State of API Security 2024
The API Secret Sprawl

How we discovered over 18,000 API secret tokens by analyzing 1 million domains on the web

escape

18K

API secrets found in actual usage scenarios of applications: from APIs to front-end code, including components running in the background, such as JavaScript

7.7K

of exposed secrets are classified as **highly-critical**

$20M

could have been accessed via exposed Stripe tokens

Key Takeaways

Our study addresses the escalating challenge of API secret sprawl. Beyond public code, our focus extends to real-world applications, ensuring a comprehensive understanding of API vulnerabilities. The diversity of exposed secrets, from AI service keys to financial access and communication tools, emphasizes the widespread challenge of keeping sensitive information secure.

Tristan Kalos, CEO of Escape

Criticality: In an extensive analysis of 1 million popular domains, the Escape research team uncovered over **18,000 exposed API secrets**. Among these, **41% were deemed highly critical,** highlighting the widespread challenge of API secret sprawl.

Diverse industry impact:

- The exposed secrets include hundreds of Stripe, GitHub/GitLab tokens, RSA private keys, OpenAI keys, AWS tokens, Twitch secret keys, Coinbase keys, X tokens, and Slack and Discord webhooks.
- The increasing trend of using OpenAI tokens made up 1.4% of the exposed secrets.
- 2.1% of the exposed secrets were detected due to the practice of compiling code into a single JavaScript file.
- Secrets span various sectors, impacting e-commerce platforms, financial companies, Web3, education, tech, and others.

Substantial financial risks and ease of exploitability: The identification of unprotected **Stripe API tokens**, capable of easily accessing an **estimated $20 million**, highlights the urgency for businesses to prioritize and enhance their security measures.

Call to action: This extensive exposure of API secrets underscores a critical security issue. Centralizing token management, enforcing rotation policies, segmenting access, intensifying security training, and leveraging automated testing tools are essential steps to mitigate these risks.

Contents

$4.45M

The global average cost of a data breach in 2023*

10M

secrets detected in public GitHub commits in 2022**

2023 ALARMING NUMBERS

Recent reports, including GitGuardian's 'The State of Secret Sprawl,' indicate a 67% increase in secret sprawl in 2023 alone, with 10 million new cases of secret exposure in GitHub. This issue extends beyond GitHub, affecting all aspects of software development and operation.

BEWARE, IT COMES: THE HIDDEN DANGERS OF RISING API SECRET SPRAWL

Secret sprawl has emerged as a significant challenge in recent years, particularly prevalent in open-source projects. It occurs when sensitive data, such as API keys, inadvertently becomes exposed. The frequency of such incidents has escalated significantly since 2022.

To combat this issue, various tools have been developed to detect these inadvertent disclosures. These tools scan code updates and alert developers to potential secret exposures. Notable examples include Gitleaks (Open-Source), GitGuardian, TruffleHog, and SonarSource.

The fallout from secret sprawl can be severe, offering unauthorized individuals complete access to essential systems, potentially leading to financial loss, data breaches, and severe damage to a company's reputation. Moreover, uncovering such vulnerabilities during compliance checks could result in the loss of critical certifications.

The primary cause of secret sprawl is usually an oversight by developers who are unaware of the secret being disclosed. Identifying who is responsible for the leak, determining at what stage it occurred, and locating the exposed secret can be challenging. This problem is often attributed to a lack of knowledge or failure to adhere to established best practices.

The risk of secret exposure is growing with the global increase in code creation and the rise of new programming methodologies. For instance, the rapid adoption of AI in development processes has increased the likelihood of secret exposure.

Given these trends, the issue of secret sprawl is garnering increasing attention. It is now widely recognized as a significant and pervasive challenge, requiring more stringent management of secrets throughout the entire SDLC.

escape

*according to IBM Cost of a Data Breach Report 2023
**according to GitGuardian the State of Secret Sprawl Report

A CASE FOR APIS: 2023 SECURITY BREACHES INVOLVING EXPLOITATION OF API KEYS

In February 2023, publicly accessible environment files were discovered on Lowe's Market website. These files leaked access tokens to AWS S3 buckets and API keys to third-party services. One of these leaked keys, GrocerKey API, allowed access to partial credit card information, addresses, and top-spending users, as well as the ability to send unsolicited orders, issue refunds, launch ad campaigns, reset passwords, and check in-store and in-app balances.
Such exposure could potentially allow threat actors to access sensitive user information and manipulate website functionality.

In the case of Microsoft, a cyberattack involved the advanced persistent threat (APT) actor, Storm-0558, who gained access to unclassified email data from various government agencies. This was achieved by discovering a leaked Microsoft Account Consumer Key, which allowed the threat actor to forge access tokens to enterprise email accounts. This incident underscores the importance of secure handling and regular rotation of API keys and access tokens.

OpenSea, an NFT marketplace, notified their customers of a breach with a third-party vendor. The data breach could have a significant impact since OpenSea is the second-largest non-fungible token (NFT) marketplace by trading volume (36.5%) after Blur (56.8%), which launched only a year ago. This incident highlights the risks associated with third-party integrations and the importance of securing API tokens that provide access to such services.

Sumo Logic, a U.S.-based cloud data analytics firm, experienced a security breach where an attacker accessed their AWS account using compromised credentials. Detected on November 3, the company responded by securing the exposed infrastructure and rotating all potentially compromised credentials, including API keys. While no direct impact on their networks or customer data was reported, Sumo Logic advised customers to rotate their API access keys and reset various credentials as a precaution.

Methodology

Development of the Web Spider

To tackle the complex task of scanning 1 million domains, we developed a specialized web spider. This tool was built using **Golang**, chosen for its excellent input/output (I/O) throughput, good productivity, and strong support for concurrency. These features of Golang were crucial for efficiently processing large volumes of web data.

For networking, we relied on a library named **fasthttp** (fasthttp GitHub), known for its high performance. **fasthttp** was instrumental in enabling our spider to handle numerous network requests swiftly and effectively.

To interpret and analyze JavaScript found on web pages, we used **tree-sitter** (Tree-sitter), a parser generator tool and an incremental parsing library.

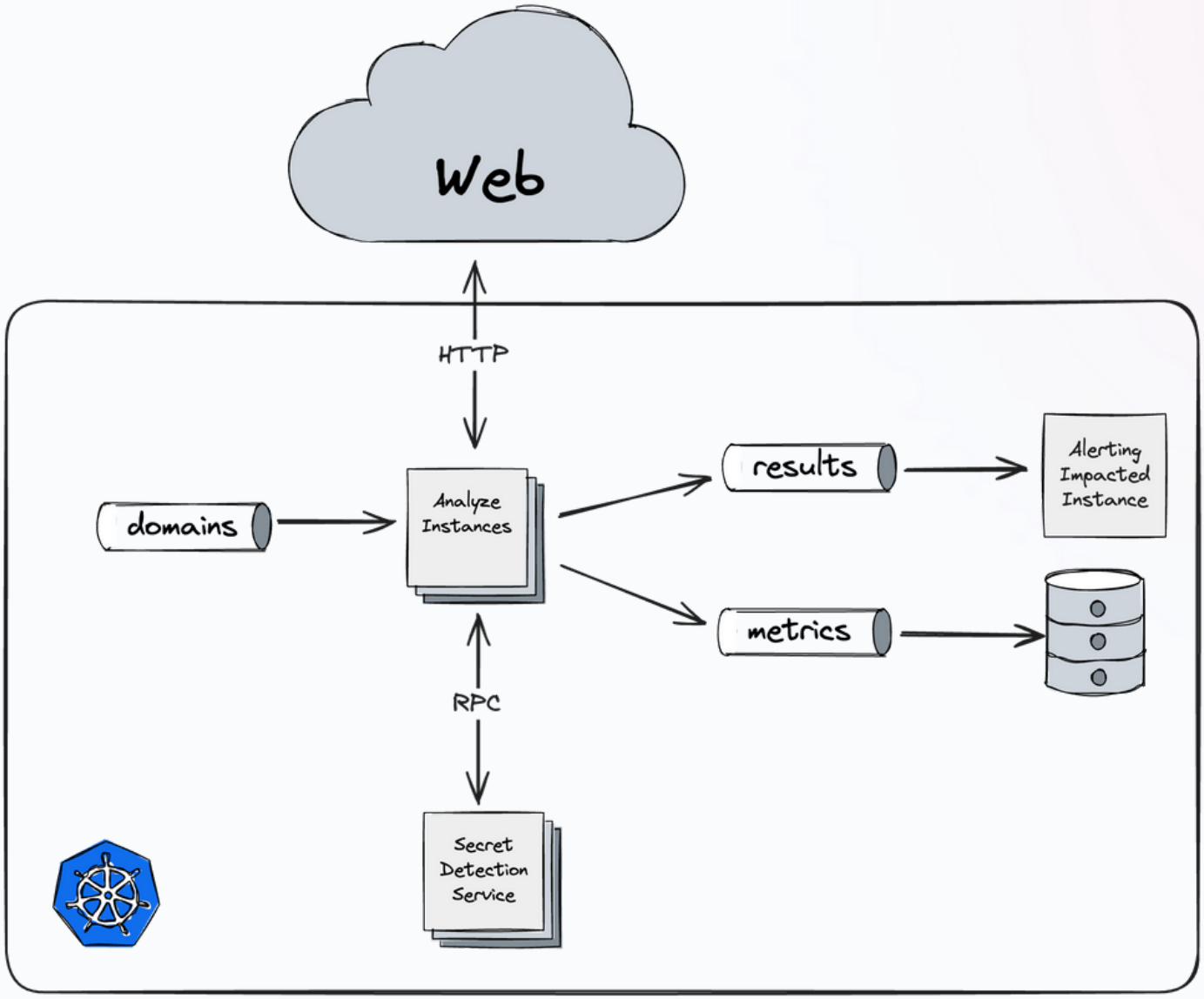
It helped us build a robust mechanism to understand and process JavaScript code, which is a critical component in modern web applications.

The Golang-based spider was containerized and designed to **listen on a Kafka (Redpanda) stream**. This setup allowed for scalable and efficient handling of data streams.

In terms of secret analysis, we incorporated an existing **Python-based service that we regularly use at Escape**. This tool, which employs natural language processing, was **accessed via gRPC**. The whole solution was **deployed on a Kubernetes cluster**, leveraging the orchestration.

Redpanda

Methodology



Our Development Process

Methodology

Data Gathering Strategy

For our comprehensive analysis, we chose to examine the 1 million most popular domains.

The list of these domains was sourced from the **Majestic Million dataset**, which ranks websites based on the number of referring subnets. This ranking offered us a diverse set of domains to study, spanning various sectors and sizes.

We acknowledge a potential bias in our approach. Typically, larger domains with more resources might have better security measures, possibly leading to fewer instances of secret sprawl. In contrast, smaller websites or those without dedicated security teams might be more prone to such issues. However, our study focused on the most popular domains without specifically addressing this bias.

Alternatively, with over 365 million domain names reported across the internet, our sample size becomes relatively small, potentially leading to greater volatility in the number of findings.

The **data collection was a one-time process**, based on the latest available list from the Majestic dataset. During the collection, we encountered several limitations. To respect legal and ethical boundaries, we deliberately excluded certain types of domains. This included governmental, educational, and health-related domains, as regular users are not typically authorized to explore these. This decision ensured that our study adhered to the ethical norms of web crawling and data collection.

By focusing on domains that are accessible to the general public, our study provides insights into the state of secret sprawl in the broader, more publicly engaged segments of the internet. This focus enables a comprehensive evaluation of security practices and the challenges encountered in a diverse array of online platforms and environments. Through this lens, we gain a deeper understanding of how secret sprawl impacts various sectors and what this means for the broader digital security posture.

Methodology

Data Collection Process

4

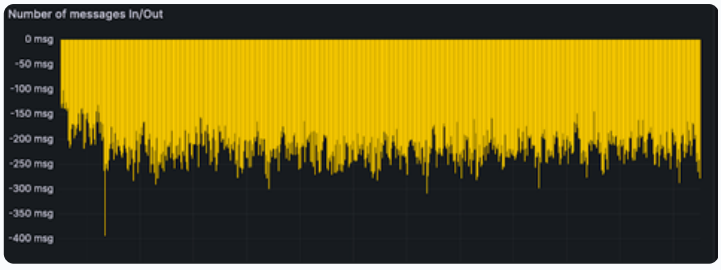
scanned domains per second

69 hours

Total scanning time

Our data collection process was a significant undertaking, both in terms of scale and technical complexity. To manage this, we deployed our containerized web spider on a Kubernetes cluster. The cluster was capable of scaling up to **150 concurrent worker instances**. This level of scalability was crucial for effectively managing the immense task of scanning 1 million domains, allowing us to distribute the workload efficiently and process a vast amount of data.

The collection spanned over 69 hours, with our system analyzing an average of 4 domains per second. This pace resulted in a **total sum duration of approximately 30,686,535 seconds for the entire operation**. On average, each domain, including its subdomains, was analyzed in about 32 seconds. This comprehensive approach ensured that we not only looked at the primary domain but also dived into the numerous subdomains associated with each, providing a more complete picture of the web landscape.



Methodology

189.5M

URLS scanned

$100

Project computing cost

In total, **our process led us to visit 189,466,870 URLs.** This extensive coverage was key to ensuring that our analysis was as thorough and inclusive as possible. By examining such a large number of URLs, we were able to gain deep insights into the current state of secret sprawl across a wide spectrum of the internet.

Also, we started this project by making a new tool as a test. It was impressive how quickly this tool was made – just three days by one engineer. Combining this quick tool development with the **project's computing cost of only about $100** shows how, in today's world, we can get big results, build solutions without spending a lot of money or time.

Methodology

Data Cleanup and Verification

One of the most challenging aspects of our study was, for sure, the data cleanup and verification process. While we could not verify the tokens ourselves, we made sure each one was classified accurately. A common pattern we noticed is that many tokens have specific prefixes. For instance, Stripe tokens have various prefixes, but we focused particularly on **live secret keys, identified by the prefix 'sk_live_'.**

To improve the accuracy of our findings, we refined our heuristics to filter out only classified information. This meant paying special attention to high-entropy keys, which often represent proprietary, undocumented tokens or false positives, and filtering them even more from our primary dataset. This approach helped us focus on the most relevant and potentially impactful tokens.

Verification of the tokens was a crucial step, and it was carried out by the token owners themselves. We alerted the owners only when our system was highly confident about the findings. This was a delicate balance to maintain – ensuring the accuracy of our alerts without the ability to test the keys ourselves. We had to be very sure before alerting affected parties to avoid any false alarms.

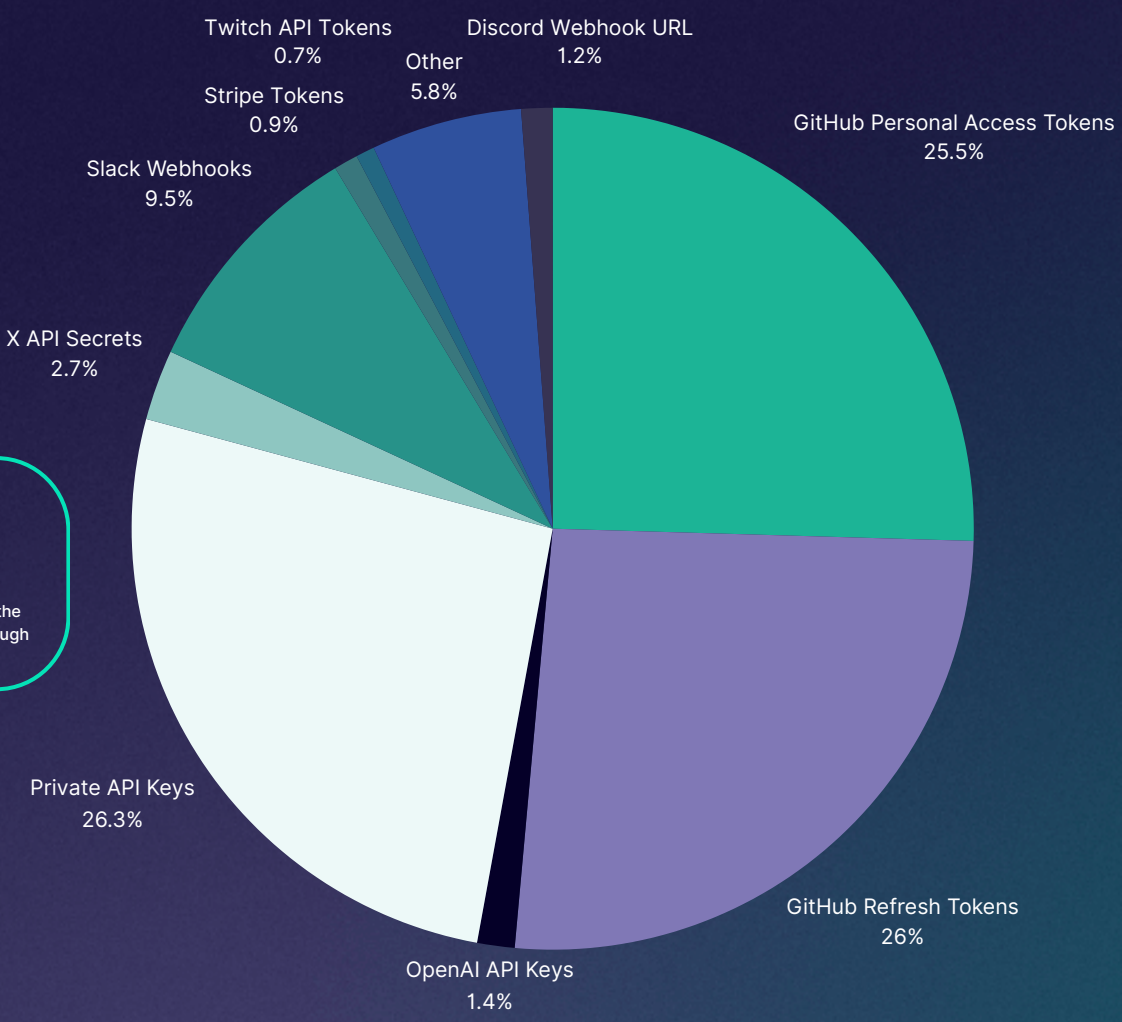
This cleanup and verification process was an intricate part of our study, requiring a nuanced understanding of token patterns and careful judgment to minimize false positives. Our method aimed to provide reliable and actionable insights to those whose security might have been compromised.

Findings

18,458
Total secrets found

7,737
Highly critical

Tokens providing access to sensitive data by known category



$17M
While representing one of the smallest %, the largest amount could have been stolen through one of the exposed Stripe tokens

escape

Findings

1.7

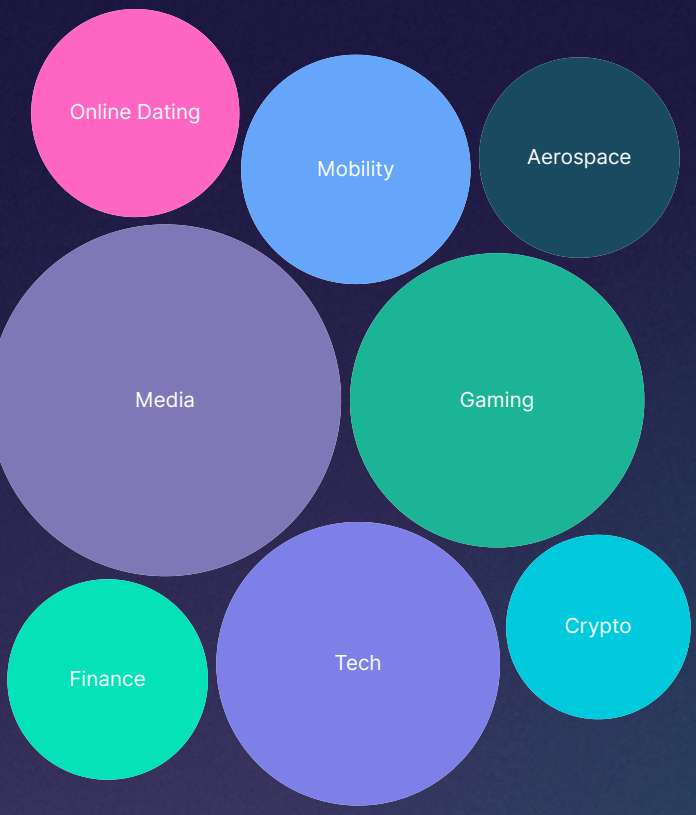
average exposed secrets per domain

409

domains are above the average

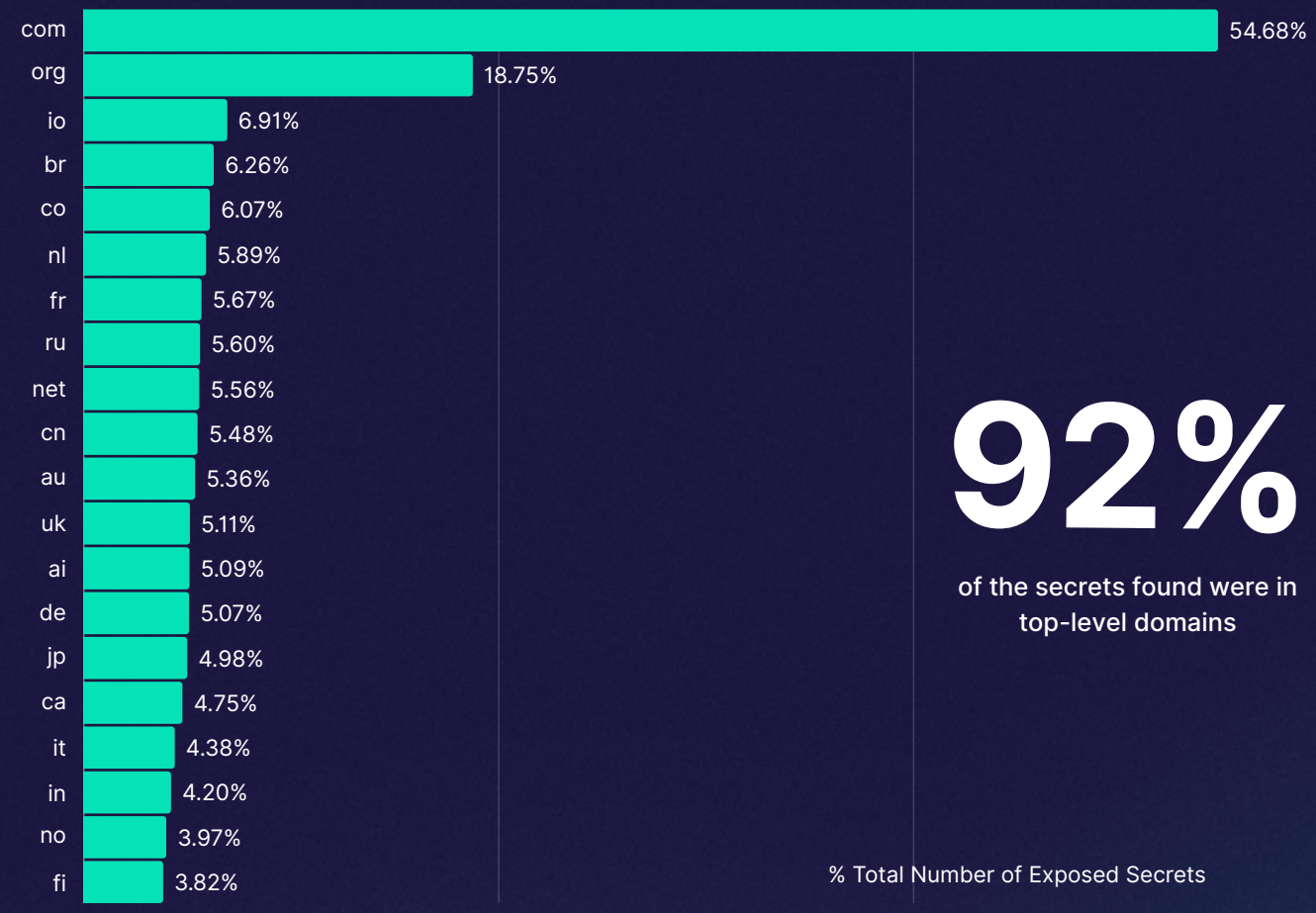
28

biggest number of secrets exposed per domain
(a gaming industry company located in the US)



Industries of the top 10 domain names with the highest number of exposed secrets

Findings



92%

of the secrets found were in top-level domains

Top 20 domain extensions with the highest number of exposed secrets

While .com domains continue to lead in market share, contributing to their top ranking for exposed secrets, it's noteworthy that the .ai domain is also prominent among established domains.
The .ai domain's rise is particularly significant, seeing a remarkable 160.77% growth from July 2022 to July 2023, expanding from 75,314 to 196,292 registrations (Domain Name Stat).
This surge, largely attributed to the popularity of ChatGPT, has made .ai domains increasingly susceptible to secret leaks. This trend reflects the growing relevance of AI-focused domains on the web and their security challenges.

Brazil's top-level domains rank #1 with

6.26%

of total domain extensions with exposed secrets

In the EU, Dutch top-level domains secure the #1 spot with

5.89%

of total domain extensions with exposed secrets

escape

High-Stake Findings: Exposed Stripe API Tokens

$20M

could have been accessed via exposed Stripe tokens

In our research, we uncovered a significant security lapse – unprotected Stripe API tokens on the internet, including **one token valued at $17M**, associated with an e-commerce company in North America (name protected for legal purposes). Despite Stripe's reputation for secure payment handling, it's evident that many individuals underestimate the gravity of protecting Stripe keys. These keys, when compromised, offer access to substantial financial assets. Our estimate indicates that the **discovered keys could potentially access up to $20 million.**

Upon notifying the organizations responsible for the exposed keys, we encountered widespread surprise. Promptly, they expressed eagerness to rectify the issue and sought our guidance. This highlights the urgent need for enhanced awareness and education on securing critical assets like Stripe keys.

While some developers lacked knowledge of how to protect these keys, they were thankful for our help. Quickly discontinuing the use of the exposed keys, they adopted more secure practices. This positive response underscores a collective commitment to learning and improvement within the ecosystem.

Our findings emphasize the ongoing importance of educating developers and organizations about securing payment keys, such as those used by Stripe. Companies need to understand the risks and know what to do to protect their online information. The discovery of these Stripe keys serves as an important reminder of the meticulous attention required when building online applications.

Standout Scenarios

Javascript Packing Trend

We've uncovered a substantial security risk tied to JavaScript code management. Specifically, some developers opt to compile all code, including sensitive setup files, into a single extensive JavaScript file for convenience. However, this approach unwittingly exposes crucial secrets to unauthorized access, which is critical for the seamless operation and intercommunication of the application.

35%

of the exposed secrets were found in a JavaScript file

2.1%

of the exposed secrets were detected due to the practice of compiling code into a single JavaScript file

Our findings extend beyond GitLab tokens; we've uncovered special keys and tokens crucial for essential app functions. Notably, these include keys granting access to databases, underscoring the broader spectrum of vulnerabilities associated with such coding practices.

Mitigating the JavaScript Packing Trend poses challenges for conventional secret scanning methods. This difficulty arises due to the common separation of build and deployment pipelines, with both being challenging to identify and integrate scans into. Additionally, some pipelines involve manual deployments, further complicating the scanning process.

On top of that, the issue extends beyond the build phase. Secrets can be dynamically loaded and exposed at runtime. In this scenario, the initial build may not contain any secrets, but the running system can dynamically load and expose them during execution.

Standout Scenarios

Decoding AWS Tokens

Tackling AWS tokens was a different, nuanced challenge. These tokens occasionally are meant to be public for a short time, particularly when granting temporary access to AWS S3 storage.

Deciphering the intended secrecy of these tokens required a meticulous examination of their use cases. We scrutinized their functionalities to ascertain whether they were designated as confidential or designed for temporary public exposure, enriching our classification process with contextual insights.

Standout Scenarios

Exposed ChatGPT Tokens: Importance of AI Projet Security

1.4%

of the exposed secrets were OpenAI API tokens

In our previous and ongoing research, we've observed challenges with new AI projects developed swiftly for AI integration. Occasionally, these projects inadvertently expose their AI tokens in publicly accessible website sections.

When building AI websites too quickly, token security may be overlooked. However, if discovered, these tokens could enable unauthorized usage of AI services. This can lead to unforeseen financial implications for website owners.

To protect these tokens, it's best to store them in a secure backend. Implementing rate limiting, such as Cloudflare WAF Rate limiting, adds an additional layer by restricting the frequency of information requests, helping in cost control.

For revenue-focused websites, user registration is beneficial. With registration, it's easier to monitor each user's usage.

Recommendations for mitigating risks

This extensive exposure of API secrets underscores a critical security issue. Immediate, strategic actions are necessary. Businesses must acknowledge the gravity of secret sprawl and implement rigorous measures to counter it.

Here are the essential steps to mitigate these risks:

- **Centralize Token Management:** Centralizing token management enables secure storage, access, and rotation. Consolidating all tokens in one location allows you to monitor their usage comprehensively, identifying potential vulnerabilities in your system.
- **Rotate Tokens Regularly:** Regularly rotating tokens mitigates the risk in case of compromise. For instance, AWS Secrets Manager supports the automated rotation of secrets.
- **Assign Tokens to Specific Teams or Services:** Ensure that only necessary personnel or services have access to each token by assigning them to specific teams or services.
- **Create a Revocation Process:** Establish a clear revocation process to promptly revoke tokens in the event of a compromise.
- **Grant Correct Permissions:** Grant only the necessary permissions for each token to minimize potential damage.
- **Limit Token Scope:** Restrict the access scope of each token within your system.
- **Monitor Usage Patterns:** Keep a vigilant eye on how tokens are used to identify any unusual activity.
- **Educate Your Internal Teams:** Ensure that all team members understand the importance of token security and adhere to best practices. Consider enhancing the security experience through gamification or implementing a Security Champion Program, following The Security Champion Program Success Guide.

These measures are crucial for protecting your API tokens and maintaining secure and compliant systems. For more detailed information, you can refer to the resources provided by AWS Secrets Manager and Hashicorp.

Recommendations for mitigating risks

Employing automated solutions for Continuous Attack Surface Testing (CAST)

Leveraging automated tools for continuous attack surface testing is crucial for maintaining system security.

These tools constantly check for security risks, including issues in APIs, websites, and other online components. CAST operates independently, identifying hidden mistakes or leaks that may go unnoticed by system administrators. When a problem is detected, CAST facilitates quick resolution, instilling trust and ensuring smooth and secure operations.

Without continuous attack surface testing, it's challenging to monitor the entire system, making it susceptible to unnoticed vulnerabilities. CAST intelligently identifies safe and potentially problematic areas, eliminating weak spots and enhancing overall system security.

Conclusion

Securing all your APIs is hard. It’s even harder when your keys and tokens get exposed involuntarily in real-world settings - from APIs to frontends. Your organization is now not only prone to data breach risks but also to severe financial implications.

Our study reveals that API secret sprawl extends across a diverse array of websites, industries, and domain types. Even modern tech industries are not exempt.

One of the most striking findings was the exposure of Stripe API tokens, which potentially gave unauthorized access to $20 million. This highlights the real and present dangers for each organization and the necessity of thoroughly **testing applications before and after their deployment.**

Moreover, we have identified emerging trends, such as the security demands of AI-driven projects and the risks associated with consolidating code into single JavaScript files. These findings suggest that the threat of API secret sprawl may escalate alongside the growth and complexity of new technologies.

Organizations must respond fast, adopting best practices for risk mitigation and integrating continuous, automated testing of their applications.

Are you looking to improve your organization's security posture, need to ensure compliance with standards like HIPAA, GDPR, and PCI DSS, or looking for guidance tailored to your specific application needs?

Our team is here for you. We'll help identify your online exposures and provide strategic advice on implementing effective security practices. Feel free to reach out!