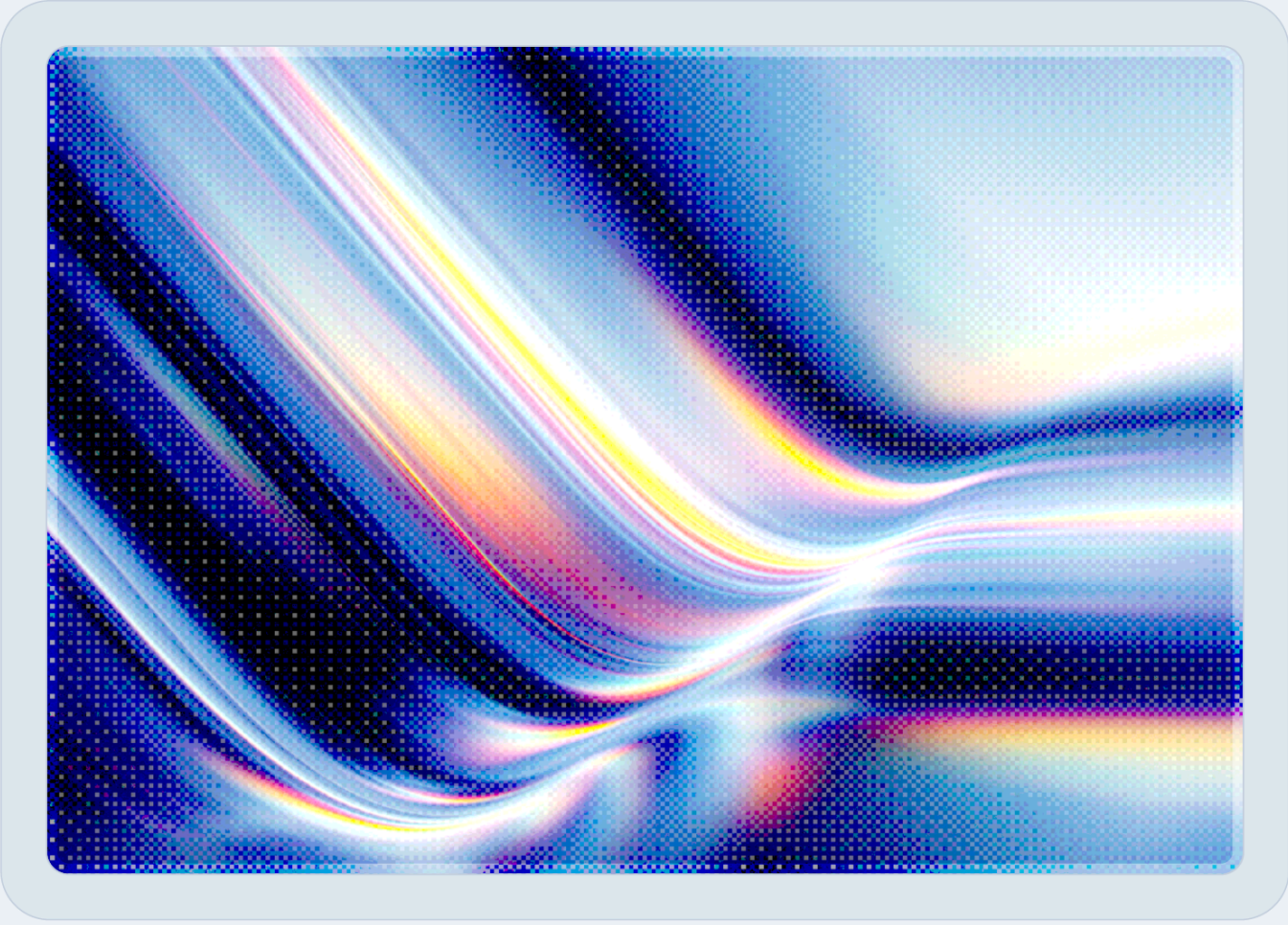
CSAI Foundation | Cloud Security Alliance

GlassWorm Takedown: Developer Supply Chain Attack Campaigns

2026-05-28

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- On May 26, 2026, CrowdStrike, Google, and the Shadowserver Foundation simultaneously dismantled all four command-and-control channels of the GlassWorm botnet — a developer-targeting infrastructure whose operators, assessed as likely Russian-based from CIS locale checks and code artifacts, have not been formally attributed to any named threat group [1][2][5].
- The broader developer-targeting campaign, active since at least early 2025, encompassed both the GlassWorm botnet's trojanized VS Code and OpenVSX extensions and the TeamPCP threat group's concurrent Shai-Hulud worm — with malicious npm and PyPI packages, poisoned GitHub Actions workflows, and stolen credentials cascading into unauthorized access to source code repositories, cloud platforms, and CI/CD pipelines [3][4][6].
- The GlassWorm botnet deployed a deliberately redundant four-channel C2 architecture — combining Solana blockchain transactions, BitTorrent DHT, Google Calendar, and traditional VPS servers — designed so that any single channel could be disrupted without losing access to infected hosts [1][5].
- The scale of downstream harm reflects the multiplier effect of developer compromise: stolen credentials were used to poison hundreds of additional packages, ultimately enabling the breach of over 300 Cisco source code repositories [7], unauthorized access to internal code repositories at OpenAI [18], and the compromise of Mistral AI's codebase management infrastructure [19].
- Organizations should immediately audit network logs for connections to the CrowdStrike sinkhole address (164.92.88.210), review installed developer tooling for known-malicious extensions and packages, and rotate any credentials stored in developer environments [1][9].

Background

Software supply chain attacks have evolved along a clear logic: rather than attacking hardened production systems directly, adversaries compromise the developers who build them. A single infected developer workstation can provide access to source code repositories, cloud credentials, CI/CD pipeline tokens, package registry publishing rights, and infrastructure-as-code files — granting potential lateral reach that may extend to downstream consumers of that developer's software. The GlassWorm takedown, executed on May 26, 2026, brought this threat into sharp public focus by dismantling infrastructure that had been quietly harvesting developer access since at least early 2025 [1][2].

The operators of the GlassWorm botnet have not been formally identified in CrowdStrike's public disclosure; the malware's behavior of silently exiting on machines located in CIS countries, combined with Russian-language code comments in analyzed samples, led CrowdStrike to assess its operators as likely Russian-based [1][2][5]. A concurrent supply chain attack campaign, separately documented, has been publicly attributed to the threat group tracked as TeamPCP (also observed under the aliases PCPcat, DeadCatx3, ShellForce, and CipherForce) [3][10]. TeamPCP's primary offensive tool was the Shai-Hulud worm — a self-propagating capability that used stolen developer credentials to automatically publish malicious versions of software already trusted across npm and PyPI, amplifying its reach without requiring manual intervention for each individual compromise [3][6][12]. Both campaigns targeted the developer ecosystem from different angles: GlassWorm built persistent footholds on individual machines through its RAT capability, while TeamPCP's Shai-Hulud automated downstream supply chain compromise using harvested credentials.

Security Analysis

How the Campaign Unfolded

Developer-targeting operations across both campaigns followed a discernible escalation pattern. The GlassWorm botnet's operators distributed GlasswormRAT through trojanized IDE extensions targeting not only VS Code but also Cursor, Positron, Windsurf, VSCodium, and other VS Code-compatible editors, publishing through both the Microsoft VS Code Marketplace and OpenVSX, a vendor-neutral open-source alternative hosted by the Eclipse Foundation [1][4]. At one documented point, 72 malicious extensions were simultaneously present on OpenVSX, and the combined campaign ultimately reached

over 400 code repositories across GitHub, npm, the VS Code ecosystem, and OpenVSX [4][11]. These extensions installed GlasswormRAT — a cross-platform Node.js remote access tool capable of information theft, credential harvesting, and command execution on Windows, macOS, and Linux [1][2].

From compromised developer workstations, attackers harvested a broad credential portfolio: GitHub personal access tokens, npm publishing tokens, cloud provider credentials (AWS, GCP, Azure), SSH keys, Kubernetes secrets, database credentials, and cryptocurrency wallet keystores [3][9]. These credentials then powered TeamPCP's Shai-Hulud worm. Shai-Hulud used harvested package registry tokens to automatically publish malicious versions of software already trusted by developers — compromised packages arrived silently through routine dependency updates, executing payloads via postinstall hooks and setup scripts [6][15].

Several high-profile compromises document the cascade effect. The threat actors compromised Trivy, the widely deployed open-source vulnerability scanner developed by Aqua Security, injecting malicious code into its binary distribution, GitHub Actions workflows, and container images [14]. They uploaded malicious versions of LiteLLM, a popular AI routing library, to PyPI — one version used an unconventional persistence technique, hiding a malicious `.pth` file that caused the payload to execute automatically every time any Python process initialized on the host [3][15]. In May 2026, a poisoned Nx Console VS Code extension (version 18.95.0) enabled the exfiltration of approximately 3,800 GitHub internal repositories within an eleven-minute window [16][20]. The May 18 "Megalodon" campaign injected malicious CI/CD workflow files into more than 5,500 public GitHub repositories within a six-hour automated window [17]. Cisco source code was affected when credentials stolen through earlier supply chain infections were used to clone over 300 internal repositories from GitHub [7]. OpenAI disclosed that two employee devices were compromised in the May 11 Mini Shai-Hulud TanStack attack [8], with internal source code repositories accessed and code-signing certificates requiring rotation across iOS, macOS, Windows, and Android [18].

The GlassWorm C2 Architecture

A defining characteristic of the GlassWorm botnet was its architecture for resilience against takedown. The operators built four parallel channels that allowed any single one to be disrupted while the others remained operational — a design that ultimately required the response team to coordinate simultaneous disruption of all four channels before the takedown could succeed [1][5].

The first channel encoded C2 server addresses in the memo fields of Solana blockchain transactions — an immutable, publicly accessible dead drop that cannot be taken offline through conventional domain seizure or hosting provider notification [1][5]. The second channel queried the BitTorrent distributed hash table (DHT) network for configuration data stored against hardcoded public keys, leveraging a global peer-to-peer network with no central point of failure [1][5]. The third channel used Google

Calendar event titles as a dead drop, embedding Base64-encoded C2 paths in the titles of calendar events that the malware could silently retrieve [1][5]. Traditional VPS-hosted C2 servers served as the fourth channel, handling final payload delivery once a victim's machine had resolved the active C2 address through one of the other channels [1][2].

This four-channel design required joint action from CrowdStrike's Counter Adversary Operations team, Google (to disable the Calendar-based channel), and the Shadowserver Foundation [1][2]. The operation executed on May 26, 2026 at 14:00 UTC [1][5]. Following the disruption, compromised machines can no longer receive new instructions or payloads; they now beacon to a CrowdStrike-operated sinkhole at IP address 164.92.88.210, allowing defenders to identify infected hosts through network log analysis [1][9].

Strategic Significance for the Security Community

Some previous developer-targeting supply chain attacks focused on one-time credential theft or a single compromise vector, whereas the combined GlassWorm and TeamPCP model differs in treating developer machines as persistent beachheads rather than one-time credential sources. GlassWormRAT's presence on an infected machine is not merely a credential-collection event; it provides ongoing access to the developer's full workspace, meaning the attackers can observe new credentials as they are created, intercept repository changes in real time, and introduce malicious code directly into development workflows before software reaches any formal review or signing process [1][3].

The campaign also provides concrete evidence that security tooling itself represents a high-value attack surface. TeamPCP's compromise of Trivy and KICS — both widely deployed in CI/CD pipelines to validate security posture — illustrates the adversarial insight that security scanning tools are implicitly trusted by organizations: they run with elevated privileges, they are often exempt from the scanning controls applied to other third-party software, and they communicate with sensitive internal systems as part of their normal function [13][14]. Weaponizing a scanner means that the act of performing a security check can become the vector of compromise. This is a pattern security architects must account for explicitly when designing pipeline trust models.

The release of the Shai-Hulud worm's source code by TeamPCP earlier in May 2026 adds a further dimension [6]. When well-developed attack tooling enters the public domain, the barrier to replication collapses. Security teams should anticipate that imitator campaigns leveraging Shai-Hulud's architecture are likely to emerge in the months following its public release, as has been observed historically when mature offensive tooling enters the public domain.

Recommendations

Immediate Actions

Organizations should treat the GlassWorm disclosure as grounds for urgent triage rather than a background news item. The first priority is to search network logs and endpoint telemetry for connections to 164.92.88.210 — any such connections indicate a host was infected with GlasswormRAT and should be treated as fully compromised [1][9]. All developer credentials (GitHub PATs, npm publish tokens, cloud access keys, SSH keys, Kubernetes service account tokens) on affected hosts should be rotated immediately; the assumption should be that any token present on an infected machine was harvested.

Separately, security teams should audit installed VS Code and compatible IDE extensions against the known-malicious extension list published in CrowdStrike's disclosure [1][2]. The campaign specifically targeted developers using Cursor, Positron, Windsurf, and VSCodium in addition to VS Code itself, so audits must extend beyond Microsoft's official marketplace to any extension sources developers may have used. Package installation logs should be reviewed for any npm or PyPI packages installed from compromised sources during the campaign's active window (early 2025 through May 2026), with particular attention to postinstall hooks and `.pth` files that could indicate Shai-Hulud-style persistence.

Short-Term Mitigations

Organizations should extend supply chain controls to cover the developer tooling layer explicitly. IDE extensions and developer CLI tools warrant the same vetting and approval processes applied to third-party software libraries: a formal allowlist of approved extensions, centrally managed and enforced, reduces the attack surface that campaigns like GlassWorm exploit. CI/CD pipeline credentials should be scoped to the minimum necessary permissions and rotated on a defined schedule, rather than treated as long-lived secrets. Pipeline steps that invoke external scanners or tools — including security tools like Trivy — should run in isolated, network-restricted environments where the tools' own network activity can be monitored.

Credential detection tooling should be deployed across source code repositories, CI/CD logs, and artifact registries to identify and invalidate secrets that may have been committed or exposed during the active campaign window. GitHub's secret scanning feature and equivalent tooling on other platforms can surface historical exposures that developers may not be aware of.

Strategic Considerations

Supply chain security controls have historically focused primarily on the artifact level — scanning packages and containers for known vulnerabilities — rather than at the developer environment level, where the compromises that enable the most damaging attacks actually begin. Addressing this requires treating the developer workstation as a security boundary in its own right: enforcing endpoint security controls on development machines, restricting the ability to install extensions or packages outside approved channels, and monitoring for anomalous developer behavior such as unusual credential usage or unexpected pushes to package registries.

At the ecosystem level, the campaign highlights structural weaknesses in how open-source package registries validate publishers. A stolen npm token provides the same trust as a legitimate one; the registry has no way to distinguish between a developer voluntarily publishing a new version and an attacker using a stolen credential to do the same. Emerging standards for software attestation — including Sigstore and similar tools — can reduce the utility of stolen publishing credentials by requiring cryptographic proof of provenance. The extent to which Shai-Hulud defeated existing provenance attestation mechanisms in some cases warrants close attention from the maintainers of those systems [6].

CSA Resource Alignment

This incident maps directly to several layers of CSA's AI and cloud security frameworks. The MAESTRO threat modeling framework identifies developer toolchain compromise as a critical entry point into AI development pipelines — an insight the GlassWorm campaign concretely exemplifies. TeamPCP's compromise of LiteLLM, an AI model routing library used in production AI systems, represents precisely the kind of Layer 5 (Agent Orchestration) supply chain risk that MAESTRO flags. The actual TeamPCP LiteLLM payloads focused on credential harvesting and persistence [3][15], but the attack surface they represent is directly relevant to production AI systems: malicious code introduced into AI middleware libraries could theoretically alter model routing, inject prompts, or exfiltrate model inputs — underscoring why the supply chain integrity of AI-adjacent dependencies warrants the same scrutiny as AI models themselves.

CSA's AI Controls Matrix (AICM), which encompasses the Supply Chain Management and Transparency (STA) controls from the Cloud Controls Matrix (CCM), provides relevant control guidance for third-party software risk and the assurance standards organizations should apply to externally sourced components. The GlassWorm campaign demonstrates that the developer toolchain — IDE extensions, CI/CD plugins, and open-source dependencies — must be treated as third-party supply chain components subject to

the same AICM controls as production software. The AICM's Application and Interface Security controls similarly cover secure software development lifecycle requirements that, if applied rigorously, would reduce the attack surface available to GlassWorm-style campaigns.

The CSA STAR program offers a mechanism for cloud service providers and development platform vendors to demonstrate supply chain security posture. Organizations selecting CI/CD platforms, package registries, or IDE ecosystems should evaluate STAR-registered providers' disclosures regarding supply chain controls, particularly around credential management and artifact integrity.

CSA's Zero Trust guidance is applicable to the CI/CD layer specifically: the principle that no credential or token should be implicitly trusted solely because it originates from an internal pipeline maps directly to the countermeasures needed against stolen-token abuse. Enforcing short-lived credentials, requiring multi-factor confirmation for high-impact operations (such as publishing a new package version), and instrumenting pipeline behavior for anomaly detection are all consistent with Zero Trust principles and directly relevant to the GlassWorm threat model.

References

[1] CrowdStrike. "Disrupting Glassworm: Inside CrowdStrike's Takedown of a Developer-Targeting Botnet." CrowdStrike Blog, May 2026.

[2] BleepingComputer. "Glassworm botnet disrupted after resilient C2 infrastructure takedown." BleepingComputer, May 2026.

[3] Oligo Security. "Open Source Supply Chain Attacks in 2026: Inside the TeamPCP Campaign." Oligo Security Blog, 2026.

[4] The Hacker News. "GlassWorm Supply-Chain Attack Abuses 72 Open VSX Extensions to Target Developers." The Hacker News, March 2026.

[5] The Register. "CrowdStrike, Google shatter Glassworm botnet." The Register, May 27, 2026.

[6] Datadog Security Labs. "Shai-Hulud Goes Open Source." Datadog Security Labs, 2026.

[7] HivePro. "TeamPCP's Automated Supply Chain: From Trivy to LiteLLM in a Multi-Ecosystem Breach." HivePro Threat Advisory, 2026.

[8] StepSecurity. "TeamPCP's Mini Shai-Hulud Is Back: A Self-Spreading Supply Chain Attack Compromises TanStack npm Packages." StepSecurity Blog, May 2026.

[9] Cybersecurity Dive. "Coordinated operation takes down Glassworm botnet." Cybersecurity Dive, May 2026.

[10] ReversingLabs. "Team PCP's Mini Shai-Hulud tears at open-source trust." ReversingLabs Blog, 2026.

[11] BleepingComputer. "GlassWorm malware hits 400+ code repos on GitHub, npm, VSCode, OpenVSX." BleepingComputer, 2026.

[12] Security Boulevard. "Mini Shai-Hulud: Frequently asked questions about the TeamPCP npm and PyPI supply chain campaign." Security Boulevard, May 2026.

[13] Trend Micro. "Analyzing TeamPCP's Supply Chain Attacks: Checkmarx KICS and elementary-data in CI/CD Credential Theft." Trend Micro Research, 2026.

[14] Palo Alto Networks Unit 42. "Weaponizing the Protectors: TeamPCP's Multi-Stage Supply Chain Attack on Security Infrastructure." Unit 42 Threat Intelligence, 2026.

[15] Deepwatch. "Software Supply Chain Alert: LiteLLM & Trivy Attack Actions." Deepwatch Labs, 2026.

[16] The Tech Marketer. "TeamPCP Supply Chain Attack 2026: One VS Code Extension Gave Hackers 3,800 GitHub Internal Repositories." The Tech Marketer, 2026.

[17] Rescana. "Megalodon Supply Chain Attack: TeamPCP Compromises 5,561 GitHub Repositories via Malicious CI/CD Workflows." Rescana, 2026.

[18] OpenAI. "Our response to the TanStack npm supply chain attack." OpenAI, May 2026.

[19] BleepingComputer. "TeamPCP hackers advertise Mistral AI code repos for sale." BleepingComputer, May 2026.

[20] The Hacker News. "GitHub Internal Repositories Breached via Malicious Nx Console VS Code Extension." The Hacker News, May 2026.