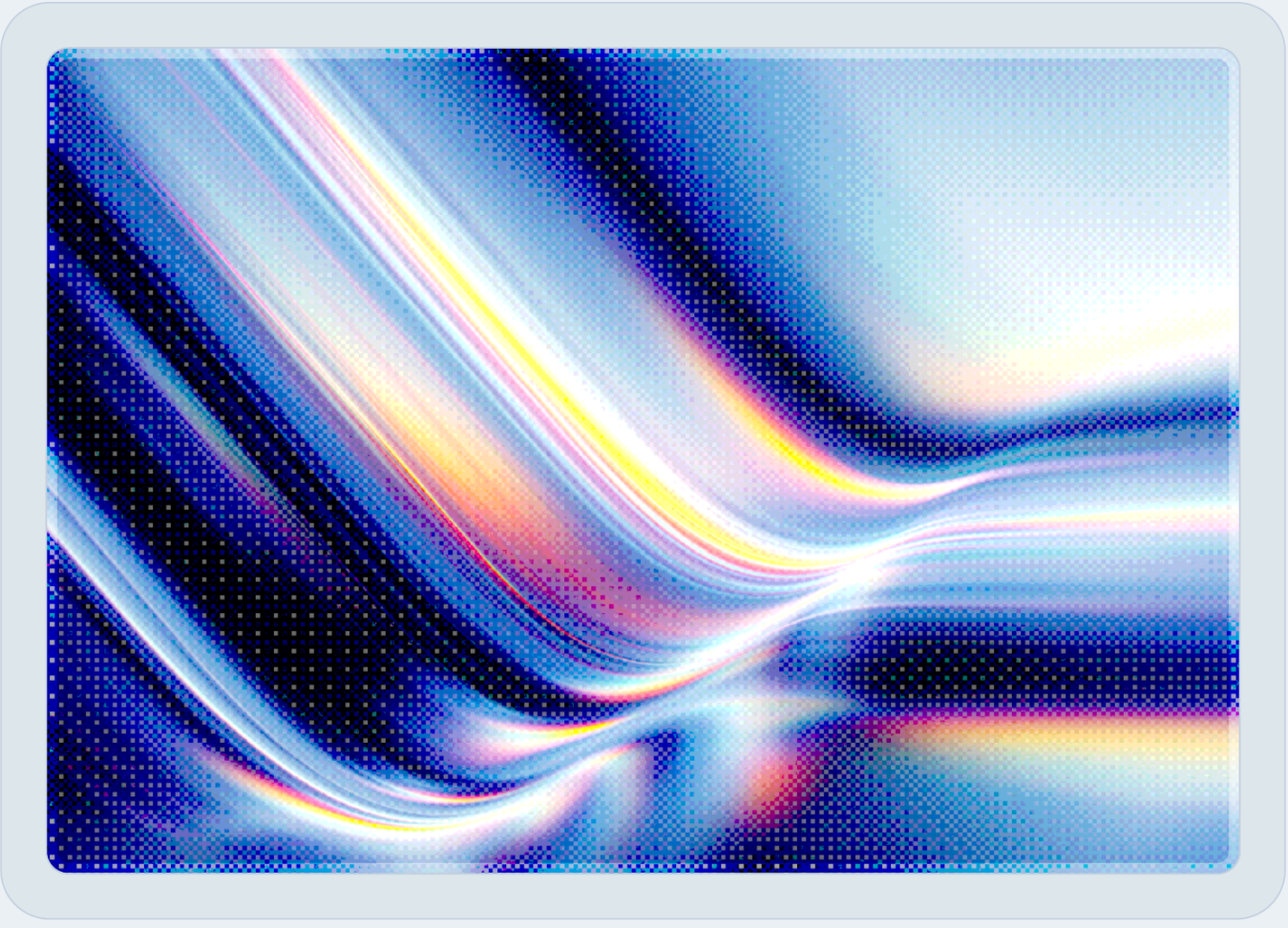
CSAI Foundation | Cloud Security Alliance

Credentials as the Payload: Secret Harvesting and Revocation Debt

Why Exploitation Now Targets Secrets, and Why Patching and Rotation Alone No Longer End an Incident

2026-11-27

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

Evidence from incident response reporting and recent supply chain campaigns indicates that, for a growing set of intrusions, the objective has shifted from the compromised system to the credentials that system can reach. Mandiant's M-Trends 2026 reports that threat actors are "harvesting long-lived OAuth tokens and session cookies" to bypass standard defenses, that attackers who compromise SaaS vendors steal hard-coded keys and personal access tokens to pivot into downstream customer environments, and that prior compromise is now the top initial infection vector in ransomware operations at 30% [1]. The software supply chain worms of 2025 and 2026, from Shai-Hulud through ChainDrop and the November 2026 wfx-loot campaign, were built primarily as credential collectors rather than as destructive or ransomware payloads [3][9][11]. The second Shai-Hulud wave did carry a wiper, but it fired only when the worm could not find tokens to steal or a channel to exfiltrate them, which underscores that theft was the primary objective [19]. In these operations the exploited package, repository, or server is the delivery vehicle, and the secrets it exposes are the product.

This shift creates what this note calls *revocation debt*, a framing of our own rather than an established industry term: the accumulated stock of credentials that have been exposed but never invalidated. GitGuardian found that more than 64% of secrets confirmed valid in 2022 were still valid when retested in January 2026 [2]. Mandiant's investigation of the 2024 Snowflake customer intrusions found that at least 79.7% of the accounts the attacker used had prior credential exposure, with some credentials traceable to infostealer infections from November 2020 [7]. Each unrevoked credential is a liability that can be drawn on years after the original exposure, by actors who had nothing to do with stealing it.

Revocation debt also changes what it means to close an incident. Patching a vulnerability does not invalidate keys stolen through it; CISA's guidance for the 2025 SharePoint "ToolShell" exploitation required organizations to rotate ASP.NET machine keys before and after patching [6]. GitGuardian observed that some victims of the GhostAction campaign had been compromised in earlier waves, apparently never rotated their secrets, and were hit again with the same access [20]. The November 2026 wfx-loot campaign showed a related pattern: Aikido Security found that accounts compromised in GhostAction were significantly more likely to be hit by wfx-loot, although the reason for that overlap has not been confirmed [3][18]. Organizations should treat confirmed credential invalidation, following code remediation, as the closing step of any incident involving code execution or account takeover, and should measure and pay down their revocation backlog as a standing program.

Background

Credential abuse has been a leading breach vector for years, but the mechanics have changed. Verizon's 2025 Data Breach Investigations Report found credential abuse to be the most common initial access vector at 22% of breaches, narrowly ahead of vulnerability exploitation at 20% [14]. Mandiant's 2026 data shows exploits leading initial infection vectors at 32% of intrusions, while prior compromise, meaning access obtained in an earlier intrusion, ranked third at 10% of intrusions overall and was the top vector in ransomware operations at 30%, double its 2024 share of 15% [1]. The same report found that the median time between initial access and handoff to a secondary threat group fell from more than eight hours in 2022 to 22 seconds in 2025 [1]. Taken together, these figures suggest an ecosystem in which access and the credentials that sustain it are traded as commodities between specialized actors.

The volume of exposed secrets has grown alongside that market. GitGuardian counted 28.65 million new hardcoded secrets in public GitHub commits during 2025, a 34% year-over-year increase and the largest single-year jump the company has recorded [2]. Secrets for AI services alone reached 1,275,105, up 81%, and 24,008 unique secrets appeared in Model Context Protocol configuration files, of which 2,117 were confirmed valid [2]. GitGuardian also reported that internal repositories were roughly six times more likely than public ones to contain hardcoded secrets, and that about 28% of incidents originated entirely outside code, in collaboration tools such as Slack, Jira, and Confluence [2]. The growth in non-human identities, including service accounts, CI tokens, OAuth grants, and AI agent credentials, means that each developer workstation or build runner now holds far more reusable access than it did a few years ago. Readers should note that much of the quantitative evidence in this field, including figures cited in this note, comes from vendors such as GitGuardian, Aikido Security, StepSecurity, Tenable, and Whiteintel that sell products addressing the problem they measure.

Attackers have adapted their tooling to that density. The first Shai-Hulud worm, the subject of a September 2025 CISA alert, compromised the npm registry, sent stolen credentials to attacker infrastructure, and posted them publicly to GitHub repositories; CISA's guidance centered on immediate rotation of developer credentials [9]. GitGuardian's analysis of the second Shai-Hulud wave found 33,185 unique secrets across 6,943 compromised machines, 59% of which were CI/CD runners rather than personal workstations [2]. Successor campaigns in 2026 refined the model. Tenable reported that Mini Shai-Hulud compromised more than 170 npm and PyPI packages, including packages that carried valid SLSA Build Level 3 provenance attestations [10], and Microsoft described ChainDrop in August 2026 as a self-propagating credential-stealing worm affecting more than 400 npm packages from unrelated publishers [11]. StepSecurity reported that ChainDrop used Ethereum transactions for command and control, which complicates takedown [12].

Security Analysis

Exploitation as a Collection Mechanism

The common thread across recent campaigns is that code execution is spent on collection. The wfx-loot campaign, which began on November 10, 2026, illustrates the pattern in its most developed form. Using valid stolen GitHub credentials rather than any vulnerability in GitHub Actions, an attacker operating 4,837 compromised accounts probed 61,764 repositories [3][4][18]. Aikido Security, a supply chain security vendor, reported that the stealer payload executed in roughly 1,800 of them; its own report gives figures of both 1,806 and 1,835 [18]. The payload serialized every repository secret and variable, scraped the memory of the Actions runner processes to recover decrypted secrets that the workflow never referenced, and collected cloud credentials, SSH and GPG keys, registry tokens, Terraform state, and API keys for AI providers including OpenAI, Anthropic, and Gemini [4][5]. The malicious change was then reverted within about a minute and the workflow run and logs deleted, often leaving only a gap in run numbers as evidence [3][5].

Server-side exploitation follows the same logic. The ToolShell campaign against on-premises SharePoint in July 2025 used code execution to extract ASP.NET machine keys, which protect ViewState, cookies, and session data, and CISA warned that malicious payloads "can be used to obtain machine keys" that would allow continued access [6]. Mandiant's 2026 reporting adds that edge-device intruders are using native packet capture to intercept plaintext credentials in transit and that the QUIETVAULT credential stealer checks targeted machines for local AI command-line tools and runs prompts to search for configuration files [1]. The last observation is notable because it shows attackers using a victim's own AI tooling as a discovery aid for secrets, a technique that is likely to spread as agentic tools gain broad file system access.

Third-party integrations extend the reach of each stolen credential. In August 2025, the actor tracked as UNC6395 used OAuth tokens stolen from the Salesloft Drift integration to export data from customer Salesforce instances, and Google Threat Intelligence observed the actor searching that data for further secrets, including AWS access keys, passwords, and Snowflake access tokens [8]. A single compromised integration therefore yielded both data and a second generation of credentials belonging to downstream organizations. Credential reuse leading to further compromise is not new, as pass-the-hash and lateral movement with harvested domain credentials long predate SaaS. This cross-organizational cascade is what sets current credential harvesting apart: a successful collection can seed intrusions at organizations the attacker never directly targeted.

The Anatomy of Revocation Debt

Revocation debt accumulates because exposure is fast and cheap while invalidation is slow and organizationally expensive. A leaked secret can be copied in seconds, but revoking it requires knowing that it exists, who owns it, what depends on it, and how to replace it without causing an outage. A 2024 CSA survey on the state of non-human identity security found that only 20% of organizations had formal processes for offboarding API keys and that nearly 40% took weeks or longer to do so [16]. Under those conditions, credentials outlive the incidents that exposed them, the employees who created them, and sometimes the systems they were issued for.

The persistence of that debt is measurable. GitGuardian's retest of credentials first confirmed valid in 2022 found that the validity rate declined only modestly, from almost 70% in January 2025 to above 64% in January 2026 [2]. The Snowflake intrusions show the consequence: credentials stolen by infostealers as early as 2020 were still valid, had not been rotated, and were usable without multi-factor authentication when UNC5537 used them in 2024 against approximately 165 potentially exposed customer organizations [7]. In the Miasma compromise of Red Hat npm packages, disclosed in June 2026, threat intelligence firm Whiteintel reported that a Red Hat GitHub credential had appeared in infostealer logs weeks before the breach, which may have enabled the account takeover [13][22]. These cases suggest that the harvest and the exploitation of a credential are frequently separated in time and in actor. That separation makes revocation, or credentials that expire or are bound to context so they cannot be replayed, the controls that most reliably break the chain.

GitHub Actions campaigns demonstrate how unpaid debt compounds. GitGuardian reported that the October 2026 GhostAction wave reached roughly 1,900 compromised accounts, some of them victims of earlier waves whose secrets had never been rotated, and that in some cases the attacker simply edited a workflow left behind by an earlier wave [20][24]. Aikido found that accounts compromised in GhostAction were significantly more likely to be hit by wfx-loot, and in more than 100 repositories the new payload was injected into GhostAction's own workflow files [3][5][18]. Aikido assessed the two campaigns as probably the work of different actors, and Aikido and StepSecurity both suggested that stolen GitHub credentials are sold or shared among multiple groups [5]. One plausible explanation for the overlap, not yet confirmed, is that GitHub account tokens exposed in the first campaign were never revoked and were later reused or resold.

Why Patching and Rotation Are Not Enough

Two common responses to credential exposure leave debt in place. The first is treating the vulnerability patch as the end of the incident. CISA's ToolShell guidance prescribed a rotate, patch, rotate sequence for machine keys followed by an IIS restart, and it warned that malicious modules left in configuration files

would persist and reload after the restart [6]. A patched server whose machine keys have been stolen remains exposed, because the attacker no longer needs the vulnerability to forge authenticated requests.

The second is rotating only the secrets an incident obviously touched. The wfx-loot payload reached beyond the secrets a workflow referenced by reading runner memory, and it established cloud persistence that survives secret rotation: it created long-lived AWS access keys, rewrote IAM role trust policies to allow any AWS principal to assume them, and added federated identity credentials to Azure applications [5][18]. Rotating the repository secrets in such an environment leaves the attacker holding new, independent credentials that were never in the repository. Microsoft and StepSecurity guidance on ChainDrop similarly stressed ordering (containment, inventory, rotation from a clean host, removal of persistence, and rebuild), because rotating on a still-compromised host issues fresh credentials to the attacker [11][12].

The following table summarizes how recent incidents illustrate each dimension of the problem, pairing what each campaign harvested with what remained exploitable after the initial fix.

Incident	What was harvested	What outlived the initial fix
Snowflake customer intrusions (2024) [7]	Customer database credentials from infostealer logs dating to 2020	Passwords never rotated; no MFA to blunt reuse
SharePoint ToolShell (2025) [6]	ASP.NET machine keys	Keys remained valid after patching unless rotated
Salesloft Drift (2025) [8]	OAuth tokens; AWS keys and Snowflake tokens found in exfiltrated data	Downstream secrets stored in CRM records
Shai-Hulud waves (2025) [2][9]	Developer, npm, GitHub, and cloud credentials, largely from CI runners	Published tokens reusable by any actor
GhostAction waves (2025–2026) [20]	Repository secrets, including registry tokens and cloud keys	Unrotated credentials and leftover workflows reused in later waves
wfx-loot (November 2026) [3][5][18]	All CI secrets, runner memory, cloud and AI provider keys	AWS trust-policy backdoors and Azure federated credentials that survive rotation

Implications for AI and Agentic Systems

AI service credentials have become a routine and fast-growing target of credential harvesting. GitGuardian counted 1.28 million exposed AI-service secrets in 2025, up 81%, and the wfx-loot payload specifically collected keys for major AI providers [2][4][5]. GitGuardian also found that commits co-authored with Claude Code showed a 3.2% secret-leak rate against a 1.5% baseline, and together with the volume of valid secrets in MCP configuration files, these figures indicate that AI-assisted development is adding to the exposure base [2]. GitGuardian cautions that the commit-level figure should not be read simply as a tool failure, since developers still decide what is pushed [2]. Agentic systems compound the issue because an agent typically holds several delegated credentials at once, and an attacker who obtains its configuration inherits all of them. The QUIETVAULT observation [1] suggests that at least some adversaries already treat local AI tooling as both a target and an instrument for credential discovery.

Recommendations

Immediate Actions

Organizations should first check for exposure to the active wfx-loot campaign. Security teams should search every repository, including private ones, for activity since November 10, 2026, matching the `wfx-docs` branch, the "docs: dependency audit notes" probe pull request, the documented commit messages ("ci: refresh build cache," "chore: add ci helper," and "chore: add dependency audit workflow"), and gaps in GitHub Actions run numbers [3][5][18]. Any account associated with even a probe should have its personal access tokens, OAuth grants, SSH keys, and sessions revoked and should be re-enrolled with phishing-resistant MFA [5]. Where the payload ran, every secret the job could reach must be rotated, not only those it referenced, and cloud accounts must be checked for trust policies containing `"AWS": "*"`, unexpected `CreateAccessKey` calls, and `wfx-ci-*` federated credentials in Azure [5][18].

More broadly, incident response playbooks should be amended so that no incident involving code execution, malicious package installation, or account takeover is closed until credential invalidation is confirmed. That confirmation should include a check that the revoked credential no longer authenticates, since rotation that creates a new secret without disabling the old one leaves the debt in place. Rotation should be performed from a known-clean host after persistence is removed [11].

Short-Term Mitigations

Over the next quarter, organizations should build an inventory of their revocation debt and begin paying it down. A practical starting point is to enumerate non-human credentials by type, owner, age, and last use, then retire or rotate any long-lived credential with no identified owner. Beyond that baseline, the priorities in the following table reflect where recent campaigns have concentrated their collection.

Credential area	Priority actions
CI/CD runners	Replace stored cloud keys with short-lived OIDC federation, scope federation trust to specific repositories and branches, and restrict runner egress.
Developer tokens	Enforce expiry and fine-grained scopes on personal access tokens and npm or PyPI publish tokens, and require trusted publishing where registries support it.
SaaS integrations	Inventory third-party OAuth grants, remove unused ones, and monitor integration tokens for unusual query volume, as Mandiant recommends [1].
Infostealer exposure	Monitor infostealer log sources for corporate domains and force resets for any credential that appears, regardless of age [7][13].
AI tooling	Remove secrets from MCP and agent configuration files in favor of runtime injection from a secrets manager [2].

Detection should also be adjusted to treat the use of a valid credential from an unexpected context, rather than the theft itself, as the primary signal, since campaigns such as wfx-loot are designed to leave little trace in the victim environment [3].

Strategic Considerations

The durable fix for revocation debt is to stop issuing credentials that can accumulate. Ephemeral, workload-bound credentials such as federated OIDC tokens and short-lived STS sessions expire on their own, which converts revocation from a manual task into a property of the system. Where long-lived secrets remain unavoidable, organizations should treat revocation capability as a design requirement: each credential needs a documented owner, a tested replacement path, and a revocation procedure that can be executed within hours, consistent with the under-24-hour remediation target in CSA's non-human

identity guidance [15]. Security leaders should consider reporting revocation debt as a metric, such as the count and age of exposed-but-valid credentials and the median time from exposure to invalidation, alongside conventional vulnerability metrics.

Vendor and platform risk management also needs to account for credentials held on an organization's behalf. The Salesloft Drift incident showed that customer secrets stored in a SaaS platform's records became a second-stage harvest [8][21], and Mandiant's 2026 findings describe vendor compromise as a routine path to downstream environments [1]. Contracts and assessments should ask how vendors store customer tokens, how quickly they can revoke them at scale, and whether they notify customers of token exposure. For AI agents, organizations should apply the same discipline from the outset, issuing each agent its own scoped, short-lived identity rather than sharing human or service credentials.

CSA Resource Alignment

CSA's research note *The wfx Campaign: Hijacked GitHub Accounts Harvest CI Secrets* provides the detailed indicators, cloud persistence techniques, and remediation steps for the campaign discussed throughout this note [5]. That analysis reached the same core conclusion presented here, namely that rotating repository secrets without revoking the developer's own tokens and sessions leaves an incident incomplete, and readers responding to wfx-loot should use it as the operational companion to this broader assessment. CSA's companion note *GhostAction Returns: Maintainer Takeover Steals Workflow Secrets* covers the earlier campaign whose victims overlapped with wfx-loot [24].

Several other CSA research notes examine incidents analyzed here in greater depth. *ShinyHunters' OAuth Pivot: A Year of SaaS Supply-Chain Breaches* traces the OAuth token theft pattern behind the Salesloft Drift intrusions and informs the vendor token-custody recommendations above [21]. *Miasma: Red Hat npm Supply Chain Worm* covers the Red Hat package compromise and its infostealer link [22], and *Private-CISA: GovCloud Leak and the Hollowing of U.S. Cyber Defense* offers a case study of static-credential exposure and contractor identity governance [23].

CSA's *Defining Non-Human Identity* guidance establishes the lifecycle and governance model that revocation debt violates [15]. It sets out lifecycle management for service accounts, workloads, and AI agents, recommends ephemeral credentials and workload identity federation over static secrets, and proposes illustrative operational targets for mature programs, such as complete ownership assignment, high rotation compliance, and a mean time to remediate under 24 hours. Measuring an organization against those targets is a direct way to quantify its revocation backlog.

CSA's 2024 survey report *The State of Non-Human Identity Security* supplies the organizational evidence for why the debt accumulates, including the finding that only 20% of organizations had formal API key offboarding processes and nearly 40% needed weeks or more to complete it [16]. Those process gaps are a plausible contributor to the long credential validity periods documented by GitGuardian and Mandiant.

At the control level, the CSA AI Controls Matrix (AICM) v1.1 addresses these risks through its Identity and Access Management domain, which covers credential lifecycle, rotation, and revocation for human and non-human identities, and through its Threat and Vulnerability Management domain [17]. Organizations extending identity governance to AI agents and MCP-connected tooling can use AICM to map the recommendations above to auditable controls that verify agent credentials are scoped, short-lived, and revocable.

References

[1] Kutscher, Jurgen. "M-Trends 2026: Data, Insights, and Strategies From the Frontlines." Google Cloud / Mandiant, March 23, 2026.

[2] Nabiullina, Anna and Winqwist, Carole. "The State of Secrets Sprawl 2026: AI-Service Leaks Surge 81% and 29M Secrets Hit Public GitHub." GitGuardian, March 17, 2026.

[3] Zorz, Mirko. "Check your GitHub repos: A secret stealer hit tens of thousands of them." Help Net Security, November 12, 2026.

[4] Kovacs, Eduard. "Attacker Targets 60,000 GitHub Repositories in Credential Theft Campaign." SecurityWeek, November 12, 2026.

[5] Cloud Security Alliance. "The wfx Campaign: Hijacked GitHub Accounts Harvest CI Secrets." CSA Labs, November 12, 2026.

[6] CISA. "UPDATE: Microsoft Releases Guidance on Exploitation of SharePoint Vulnerabilities." Cybersecurity and Infrastructure Security Agency, July 20, 2025 (last revised August 6, 2025).

[7] Mandiant. "UNC5537 Targets Snowflake Customer Instances for Data Theft and Extortion." Google Cloud Threat Intelligence, June 10, 2024.

[8] Google Threat Intelligence Group. "Widespread Data Theft Targets Salesforce Instances via Salesloft Drift." Google Cloud Threat Intelligence, August 26, 2025.

[9] CISA. "Widespread Supply Chain Compromise Impacting npm Ecosystem." Cybersecurity and Infrastructure Security Agency, September 23, 2025.

[10] Tenable Research. "Mini Shai-Hulud Supply Chain Attack CVE-2026-45321 FAQ." Tenable, May 21, 2026.

[11] Microsoft Threat Intelligence. "ChainDrop supply chain compromise: Anatomy of a self-propagating worm." Microsoft Security Blog, August 4, 2026.

[12] StepSecurity. "ChainDrop npm Worm: Bun-loaded CI/CD credential harvester with Ethereum dead-drop C2." StepSecurity, August 2026.

[13] Lakshmanan, Ravie. "Miasma Supply Chain Attack Compromises Red Hat npm Packages with Credential-Stealing Worm." The Hacker News, June 1, 2026.

[14] Verizon. "2025 Data Breach Investigations Report." Verizon Business, April 2025.

[15] Cloud Security Alliance. "Defining Non-Human Identity." Cloud Security Alliance, July 22, 2026.

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

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

[18] Aikido Security. "Compromised GitHub accounts push CI secret stealer to tens of thousands of repositories." Aikido Security, November 2026.

[19] Kaspersky GReAT. "Shai Hulud 2.0, now with a wiper flavor." Securelist, November 2025.

[20] The Hacker News. "GhostAction Compromises About 1,900 GitHub Accounts, Targets Over 30,000 Repositories." The Hacker News, October 2026.

[21] Cloud Security Alliance. "ShinyHunters' OAuth Pivot: A Year of SaaS Supply-Chain Breaches." CSA Labs, July 16, 2026.

[22] Cloud Security Alliance. "Miasma: Red Hat npm Supply Chain Worm." CSA Labs, June 3, 2026.

[23] Cloud Security Alliance. "Private-CISA: GovCloud Leak and the Hollowing of U.S. Cyber Defense." CSA Labs, 2026.

[24] Cloud Security Alliance. "GhostAction Returns: Maintainer Takeover Steals Workflow Secrets." CSA Labs, October 2026.