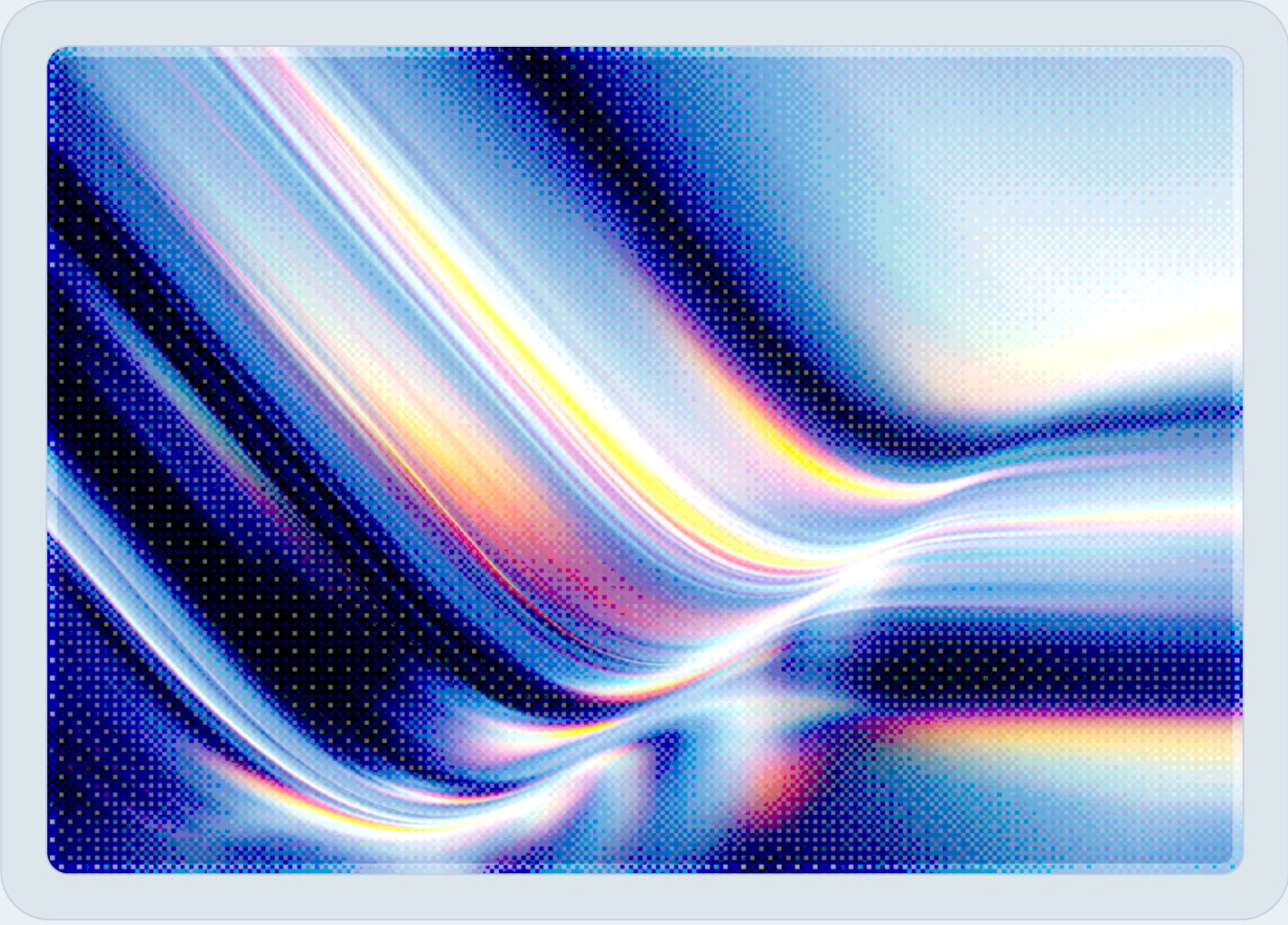
CSAI Foundation | Cloud Security Alliance

Shai-Hulud's Credential Scan Now Targets AI Tool Configs

2026-09-05

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

The latest variant in the Shai-Hulud worm lineage, propagated through a compromised `keyv@6.0.0` npm package published on August 4, 2026, now scans 469 distinct credential locations on an infected host — more than double the 189 locations checked by the family's earlier variants [1][2]. GitGuardian, the secrets-detection firm that reverse-engineered the payload, reports that the expanded scan list includes configuration paths for a wide range of AI development tools, among them Cursor, OpenClaw, OpenAI Codex, OpenCode, Gemini, and Hermes, alongside CI/CD systems such as ArgoCD, Jenkins, and CircleCI and cloud providers including Hetzner, Alibaba Cloud, and Tencent Cloud [1][2]. The finding matters less for the specific tool list, which will inevitably grow further, than for what it confirms about attacker methodology: credential-harvesting malware is now built to enumerate AI coding assistants and agent configurations as a matter of course — a design choice that only makes sense if attackers expect developers to have accumulated exploitable secrets there. This research note examines what changed in the August campaign, situates it against the pattern CSA has tracked across the Shai-Hulud family since September 2025, and updates guidance for securing AI tool configuration as a first-class credential surface rather than an afterthought.

Background

Shai-Hulud first surfaced in September 2025 as a self-propagating npm worm that harvested build-environment secrets and used stolen publishing tokens to infect the next wave of packages, compromising dozens of libraries — including several maintained by CrowdStrike — within its first days of activity [3]. A more destructive second wave, publicized by its operators as "Shai-Hulud V2," emerged in November 2025 and compromised more than 700 packages, exposing roughly 14,000 secrets across 487 organizations within hours of detection [4]. The family has continued to mutate through a series of "Mini Shai-Hulud" and toolkit-derived campaigns since, including the Miasma worm that used the original group's open-sourced framework to compromise Red Hat's npm packages in June 2026 [5]. CSA has tracked this lineage extensively, including the arrest of two alleged operators behind the related TeamPCP campaign in August 2026 and the broader ecosystem risk that persists even after individual actors are taken into custody [6].

The August campaign analyzed here follows the same self-propagating design but marks a distinct escalation in reconnaissance scope. GitGuardian's timeline shows the malicious `keyv@6.0.0` package published at 9:35 a.m. UTC on August 4, 2026, with the last confirmed malicious release following roughly 27 hours later; in that window the infection propagated through more than 800 downstream packages and thousands of versions, reaching dependents such as `@cacheable/memory`, `cacheable-request`, and `flat-cache` and affecting organizations including OneReach, Ornikar, Qlik, and Picsart [1][2]. Rather than introducing a new propagation technique, the payload's principal change was reconnaissance breadth: its credential-location list grew from 189 to 469 entries, with the largest increases on Linux (89 to 290 locations) and Windows (12 to 50 locations), and the malware now enumerates secrets across every user account on a system rather than confining its search to the current user's home directory [2].

Security Analysis

CSA assesses that the addition of AI tool configuration paths to Shai-Hulud's scan list reflects the malware catching up to where developers have already put their credentials, rather than a novel attack surface the worm invented. AI coding assistants and agent frameworks tend to store API keys, MCP server definitions, and provider tokens in predictable configuration directories — the same category of file that a worm targeting `.env` files, shell history, and cloud CLI configs was always designed to sweep. The Hacker News, reporting independently of the August campaign, observed that attackers are "increasingly finding access keys in the configuration used by AI development tools" [1]. CSA reads this as a symptom of rapid, largely unmanaged AI-tool adoption across development organizations rather than any AI-specific vulnerability in the malware itself. What the August variant demonstrates is that this pattern has now been operationalized at scale: AI tool configuration enumeration is no longer confined to bespoke campaigns targeting AI infrastructure directly, but has become a default checklist item in a widely propagated credential-harvesting worm.

This escalation is consistent with — and reinforces — a pattern CSA has flagged in earlier Shai-Hulud-family analysis (see CSA Resource Alignment, below) [7][8]. A February 2026 campaign tracked by Socket Research, described as "Shai-Hulud-style" but not confirmed to share the same codebase or operators, went further still by actively tampering with AI toolchains rather than merely reading their configuration: it planted hidden Model Context Protocol server entries in Claude Code, Cursor, and other assistant configurations containing prompt-injection instructions designed to cause the AI tool itself to silently exfiltrate SSH keys, cloud credentials, and environment secrets on the attacker's behalf [9]. That campaign also harvested API keys for nine separate LLM providers, indicating that attackers view AI tool credentials as valuable independent of any other objective, not merely as a stepping-stone to

conventional cloud or CI/CD access [9]. Read together, CSA assesses that the passive AI-config scanning in the August keyv campaign and the active AI-toolchain manipulation in February illustrate two points on a plausible attacker-maturation curve, even though the two campaigns are not confirmed to be directly linked: enumerate first, manipulate later where the return justifies the effort.

The practical significance for defenders is that AI tool configuration has joined `.env` files and shell history as a location security teams must assume is routinely scanned by commodity malware, not a niche corner of the developer environment. Because these configuration files frequently hold long-lived API keys for the AI provider itself alongside references to other systems the agent has been granted access to — repositories, cloud accounts, internal APIs — their compromise can cascade in the same way a stolen npm publishing token does, converting a single infected developer workstation into a foothold across every system the AI tool was configured to reach. GitGuardian's assessment that "attackers have stopped trying to break trust relationships and started using the credentials that already make those relationships work" captures this shift [1][2]. CSA assesses that this applies with particular force to AI tooling, where trust relationships are often established quickly, during rapid tool adoption, and audited rarely.

Recommendations

Immediate Actions

Organizations should inventory every AI coding assistant, agent framework, and MCP-connected tool in active use across development teams and locate the credential and configuration files each one maintains, using the tool list GitGuardian identified — Cursor, OpenClaw, OpenAI Codex, OpenCode, Gemini, and Hermes — as a starting point rather than an exhaustive list [1][2]. Any organization that consumed `keyv`, `@cacheable/memory`, `cacheable-request`, `flat-cache`, or their dependents between August 4 and August 6, 2026 should treat credentials present in affected build and developer environments during that window as compromised and rotate them without waiting for confirmation of specific exposure.

Short-Term Mitigations

Security teams should extend existing secrets-scanning programs to explicitly cover AI tool configuration directories rather than assuming general `.env` and shell-history coverage is sufficient, since these tools frequently use nonstandard file locations and formats that generic scanners may miss. Development organizations should also apply the same install-script allowlisting and CI/CD credential-

scoping discipline that CSA has previously recommended for the broader Shai-Hulud family to AI tool installation and update processes specifically, since these tools are installed and updated through the same package-manager mechanisms the worm exploits [6].

Strategic Considerations

The steady expansion of Shai-Hulud's credential scan list — from 189 to 469 locations in under a year, with each increment adding newly popular developer tools — indicates that commodity credential-harvesting malware is likely to continue tracking developer tool adoption closely, and that AI coding assistants and agent frameworks should be governed under the same least-privilege and monitoring standards already applied to cloud IAM and CI/CD systems, rather than treated as a separate, lower-scrutiny category. Organizations building or expanding AI-assisted development programs should require that any tool capable of holding provider API keys or repository access tokens be included in existing secrets-management and endpoint-monitoring programs from the outset, rather than added retroactively after an incident demonstrates the gap.

CSA Resource Alignment

This finding extends the threat picture in CSA's whitepaper npm Supply Chain Under Siege: TeamPCP, Miasma, and npm v12, which documented the broader TeamPCP and Miasma campaign chronology and the npm v12 structural defenses now emerging in response; the August keyv incident confirms that the install-script and dependency-source hardening described there remains directly relevant, since the credential-scanning payload still depends on unrestricted script execution during package installation to reach the developer environment in the first place [7]. It also builds on two prior CSA threat intelligence reports that anticipated this trajectory. Miasma: Red Hat npm Supply Chain Worm established that Shai-Hulud-derived tooling already reached beyond conventional cloud and CI/CD credentials into developer environments broadly [8], and Miasma and IronWorm: Self-Replicating Worms Targeting AI Credentials, published June 9, 2026, documented npm worms systematically harvesting API keys for Anthropic, OpenAI, Google Gemini, and other AI providers directly from AI coding tool configurations [10]. The August variant's explicit enumeration of AI tool configuration paths is a direct continuation of that trajectory, extending passive credential scanning to a far larger and more diverse set of AI development tools than either prior campaign targeted.

More broadly, the governance gap this campaign exposes — AI development tools accumulating credential access without commensurate scrutiny — maps to the Supply Chain Management, Transparency, and Accountability and Identity and Access Management domains of CSA's AI Controls

Matrix (AICM) v1.1, the current superset of the Cloud Controls Matrix [11]. Organizations extending secrets-management programs to cover AI tool configuration should use AICM's STA and IAM control families as the baseline for scoping credential lifecycle, least-privilege access, and monitoring requirements for these tools alongside traditional cloud and CI/CD infrastructure.

References

[1] The Hacker News. "Shai-Hulud's Reach Just Grew to 469 Credential Locations. Here's What That Means." The Hacker News, September 2026.

[2] GitGuardian. "Mini Shai-Hulud's Latest Wave: 280 New Places." GitGuardian Blog, August 2026.

[3] Unit 42, Palo Alto Networks. "'Shai-Hulud' Worm Compromises npm Ecosystem in Supply Chain Attack." Unit 42, September 2025 (updated November 2025).

[4] Zscaler ThreatLabz. "Shai-Hulud V2 Poses Risk to npm Supply Chain." Zscaler, December 2, 2025.

[5] The Register. "Miasma worms its way onto GitHub as attack kit goes open source." The Register, June 9, 2026.

[6] Brian Krebs. "Two Alleged 'TeamPCP' Hackers Arrested in Australia." KrebsOnSecurity, August 2026.

[7] Cloud Security Alliance AI Safety Initiative. "npm Supply Chain Under Siege: TeamPCP, Miasma, and npm v12." CSA Lab Space, June 2026.

[8] Cloud Security Alliance AI Safety Initiative. "Miasma: Red Hat npm Supply Chain Worm." CSA Lab Space, June 2026.

[9] Socket. "SANDWORM_MODE: Shai-Hulud-Style npm Worm Hijacks CI Workflows and Poisons AI Toolchains." Socket, February 20, 2026.

[10] Cloud Security Alliance AI Safety Initiative. "Miasma and IronWorm: Self-Replicating Worms Targeting AI Credentials." CSA Lab Space, June 9, 2026.

[11] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2026.

