CSAI Foundation

Cloud Security Alliance AI Safety Initiative

TeamPCP: Trojanized Security Tools Backdoor AI Infrastructure

A Cascading Supply Chain Campaign Targeting LLM Gateways and CI/CD Pipelines

Unofficial AI-assisted Research

2026-04-06

Key Takeaways

- TeamPCP executed a precisely sequenced, cascading supply chain campaign between March 19–27, 2026, compromising Trivy, Checkmarx KICS, LiteLLM, and the Telnyx Python SDK in succession across three distribution platforms — GitHub Actions, OpenVSX, and PyPI — in five discrete compromise events.
- The campaign's most consequential step was the trojanization of LiteLLM, a universal LLM proxy gateway with 97 million monthly PyPI downloads [1] that processes AI traffic to OpenAI, Anthropic, Google Gemini, AWS Bedrock, and dozens of other LLM providers simultaneously — making a compromised LiteLLM deployment a single point of exposure for all LLM provider credentials configured in that instance, potentially spanning an organization's entire AI provider stack.
- Credentials stolen from one upstream victim's CI/CD pipeline were used to compromise the next downstream target, creating a trust-chain vulnerability. The cascade progressed across four ecosystems in eight days, compressing the available response window between each stage.
- TeamPCP harvested an estimated 300 GB of data from over 500,000 infected systems [2], including LLM provider API keys, cloud credentials, Kubernetes service account tokens, and CI/CD pipeline secrets.
- Organizations using AI infrastructure built on Python-based LLM frameworks must treat their CI/CD pipelines and package dependencies as adversarially contested surfaces, not trusted internal resources.

Background

TeamPCP is a financially motivated threat actor cluster that emerged in mid-2025. Within six months of emergence, the group had compromised four software distribution ecosystems and an estimated 500,000 systems; Cyble researchers have characterized TeamPCP as one of the most active cloud-native supply chain threat groups documented to date [3]. The group is tracked under multiple aliases including DeadCatx3, PCPcat, PersyPCP, ShellForce, and CipherForce, reflecting the evolution of both its tooling and operational scope [3]. What began as an automated worm campaign targeting

misconfigured Docker APIs and exposed Kubernetes control planes evolved, within six months, into a sophisticated multi-stage operation that systematically weaponized the software tools developers trust most.

The group's initial phase, active between December 2025 and February 2026, established its operational template. TeamPCP deployed automated scanning infrastructure — including tools named `scanner.py`, `kube.py`, and `pcpcat.py` — to enumerate CIDR ranges for exposed container orchestration services, Ray distributed compute dashboards, and Redis instances [4]. The campaign exploited CVE-2025-55182, a critical 10.0-severity remote code execution vulnerability in React/Next.js applications dubbed "React2Shell" [5], compromising an estimated 60,000 servers and building both a cryptomining botnet and a credential harvest pipeline. During this period, TeamPCP also deployed the first generation of CanisterWorm, a self-propagating npm worm whose defining innovation was the use of an Internet Computer Protocol (ICP) decentralized canister as a censorship-resistant command-and-control dead-drop. Mend.io researchers assessed this as the first documented instance of this technique in a supply chain attack [6]. Because no single hosting provider controls ICP canisters, conventional takedown requests have limited efficacy against this C2 architecture, as no single hosting provider controls ICP canister execution.

By March 2026, TeamPCP had accumulated a substantial cache of stolen CI/CD tokens, developer credentials, and organizational access — the raw material for what security researchers would come to call the "Five-Day Siege."

Security Analysis

The Cascade Architecture

What makes the March 2026 campaign analytically significant is not any single exploit, but the architectural logic connecting them. TeamPCP did not select targets randomly. Each compromise was chosen because its CI/CD pipeline consumed the output of the previous victim, creating a downstream trust chain that functioned as both a propagation mechanism and an attribution-evasion technique. Defenders investigating the LiteLLM compromise, for example, found no direct attack on LiteLLM's own infrastructure — only the consequence of trusting a security tool that had already been silently subverted.

Step One: The Trivy GitHub Actions Compromise (March 19, 2026)

The cascade originated with Trivy, the open-source vulnerability scanner maintained by Aqua Security and one of the most widely deployed container security tools in enterprise CI/CD pipelines. In late February 2026, an AI-powered autonomous bot later attributed to TeamPCP — designated "hackerbot-claw" by researchers — targeted CI/CD workflows at Microsoft, Datadog, and Aqua Security repositories, opening innocuous-looking pull requests designed to steal GitHub Personal Access Tokens [7]. Aqua Security disclosed the intrusion on March 1 and rotated credentials; however, researchers subsequently assessed that the rotation was incomplete, and TeamPCP appears to have retained residual access [8].

On March 19, using that retained access, TeamPCP force-pushed malicious commits to 76 of 77 version tags in the `aquasecurity/trivy-action` GitHub repository and all tags in `aquasecurity/setup-trivy` [8]. The technique exploited a fundamental characteristic of how GitHub Actions resolves tags at runtime: any pipeline that referenced `aquasecurity/trivy-action@v0.35.0` or any other pinned tag would automatically begin executing attacker-controlled code without any modification to its own workflow file. The malicious commits were engineered to deceive human reviewers — reusing original author metadata, commit messages, and timestamps while substituting only the `entrypoint.sh` entrypoint.

The payload, which researchers named the "TeamPCP Cloud Stealer," was designed for maximum credential breadth. It dumped the `Runner.Worker` process memory to harvest secrets injected by GitHub Actions' own secret-handling mechanism, then swept the runner filesystem for SSH keys, cloud credentials (AWS, GCP, Azure), Kubernetes secrets, Docker configurations, pipeline tokens, shell history, and cryptocurrency wallet files [9]. All collected material was encrypted using AES-256 combined with RSA-4096 and exfiltrated to attacker-controlled infrastructure. Post-compromise lateral movement used the Sliver open-source C2 framework, with at least one confirmed node at `67.217.57.240` [10].

Step Two: Checkmarx KICS and the OpenVSX Extensions (March 23, 2026)

Trivy's compromise yielded credentials from dozens of organizations whose pipelines had consumed the malicious action during its window of exposure. Among those was a Checkmarx CI/CD pipeline. Using credentials stolen via that Trivy-infected pipeline, TeamPCP on March 23 force-pushed malicious commits to all 35 version tags of `checkmarx/kics-github-action` and poisoned `checkmarx/ast-github-action@v2.3.28` [11]. The group simultaneously published two trojanized VS Code extensions to the OpenVSX registry within 12 seconds of each other: `ast-results-2.53.0.vsix` and `cx-dev-assist-1.7.0.vsix` [12]. These extensions actively

checked the developer's environment for GitHub, AWS, GCP, and Azure credentials before fetching second-stage payloads from attacker-controlled `checkmarx[.]zone` infrastructure. The exposure window lasted approximately 13 hours before remediation [11]. Notably, the official VS Code Marketplace was not affected — the attack was scoped to OpenVSX, the alternative registry used by open-source IDE distributions including VSCodium.

Step Three: LiteLLM and the AI Gateway Compromise (March 24, 2026)

The most consequential step for AI infrastructure security was the trojanization of LiteLLM. LiteLLM is a universal LLM proxy library that provides a unified Python interface for routing API calls to OpenAI, Anthropic, Google Gemini, AWS Bedrock, Cohere, and over 100 other LLM providers [13]. Organizations deploy it as an internal AI gateway, and it processes approximately 97 million monthly PyPI downloads. At the time of the compromise, Hunt.io scanning identified 33,688 internet-facing LiteLLM deployments [1].

LiteLLM's CI/CD pipeline used Trivy for security scanning [14]. When Trivy was poisoned on March 19, LiteLLM's pipeline executed the malicious action during its normal security scanning workflow, and the TeamPCP Cloud Stealer harvested the LiteLLM maintainer's PyPI publish credentials along with everything else it found. On March 24 — five days after the Trivy compromise — TeamPCP used those credentials to publish malicious versions `1.82.7` and `1.82.8` to PyPI [14]. The packages remained live for approximately 40 minutes before PyPI quarantined them [14].

The technical sophistication of the LiteLLM payload warrants specific attention. Rather than injecting malicious code into a script that would only run when explicitly invoked, the payload abused Python's `.pth` file mechanism — planting a file named `litellm_init.pth` in the site-packages directory that executes a double-base64-encoded payload on every Python interpreter startup [15]. Any Python process on the system, not merely LiteLLM itself, would trigger the malware on launch. The environmental reach of a compromised LiteLLM installation was also unusually broad: because LiteLLM is configured with credentials for every LLM provider an organization uses, the malware's credential harvesting scope included OpenAI API keys, Anthropic API keys, Cohere keys, AWS credentials, GCP and Azure service account tokens, and Kubernetes service account tokens — effectively the entire AI provider stack of any organization that had installed the package during the exposure window [16]. Exfiltration was identifiable by the HTTP header `X-Filename: tpcp.tar.gz` sent to `models.litellm.cloud` [15].

Step Four: Telnyx and CVE-2026-33634 (March 27, 2026)

Three days after the LiteLLM attack, TeamPCP published malicious versions `4.87.1` and `4.87.2` of the Telnyx Python SDK, a telecommunications API library with 742,000 documented downloads [17]. The intrusion vector in this case was CanisterWorm, which had found a Telnyx npm token in a previously compromised CI environment. The Telnyx payload, assigned CVE-2026-33634, introduced a novel delivery technique: the malicious code in `telnyx/_client.py` fetched audio files – `hangup.wav` on Windows and `ringtone.wav` on Linux and macOS – from a C2 server at `83[.]142[.]209[.]203:8080` and used XOR decoding to extract third-stage executables concealed within the WAV data via steganography [18]. On Windows, the decoded payload dropped as `msbuild.exe` in the Startup folder for persistence. On Linux and macOS, it decoded a Python-based credential collector that exfiltrated data using AES-256-CBC. Security researchers assessed this use of audio steganography for payload delivery as a meaningful evasion advance relative to conventional payload delivery mechanisms, as WAV file content is less commonly inspected by network security tools than executable formats [18].

Post-Compromise Exploitation and Confirmed Impact

With the credential harvest from four compromised ecosystems in hand, TeamPCP transitioned to active exploitation. The group used TruffleHog to validate stolen AWS access keys and Azure secrets at scale, then began cloud discovery operations within 24 hours of credential validation – enumerating ECS clusters, EKS task definitions, and AWS Secrets Manager entries [19]. Post-compromise C2 infrastructure included AdaptixC2 (beaconing to `83.142.209.11:443`) and Havoc C2 operating in parallel – an architectural choice researchers assessed as providing C2 redundancy [10]. The use of two frameworks in parallel provided resilience against detection and blocking of either channel in isolation.

The first confirmed organizational victim disclosure was Mercor, an AI recruiting platform, breached via the LiteLLM supply chain compromise [19]. Databricks reported an ongoing investigation into a potential compromise [19]. Separately, alleged AstraZeneca clinical data – potentially containing PHI subject to HIPAA and GDPR – appeared in material attributed to the campaign's data release operations [20]. TeamPCP simultaneously established two ransomware monetization tracks: a proprietary CipherForce ransomware program and an affiliate relationship with the Vect Ransomware-as-a-Service program advertised on BreachForums [21]. Researchers at vx-underground reported an aggregate figure of 300 GB of data exfiltrated from over 500,000 infected systems across the campaign's full duration [2].

Recommendations

Immediate Actions

Organizations that ran any pipeline using `aquasecurity/trivy-action`, `checkmarx/kics-github-action`, `checkmarx/ast-github-action`, or any LiteLLM PyPI package between March 19 and March 27, 2026 should treat those environments as potentially compromised and initiate credential rotation as a precautionary measure, regardless of whether active indicators of compromise are present. Every LLM provider API key, cloud service credential, and CI/CD secret accessible from those environments should be rotated immediately. Organizations should also audit Python `site-packages` directories on affected systems for the presence of `litellm_init.pth` or other unexpected `.pth` files, which represent the persistence mechanism from the LiteLLM payload. Network logs should be reviewed for outbound connections to `models.litellm.cloud`, `checkmarx[.]zone`, and `83.142.209.203:8080`.

Short-Term Mitigations

The structural vulnerability TeamPCP exploited — the practice of referencing GitHub Actions by mutable version tags — has a known and deployable fix. Organizations should pin all GitHub Actions references to immutable commit SHA hashes (e.g., `uses: aquasecurity/trivy-action@sha256:abc123...`) rather than floating tags. Tag references can be silently rewritten by anyone with repository write access, as this campaign demonstrated; commit SHAs cannot. Teams using tools such as Dependabot or Renovate can automate SHA-pinning and keep those pins current.

Similarly, PyPI package references in CI/CD pipelines should be pinned to specific version hashes using `pip`'s `--require-hashes` flag or a lockfile format such as `pip-compile` with hash verification enabled. Hash verification ensures that the package installed matches a previously reviewed artifact — preventing silent substitution of a known-good version. It does not protect against pinning to a version that was itself malicious at the time of initial adoption, which is why continuous monitoring of package integrity and timely response to supply chain alerts remain essential complements to hash-pinning.

The use of a decentralized ICP canister for C2 infrastructure by CanisterWorm warrants specific attention in network security policy. Traditional domain-block and IP-block C2 defense strategies have limited efficacy against canister-based C2. Organizations should evaluate whether their network egress policies include monitoring for ICP canister protocol traffic, which operates over HTTPS to ICP boundary nodes.

Strategic Considerations

The LiteLLM compromise reveals a structural risk that is likely to persist and intensify as AI adoption accelerates: the aggregation of LLM provider credentials in a single proxy layer creates a uniquely attractive target. Organizations that deploy LiteLLM or architecturally equivalent AI gateways — Portkey, OpenRouter, or internal proxy implementations — should apply the same least-privilege and secret rotation disciplines to their LLM credential stores as they would to cloud root credentials. LLM API keys with access to production AI workloads should be scoped to specific models and use cases wherever provider APIs support it, and should be rotated on a regular schedule independent of suspected compromise.

More broadly, the TeamPCP campaign demonstrates that security tooling within CI/CD pipelines carries the same supply chain risk surface as application dependencies. Security scanners, linting tools, and test frameworks in pipelines should be subjected to the same supply chain vetting that organizations apply to application dependencies — evaluated not only for the vulnerabilities they detect, but also for the trust they are implicitly granted within the pipeline environments where they run.

CSA Resource Alignment

The TeamPCP campaign maps directly to several areas of active CSA AI Safety Initiative research and guidance. The MAESTRO threat modeling framework for agentic AI systems [22] specifically addresses trust boundary violations in multi-component AI pipelines; the LiteLLM compromise is a concrete instantiation of MAESTRO's "orchestration layer compromise" threat scenario, in which a single poisoned component gains visibility into all downstream provider interactions. Organizations using MAESTRO to model their AI infrastructure should explicitly include the package and dependency supply chain as a trust boundary requiring integrity verification.

The CSA Cloud Controls Matrix (CCM) and AI Control Matrix (AICM) address supply chain security under the Supply Chain Management and Transparency (STA) control domain. The TeamPCP campaign provides direct evidence for the specific controls around third-party component integrity verification (STA-09) and CI/CD pipeline security (STA-08). The AICM, as a superset of CCM incorporating AI-specific risks, additionally provides controls relevant to AI gateway security and credential management for LLM provider integrations.

CSA's Zero Trust guidance [23] is directly applicable to the post-compromise exploitation phase of this campaign, in which TeamPCP used validated cloud credentials to enumerate and access cloud environments without raising alerts. A Zero Trust architecture that continuously validates both the

identity of the requesting principal and the integrity of the requesting workload — rather than trusting a valid credential unconditionally — would limit the lateral movement opportunity that stolen CI/CD credentials provided. The CSA Trusted Cloud Initiative and STAR program provide assessment frameworks that organizations can use to evaluate third-party AI tool vendors against these supply chain security standards.

References

[1] Hunt.io. "33K Exposed LiteLLM Deployments and TeamPCP C2 Servers." Hunt.io Blog, March 2026.

[2] vx-underground. Cited in: Datadog Security Labs. "LiteLLM and Telnyx Compromised on PyPI: Tracing the TeamPCP Campaign." Datadog Security Labs, March 2026.

[3] Cyble. "TeamPCP Threat Actor Profile." Cyble Research, March 2026.

[4] iSec News. "Worm-Driven TeamPCP Campaign Compromises Cloud-Native Infrastructure at Scale." iSec News, February 9, 2026.

[5] Rescana. "TeamPCP Worm Targets Docker, Kubernetes, Ray, and Redis via React2Shell (CVE-2025-55182)." Rescana, February 2026.

[6] Mend.io. "CanisterWorm: The Self-Spreading npm Attack That Uses a Decentralized Server to Stay Alive." Mend.io Blog, March 2026.

[7] The Hacker News. "Trivy Security Scanner GitHub Actions Compromised in Supply Chain Attack." The Hacker News, March 2026.

[8] Wiz. "Trivy Compromised by TeamPCP: Supply Chain Attack on GitHub Actions." Wiz Blog, March 2026.

[9] CrowdStrike. "From Scanner to Stealer: Inside the trivy-action Supply Chain Compromise." CrowdStrike Blog, March 2026.

[10] Flare.io. "Threat Alert: TeamPCP — Cloud-Native Ransomware." Flare.io Blog, 2026.

[11] Sysdig. "TeamPCP Expands: Supply Chain Compromise Spreads from Trivy to Checkmarx GitHub Actions." Sysdig Blog, March 2026.

[12] ReversingLabs. "Inside the TeamPCP Cascading Supply Chain Attack." ReversingLabs Blog, March 2026.

[13] LiteLLM. "Security Update: Suspected Supply Chain Incident — March 2026." LiteLLM Official Blog, March 2026.

[14] Snyk. "How a Poisoned Security Scanner Backdoored LiteLLM." Snyk Blog, March 2026.

[15] Sonatype. "Compromised litellm PyPI Package Delivers Multi-Stage Credential Stealer." Sonatype Blog, March 2026.

[16] Datadog Security Labs. "LiteLLM and Telnyx Compromised on PyPI: Tracing the TeamPCP Campaign." Datadog Security Labs, March 2026.

[17] Aikido. "Popular Telnyx Package Compromised on PyPI by TeamPCP." Aikido Blog, March 2026.

[18] MrCloudBook. "TeamPCP Supply Chain Attack — Telnyx, CanisterWorm Full Analysis (CVE-2026-33634)." MrCloudBook, March 2026.

[19] SANS Internet Storm Center. "TeamPCP Supply Chain Campaign Update 005: First Confirmed Victim Disclosure, Post-Compromise Cloud Enumeration." SANS ISC, March 2026.

[20] SANS Internet Storm Center. "TeamPCP Supply Chain Campaign Update 004: Databricks Investigating Alleged Compromise, TeamPCP Runs Dual Ransomware Operations." SANS ISC, March 2026.

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

[22] Cloud Security Alliance. "MAESTRO: Multi-Agent Environment, Security, Threat, and Risk Ontology." CSA AI Safety Working Group, 2025.

[23] Cloud Security Alliance. "Zero Trust Advancement Center." CSA, 2025.