CSAI

cloud security alliance®

CSAI Foundation

Cloud Security Alliance AI Safety Initiative

TeamPCP: Supply Chain Attacks on AI Development Infrastructure

How March 2026's multi-wave campaign weaponized security scanners and LLM gateways to harvest enterprise AI credentials at scale

Unofficial AI-assisted Research

2026-04-09

Key Takeaways

- Between March 19 and March 27, 2026, the threat actor TeamPCP executed a four-wave supply chain campaign that compromised Trivy, Checkmarx KICS, LiteLLM, and the Telnyx Python SDK — targeting both security scanning infrastructure and AI development tooling in a sequence that analysis suggests was deliberately planned to cascade through connected tooling ecosystems.
- LiteLLM's architecture as a unified gateway to more than 100 large language model APIs created an exceptionally high-value target: a single compromised package could expose credentials for any or all major LLM providers — including OpenAI, Anthropic, AWS Bedrock, Google Vertex AI, and Azure OpenAI — for any organization that had those providers configured and installed the poisoned versions.
- TeamPCP exploited a structural vulnerability in modern CI/CD pipelines: security scanners occupy privileged positions with access to cloud credentials, secrets, and deployment tokens. Compromising a scanner is effectively compromising every pipeline in which it runs — the scanner processes the same environment variables, cloud tokens, and secrets used in each build it executes.
- CVE-2026-33634 (CVSS 9.4) formalizes the Trivy compromise. Organizations that ran affected Trivy or KICS GitHub Actions between March 19–27, or installed LiteLLM versions 1.82.7 or 1.82.8, should treat all accessible pipeline credentials as compromised and rotate immediately.
- As of late March 2026, TeamPCP has announced a monetization pivot in partnership with the Vect ransomware-as-a-service operation, converting the harvested credential corpus into ransomware deployments. The threat from the March campaign extends well beyond the initial exfiltration window.

Background

The TeamPCP Threat Actor

TeamPCP — tracked by Google's Threat Intelligence Group as UNC6780 and also identified under aliases PCPcat, ShellForce, and DeadCatx3 — emerged as a distinct threat actor in late 2025, though infrastructure analysis suggests operational activity dating to at least September 2025 [1]. The group is assessed as financially motivated, with campaigns demonstrating a level of operational sophistication atypical of opportunistic actors: multi-stage payload delivery, decentralized command-and-control infrastructure built on the Internet Computer Protocol (ICP), and deliberate pre-positioning through incomplete credential rotation exploitation [2].

The March 2026 campaign was not improvised. TeamPCP obtained credentials for the Aqua Security GitHub service account used to maintain Trivy in late February 2026 — a breach that Aqua Security detected but addressed with only partial secret rotation [3]. TeamPCP retained the unrotated secrets for three weeks before activating them, suggesting the group had pre-identified LiteLLM as a downstream target and was waiting for an opportune moment to cascade through connected tooling ecosystems. This deliberate patience distinguishes the campaign from typical opportunistic supply chain attacks.

Why AI Development Tooling Is a High-Value Target

To understand the March 2026 campaign, it is necessary to understand a structural feature of modern AI development pipelines: credential aggregation. Organizations building AI-powered products typically depend on a small number of gateway libraries that centralize API authentication across many model providers. LiteLLM is among the most widely deployed examples — a Python library that presents a unified OpenAI-compatible interface to more than 100 LLM providers. Deploying LiteLLM in production typically means storing API keys for OpenAI, Anthropic, AWS Bedrock, Google Vertex AI, and Azure OpenAI in a single environment, often alongside Kubernetes secrets and cloud IAM credentials needed to manage inference infrastructure [4].

From an attacker's perspective, this aggregation inverts the economics of credential theft. Rather than compromising individual provider accounts one at a time, targeting a widely deployed AI gateway library delivers access to an organization's entire AI infrastructure in a single package installation. LiteLLM reported approximately 95 million monthly PyPI downloads and deployment in an estimated 36% of monitored cloud environments at the time of the compromise [5]. The potential credential surface was correspondingly vast.

Security scanning tools compound this risk. Modern CI/CD pipelines route vulnerability scanners like Trivy and infrastructure-as-code analyzers like KICS through every build — meaning these tools routinely process the same environment variables, cloud tokens, and secrets used by production deployments. An organization that follows security best practices by scanning every container image is, in effect, handing its most sensitive credentials to the scanner on every build. TeamPCP understood and systematically exploited this dynamic.

Security Analysis

Wave One: Trivy and the Weaponized Scanner (March 19–22)

The campaign began at 18:22 UTC on March 19, when TeamPCP force-pushed malicious code to the Aqua Security GitHub release infrastructure using retained credentials from the February breach. Version 0.69.4 of the Trivy binary — distributed through GitHub Container Registry, Amazon ECR Public, Docker Hub, and official Debian and RPM packages — contained a multi-stage credential stealer [6]. Simultaneously, 76 of 77 version tags for `aquasecurity/trivy-action` and all seven tags for `aquasecurity/setup-trivy` were force-pushed with malicious GitHub Actions workflows.

The malicious payload operated covertly. After harvesting credentials, it continued executing the legitimate Trivy scan — returning expected results and generating no user-visible anomaly. The compromise remained active through March 23, creating a four-day window during which the most diligent organizations, those running security scans on every commit, had the greatest credential exposure — underscoring that the root problem is mutable-tag deployment, not scanning frequency. The payload dumped runner process memory to extract GitHub Actions secrets that the platform's masking mechanism would otherwise protect, scanned more than 50 file paths for SSH keys, cloud credentials, and Kubernetes configurations, and exfiltrated the collected data encrypted with AES-256-CBC and RSA-4096 to an attacker-controlled domain (`scan.aquasecurtiy.org` , a typoquatted domain mimicking Aqua Security's legitimate infrastructure) [7].

CVE-2026-33634, assigned a CVSS score of 9.4, formally captures this compromise. Safe Trivy versions are v0.69.3 and earlier. For GitHub Actions workflows, the safe references are `aquasecurity/trivy-action@0.35.0` and `aquasecurity/setup-trivy@0.2.6` , though pinning to these tags is only safe if the reference is a full commit SHA rather than a mutable label [8].

Wave Two: KICS and Cross-Vendor Lateral Movement (March 23)

On March 23, TeamPCP used npm tokens harvested from the Trivy pipeline compromise to pivot to Checkmarx's GitHub Actions infrastructure — demonstrating that the stolen credentials from one vendor's CI/CD system were sufficient to authenticate against another vendor's publishing infrastructure. All 35 version tags of `checkmarx/kics-github-action` and version 2.3.28 of `checkmarx/ast-github-action` were force-pushed with payloads deploying a three-stage infection sequence [9]. Additionally, two Checkmarx VS Code extensions — `checkmarx.ast-results` v2.53 (approximately 36,000 downloads) and `checkmarx.cx-dev-assist` v1.7.0 — were published to the OpenVSX Registry with malicious payloads targeting developer workstations rather than CI/CD infrastructure [10].

This cross-vendor lateral movement has significant implications for how organizations assess the independence of their vendor supply chains. Organizations that had already rotated secrets after the Trivy disclosure on March 20 may not have recognized that KICS workflows running in the same pipelines were freshly compromised. The assumption that vendor supply chains are isolated from each other is demonstrably incorrect when multiple vendors' automation relies on shared token namespaces or overlapping CI/CD environments.

Wave Three: LiteLLM and the AI Gateway Compromise (March 24)

The campaign's highest-credential-yield action occurred on March 24 at approximately 10:52 UTC, when two malicious LiteLLM versions — 1.82.7 and 1.82.8 — were published to PyPI within 13 minutes of each other using API tokens obtained during the Trivy compromise [11]. The two versions employed distinct but complementary delivery mechanisms designed to maximize persistence.

Version 1.82.7 dropped a double-Base64-encoded payload to disk as `p.py`, executing when the LiteLLM proxy was invoked. Version 1.82.8 escalated to a more severe persistence mechanism: a file named `litellm_init.pth` placed in the Python site-packages directory. Python's interpreter processes `.pth` files automatically during interpreter initialization — before any user code executes and regardless of whether LiteLLM is explicitly imported. This means the backdoor installed by version 1.82.8 activated on every Python process invocation across the affected host, not merely when LiteLLM itself was called [4]. An organization that installed the compromised package and subsequently pivoted to a different LLM library could still be running the TeamPCP payload with every Python execution — if the `litellm_init.pth` file was not explicitly removed from the Python site-packages directory during the transition.

The payload's credential harvest targeted the full scope of what LiteLLM environments typically hold: API keys for all configured LLM providers, SSH keys, AWS and GCP and Azure cloud credentials, Kubernetes service account tokens, CI/CD secrets, Docker registry credentials, database connection strings, and cryptocurrency wallet files. Exfiltration was encrypted with AES-256, the key further protected by an embedded RSA public key, and transmitted to `models.litellm.cloud` (version 1.82.8) or `checkmarx.zone` (version 1.82.7) — both attacker-controlled domains designed to blend with legitimate tooling infrastructure [12]. PyPI quarantined the malicious packages approximately 40 minutes after publication. However, organizations that had already installed those versions during that window, or that had cached them in internal artifact repositories, remained exposed.

Wave Four: Telnyx SDK and the Persistence Campaign (March 27)

On March 27, versions 4.87.1 and 4.87.2 of the Telnyx Python SDK were published with a steganographic payload — malicious code embedded within a WAV audio file included in the package, a technique designed to evade static analysis tools that inspect code but do not analyze binary media assets [10]. With more than 3.75 million cumulative PyPI downloads, Telnyx represents a communications and telephony integration layer rather than AI infrastructure specifically, but the same credential sweep logic applied: any environment running Telnyx alongside AI workloads was at risk of simultaneous exfiltration of both communications API keys and LLM credentials.

The Ransomware Monetization Pivot

Following the March 27 Telnyx compromise, TeamPCP paused active supply chain operations. The pause proved to be strategic repositioning: in late March, TeamPCP announced a formal partnership with Vect, a newly emerged ransomware-as-a-service operation, via BreachForum, confirming the strategic direction the pause implied [13]. The first confirmed Vect ransomware deployment using TeamPCP-sourced credentials has been reported. Researchers assess that an estimated 300 GB of stolen credentials from the campaign remain in TeamPCP's possession, providing a long-duration monetization inventory that the group has stated publicly it intends to exploit over an extended timeline.

The pivot from credential theft to ransomware deployment significantly extends the threat horizon for organizations affected by the March campaign. Rotation actions taken in March and early April are necessary but may be insufficient if any credentials were exfiltrated before the compromise was detected and systems have not been thoroughly re-imaged. Organizations that cannot confirm clean pipeline environments from before March 19 should assume ongoing exposure.

Recommendations

Immediate Actions

Organizations that ran `aquasecurity/trivy-action` (any tag), `aquasecurity/setup-trivy` (any tag), `checkmarx/kics-github-action` (any tag), or `checkmarx/ast-github-action` at version 2.3.28 between March 19 and March 27, or that installed LiteLLM versions 1.82.7 or 1.82.8, or Telnyx SDK versions 4.87.1 or 4.87.2, should treat all secrets accessible to those workflows as fully compromised. This includes cloud provider IAM credentials, Kubernetes service account tokens, GitHub personal access tokens and deploy keys, npm and PyPI publishing tokens, LLM provider API keys (all providers), database credentials, and SSH keys present in the runner environment. Rotate all of these credentials without exception; do not attempt to enumerate which specific secrets were present during each run. The payload performed exhaustive sweeps of more than 50 credential file paths and runner process memory, so absence of a credential from a workflow's explicit configuration does not guarantee it was not harvested.

Any host that had LiteLLM 1.82.7 or 1.82.8 installed should be treated as running persistent malware regardless of whether the package has since been removed or updated. The `.pth` file mechanism used in version 1.82.8 may persist across package removal, as pip does not reliably clean non-standard files from the site-packages directory [17]. Given this persistence risk, reimaging affected hosts from a known clean state is preferable to in-place remediation. At minimum, audit the Python site-packages directory on any affected host for the presence of `litellm_init.pth`, and perform thorough credential rotation for all credentials accessible from the affected environment [14].

Short-Term Mitigations

Pin all GitHub Actions references to full 40-character commit SHAs rather than mutable version tags. Tags in GitHub Actions are not immutable references — they can be force-pushed by any party with write access to the repository, as the TeamPCP campaign demonstrated at scale. SHA pinning ensures that a workflow runs exactly the code that was reviewed and trusted at the time of pinning, and that a compromised upstream cannot substitute different code without the organization's awareness. Enable branch protection rules that prohibit force-pushes to release tags across all internally maintained GitHub repositories.

Audit all CI/CD artifact dependencies using a software bill of materials (SBOM) tool. For Python packages in particular, verify that installed versions match expected checksums before pipeline execution. Internal artifact mirrors should be configured to pin to verified-clean versions and block

automatic pull of upstream updates during incident response periods.

Strategic Considerations

The TeamPCP campaign reveals a structural vulnerability in how AI development infrastructure is typically assembled: credential aggregation without isolation. Organizations should evaluate whether LLM provider API keys need to be co-located in the same environment as CI/CD tokens and cloud IAM credentials. Where feasible, deploy LiteLLM or equivalent AI gateway libraries in dedicated, isolated environments with narrowly scoped credentials that do not also hold access to build infrastructure or source repositories.

Treat AI gateway dependencies as critical security infrastructure rather than convenience libraries. Dependencies that aggregate credentials for dozens of external services warrant the same dependency management rigor applied to authentication libraries and secrets managers — including regular supply chain audits, pin-and-verify deployment practices, and privileged access monitoring on the processes that handle them. Organizations that have not yet assessed their AI development toolchain for the concentration of credential access should do so as a near-term priority, independent of this specific campaign.

Consider implementing runtime monitoring for unexpected network egress from Python processes, particularly in CI/CD environments. The LiteLLM payload transmitted data to attacker-controlled domains that could have been detected by network-layer controls monitoring for connections to unrecognized external hosts from build runners.

CSA Resource Alignment

This campaign directly engages several areas of CSA's AI security framework. CSA's MAESTRO threat modeling framework for agentic AI systems addresses supply chain compromise of AI runtime dependencies as a deployment infrastructure threat, recognizing that poisoned dependencies in the development pipeline can cascade upward to compromise model inference authorization, API credential management, and agent orchestration layers [15]. The TeamPCP LiteLLM compromise is a concrete instance of this threat pattern: a poisoned dependency propagated through the AI development pipeline to undermine the infrastructure that AI systems depend on at runtime.

CSA's AI Controls Matrix (AICM) [19] addresses credential management for AI workloads under its Secure Development and Supply Chain controls. The AICM's guidance on dependency integrity verification — cryptographic validation of packages before installation and SBOM maintenance across

the AI development toolchain — directly addresses the mutable-tag vulnerability exploited in the Trivy and KICS compromises. Organizations should map their AI pipeline dependency practices against AICM controls as part of their post-incident review.

The Cloud Controls Matrix (CCM) v4 provides applicable controls under the Supply Chain Management (STA) and Cryptography, Encryption & Key Management (CEK) domains. Specifically, CCM STA-04 (Supply Chain Inventory) and CEK-04 (Encryption Algorithm) are directly relevant to the credential isolation and package verification requirements surfaced by this campaign [16].

CSA's Agentic Trust Framework [20] addresses the principle of least-privilege access for AI workloads — specifically, that AI inference environments should not hold credentials for systems outside their operational scope. Applying this principle to LiteLLM deployments would have limited the blast radius of the March 24 compromise: an LLM gateway operating under least-privilege would hold only the LLM provider credentials it actively requires, not the full range of CI/CD tokens and cloud IAM credentials that co-located pipeline environments may also expose.

CSA has published two prior research notes directly addressing the technical phases of this campaign: a cascade analysis published April 2 [17] and a CI/CD kill chain analysis published April 3 [18]. This research note supplements those publications with updated threat intelligence on the Vect ransomware partnership and extended guidance specific to organizations operating AI infrastructure.

References

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

[2] SANS Institute. "When the Security Scanner Became the Weapon: Inside the TeamPCP Supply Chain Campaign." SANS Institute, March 2026.

[3] Arctic Wolf Networks. "TeamPCP Supply Chain Attack Campaign Targets Trivy, Checkmarx (KICS), and LiteLLM." Arctic Wolf, March 2026.

[4] Snyk. "How a Poisoned Security Scanner Became the Key to Backdooring LiteLLM." Snyk, March 2026.

[5] Wiz. "LiteLLM TeamPCP Supply Chain Attack: Malicious PyPI Packages." Wiz Blog, March 2026.

[6] Wiz. "Trivy Compromised by 'TeamPCP'." Wiz Blog, March 2026.

[7] Penligent AI. "CVE-2026-33634 and the Trivy Supply Chain Compromise." Penligent, March 2026.

[8] Microsoft Security. "Guidance for Detecting, Investigating, and Defending Against the Trivy Supply Chain Compromise." Microsoft Security Blog, March 24, 2026.

[9] Burns & McDonnell Advisories. "March 2026 Developer Supply Chain Attack Wave: TeamPCP CI/CD Infrastructure Campaign (CVE-2026-33634)." Burns & McDonnell, March 2026.

[10] ReversingLabs. "Inside the TeamPCP Cascading Supply Chain Attack." ReversingLabs Blog, March 2026. *Covers both the Checkmarx VS Code extension compromises (Wave Two) and the Telnyx SDK steganographic payload (Wave Four).*

[11] Help Net Security. "LiteLLM PyPI Packages Compromised in Expanding TeamPCP Supply Chain Attacks." Help Net Security, March 25, 2026.

[12] Phoenix Security. "TeamPCP Attack Day 6 Backdoors LiteLLM: Your AI Gateway Just Became the Attack Vector." Phoenix Security, March 2026.

[13] Help Net Security. "TeamPCP's Attack Spree Slows, but Threat Escalates with Ransomware Pivot." Help Net Security, March 30, 2026.

[14] Bastion. "LiteLLM PyPI Supply Chain Attack: What Happened and How to Stay Safe." Bastion, March 2026.

[15] Cloud Security Alliance. "Agentic AI Threat Modeling Framework: MAESTRO." CSA Blog, February 6, 2025.

[16] Cloud Security Alliance. "Cloud Controls Matrix v4." CSA, 2021.

[17] Cloud Security Alliance AI Safety Initiative. "TeamPCP Supply Chain Cascade: When Security Tools Become Attack Infrastructure." CSA Labs, April 2, 2026.

[18] Cloud Security Alliance AI Safety Initiative. "TeamPCP: CI/CD Kill Chain from Trivy to the EU." CSA Labs, April 3, 2026.

[19] Cloud Security Alliance. "AI Controls Matrix." CSA, 2025.

[20] Cloud Security Alliance. "Agentic Trust Framework: Zero Trust for AI Agents." CSA Blog, February 2, 2026.