Corsha™

It's Time To Get Honest About Secrets Management

Corsha State of API Secrets Management Report, 2023

The Good, the Bad, and the Ugly Of API Authentication

It's time for security professionals to address the elephant in the room: There's a big problem with current API secrets management practices: They haven't proven to be effective at preventing organizations from suffering data breaches.

Many of 2022's highest-profile data breaches were due to compromised API secrets. What did Twitter, Dropbox, and Uber all have in common last year? They were the targets of attacks that either leaked or took advantage of tokens, keys, and certificates — making their APIs open season and keys to the kingdom for bad actors looking to access sensitive data.

Clearly, APIs are a red-hot target for threat actors, often over-exposed and under-protected, the gateways to treasure troves of data. But even as security teams invest in secrets management and better secrets hygiene, why are bad actors *still* able to breach that first line of defense?

It's time to take a closer look at the current state of API secrets management practices. We surveyed more than 400 security and engineering professionals to learn about their API security practices and uncovered how the evolving digital ecosystem presents a new set of challenges for many organizations. Most importantly of all, we discovered that "sound" secrets management doesn't necessarily mean "secure" secrets management.

Executive Summary

With an overflow of credentials to provision, rotate, and manage, security and engineering professionals are bogged down by a never-ending game of catch-up. The sheer amount of secrets in play and leveraged in automated processes makes it difficult to achieve effective API security, leaving organizations vulnerable to leaks and breaches.

Key Findings

Here are the key findings we found that sum up the current state of managing secrets according to security teams:

86% of respondents spend up to 15 hours a week provisioning, managing, and dealing with secrets.

Security and engineering professionals are spending a lot of time on secrets management. That means less focus on other important duties, like innovating new security protocols or spinning up new projects.

42% of respondents manage up to 250 API tokens, keys, or certificates across their network — and over 55% manage at least 51 service account tokens.

There are a lot of credentials out there to keep track of — and that's counting with internal APIs alone. Whether they're hard-coded, in a database, or in a separate file, monitoring hundreds of different credentials is a tall order for already time-crunched security teams.

Over half (53%) of respondents have already experienced a data breach with unauthorized access to their networks or apps due to compromised API tokens.

The threat here is real, not theoretical. The potential risk of data breaches associated with leaked API secrets is high. A compromised API token will likely be used by threat actors to gain unauthorized access to systems and services — and of course, sensitive data.

72% of respondents use a secrets manager yet over half (56%) of them are still concerned about a potential data breach due to their current secrets management practices.

Secrets managers are widely used yet there's a concern that current secrets management practices are not enough to effectively defend against breaches.

Major Challenges With APIs and Token Authentication

API Ecosystems are Increasingly Complex

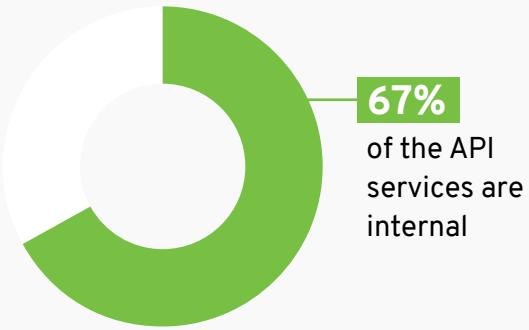
According to our survey, **67% of API services are internal**, which means they're essential to facilitating business functions. API ecosystems are increasingly broken down into a plethora of microservices each built to be good at a specific job and each communicating with others over APIs. They're also used in hybrid cloud environments, serverless environments, and other app-to-app or service-to-service communications.

Basically, APIs create contact between two machines and make transmitting data fast and seamless. They are not only the backbone of application infrastructure but also the keys to the kingdom.

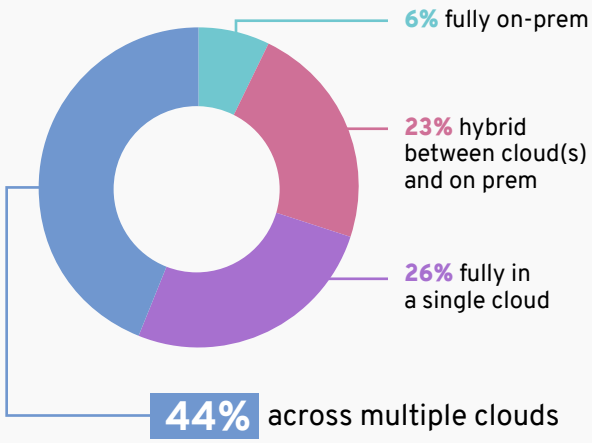
Speed is a virtue, but it also comes with a price. Cisco predicts there will be more than 500 million new digital applications in 2023. That's an amount equal to the number developed in the last 40 years. More applications means that the army of machines requiring API access will only catapult.

More machines mean more identities — which will inevitably lead to more credentials (aka those pesky but necessary tokens and secrets) to manage. Security here needs to match the pace of development and automation.

Approximately what percentage of these API services are internal (vs. third party)?



Where do you currently host your API services?



Acceleration and Expansion Caused By the Cloud

What else contributes to the internal API explosion? More companies are running in multiple cloud environments than before. Our survey shows that **44% of respondents host their API services across multiple clouds.** Plus, according to Leftronic, enterprises use on average 1,427 different cloud services. That's an 18.9% increase from Forbes' 2019 stat of nearly 1,200.

APIs are the intermediaries that make seamless work across complex digital environments possible. Even microservices and serverless architectures are API-driven — which is exactly why these internal APIs are so essential to driving critical infrastructure.

We've got good news and bad news. The good news is that APIs work well alongside new, speed-driven technologies like the cloud, containerization, and Kubernetes. The bad news? The security strategies in place to protect our API-based communication haven't been able to keep up with the level of scale and automation possible today.

Executive Summary

Getting API Authentication right — while completely necessary — isn't foolproof or easy to deal with for a multitude of reasons. According to our survey, the top three pain points in managing API secrets are:

- Working with certificate authorities
- Rotating secrets
- Provisioning secrets

What makes these three areas such great headaches for today's security teams? The methods used to address them are often dated, manual, error-prone, difficult to verify — and, as a result — dangerous.

Insider Threat and Privileged Accounts

While many security teams assign specific entitlements to API keys, tokens, and certificates, our survey discovered that more than 42% do not. That means they're granting all-or-nothing access to any users bearing these credentials.

While all-or-nothing access might be the path of least resistance in access management, it's unfortunately a security vulnerability for many organizations. This type of "world" access leaves the door open to lateral movement, leaving organizations vulnerable to breaches caused by insider threats.

That's exactly what happened with Uber. A contractor with unlimited network permissions found and stole hard-coded credentials to escalate their privileges. They gained unauthorized access by initially jumping in and laterally moving across the network.

What Good Secrets Hygiene Looks Like in an Ideal World

To manage the growing number of secrets and credentials, organizations have invested heavily in secrets management. The core tenets of effective secrets management are:

- Integrating a good secrets manager to gain overall visibility into your secrets
- Using mTLS when and where possible
- Always set a short expiry on secrets when possible
- Always sign *and* verify tokens
- Don't store or pass secrets in plaintext

Three Signs That Secrets Aren't Cutting It

According to Gartner's 2022 Hype Cycle Report, **90% of web-enabled applications have a greater attack surface area** thanks to exposed APIs. Plus, IBM estimates that compromised credentials cost organizations an average of $4.5M.

If organizations continue current API security practices, API attacks are not only going to become more common but more costly in time. That means API security strategies must become a robust first line of defense that can adapt (not just react) to the advanced tactics of a modern threat actor.

Where does that leave API authentication? In a complicated place. Here are three signs we found in our survey that — when it comes to protecting APIs — we need to up our game.

It's Hard to Keep Track of Who (or What) is Using APIs

An increase in APIs also means an increase in API credentials. It's up to security teams to faithfully track what identities use these credentials, to ensure they don't fall into the wrong hands.

However, our survey found that more than **50% of respondents have little to no visibility into the machines**, devices, or services (clients) that are leveraging the API tokens, keys, or certificates that their organization provisions.

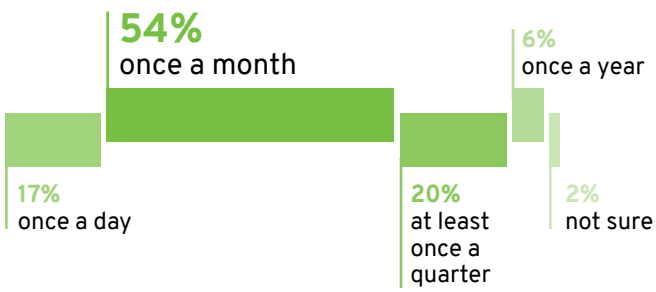
More often than not, these teams also lack visibility into their APIs proper. Without that knowledge, the entire API threat landscape becomes one big blind spot because:

- You don't have complete visibility into your API services
- You don't have complete visibility into the credentials that are provisioned to access those services
- You don't have visibility into the machines leveraging those credentials

Ultimately, limited visibility can lead to secrets that are forgotten, neglected, or left behind — which are prime targets for threat actors to quickly exploit.

Here are the real-world implications: **While 54% of respondents report that they rotate their secrets once a month, 27% report that they can take one quarter to one *year* to rotate secrets.** Amazon, referring to PCI-DSS standards, notes it's best practice to rotate secrets every 90 days. Imagine the burden though on DevSecOps teams to keep that up!

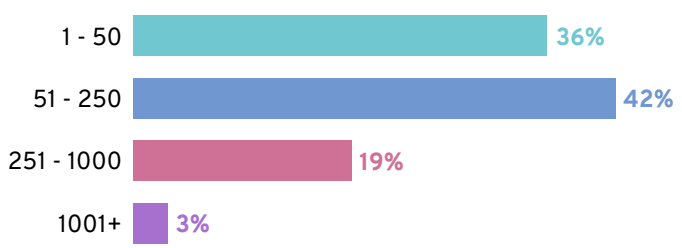
On average, how frequently do you rotate your API tokens/ keys/certificates?



Remember, **42% of our respondents state that they manage up to 250 API tokens**, keys, or certificates across their networks. A growing mountain of credentials sprayed across multiple environments, combined with a lack of visibility into what is using secrets, hints at a security strategy that relies on luck vs. bullet-proof practices.

Plus, secrets management is just one very small part of a security or engineering team's job — a part that, if left unchecked, can easily start to eat away at every organization's greatest resource: time.

How many API tokens/keys/ certificates do you or your team manage across your network?



Teams Are Already Spending A Ton of Resources Managing API Secrets

Security and development teams operate best with clear, streamlined, and efficient workflows and always seek out ways to automate themselves out of a job. This frees them up to do strategic work and keep business going above and beyond.

But according to our survey, a perfect workflow may feel like a distant fantasy when secrets management is involved.

Postman's 2022 State of the API Report found that **67% of respondents spend over 10 hours a week developing APIs**. Meanwhile, **86% of our survey respondents stated that they spend up to 15 hours a week provisioning, managing, or dealing with issues related to secrets**. That means, according to our data, security teams are focusing at least 800 hours each *year* on rote (but necessary) tasks of secrets management.

Imagine all the strategic, innovative, and new projects your teams could tackle with those 800 extra hours. Then imagine then spending it rotating tokens and secrets instead.

Pretty frustrating, right?

How Much is Secrets Management Costing You?

For key decision-makers, time is also money. The cost of labor for managing secrets can make an unwieldy dent in your organization's budget.

Nearly 86% of our respondents said they spend up to 15 hours each week managing secrets across their network. According to Indeed, the average software developer's salary breaks down to about $120,000 a year, or $57 an hour. That means each respondent could be spending up to $44,460 per year on secrets management.

API Secrets Are Leaking Anyway

Over half of our survey respondents have already experienced a data breach due to stolen or leaked API credentials. Despite all the resources spent on API secrets management, organizations are still plagued by API breaches. When it comes to today's threat landscape, many teams may feel as if breaches are a matter of when not if.

Why do breaches and leaks persist even when security teams are carrying out top-of-the-line secrets management? It's because secrets management is simply a fragile system. There are tons of credentials to keep track of, and they must be kept track of 24/7, which can lead to huge challenges with visibility.

We see API secrets getting leaked out of:

- Code repositories
- Versioning control
- CI build systems
- Test artifcats
- Cloud environments

And many other spaces throughout an organization's footprint. Plus, if you're operating in two different clouds, you're likely also operating in two different secrets management systems. That's a lot of vulnerable resources to management for one security and engineering team.

With one simple oversight, organizations can fall victim to not only internal leaks but also third-party compromises that expose their API credentials.

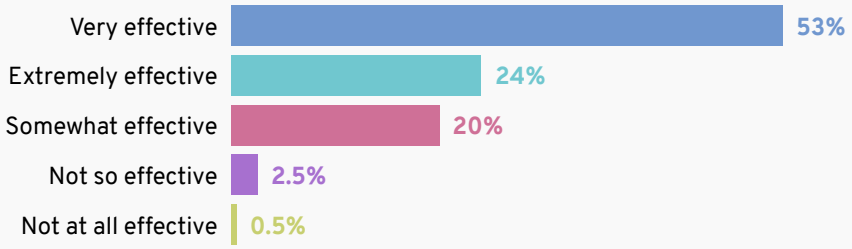
Take November 2022's Algolia API leak, for example. Over 11,000 websites and applications use the Algolia API to integrate search engines with discovery and recommendation features. 1,550 mobile apps connected to the API — due to their own poor secrets hygiene — leaked 32 secrets and 57 unique admin keys. Algolia's apps were downloaded over *3 million times*, introducing high risk to its millions of users and their data.

Even if Algolia's API security practices were top-of-the-line, the company would not be protected from third-party collaborators dropping the ball. Defending against third-party risks must be top of mind for security teams — yet modern strategies for API security haven't caught up.

When It Comes To API Security, "Good" Doesn't Always Mean "Safe"

Security teams are today's digital threat bounty hunters. They've adapted well to working with what they already have — and that includes devoting more resources to defending their APIs. That might be why **77% of our survey respondents believe that their organization is very to extremely effective at managing their API tokens, keys, secrets, and certificates**.

How many API tokens/ keys/certificates do you or your team manage across your network?



But the data speaks for itself. Even the most robust modern secrets management isn't enough to prevent your APIs from being exploited. That might be why over half of respondents are *still* concerned about a potential data breach due to their current secrets management practices.

The Bottom Line

Despite security teams' best efforts to shore up their secrets hygiene, breaches are still an all too common occurrence and anxiety. Organizations would benefit from a stronger and automated answer to their API authentication woes — an added factor to API authentication to protect their data from today's savvy and opportunistic bad actors.

Staying Secure in a Sea of New Machines

The proliferation and reliance on M2M communication is greater than ever before, and security teams are already struggling to keep up. Remember that Cisco stat: there will be 500 million new digital applications in 2023 — so the work is about to get much harder, not easier.

As our API ecosystem grows, so will the gap between secrets management and the standard of security that APIs *actually* need.

Modern threat actors and business needs render the model of relying solely on secrets and token authentication outdated. It's time for organizations to move on from hinging on secrets as their first line of API security defense.

How can security teams take their API security strategies to the next level? By adding a clear additional factor to API authentication — like MFA. Static secrets are increasingly harder to manage, meaning more vulnerable, as API usage continues to explode.

Many organizations already mandate MFA for human user validation due to challenges with static passwords. The challenges with the bearer model and its keys, tokens, and credentials are no different. It's time for API security to get an update and follow the leading security trends.

That way, with another layer of identity validation at play, organizations can rest easier and look ahead to a better frontier of API security.

When it comes to securing APIs, we have a lot more details to spill. Request a demo with us to get the inside scoop.

About Corsha

Corsha fully automates multi-factor authentication (MFA) for APIs to better secure machine-to-machine communication. Our product creates dynamic identities for trusted clients, and adds an automated, one-time use MFA credential to every API call, ensuring only trusted machines are able or leverage keys, tokens or certificates across your applications, services, and infrastructure. Halt and resume access to a machine or group of machines without revoking secrets or impacting other workloads, leaving compromised secrets are rendered useless using Corsha.

API-first ecosystems are driven by the machines that power them. Whether those are Kubernetes pods, containers, virtual machines, physical servers, IoT devices, or other form factors, risk is shifting from human to machine as we automate more, and securing communication between machines often becomes an afterthought. Today, API secrets like keys, tokens and certificates are used as a way to broker access between machines, but these static secrets are often shared, rarely rotated and are being leaked in CI pipelines, logs and code repositories at an alarming rate.

Corsha is taking all the goodness of MFA, and using the same principals like one time use credentials to secure APIs. This provides teams security, visibility and control into the machines that are accessing your APIs and the ability to revoke API access at the drop of a hat. For more information, visit: corsha.com

Survey Methodology

The Corsha State of API Secrets Report, 2023 surveyed over 400 respondents from U.S.-based companies with over 50 employees in November 2022. More than 90% of respondents worked in the developer and security space.