Exposed Developer Secrets Surge: AI Drives 34% Increase in 2025

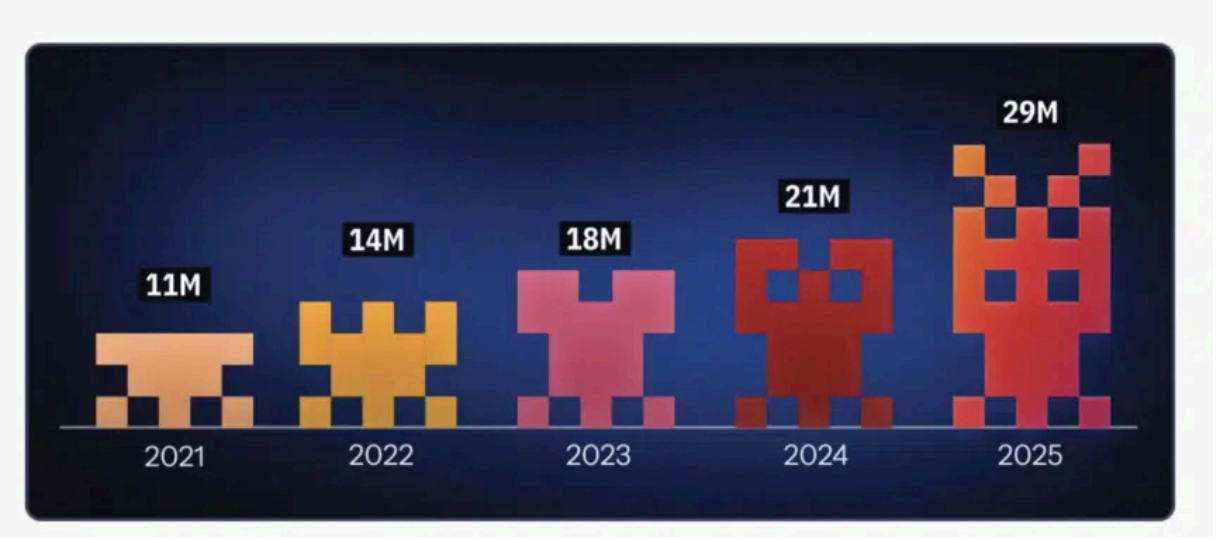
securityledger.com/2026/03/exposed-developer-secrets-surge-ai-drives-34-increase-in-2025

Paul Roberts March 17, 2026

Incidents of exposed developer secrets exploded in 2025, jumping by 34% from the previous year, as generative AI and coding assistants fueled dramatically faster software development—and a corresponding surge in exposed credentials.

That's the conclusion of GitGuardian's newly released **"State of Secrets Sprawl 2026"** report, which analyzes billions of commits on GitHub and other development platforms. The findings of the fifth annual report find that the rise of AI-assisted development comes with a cost: more hard-coded secrets embedded in source code and development tools – a leading cause of security compromises.

The numbers are stark. GitGuardian identified **28.76 million new secrets exposed in public GitHub commits in 2025**, a **34% increase over the previous year** and the largest annual jump the company has recorded. The spike reflects a broader transformation in software creation, as AI tools lower the barrier to coding. That has more people are writing software—often without formal knowledge of software development, let alone secure coding practices.



Developer secrets detected on public GitHub repos by year. (Image courtesy of GitGuardian)

> "AI scaled software delivery, and secrets sprawl is accelerating faster than remediation," the report notes in its executive summary.

More than a million AI-related credential leaks in 2025

The most dramatic increases are tied to AI infrastructure itself.

GitGuardian detected **1.27 million leaked secrets linked to AI services in 2025**, representing an **81% year-over-year increase**. These include API keys for large language models, vector databases, orchestration platforms, and other tools that underpin modern AI applications. Eight of the ten types of secrets with the fastest growth rates were tied to AI services.

Part of the reason: AI ecosystems are expanding rapidly and security controls often lag behind adoption. New platforms and services frequently lack mature safeguards such as push-protection scanning or automated revocation workflows.

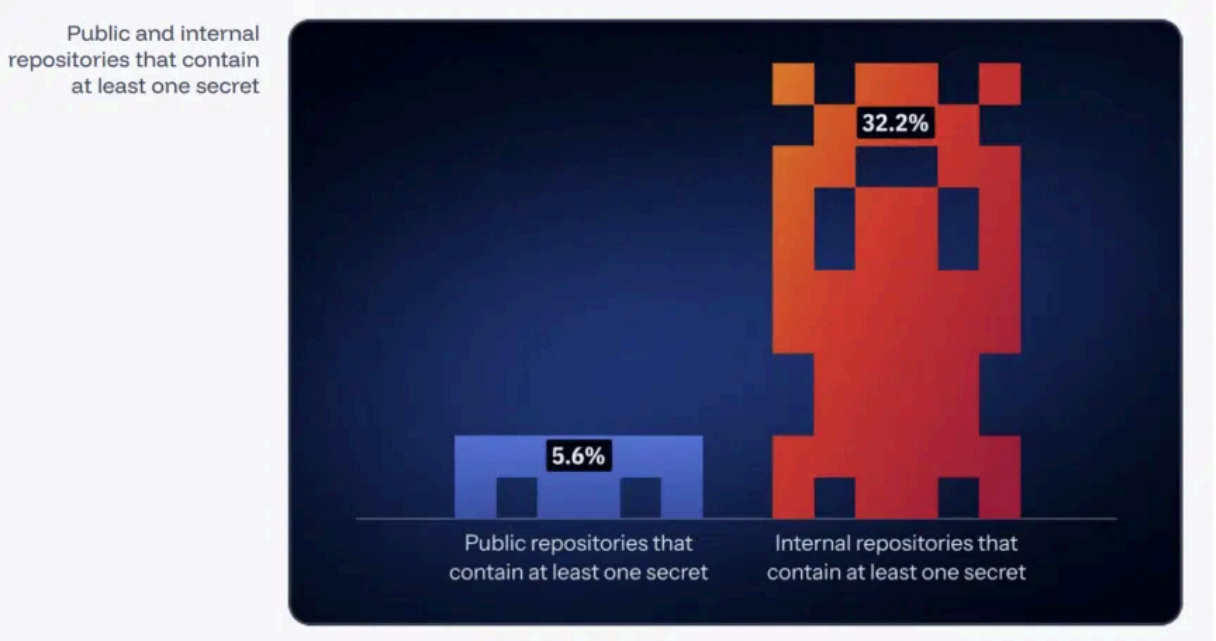
One example cited in the report is the sudden rise of new AI vendors whose credentials slip past existing detection systems. In just one example, GitGuardian researchers found **more than 113,000 leaked API keys for the emerging AI provider DeepSeek** in public repositories during 2025. At the same time, AI coding assistants themselves may contribute to the problem. Developers using tools like Claude Code leaked secrets in **roughly 3.2% of commits—more than double the baseline leak rate of 1.5% across GitHub**.

That doesn't necessarily mean AI tools are inherently insecure. Instead, the data underscores a familiar reality in security: automation amplifies both productivity and mistakes.

Internal systems: the bigger blind spot

Public GitHub repositories tend to get the most attention in discussions about exposed credentials. But GitGuardian's data suggests the more dangerous exposures often happen inside organizations.

According to the report, **internal repositories are roughly six times more likely to contain hardcoded secrets than public ones**.



GitGuardian data shows that 32% of internal repositories contained at least one hardcoded secret – six times the rate of public repositories. (Image courtesy of GitGuardian.)

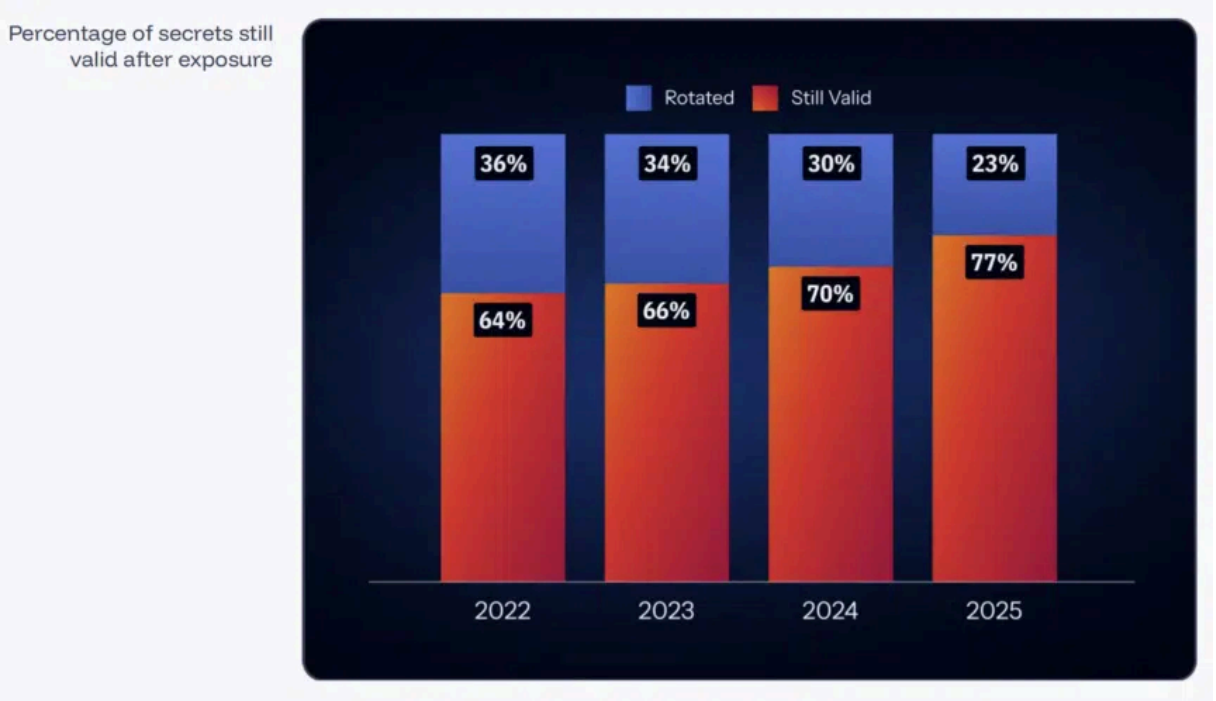
Private infrastructure also appears to contain more dangerous leaks. Secrets discovered in **self-hosted GitLab and Docker environments were three to four times more likely to be valid credentials**, meaning attackers could potentially use them immediately.

And not all leaks occur in code.The study found that **28% of credential exposures originate outside repositories entirely**, appearing instead in tools such as Slack, Jira, and Confluence. These leaks—often the result of pasted configuration files or troubleshooting logs—were **13% more likely to be categorized as critical** than secrets found directly in source code.

The takeaway: as software development workflows expand to include collaboration platforms, AI agents, and integrated pipelines, the boundaries of “the codebase” are dissolving.

The persistence problem: secrets rarely die

Even when exposed credentials are discovered, organizations often fail to revoke or rotate them. GitGuardian tracked credentials originally identified in 2022 and found that **64% remained valid and exploitable as of early 2026**.



More than 60% of secrets detected by GitGuardian in 2022 were still valid in 2025. (Image courtesy of GitGuardian)

That persistence is one of the most troubling findings in the report. In theory, revoking a compromised API key should be a routine task. In practice, organizations often hesitate to invalidate credentials tied to critical systems, fearing service outages or operational disruption.

As a result, leaked secrets can remain usable for years. The report describes these long-lived credentials as “durable access paths”—effectively leaving a door open for attackers long after a breach or exposure occurs.

Threats go local as AI agents expand the attack surface

The growth of AI development environments is also shifting where secrets live, with local developer systems posing a growing risk, GitGuardian found. That’s due to the fact that modern AI agents require access to local credentials like API tokens, database passwords, or cloud service keys to be able to perform- and automate tasks across systems. (See: ClawCode!)

This trend effectively expands the credential perimeter from centralized infrastructure to developer laptops and local environments.As GitGuardian CEO Eric Fourrier notes in the report: “AI agents need local credentials to connect across systems, turning developer laptops into a massive attack surface.”

That shift raises new risks, including prompt-injection attacks that trick AI agents into revealing or misusing sensitive credentials. It also complicates the traditional security

model. Instead of protecting a handful of centralized secrets stores, organizations must now manage sprawling webs of machine identities and tokens.

The next frontier: governing machine identities

What's the fix for the explosion of leaked secrets and development pipeline risks? GitGuardian argues that the security community needs to rethink how it approaches credentials altogether.

The current approach —detecting exposed secrets in code—is important, but addresses only part of the problem. A bigger challenge is managing the lifecycle of what the report calls **non-human identities (NHIs)**. That refers to things like service accounts, automation tokens, and API credentials that allow software systems to interact.

As AI-driven development accelerates, the number of these identities is exploding. The report calls for organizations to move beyond simple detection toward comprehensive governance: identifying which machine identities exist, who owns them, what they can access, and how their credentials are managed.

> Security teams must shift from asking "Where are my leaked secrets?" to "What non-human identities exist in my environment—and what can they access?"

In other words, the problem isn't just secrets sprawl. It's identity sprawl—now amplified by AI-powered software creation.

AI's "gift"? More code…and more risks

The GitGuardian data paints a clear picture: AI is accelerating both software innovation and security risk.

More developers, more code, and more automation are producing an unprecedented number of credentials embedded in modern software stacks. Meanwhile, remediation practices and identity governance are struggling to keep up.

If current trends continue, exposed secrets will remain one of the most reliable—and preventable—paths into corporate infrastructure.

And in an AI-driven development ecosystem, the scale of that problem is only growing.