't translated that OWASP API Top 10 focus into an API security strategy. The time is now for CISOs to insist that security teams devise and implement an API security strategy. One approach is to assemble an API task force to craft a comprehensive approach to writing, managing, and securing APIs – across both pre-production and runtime.

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



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



Current API security approaches heavily rely on pre-production lifecycle phases

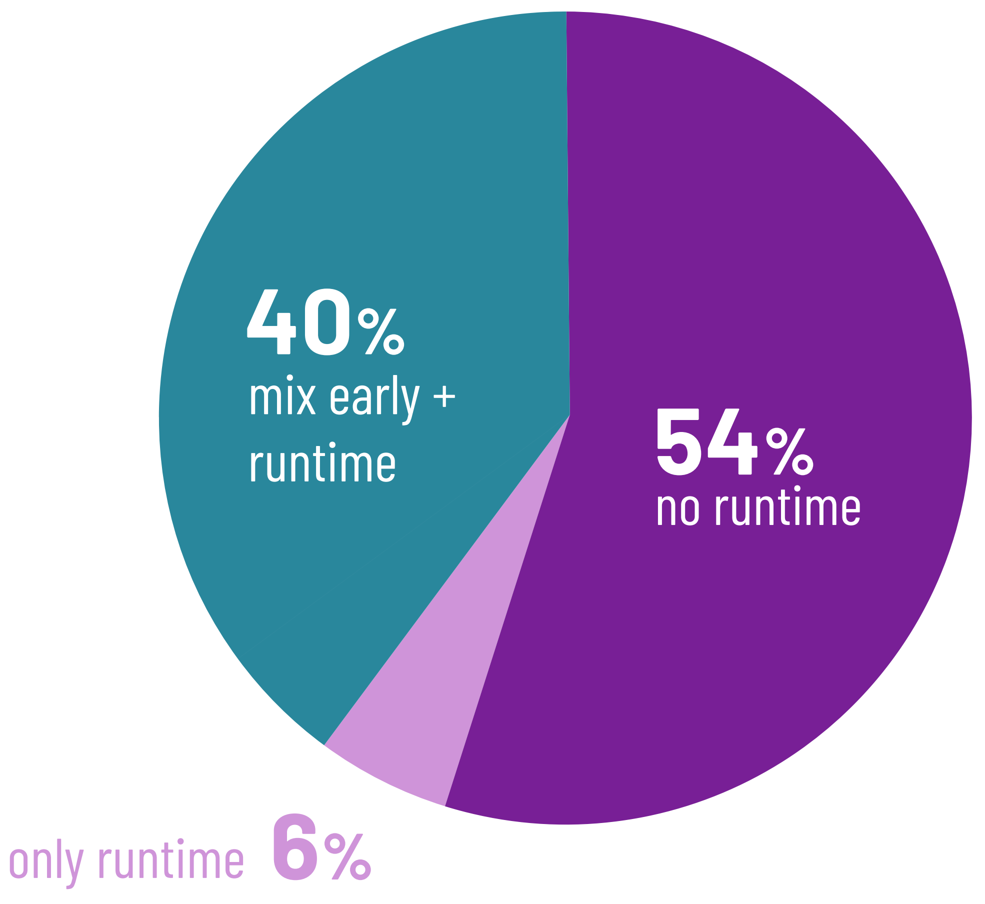
More than half of respondents apply no API security tactics during runtime

"Shift left" is a major and worthwhile goal for application security and DevSecOps initiatives. Improving secure development practices, scanning, and thorough testing can help improve the security of APIs as well as mobile and web apps. With only 46% of respondents applying runtime protection and 90% of respondents experiencing a security issue in production APIs (page 3), this overreliance on pre-production security tactics is leaving organizations vulnerable.

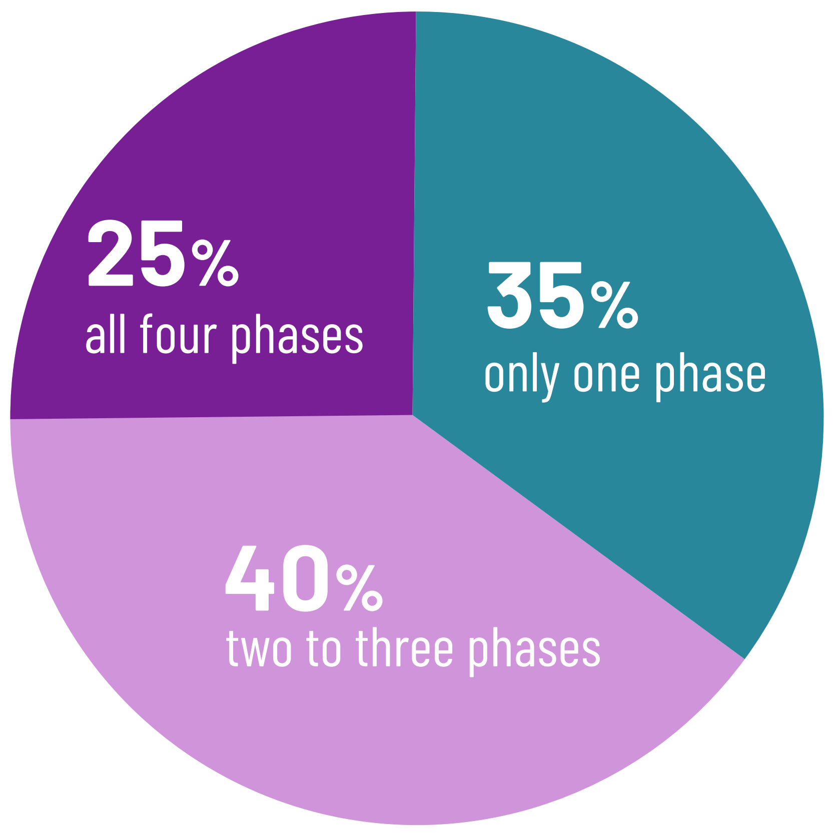
Only 40% of respondents are blending pre-production and runtime protections, and only 25% are following best practice and applying security tactics across every phase of the API lifecycle.

Full protection of APIs requires continuous improvement and coordinated efforts between security and development teams, spanning the full API lifecycle.

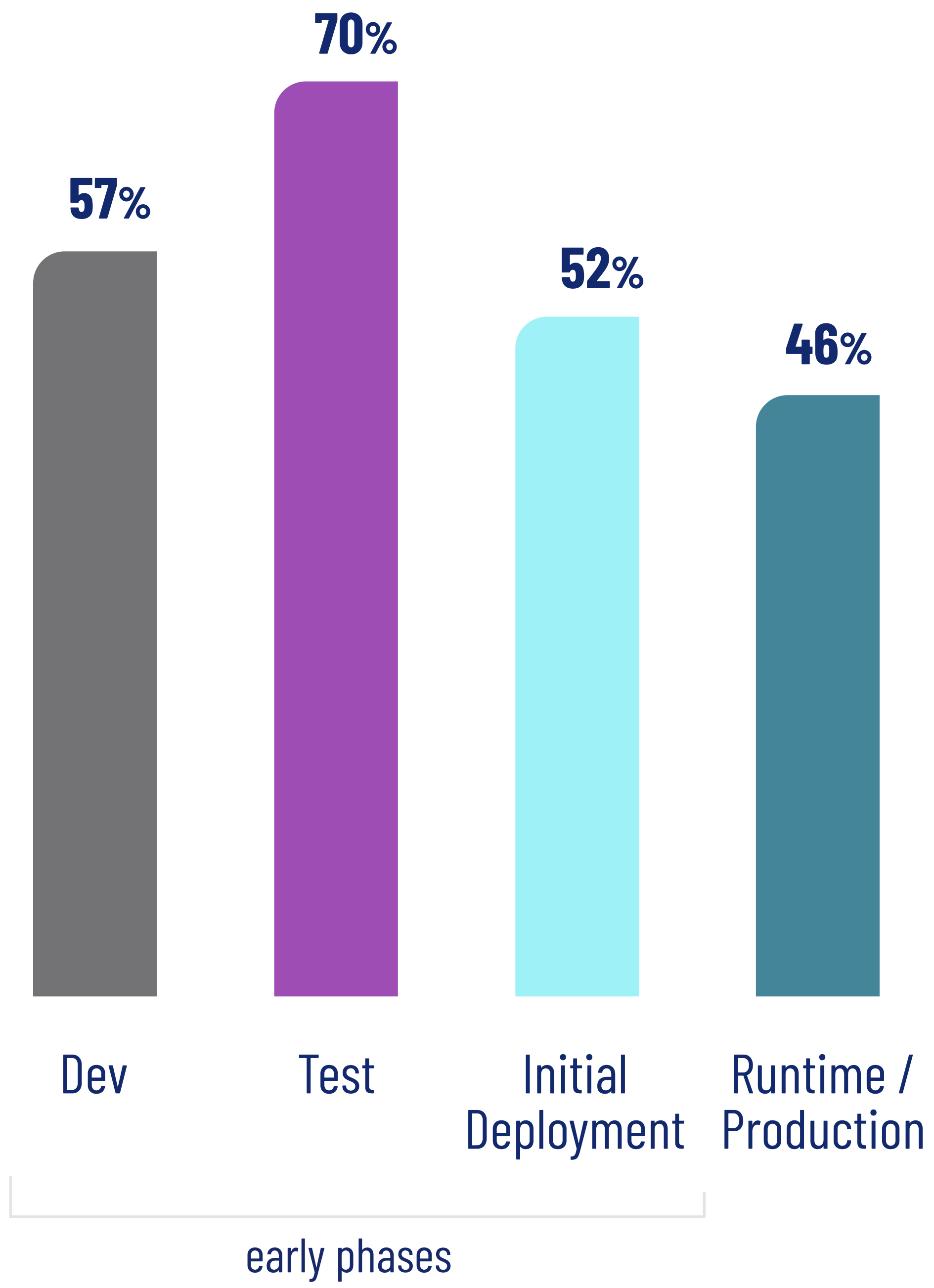
Mix by stages of lifecycle phases



Mix by number of lifecycle phases



At what phase(s) in the dev lifecycle does your company identify and remediate API security gaps (pick all that apply)?



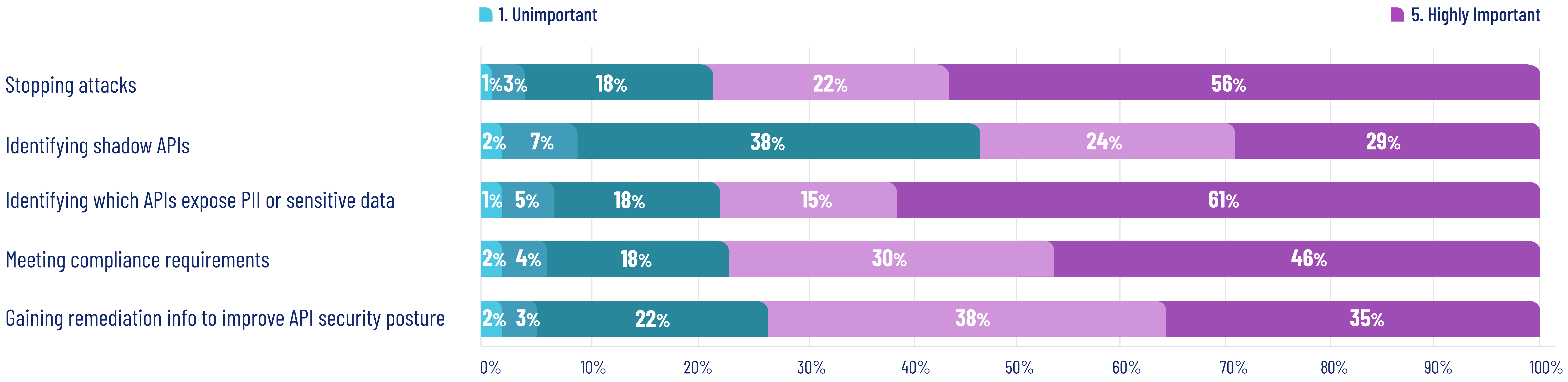
Runtime protections top the list of desired API security capabilities

Identifying exposed data and stopping attacks are highly important to more than half of respondents

Respondents are looking for full-featured API security platforms, solutions that provide a wide range of capabilities. Of the five use cases cited, only “identifying shadow APIs” was not rated as fairly or highly important by more than 70% of respondents. Possibly this low rate of concern indicates a lack of awareness about how frequently APIs can be created and deployed by developers without IT or security oversight.

Identifying which APIs expose PII or other sensitive data was the most valued use case, with 61% rating it highly important. Stopping attacks was the second most popular, with 56% responding that it is highly important. Compliance is another major driver for API security platforms, with nearly half of respondents citing that capability as highly important. The data shows that when asked about the importance of different use cases, protection across all phases of the API lifecycle is essential.

On a scale of 1 to 5, how would you rate the value of each of these attributes of an API security platform?



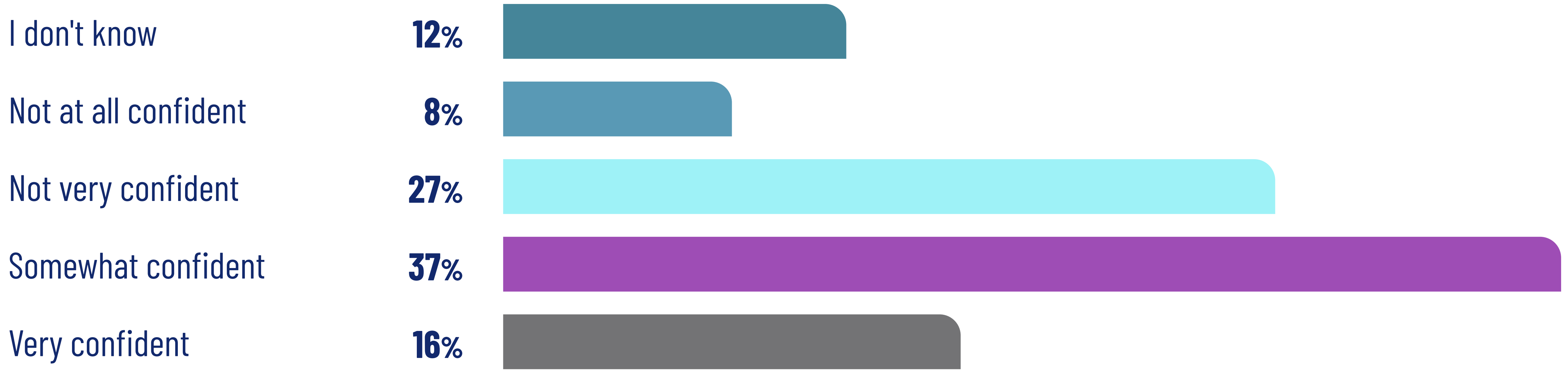
83% of organizations lack confidence in their API inventory

Despite the prevalence of popular API documentation tools, organizations are struggling with API inventory gaps

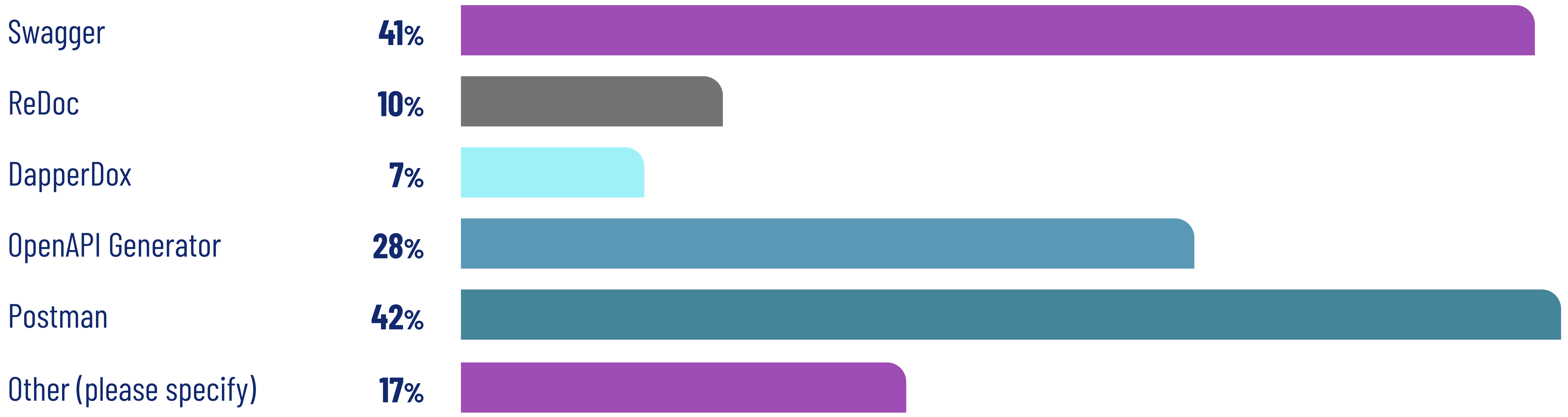
A comprehensive, up-to-date API inventory and accurate API documentation are essential for security. However, 83% of organizations lack confidence in their API inventory, and the most popular tools they employ for cataloging APIs depend on humans for accuracy. Given the speed of development, API documentation is often missing, incomplete, or inaccurate. These gaps contribute to unknown data exposure and unrealized risk, and they complicate compliance efforts.

Organizations need automated tools to align documentation efforts with the speed of application rollouts and the frequency of updates as part of modern DevOps practices. Such automation, with continuous feedback between security and development teams, is crucial to ensure documentation accuracy and API security.

How confident are you that your API inventory is complete?



What mechanism(s) do you use to inventory your APIs? (pick all that apply)

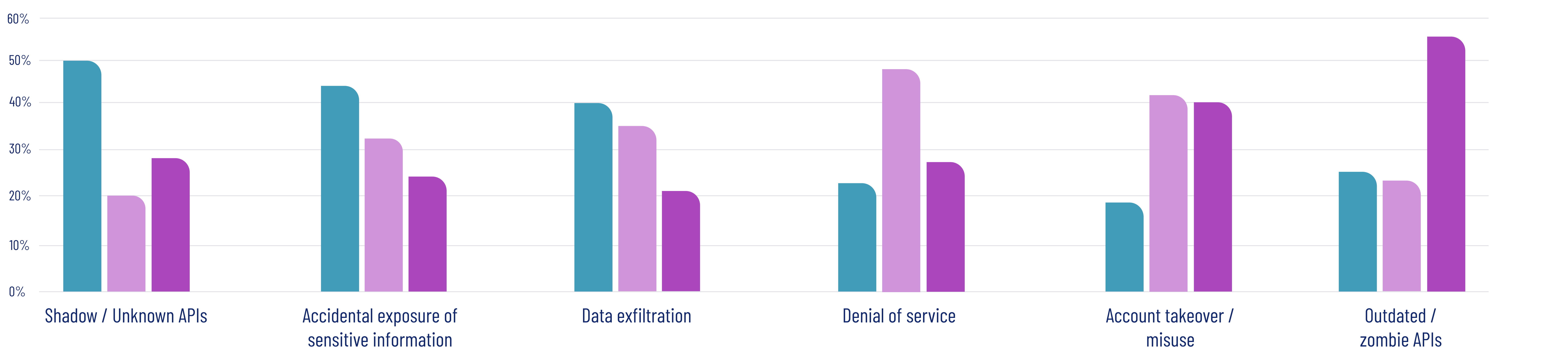


Outdated and zombie APIs present the greatest perceived risk

More than half of respondents rate these APIs as their first or second greatest concern

As agile development has taken hold, all aspects of applications are changing frequently, including APIs. Given the lack of confidence in the completeness of API inventories (page 9), it's not surprising that respondents are most concerned about outdated and zombie APIs. The second-highest concern, account takeover or misuse, also seems intuitive given the risk to an organization should an attacker successfully breach an API and break into an account. These types of successful attacks can result in fraudulent transactions and other activities that impact customer and user confidence and trigger expensive fines. Shadow APIs rank lowest on the list of perceived risks, which could be a result of the phenomenon that organizations consistently overestimate the accuracy of their API inventory. For example, across multiple customer deployments, the Salt Security platform has detected anywhere from 40% greater to 8 times the number of APIs than the organization had documented.

Please rank the following risks related to API security

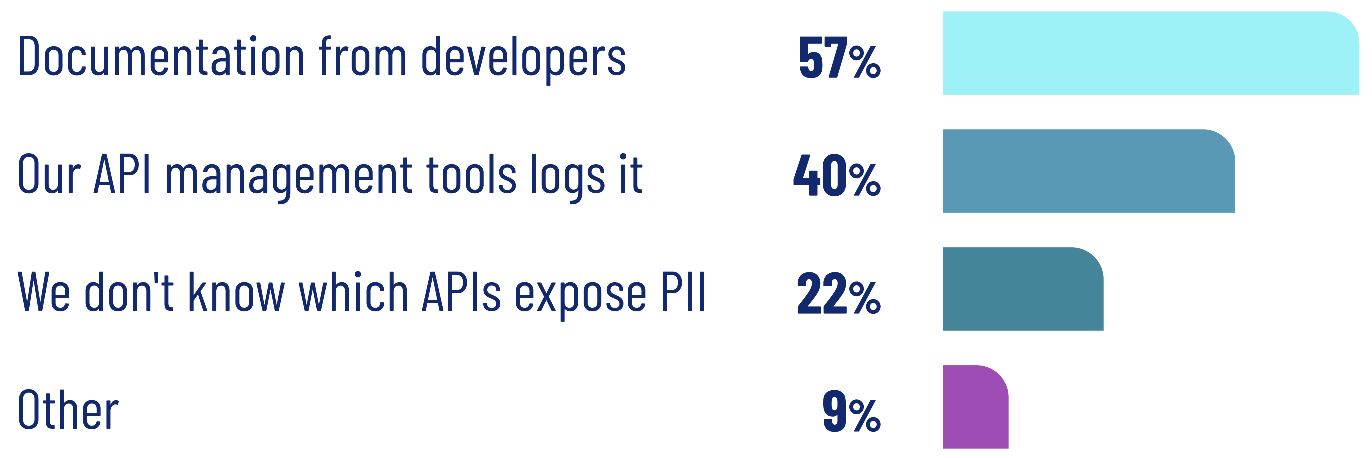


82% of organizations lack confidence they know API details such as exposed PII

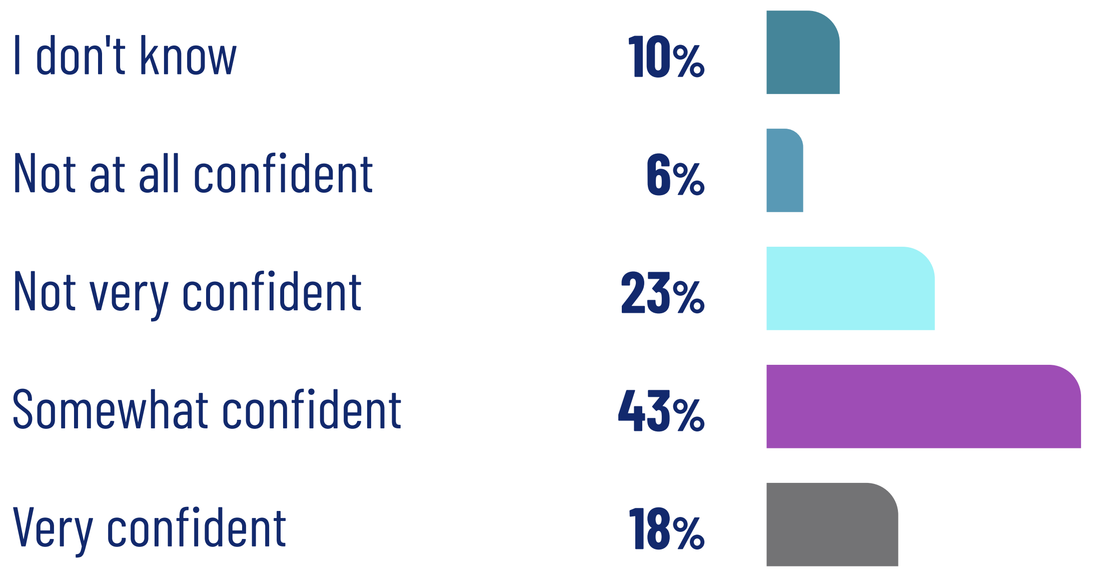
PII exposure represents significant risk for organizations

Nearly a quarter of organizations admit they have no way to know which APIs expose PII – a direct result of an incomplete API inventory and inaccurate documentation. The majority of organizations depend on developer-created documentation and/or API gateways to understand PII exposure (page 9) and clearly lack confidence that these approaches are complete and provide enough details. Most organizations with API gateways have multiple platforms, often from a mix of providers, and no consolidated API management, making it difficult to gain a definitive view of all production APIs. Developer documentation often does not exist or lags behind the deployment of APIs and is not complete. As a result, policies, even if defined within API management and API gateways, may be inadequate to protect a given API. The Salt Security platform automatically identifies which APIs expose PII or sensitive data, and across all our customers, 91% of APIs expose sensitive data, including PII, account numbers, and other data that are a valuable target for attackers.

How do you know which APIs expose sensitive data or PII? (pick all that apply)

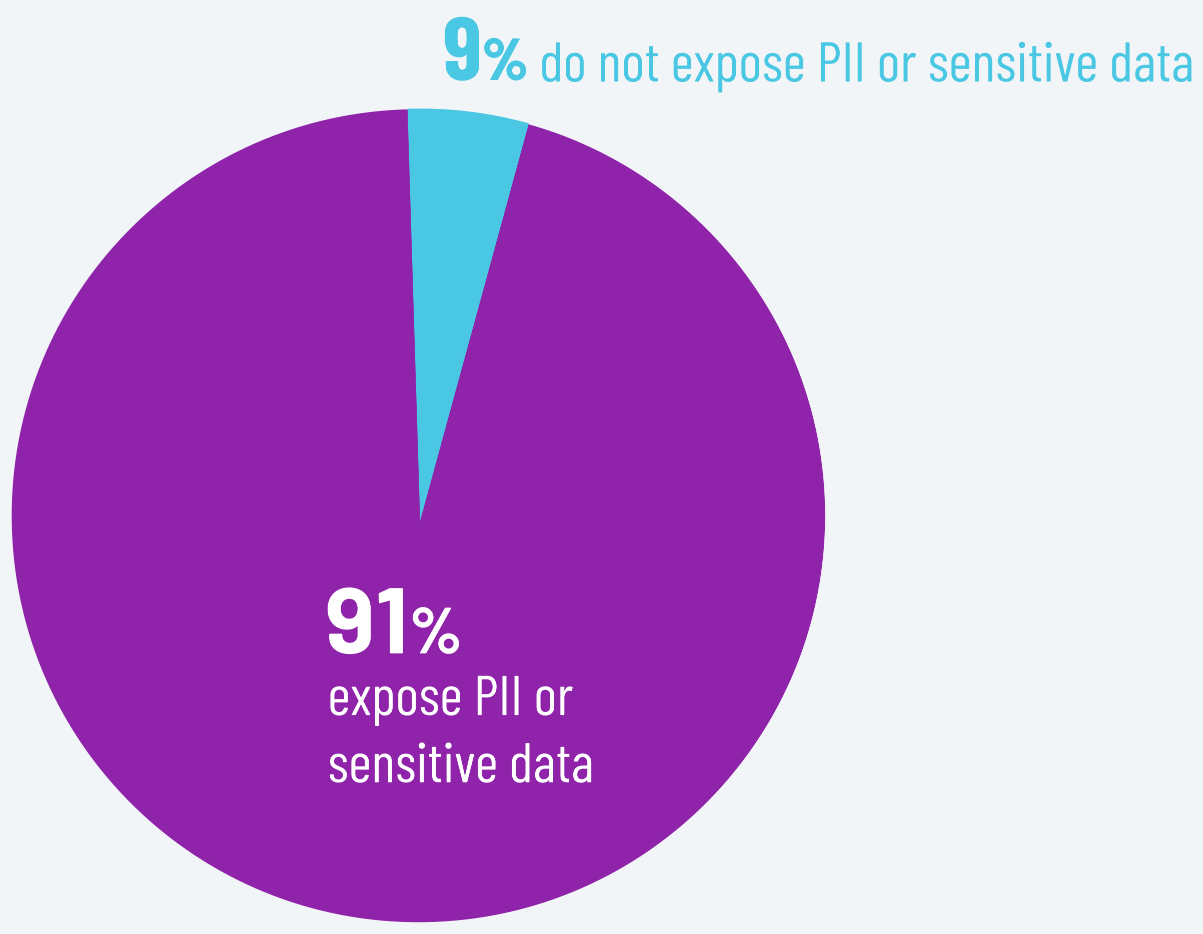


How confident are you that your API inventory provides enough detail about your APIs, including exposure of sensitive data or PII?



Salt customer data

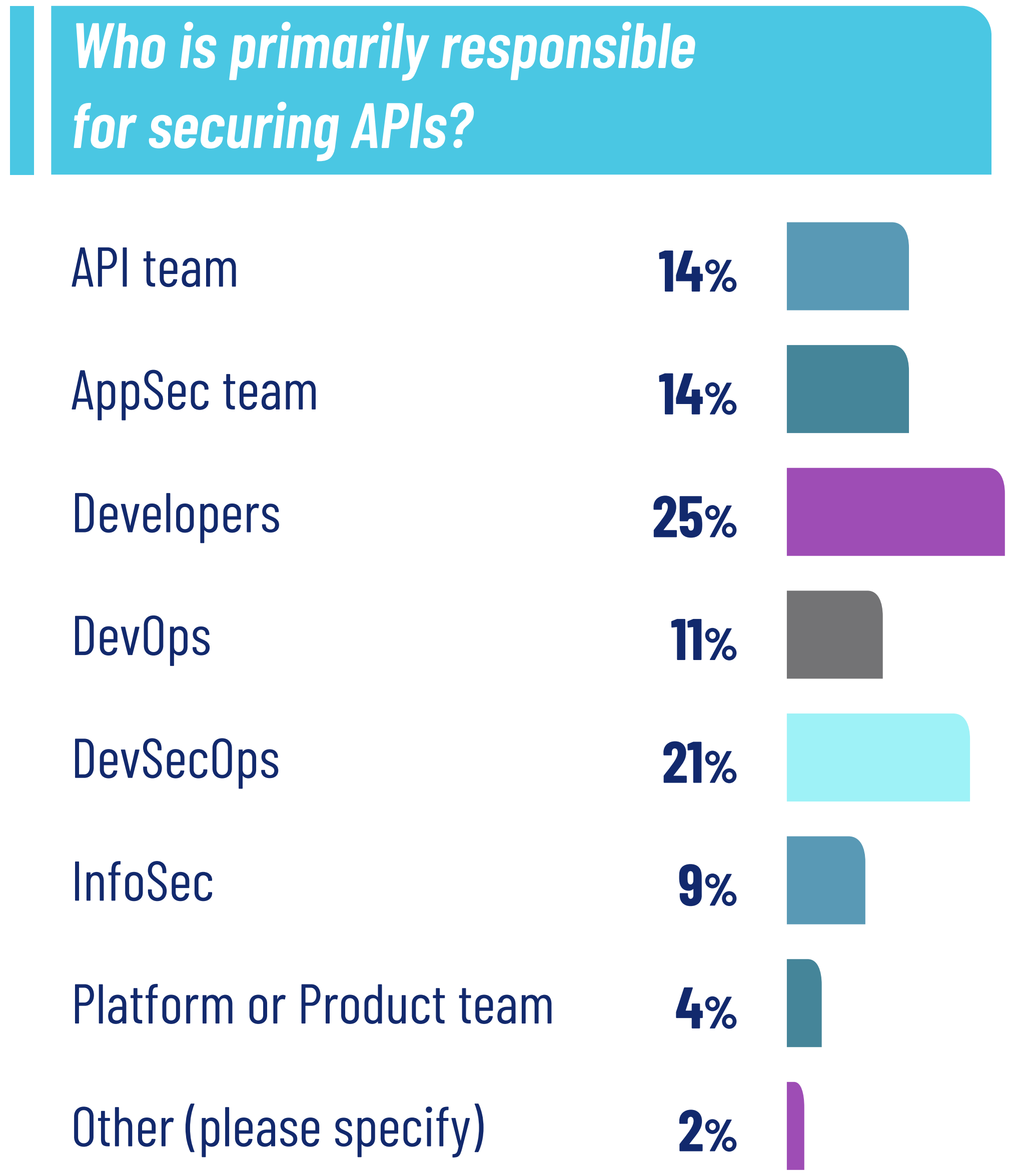
Number of APIs that expose PII or sensitive data

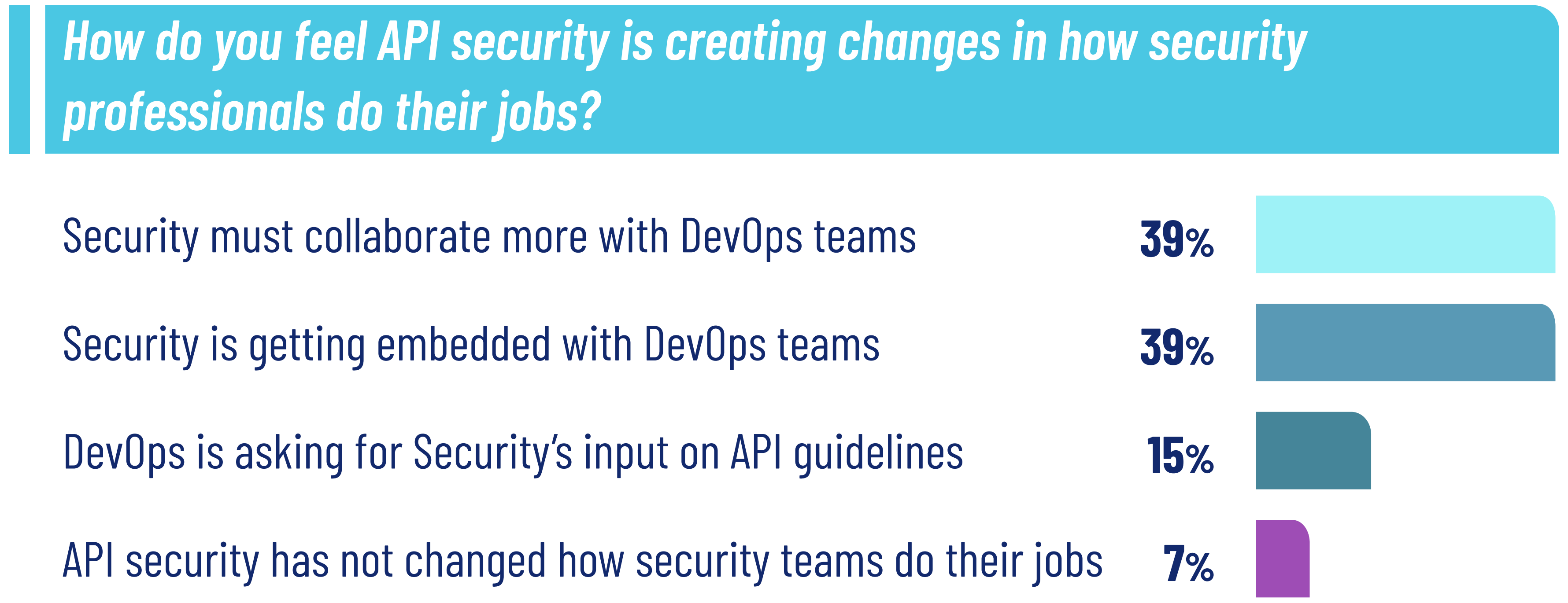


More than a third of respondents say developers and DevOps are responsible for API security

Successful strategies will depend on collaboration between security and dev teams

Although more than half of our survey respondents identify themselves as being in a security role (page 14), 36% of respondents say developers or DevOps teams hold primary responsibility for securing APIs. With current API security approaches heavily relying on pre-production tactics, and 91% of respondents experiencing an API security incident last year, it's clear that DevOps efforts alone are not enough to protect APIs. Respondents also cite a strong desire for more integration and collaboration between development and security teams when it comes to securing APIs. Developers are responsible for creating secure code and for eliminating vulnerabilities, but they need input from security teams on what needs to be remediated, how, and in what priority order. Security teams, in addition to implementing effective runtime security tools, should also look for API security platforms that offer valuable insights into how attackers are attempting to breach an environment or exfiltrate data using APIs. They can share these insights with developers to improve API security posture and create more secure development practices moving forward.


Who is primarily responsible for securing APIs?


How do you feel API security is creating changes in how security professionals do their jobs?

Implications for API security

Both the survey results and the data from the Salt Security API security SaaS platform show that organizations are struggling to keep up with the security risk that APIs present. 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, provides a broad range of protections, and fosters collaboration across teams.

1. Augmenting WAFs and API gateways is essential

Longevity can breed complacency, and APIs have been around for decades. Too many organizations think they've "got it covered" with WAFs and API gateways, but successful API attacks continue to increase, proving these older technologies provide insufficient protection.

2. Overreliance on dev teams and pre-prod checks is not working

As with many application security projects, API security has often started with depending on development teams. This approach is not enough – it's clearly not preventing security problems in production APIs. Attacks targeting APIs are on the rise, and developers will continue to deploy new and changed APIs at a rapid pace. While "shift left" efforts to improve API security should continue, organizations must also "shift right" and augment these tactics with runtime protection for APIs.

3. A full lifecycle approach is essential

Organizations need to improve security at every phase of the API lifecycle and should especially ensure protection against vulnerabilities in production. Organizations should vet APIs as they're developed, automate pre-production scanning in build pipelines, apply thorough manual testing as time permits or as mandated by regulation, and deploy runtime protection. To deliver full efficacy, an API security platform should also include a closed-loop system, with a means of providing developers feedback to quickly remediate vulnerabilities found in production so the organization can continuously improve its API security posture.

4. Automation is critical

Given the speed of agile development methods and DevOps practices, the volume of APIs in use, and the rate of change to internal and external APIs, properly securing APIs requires automation at every phase of the API lifecycle. Techniques that depend on manual efforts will not scale in today's DevOps and cloud-native world and will hamper application deployments. API security strategies must include solutions that can deploy anywhere with minimal impact, use automation to continuously inventory APIs, identify sensitive data exposure, baseline normal API behavior, and adjust to ongoing API changes. Automation across all these capabilities is essential to providing an accurate understanding of the rapidly changing attack surface and the ability to pinpoint and stop attackers.

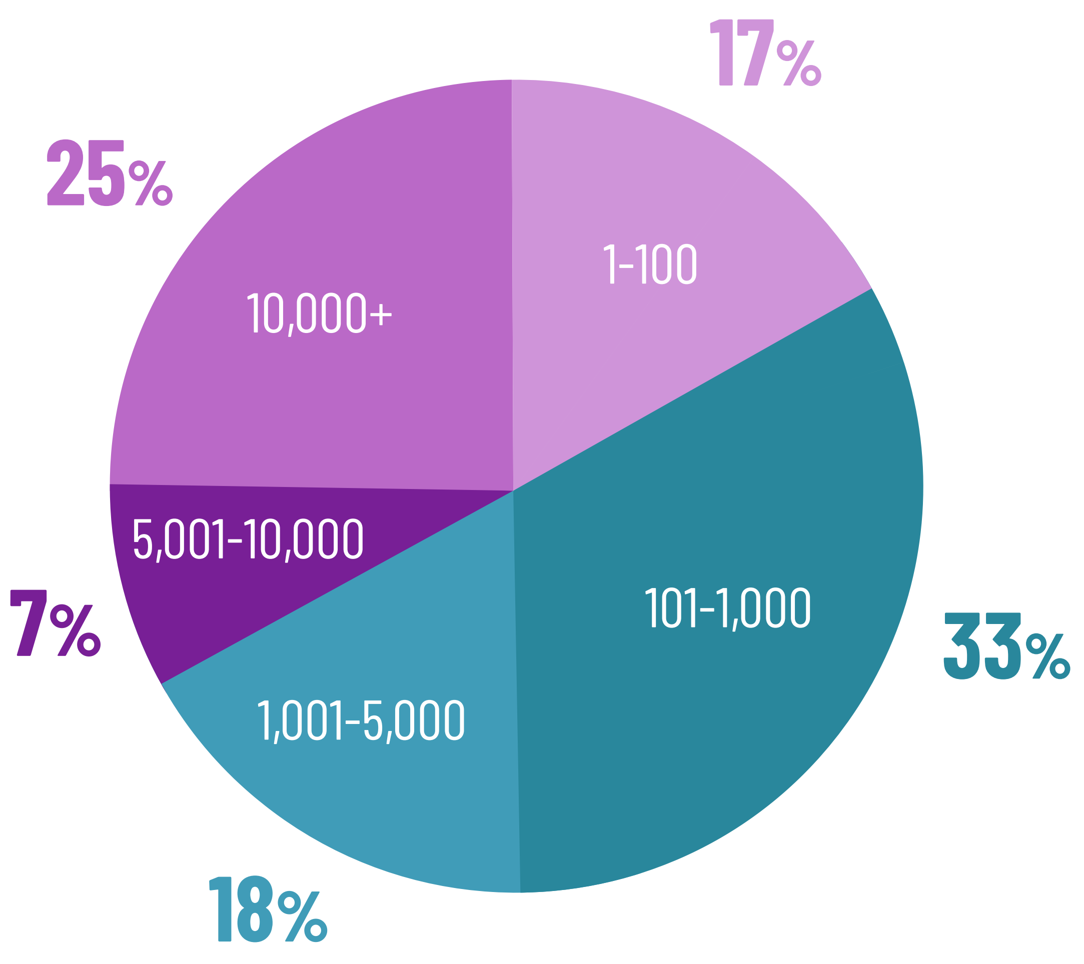
5. You can't prevent attackers from targeting APIs, but you can stop them before they succeed

Since APIs connect customers and partners to valuable data and services, they make an increasingly attractive target for attackers. You cannot prevent all vulnerabilities from making their way into production APIs, and you can't stop attackers from targeting APIs. However, you can stop attackers before they reach their objective, and you can continuously improve your organization's API security posture. Unlike other security risks, where a single successful effort unlocks the reward, API attacks are often low and slow as attackers build out their understanding of your environment. This reality means you have time to identify attackers - provided you have adequate detection and protection mechanisms - see their minor successes, learn from their behaviors, block them before they ultimately succeed, and keep your most critical services and data safe.

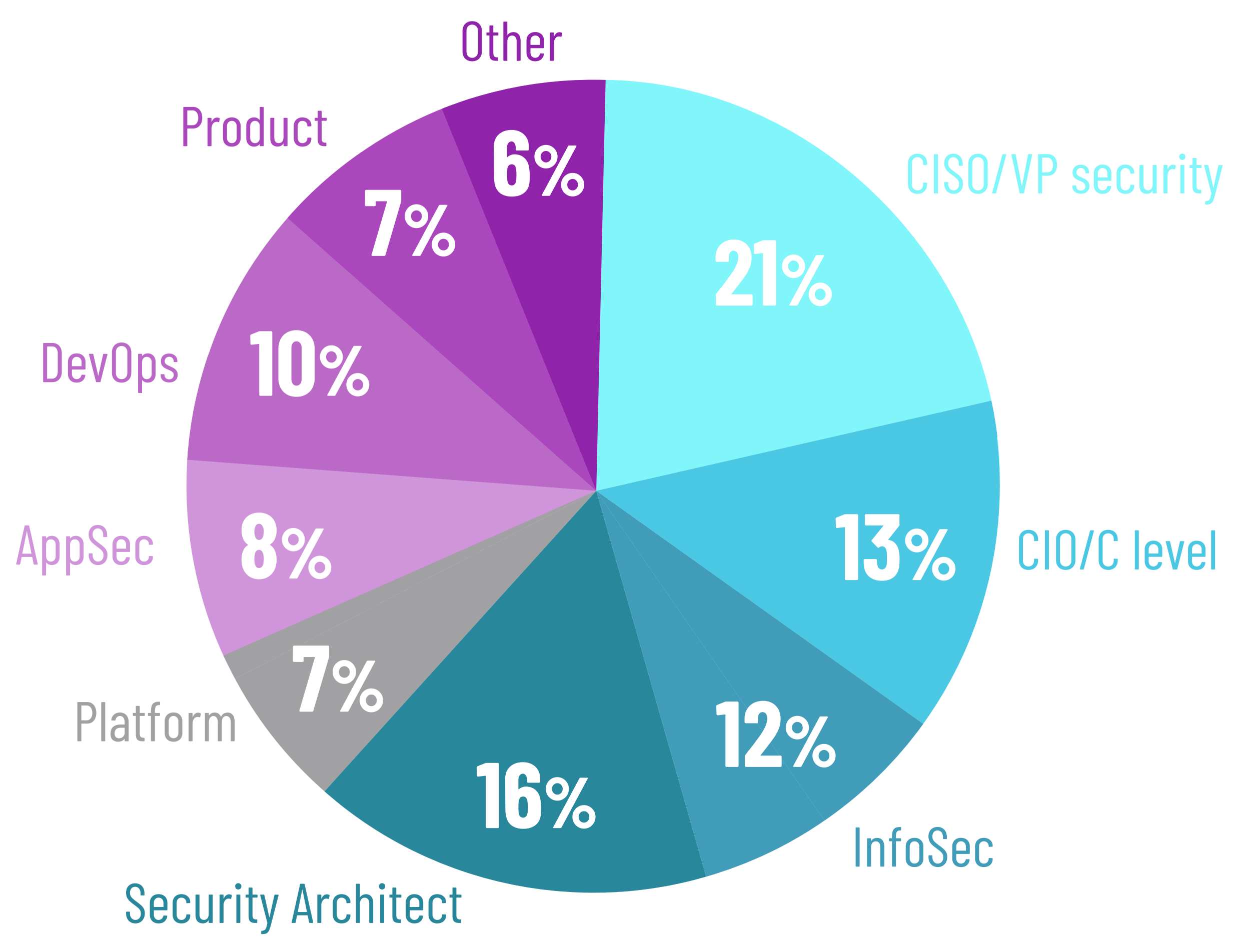
Demographics

These report findings are a combination of live Salt customer data and the survey responses of approximately 200 respondents. The survey respondents are well distributed across a range of job responsibilities, industries, and company sizes. More than half (57%) hold roles in security, 13% are CIOs, and another 24% sit on platform, DevOps, or product teams. Technology and financial services companies – widely viewed as at the forefront of API use – make up 60% of respondents. Companies large and small are evenly represented.

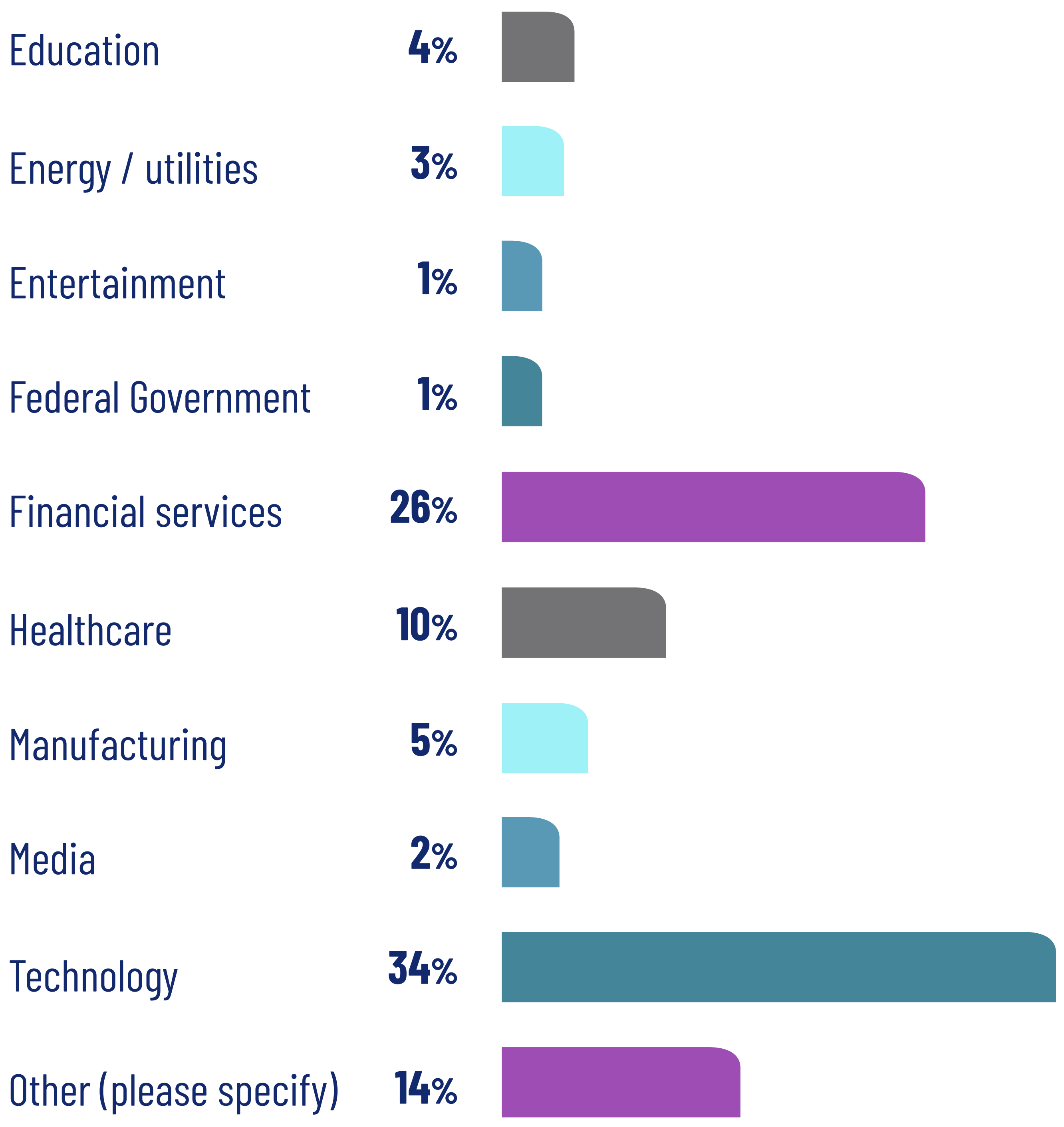
Size of company



What area best represents your functional role?



Industry



About Salt Security

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

The Salt Security API Protection Platform is the industry's first patented solution to prevent the next generation of API attacks, using machine learning and AI to automatically and continuously identify and protect APIs. Deployed in minutes, the Salt platform learns the granular behavior of a company's APIs and requires no configuration or customization to pinpoint and block API attackers.

The Salt platform provides three critical advantages that enable complete protection of APIs across across the full development lifecycle:

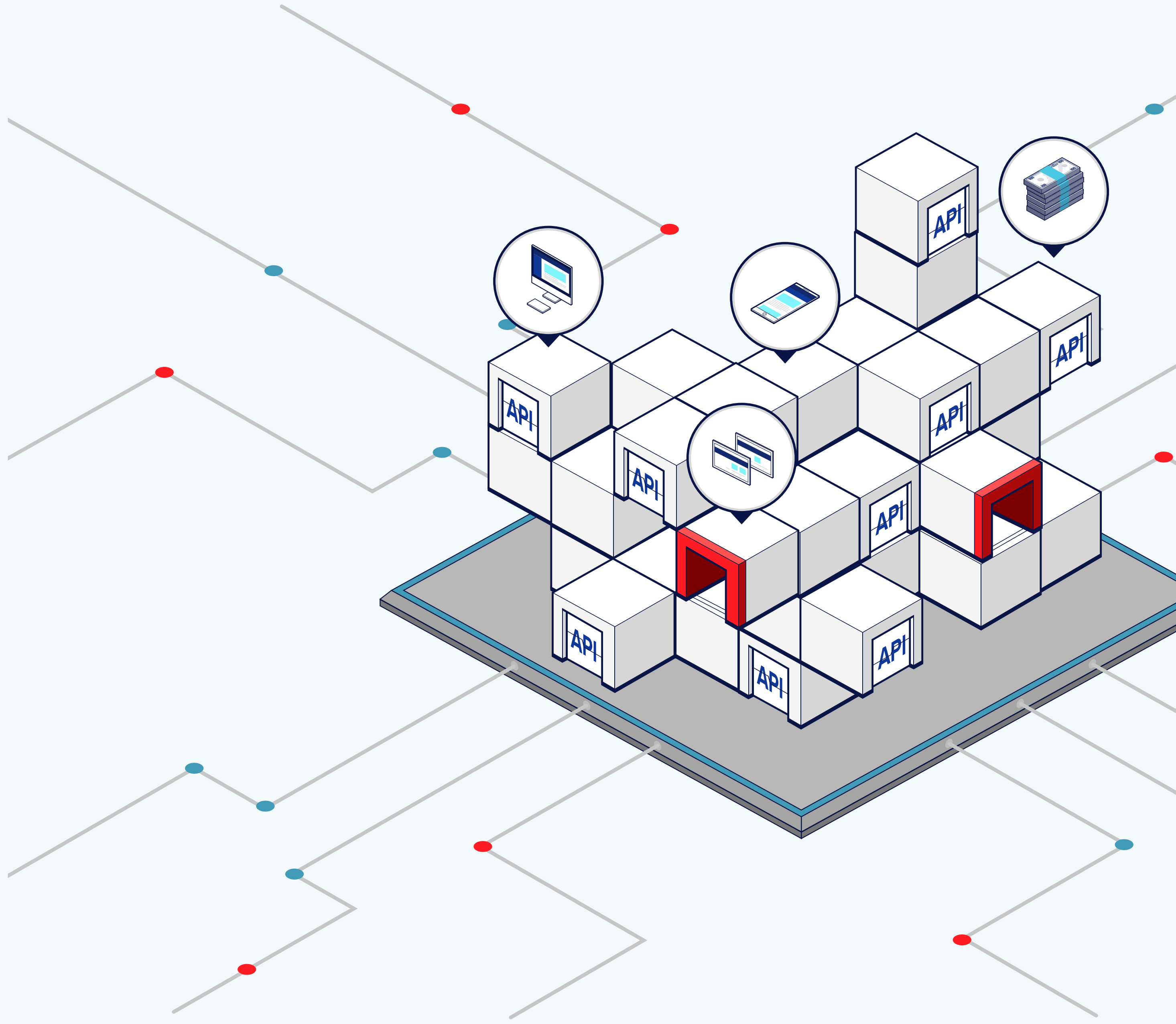
- **Holistic coverage** – we cover all your APIs across all your environments, including load balancers, API gateways, WAFs, and Kubernetes clusters, running on prem or in the cloud. And we deploy with no application or network changes and no configuration.
- **Big data and AI engine** – every one of your APIs is unique. Salt understands the unique logic of your APIs. We apply ML and AI to baseline your APIs and isolate anomalies, differentiating between "different" and "malicious." All without false positives.
- **Context-based detection** – Salt combines our ubiquitous coverage and big data engine to discover all your APIs, see the sensitive data they expose, find and stop attackers, and capture remediation insights for dev teams to improve API security posture.

Ready to see Salt in action?

Request a personalized demo to see how the Salt Security API Protection Platform can protect the APIs at the heart of your business innovation.

Request a Demo

info@salt.security
www.salt.security



SALT
SECURITY