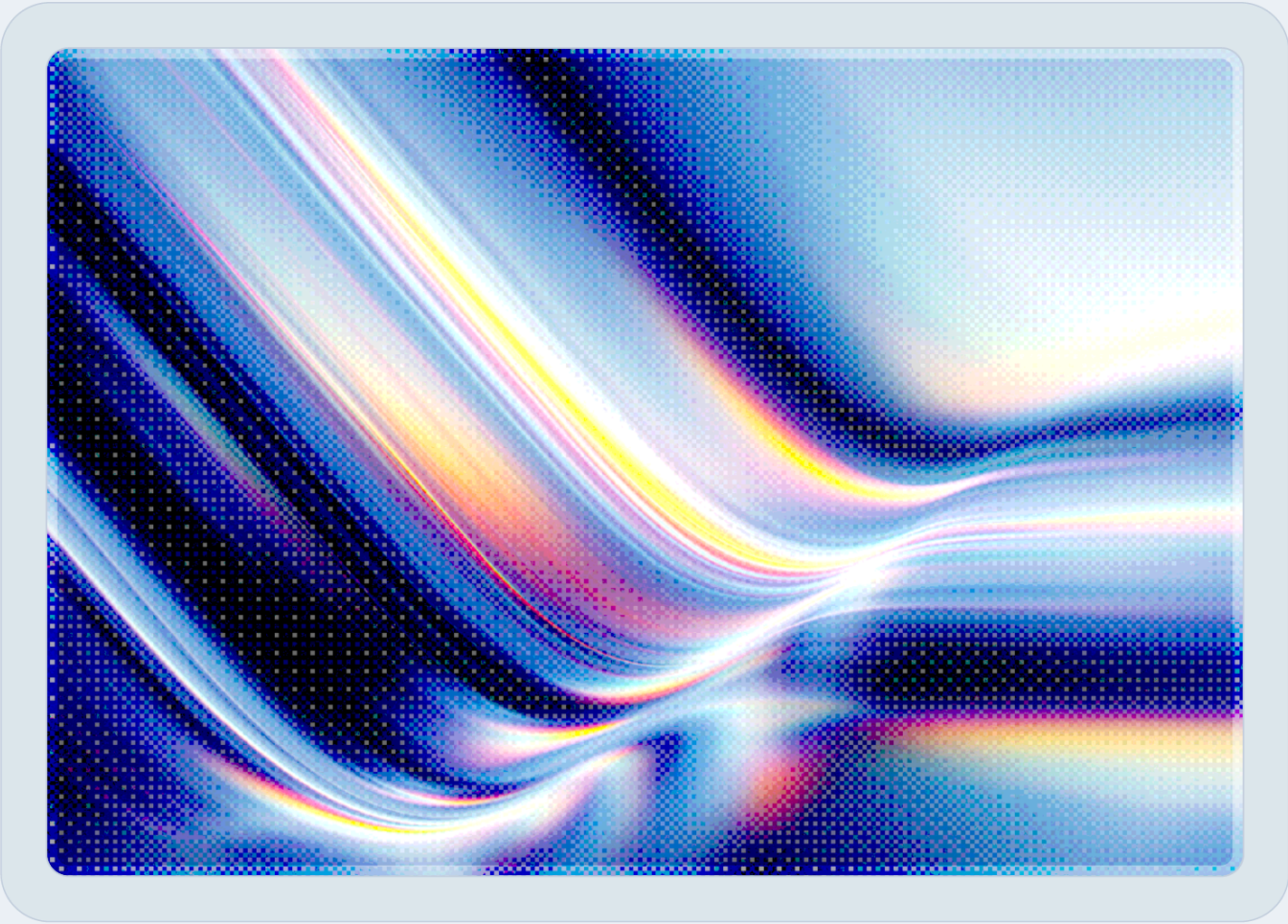
CSAI Foundation | Cloud Security Alliance

The Shai-Hulud Playbook: TeamPCP and Supply-Chain Ecosystem Risk

2026-09-01

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

The arrest of two Western Australian men on August 26, 2026, gives defenders a rare law-enforcement-confirmed account of the actors behind Shai-Hulud and the broader TeamPCP campaign that convulsed the npm ecosystem for the better part of a year [1][2]. Ruben Ian Thomson, 21, and Louis Michael Gaebler, 23, were charged with a combined fourteen cybercrime offenses after a joint investigation by the Australian Federal Police, the FBI, and Western Australia Police concluded that malware distributed under the TeamPCP banner had compromised more than 1,000 organizations, exposed roughly 500,000 credentials, and exfiltrated at least 300 gigabytes of data [2][3]. The case record suggests this was not a nation-state operation executed with exquisite tradecraft, but a small, loosely affiliated group of young, low-resourced actors who used commodity tooling and large language models to narrow the skills gap that once separated hobbyist hackers from capable adversaries — captured in one investigator's description of the group as "high capability, low operational discipline" [1].

This gap between low sophistication and high impact is, in our view, the central lesson for the open-source ecosystem. TeamPCP did not need novel exploits to compromise GitHub, LiteLLM, Red Hat, Checkmarx, and hundreds of other packages and organizations; it needed phished maintainer credentials, permissive CI/CD configurations, and a self-propagating worm design that let each compromise fund the next. The group's decision to open-source its Shai-Hulud tooling in mid-2026 then triggered a second wave of copycat campaigns — Miasma and others — carried out by actors with even less skill than the original crew, confirming that the arrests close a chapter on specific defendants without closing the underlying vulnerability [4]. This research note examines what the TeamPCP case reveals about ecosystem-level supply chain risk, assesses the defensive measures that have emerged in response, and offers guidance for organizations that depend — almost universally, whether they realize it or not — on the same open-source infrastructure TeamPCP exploited.

Background

Shai-Hulud first surfaced in September 2025, when researchers identified a self-propagating worm embedded in the npm package rxnt-authentication and rapidly spreading across the JavaScript ecosystem by harvesting secrets from CI/CD pipelines and cloud metadata endpoints [5][12]. The malware's five-stage attack chain — phish a maintainer, harvest build-environment credentials, exfiltrate them, then use the stolen npm token to infect the next wave of packages — allowed it to compromise

dozens of packages, including several maintained by CrowdStrike, within its first days of operation [5]. A more aggressive second wave, which its operators called "The Second Coming," emerged in late November 2025 and within hours had compromised more than 700 npm packages, generated over 27,000 malicious GitHub repositories, and exposed roughly 14,000 secrets across 487 organizations, adding preinstall execution and destructive fallback behavior when credential theft failed [6].

Investigators have since attributed the Shai-Hulud campaigns, along with a chain of related operations, to TeamPCP, a cybercrime syndicate that researcher Austin Larsen has described not as a hierarchical organization but as "a peer community of individually-skilled actors" drawn from several overlapping criminal circles and coordinated through a Matrix chat server called Cybercats [1]. The group's most consequential operations extended well beyond the original npm worm. In March 2026, TeamPCP compromised the LiteLLM AI gateway — a proxy layer connecting to more than 100 language model providers — harvesting credentials from over 2,500 organizations and an estimated 434,000 CI/CD pipelines in a single incident [1][13]. Between that campaign and its next major operation, CSA researchers tracked a further two-wave assault on the same ecosystem: a first wave, later named "Mini Shai-Hulud," compromised 172 packages across 404 malicious versions between April 29 and May 12, 2026, and was followed six days later by "Megalodon," which pushed 5,718 malicious commits across 5,561 repositories in under six hours — evidence that the same self-propagating design could be redeployed against new targets faster than the original campaign had been contained [14]. Two months after the LiteLLM compromise, the group had also breached at least 3,800 GitHub repositories after developers installed a trojanized code extension, and it targeted organizations ranging from the European Commission to the AI recruiting platform Mercor and OpenAI itself [1][3]. To sustain recruitment, TeamPCP reportedly ran a contest offering roughly $1,000 in Monero to whichever affiliate could execute the largest supply chain compromise, framed publicly as a "participation trophy" while quietly rewarding real results [1].

The case against Thomson and Gaebler came together through conventional, if painstaking, digital forensics rather than any single smoking-gun disclosure. Investigators connected years of forum aliases — including BulkDMT, EllisD25, and Persy_PCP — tied to a Gmail address registered on cybercrime forums since 2022, cross-referenced against IP addresses linked to family network infrastructure in Perth [1]. Thomson, in what investigators characterized as a striking lapse in operational security, incorporated a legitimate Australian company called "OPSEC Express" using one of his own cybercrime handles [1]. He later told Krebs via Signal that he had earned approximately $20,000 from TeamPCP activity and, in a series of unguarded messages describing his struggles with methamphetamine and homelessness, remarked that "blackhatting is fun... without qualifications, no employer will even take the time to hear you out" [1]. Thomson claimed to have led the group until stepping back in March 2026 [3]. Both men were charged on August 26, 2026, and Thomson was denied bail ahead of a September 18 court appearance [1][2].

Security Analysis

The TeamPCP case is most useful to defenders not as a story about a specific group of hackers but as a demonstration of how little sophistication is now required to inflict ecosystem-scale damage, and how that dynamic outlives any single arrest. Three structural features of the open-source supply chain made TeamPCP's campaigns possible, and none of them are addressed by removing two individuals from circulation.

The first is the self-propagating design pattern itself. Shai-Hulud's core innovation was not credential theft — phishing and token exposure are old problems — but the automation of what to do with stolen credentials. Rather than exfiltrating secrets for later use, the malware immediately used harvested npm and GitHub tokens to publish new malicious package versions under the compromised maintainer's identity, which in turn exposed the credentials of every downstream developer who installed the update. This converts a single successful phish into a chain reaction whose growth rate is bounded only by how quickly defenders can detect and interrupt it. Traditional artifact-based defenses — cryptographic provenance attestations, Sigstore signing, SLSA build-level certification — are generally understood to verify that a package was built by an authorized pipeline, but they do not typically address whether the maintainer's account or build credentials were compromised beforehand. A malicious version published through a legitimately compromised, fully authenticated pipeline carries a valid signature, which is precisely what made Shai-Hulud-derived worms difficult to distinguish from routine releases in their first hours of circulation.

The second structural feature is credential aggregation, particularly in AI development infrastructure. The LiteLLM compromise stands out because AI gateways are, by design, concentration points: a single proxy service brokering access to dozens of model providers holds API keys with value and blast radius far exceeding a typical package dependency. TeamPCP's targeting of LiteLLM, and its subsequent attention to AI coding assistant configuration files (settings for tools such as Claude Code and other MCP-connected agents) as persistence surfaces, illustrates a risk this note argues is broader: AI development tooling may function as one of the richest credential stores in the enterprise, often without commensurate access controls or monitoring. An organization that would never grant a build script unrestricted access to its cloud provider's IAM console may not realize that its AI gateway configuration effectively does exactly that.

The third feature is the commoditization effect of open-sourcing offensive tooling. When TeamPCP published its worm framework publicly in mid-2026, it did not merely disclose a technique — it distributed working attack tooling that required none of the original group's skill to deploy. The Miasma campaign against Red Hat's npm packages in June 2026, built on the released codebase, was executed with a level of technical sophistication well below what TeamPCP itself demonstrated, yet it achieved comparable

impact: CSA's own accounting identified 32 compromised @redhat-cloud-services packages exposing roughly 80,000 weekly downloads [10], while The Register separately reported upwards of 100 affected Red Hat and Microsoft projects [4], and the credential-harvesting reach extended across GitHub, AWS, Azure, GCP, and HashiCorp Vault environments [4][7]. This is the mechanism by which the arrests of Thomson and Gaebler, however significant as a law enforcement milestone, do not resolve the underlying risk: the tooling that made TeamPCP dangerous is now available to anyone willing to run it, and subsequent campaigns have already demonstrated that low-skill operators can achieve outcomes once reserved for the originators.

Taken together, these dynamics point to a threat model in which the open-source package ecosystem functions less like a supply chain with discrete, auditable links and more like an epidemiological system, where a single compromised node can seed exponential spread through automated, credential-driven propagation. In our assessment, defenders who treat supply chain risk primarily as a matter of vetting individual dependencies at intake are addressing a static threat model against a dynamic, self-amplifying one.

Recommendations

Immediate Actions

Organizations that consumed npm packages during the active Shai-Hulud, Mini Shai-Hulud, or Miasma propagation windows (September 2025 through June 2026) should treat any credentials present in affected build environments during those periods as potentially compromised and rotate them without waiting for confirmation of specific exposure [14]. Security teams should also audit CI/CD workflow permissions for overly broad OIDC token scopes and `pull_request_target` misconfigurations — both common CI/CD attack surfaces that could similarly escalate a single compromised credential into a publishing capability. Any organization operating an AI gateway or multi-provider LLM proxy should conduct a targeted credential audit of that specific system, given its demonstrated status as a high-value aggregation point.

Short-Term Mitigations

Development organizations should adopt install-script allowlisting rather than blanket execution of npm lifecycle hooks, following the model that npm v12 has now made the default; a package that never needed a preinstall or postinstall script in the past should not silently gain one in a routine update. Enabling GitHub's default three-day Dependabot cooldown for version updates — while leaving security

updates on their existing expedited path — closes the specific window that npm-based worms have exploited, since GitHub's own review of 21 major incidents since 2018 found that malicious releases were consistently caught within hours of publication, well inside a multi-day delay [8]. Teams should also extend behavioral monitoring beyond hash-based indicators of compromise; because malware in this family could plausibly evolve toward per-build payload encryption, static signatures should not be relied on as the sole detection method, and detection is more durable when it targets behavior — unexpected outbound connections during package installation, anomalous npm publish activity, or unauthorized modification of AI tool configuration files — rather than known-bad file hashes.

Strategic Considerations

The recurring lesson of the TeamPCP case is that provenance and signing controls, while valuable, answer a narrower question than they are often treated as answering: they confirm that a build pipeline produced an artifact, not that the pipeline or the identity authorizing it was not compromised. Organizations should treat cryptographic attestation as one layer within a defense-in-depth posture rather than as a sufficient control on its own, and should invest correspondingly in identity-centric protections — least-privilege credential scoping, ephemeral build runners, and Zero Trust assumptions about developer and CI/CD environments — that remain effective even when an upstream signing identity has been compromised. Because the open-sourcing of attack tooling has demonstrably lowered the skill floor required to execute ecosystem-scale campaigns, organizations should also plan for a threat landscape in which copycat activity persists indefinitely after any single group's disruption, and should build supply chain risk programs around that assumption rather than around the expectation that arrests or takedowns meaningfully reduce residual risk.

CSA Resource Alignment

This case connects directly to two recent CSA analyses of the same threat cluster, alongside CSA's earlier research documenting the campaign's origin and escalation. CSA's white paper npm Supply Chain Under Siege: TeamPCP, Miasma, and npm v12 provides the most complete technical accounting of the campaign chronology addressed in this note, tracing the TeamPCP and Miasma operations from March through June 2026 and evaluating how npm v12's install-script and dependency-source restrictions address — and fail to fully address — the propagation mechanisms described above [9]. Organizations implementing this note's short-term mitigations should treat that paper's npm v12 preparation workflow, including systematic use of `npm approve-scripts` to build an auditable allowlist, as a companion

operational guide. CSA's earlier research notes on the original Shai-Hulud worm and on TeamPCP's March 2026 campaign against developer security tooling provide additional technical depth on the events summarized in the Background section above [12][13].

CSA's threat intelligence report Miasma: Red Hat npm Supply Chain Worm offers the clearest illustration of this note's commoditization argument: it documents how a worm built on TeamPCP's publicly released codebase reproduced the original campaign's credential-harvesting reach — spanning GitHub, AWS, Azure, GCP, and HashiCorp Vault — despite being operated by less capable actors, reinforcing that the underlying risk persists independent of any single group's legal exposure [10].

More broadly, the control gaps this campaign exposed map to the Supply Chain Management, Transparency, and Accountability domain of CSA's AI Controls Matrix (AICM) v1.1, the current superset of the Cloud Controls Matrix, which addresses credential scoping, dependency governance, and build-pipeline integrity requirements directly relevant to the mitigations recommended above [11]. Organizations formalizing a supply chain risk program in response to this incident should use AICM's STA and IAM control families as the baseline against which to assess CI/CD credential handling, install-script governance, and AI toolchain access management.

References

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

[2] Help Net Security. "Two alleged TeamPCP hackers arrested over global supply chain attacks." Help Net Security, August 27, 2026.

[3] TechCrunch. "Australian police arrest two over TeamPCP hacks targeting Mercor, OpenAI, and others." TechCrunch, August 27, 2026.

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

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

[6] Zscaler ThreatLabz. "Shai-Hulud V2: npm Supply Chain Attack Analysis." Zscaler, November 2025.

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

[8] GitHub. "The case for a cooldown: Why Dependabot now waits before issuing version updates." The GitHub Blog, July 2026.

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

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

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

[12] Cloud Security Alliance AI Safety Initiative. "Shai-Hulud: npm Worm Targeting AI Developer Toolchains." CSA Lab Space, April 2026.

[13] Cloud Security Alliance AI Safety Initiative. "TeamPCP: Cascading Supply Chain Assault via Developer Security Tooling." CSA Lab Space, April 2026.

[14] Cloud Security Alliance AI Safety Initiative. "Shai-Hulud/Megalodon: A Two-Wave AI Developer Supply Chain Attack." CSA Lab Space, May 2026.

