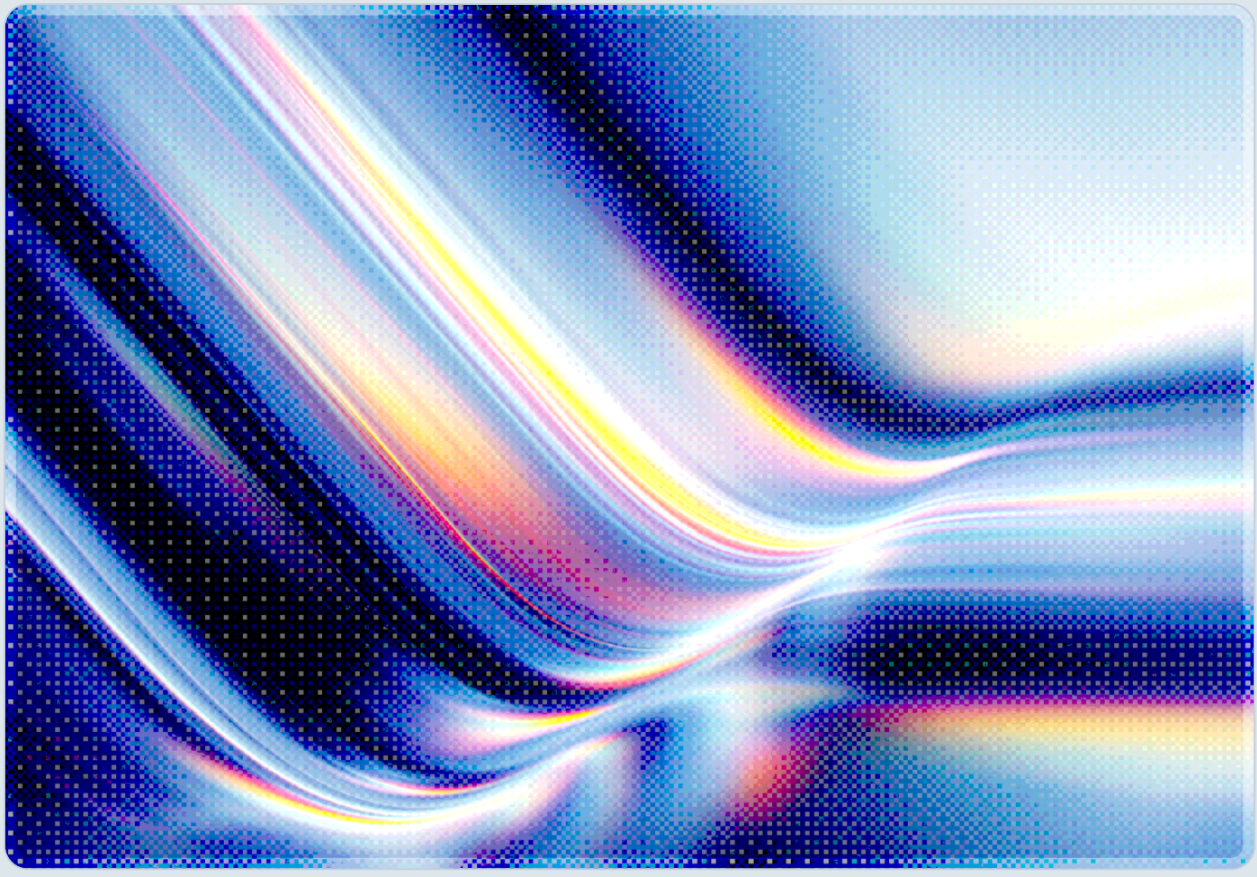
CSAI Foundation | Cloud Security Alliance

Five Minutes from Search to Key Vault: Agentic Secret Harvesting

Why Scheduled Rotation Alone No Longer Bounds Exposure, and What Must Carry the Load

2026-11-09

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

© 2026 Cloud Security Alliance. Some rights reserved.

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

This document was generated with AI assistance and has not undergone official CSA review and approval processes.

Key Takeaways

On November 5, 2026, Microsoft Threat Intelligence and Microsoft Security Research reported that threat actors are using ARTEX, an agentic offensive framework, and similar tools to search public GitHub repositories for exposed secrets and then act on what they find [1]. In one case Microsoft observed, an AI agent went from searching GitHub for client secrets to exfiltrating data from Azure Key Vault in five minutes [1]. Microsoft presents this as a single observed case, not a typical timeline. Still, it shows how little time defenders have once an agent finds an exposed secret. The agents did more than collect keys. They checked what each key could reach and enumerated subscriptions, permissions, and AI deployments. When an API call failed, they corrected the request and retried successfully within roughly 28 to 80 seconds [1].

Attackers have exploited leaked credentials within minutes for years. What agentic tooling adds is the ability to turn one leaked key into a compromise of the wider environment, including the contents of vaults, with little operator involvement and before any manual response can begin. Scheduled rotation still matters because it removes stale leaked credentials, which make up much of what harvesters find. However, a 30-, 60-, or 90-day rotation schedule cannot contain a freshly leaked credential that an autonomous agent finds, validates, and uses within minutes of discovering it. Deleting the commit or rewriting Git history does not help either, because the secret stays valid until someone revokes it [1][2]. Organizations should now assume that any secret pushed to a public surface is compromised as soon as it is exposed. Event-driven revocation that completes in seconds has to carry the load that rotation cannot. For high-value workloads, organizations should also move away from static secrets to federated, short-lived credentials, which leave far less worth harvesting.

Background

From Opportunistic Scrapers to Goal-Directed Agents

Attackers have exploited leaked credentials quickly for years. In a 2023 honeypot study, Orca Security found that an AWS key planted on GitHub was used two minutes after it was exposed. The same study found that assets exposed on GitHub, HTTP, and SSH were all discovered in under five minutes [4]. Cloud providers and code hosts have responded with automation of their own. In a test published by Palo Alto

Networks Unit 42 in September 2026, AWS attached its `AWSCompromisedKeyQuarantineV3` managed policy to an IAM user about 10 seconds after the access key was pushed to a public repository, and GitHub sent its notification about one second later [5]. This race between harvesters and platforms has long shaped how organizations defend against leaked credentials.

What changed in 2026 is what happens after discovery. Earlier exploitation was limited by how much time human operators had to act on what they found [1]. Scrapers have commonly been associated with bulk monetization such as cryptomining [4], and they ran fixed scripts against the keys they collected. According to GitGuardian's analysis of Microsoft's research, ARTEX is an agentic penetration-testing framework that ships with few tools and instructions and leaves most decisions to the agent. An operator sets a goal and can step in later to correct the agent if it drifts [2]. Microsoft observed operators submitting large volumes of GitHub queries starting in August 2026. The queries combined a target organization's name with secret-related keywords and keywords for cloud or AI service providers [1]. Discovery is therefore aimed at specific organizations. Exploitation can then proceed largely on its own, with operators stepping in only when the agent goes off course [2].

The Scale of the Exposed Surface

The exposed surface these agents search is large. GitGuardian's *State of Secrets Sprawl 2026* counted 28.65 million new hardcoded secrets in public GitHub commits in 2025, a 34% increase over the prior year [3]. Secrets for AI services grew faster still: 1,275,105 leaked in 2025, an 81% rise [3]. The same report found 24,008 unique secrets in MCP-related configuration files. It also found a 3.2% secret-leak rate in commits made with Claude Code assistance, compared with a 1.5% baseline across all public commits [3]. GitGuardian cautions that this gap should not be read as a simple tool failure, and the comparison does not account for differences in repository type or project maturity. It does indicate that secret-scanning coverage needs to pay explicit attention to AI-assisted workflows. Leaked secrets also stay usable for a long time. When GitGuardian retested in January 2026, more than 64% of the secrets it had confirmed as valid in 2022 were still valid [3].

The offensive tooling is also widely deployed. GitGuardian reports that Censys counted about 1,400 publicly exposed ARTEX instances on November 4, 2026, and that Check Point reported more than 1,800 the following day. ARTEX management interfaces do not have to be exposed to the internet, so both counts likely understate the true number [2]. Cybersecurity Dive, summarizing reporting from Microsoft and other vendors, says that CrowdStrike has linked the framework to attacks on more than 200 organizations since mid-October. It also says that ARTEX was previously used by the China-nexus actor Storm-0940 against downstream Microsoft 365 tenants [6]. Microsoft, however, did not attribute the GitHub scanning activity to a named actor [1], so readers should not assume that Storm-0940 is behind it.

Security Analysis

Anatomy of the Five-Minute Chain

Microsoft's report describes the attack chain in three stages but does not give a step-by-step breakdown [1]. First, the agent searched GitHub for exposed client secrets. Next, it used a leaked credential to obtain a token for a single-tenant OAuth application in the target organization, a step Microsoft observed in multiple cases. Finally, acting as that application, it enumerated cloud resources and exfiltrated secrets from Azure Key Vault. The stolen credential was an application secret, not a user password. As a result, no human sign-in, MFA prompt, or interactive session occurred at any point in the chain. Every step was an authenticated API call that the target's identity system treated as legitimate service-principal activity.

The secret types Microsoft listed suggest that the attackers aim to reach further credentials rather than a single resource [1]. The list includes storage account keys, App Service settings, connection strings and publishing credentials, Azure Kubernetes Service user credentials, OAuth application secrets, and AI and machine-learning connections and datastores. After gaining access, the agents signed in as service principals across dozens of applications and multiple tenants. They also used Azure VM Run Command to execute scripts, searched Azure DevOps for further secrets, and accessed SharePoint, OneDrive, and mailbox items through Microsoft 365 [1]. A vault holds the secrets for many other systems. When the leaked identity can read a vault, the scope of the compromise depends on what is in that vault, not on the original key.

Self-Correction Removes Incidental Friction

In our assessment, the agents' ability to recover from errors is the most important technical detail in Microsoft's report. In one sequence, a request to enumerate Key Vault secret metadata failed with HTTP 400 because the agent sent a data-plane API version to Azure Resource Manager (ARM). It reran the request with ARM API version 2023-07-01 and received HTTP 200 about 28 seconds later [1]. Similar recoveries took about 60 seconds for an Azure SQL firewall-rule query and about 80 seconds for an Application Insights query [1]. A conventional script would typically stop at an error like this until an operator intervened. Unusual API versions, uncommon services, and environment-specific quirks have therefore slowed attackers down, even though nobody designed them as controls. An agent that reads the error and fixes its own request is much less likely to be stopped by these obstacles.

Why Rotation Windows Collapse

Scheduled rotation and incident-driven revocation solve different problems. Rotating credentials every 30 to 90 days limits how long a credential stays usable after an undetected theft. That helps against slow, low-volume abuse and against the large number of stale secrets that remain exposed long after they leak. It does not limit the damage from a fresh credential that is found and used within minutes. The table below compares the attacker and platform timelines documented in public research with measures of how quickly enterprises can respond.

Event or measure	Observed value	Source
AWS quarantine policy attached after public GitHub push	~10 seconds	Unit 42 [5]
GitHub honeypot AWS key first used by attacker	2 minutes	Orca Security [4]
Agent recovers from a failed cloud API call	~28–80 seconds	Microsoft [1]
GitHub search to Azure Key Vault exfiltration (single case)	~5 minutes	Microsoft [1]
Organizations taking weeks or more to offboard an API key	~40% of respondents	CSA / Astrix survey [11]
Organizations with a formal process for rotating or rolling back API keys	16% of respondents	CSA / Astrix survey [11]
Secrets confirmed valid in 2022 that were still valid in January 2026	>64%	GitGuardian [3]

Attacker timelines are measured in minutes, while enterprise offboarding is measured in weeks, a difference of roughly three to four orders of magnitude. CSA's 2024 *State of Non-Human Identity Security* survey, developed with Astrix Security, found that only 20% of organizations had a formal process for offboarding and revoking API keys. Only 16% had one for rotating them, and nearly 40% took weeks or more to offboard keys [11]. Respondents named lack of credential rotation as the most common cause of non-human identity (NHI) security incidents, at 45% [11]. These figures predate the agentic campaigns, but they suggest that the 2023-era scrapers Orca observed would already have outpaced most enterprises. An organization that relies on manual response has no meaningful rotation window for a freshly leaked credential when the attacker is an agent that finishes in minutes.

Platform-side protections help only in part. Unit 42 notes that AWS's quarantine policy denies a defined list of high-risk actions but does not disable the key, so whoever holds the key can still perform actions outside that list [5]. On the Azure side, Microsoft Entra ID Protection includes a leaked-credentials detection for workload identities. It checks credentials found on GitHub, paste sites, and other sources against valid Entra credentials [14]. That detection raises the risk level of the service principal rather than revoking the secret. Its effect therefore depends on risk-based Conditional Access policies for workload identities, which require Workload ID Premium licensing and cover only single-tenant applications [1][8]. Through GitHub's secret scanning partner program, many providers are notified when their token formats appear in public repositories, and some providers revoke those tokens. Coverage still depends on each provider's participation and response [15]. As a result, automatic protections are not uniform across providers, secret types, or the many third-party and AI-provider API keys that make up a growing share of leaks. Removing a secret from the repository does not revoke it. Even after a compromised application credential is removed, tokens already issued from it remain valid until they expire [1].

Detection Opportunities Remain

Agentic speed does not make these attacks invisible, and defenders still have clear signals to look for. Microsoft recommends detecting behavior rather than relying only on network indicators [1]. Its examples include one address authenticating to three or more service principals with a mix of successes and failures, and one application requesting tokens for ARM, Key Vault, Storage, and Microsoft Graph within a short window. They also include the Entra error codes AADSTS7000215 and AADSTS7000222, which show that someone is presenting a credential for the application. A failure followed by a success should be treated as urgent [1]. Microsoft also reports that honeypots, decoys, and tarpits reduced the attack traffic that reached real assets in its testing [1]. Decoy credentials may work particularly well against agentic adversaries, because an agent that validates and pursues every key it finds will also pursue planted ones.

Recommendations

Immediate Actions

Organizations should first find out whether their secrets are already exposed. Scan public and internal repositories, including full commit history, for hardcoded credentials. Start with OAuth application secrets, service-principal credentials, and anything that grants read access to a key vault [1][2]. Treat any secret found on a public surface as compromised even if it has since been deleted [1]. Revoke or rotate it. Also rotate every secret stored in any vault that the affected identity could read, because the attacker

should be assumed to have harvested the vault's contents as well [1]. Removing a client secret does not end access through tokens that have already been issued. For high-value applications, also disable the service principal or revoke its sessions [1].

Teams should then review their logs. Microsoft observed follow-on service-principal sign-ins beginning October 28, 2026, but targeted GitHub querying began in August 2026, and Microsoft advises searching as far back as log retention allows [1]. The review should cover Entra service-principal and non-interactive sign-in logs, Azure Activity, Key Vault, Storage, Azure DevOps, and Microsoft 365 audit logs. Teams should look for the indicators and behavioral patterns Microsoft published, including failed sign-ins [1]. On AWS, teams should alert on `AttachUserPolicy` events that reference the `AWSCompromisedKeyQuarantine` policy. AWS Support cases about exposed keys should go to the security team as well as to cloud engineering [5].

Short-Term Mitigations

Over the next several weeks, organizations should turn secret-exposure response into an automated pipeline instead of a ticket queue. A detection from push protection, a scanning partner, or a provider quarantine event should trigger revocation and reissuance directly through the secrets manager. A human should review the action afterward rather than approve it beforehand. CSA's *Defining Non-Human Identity* guidance describes the capabilities this requires [10]. They include automatic rotation for security purposes and automated propagation of new credentials to dependent consumers, with validation before the old credential is retired. They also include rollback if propagation fails. Automation of this kind can cut response time from weeks to seconds, and validation and rollback reduce the risk that a rotation breaks dependent systems. Organizations should still expect to find undocumented consumers during early runs and should rehearse the pipeline before relying on it.

Preventive controls at the source and at the identity layer complement the response pipeline. Teams should enable GitHub push protection at the repository and organization level rather than relying only on the account-level default. They should also use delegated bypass so that contributors cannot push a detected secret without approval [7]. Push protection covers only supported secret patterns and requires GitHub Secret Protection for private repositories, so it should complement history scanning, not replace it. On the identity side, organizations should restrict who can register applications and apply Conditional Access for workload identities to service principals, including risk-based policies driven by leaked-credential detections [8][14]. They should also reduce service-principal permissions so that one leaked application secret cannot read every vault in a subscription [1]. Finally, decoy credentials and canary secrets in repositories and vaults provide an inexpensive early warning of agentic reconnaissance [1]. Adversaries may learn to recognize common canary-token formats, so decoys should be varied and realistic.

Strategic Considerations

The lasting fix is to stop issuing secrets that are worth stealing. Microsoft recommends moving away from secret-based authentication toward managed identities [1]. Workload identity federation lets external workloads such as CI/CD pipelines exchange a platform-issued token for short-lived cloud access without storing a client secret [9]. A credential that never exists in a repository, configuration file, or developer laptop cannot be harvested from one. Federation shifts risk rather than eliminating it, however. Trust policies must be limited to specific repositories, branches, and environments, and CI/CD runners become high-value targets because they can mint tokens. Credentials that must remain static should be stored in a vault, narrowly scoped, and monitored for suspicious behavior such as mass secret downloads, which CSA's NHI guidance specifically calls out as something to detect [10].

Organizations should also change how they measure credential risk. Rotation frequency is easy to audit, but it says little about exposure when exploitation takes minutes. Three measures track exposure more directly: the share of workloads that use federated or ephemeral credentials, the median time from secret detection to revocation, and how much each remaining static secret could expose if it leaked. AI-assisted development and agent tooling also deserve attention as sources of leakage, as GitGuardian's data on MCP configuration files and AI-assisted commits indicates [3]. Secret controls should therefore cover agent configuration files, local developer environments, and AI-provider keys alongside traditional cloud credentials.

CSA Resource Alignment

This campaign targets non-human identities, and the most directly applicable CSA guidance is *Defining Non-Human Identity*, published by the CSA Identity and Access Management Working Group [10]. It recommends managing ephemeral credentials differently from static ones. Static credentials should be kept in vaults under centralized governance and rotated automatically, which limits how long a leaked credential remains useful (its "shelf-life"). Incident response should first revoke compromised tokens and keys, then trace the identity graph to determine how far the compromise reached [10]. The ARTEX findings suggest one change in how organizations apply this guidance. A freshly leaked credential can now be exploited well within the shelf-life that scheduled rotation provides. Event-driven revocation and the move to ephemeral credentials should therefore take priority over tuning rotation intervals, while scheduled rotation continues to retire stale secrets.

CSA's *State of Non-Human Identity Security* survey report, produced with Astrix Security, provides the organizational baseline used in this note [11]. It shows that few organizations have formal processes for revoking or rotating credentials, that offboarding often takes weeks, and that lack of rotation is the most

common cause of NHI incidents. Organizations can compare their own revocation times against this baseline.

The *AI Controls Matrix (AICM) v1.1* provides a control framework for putting these recommendations into practice [12]. Its Identity and Access Management domain and its Cryptography, Encryption and Key Management domain cover the credential lifecycle, secret storage, and key rotation. Its Threat and Vulnerability Management and Logging and Monitoring domains cover the scanning and behavioral detection described above. In addition, *MAESTRO*, CSA's threat-modeling framework for agentic AI, gives defenders a way to model adversarial agents like ARTEX [13]. Threat models for any system whose credentials could appear on a public surface should assume that adversaries have the goal-directed, self-correcting capabilities Microsoft observed.

References

[1] Microsoft Threat Intelligence and Microsoft Security Research. "Threat actors use agentic AI tooling to find and abuse exposed secrets." Microsoft Security Blog, November 5, 2026.

[2] Dwayne McDaniel. "The Agents Have Now Found Your Secrets On GitHub: ARTEX Is Abusing Public Secrets Right Now." GitGuardian Blog, November 6, 2026.

[3] Anna Nabiullina and Carole Winqwist. "The State of Secrets Sprawl 2026." GitGuardian, March 17, 2026.

[4] Orca Security. "2023 Honeypotting in the Cloud Report." Orca Security, June 2023.

[5] Margaret Kelley. "From Exposure to Lockdown: How AWS Neutralizes Compromised IAM Credentials through Managed Policies." Palo Alto Networks Unit 42, September 21, 2026.

[6] David Jones. "Hackers enlist Artex to scan GitHub repositories for exposed secrets." Cybersecurity Dive, November 6, 2026.

[7] GitHub. "About push protection." GitHub Docs, accessed November 9, 2026.

[8] Microsoft. "Conditional Access for workload identities." Microsoft Learn, accessed November 9, 2026.

[9] Microsoft. "Workload identity federation." Microsoft Learn, accessed November 9, 2026.

[10] Cloud Security Alliance Identity and Access Management Working Group. "Defining Non-Human Identity." Cloud Security Alliance, 2026.

[11] Cloud Security Alliance and Astrix Security. "The State of Non-Human Identity Security." Cloud Security Alliance, September 2024.

[12] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2025.

[13] Cloud Security Alliance. "MAESTRO: Agentic AI Threat Modeling Framework." CSA Labs, accessed November 9, 2026.

[14] Microsoft. "Securing workload identities with Microsoft Entra ID Protection." Microsoft Learn, accessed November 9, 2026.

[15] GitHub. "Secret scanning partner program." GitHub Docs, accessed November 9, 2026.