**CSAI Foundation**

Cloud Security Alliance AI Safety Initiative

# TeamPCP: CI/CD Kill Chain from Trivy to the EU

How a Poisoned Security Scanner Became a Cascading Supply Chain Weapon

Unofficial AI-assisted Research

2026-04-03

## Key Takeaways

- **A misconfigured GitHub Actions workflow** in Aqua Security's Trivy repository allowed an automated attacker bot to exfiltrate a service account Personal Access Token (PAT) in late February 2026, setting the stage for a much larger compromise on March 19, 2026.
- **TeamPCP poisoned 76 of 77 version tags** in the trivy-action repository [2] and all 7 tags in setup-trivy, ensuring that every CI/CD pipeline pinned to a version tag — rather than a commit SHA — executed attacker-controlled code on its next run.
- **A five-stage payload** harvested secrets from GitHub Actions runner memory and developer filesystems, encrypted the data with RSA-4096, and exfiltrated it to a typosquatted domain; a persistent backdoor polled Internet Computer Protocol (ICP) blockchain infrastructure for second-stage commands, making conventional C2 takedown ineffective.
- **The campaign cascaded across five ecosystems** within one week: Trivy, npm, Checkmarx KICS GitHub Actions, LiteLLM, and Telnyx, linked by a shared RSA key and reuse of harvested credentials.
- **The European Commission** is the first confirmed victim of direct TeamPCP credential operationalization from the campaign; AWS API keys obtained through the March 19 Trivy compromise enabled access to EU Commission AWS accounts, resulting in 91.7 GB (approximately 340 GB uncompressed) of data stolen from 71 EU entities across a five-day window before detection.
- CVE-2026-33634 (reported CVSS 9.4 [7]) is the primary assigned identifier for the Trivy supply chain compromise and its downstream impact.
- **Safe Trivy versions**: v0.69.3 and earlier; trivy-action tag 0.35.0; setup-trivy tag 0.2.6. Organizations that ran v0.69.4 or any trivy-action or setup-trivy tag between March 19 and approximately March 24 should treat their CI/CD environments as fully compromised.

---

## Background

Trivy is Aqua Security's open-source container and filesystem vulnerability scanner, among the most widely deployed security scanning tools in modern CI/CD pipelines [17]. Its ubiquity is precisely what made it an attractive target: compromising Trivy is not an attack against a single organization but against

the security scanning layer of a large fraction of modern CI/CD pipelines simultaneously. The threat actor that executed this campaign, known as TeamPCP and tracked under aliases including PCPcat, Persy_PCP, ShellForce, and CipherForce, is assessed as a financially motivated criminal group that first surfaced in documented threat intelligence in December 2025 [1]. The group specializes in cloud-native infrastructure compromise, operating through a worm-driven methodology that targets exposed Docker APIs, Kubernetes clusters, Ray dashboards, and Redis servers, with objectives spanning credential theft, data exfiltration, ransomware deployment, and cryptocurrency mining [2].

The attack that culminated on March 19, 2026, appears to have been deliberately staged rather than opportunistic. An initial breach by an automated bot named `hackerbot-claw` in late February 2026 exploited a `pull_request_target` GitHub Actions workflow that had existed in the Trivy repository since October 2025 [3]. This trigger executes workflows from the base repository branch with elevated permissions even when the triggering pull request originates from an external fork, a well-documented GitHub Actions misconfiguration class. The bot used this to exfiltrate a PAT belonging to the `aqua-bot` service account. Aqua Security's reported response — credential rotation — appears to have left residual access paths open [17]; those surviving credentials were subsequently used to initiate the March 19 campaign [5].

The March 19 campaign should be understood as a case study in cascading supply chain compromise. TeamPCP did not need to find a novel vulnerability in any downstream consumer of Trivy. By poisoning the distribution mechanism — the GitHub tags that CI/CD pipelines resolve by name — the group transformed a single point of compromise into a persistent execution vector across every organization that ran a pipeline referencing those tags before remediation [4].

---

# Security Analysis

## The Kill Chain

At 17:43:37 UTC on March 19, 2026, TeamPCP used the surviving `aqua-bot` credentials to push a malicious tag, `v0.69.4`, to the aquasecurity/trivy repository, triggering an automated release pipeline [5]. The attacker created spoofed commits impersonating two legitimate contributors — consistent with an effort to reduce the likelihood of casual review detecting the manipulation. Using `git tag -f` with force-push capability, TeamPCP then repointed 76 of 77 version tags in aquasecurity/trivy-action [2] and all 7 tags in aquasecurity/setup-trivy to malicious commits. The malicious binary was

simultaneously published to GitHub Releases, Docker Hub, GitHub Container Registry, and Amazon ECR, ensuring coverage across the primary channels through which organizations consume Trivy in automated pipelines.

The payload executed in five stages. First, it enumerated GitHub Actions Runner processes (Runner.Worker, Runner.Listener) and read `/proc/{PID}/environ` and process memory, targeting the JSON pattern `{"value":"<secret>","isSecret":true}` that GitHub Actions uses internally to store secrets in runner memory during execution [5]. On self-hosted runners, where runners operate on developer machines or internal servers rather than GitHub-managed infrastructure, the payload executed a broad filesystem scan targeting SSH keys, AWS and cloud provider credentials, Kubernetes configuration files, Docker registry credentials, CI/CD platform tokens (GitLab, Jenkins), Terraform state and variable files, TLS certificates, and cryptocurrency wallet keys [3].

Collected data was encrypted with AES-256-CBC under a randomly generated session key, which was itself encrypted with a hardcoded RSA-4096 public key. Primary exfiltration was an HTTPS POST to `scan.aquasecurtiy[.]org` – a typosquatted domain resolving to 45.148.10.212 – with a fallback path that created a public GitHub repository named `tpcp-docs` in the victim's account to upload the encrypted payload as a release asset when the primary channel was unavailable [6]. Payload archives were consistently named `tpcp.tar.gz` across all phases of the campaign, a naming convention that has since become an indicator of compromise.

The final stage deployed a persistent backdoor: a Python loader written to `~/.config/sysmon.py` that slept five minutes after installation to defeat sandbox timeout heuristics, then polled an Internet Computer Protocol (ICP) blockchain canister (`tdtqy-oyaaa-aaaae-af2dq-cai.raw.icp0.io`) every approximately 50 minutes for second-stage payloads [7]. The choice of ICP blockchain infrastructure for command and control is significant: because the canister is hosted on a decentralized network without a traditional domain registrar or hosting provider relationship, conventional C2 takedown mechanisms – domain seizure, registrar notification, hosting provider abuse reports – are not applicable. Endpoint-based behavioral detection remains viable even when infrastructure takedown is not; the Python loader, its polling interval, and the canister URI pattern are all detectable through host-based monitoring. The backdoor also registered a `sysmon.service` systemd unit for persistence across reboots. After all stages completed, the legitimate Trivy scanner executed normally, presenting a fully successful pipeline run to developers and security teams monitoring CI/CD logs.

## Cascading Compromise Across Five Ecosystems

The credentials harvested from Trivy's GitHub Actions runners enabled TeamPCP to expand the campaign to four additional ecosystems within one week [8].

| Date | Target | Mechanism | Scope |
|---|---|---|---|
| March 19, 2026 | Trivy (Aqua Security) | Tag poisoning; malicious binary v0.69.4 | 76/77 trivy-action tags [2]; 7/7 setup-trivy tags |
| ~March 20–22 | npm (CanisterWorm) | Stolen npm publish tokens; ICP blockchain C2 | 47+ npm packages [9]; self-propagating worm |
| ~March 23 | Checkmarx KICS (`ast-github-action` v2.3.28) | Reused PATs; identical `setup.sh` stealer | C2: `checkmarx[.]zone` (83.142.209.11) [14] |
| March 24 | LiteLLM PyPI (v1.82.7, v1.82.8) | Compromised co-founder GitHub account; `litellm_init.pth` pth persistence | ~95M reported monthly downloads [8]; Kubernetes exploitation |
| March 27 | Telnyx PyPI (v4.87.1, v4.87.2) | Backdoored `telnyx/_client.py`; WAV file steganography for second-stage delivery | Millions of downloads [7][8] |

*Table scope figures sourced from [7][8][9][14] and package repository metadata. The npm package total reflects Datadog's enumeration [9]; MrCloudBook [7] reports 66+ packages, which may reflect later updated counts.*

The technical attribution linking these campaigns rests on a shared RSA-4096 public key embedded in the malicious payloads across Trivy, LiteLLM, and Telnyx [9]. This means that the private key holder — TeamPCP — is able to decrypt all credential bundles collected across all five campaign phases. The total exposure is accordingly much larger than any single component suggests. A separate Axios npm compromise identified on March 31 was initially attributed to TeamPCP but has since been re-attributed

to UNC1069, a suspected North Korean threat actor; the current assessment is that UNC1069 may have operated on credentials harvested and shared through TeamPCP's infrastructure rather than as a direct campaign extension [10].

## The European Commission Breach

The European Commission connection illustrates the compound nature of supply chain attacks: the most significant impact does not necessarily materialize at the point of initial compromise. On March 19, 2026, through the Trivy supply chain compromise, TeamPCP obtained AWS API keys with management rights over European Commission AWS accounts [11][12]. Using those credentials, the attackers deployed TruffleHog to scan for additional secrets and created new access keys on existing IAM users, a technique designed to establish persistent access without introducing new user accounts that might be detected by access reviews.

Between March 19 and March 24, TeamPCP conducted reconnaissance and data exfiltration from Commission infrastructure before the European Commission Cybersecurity Operations Centre detected anomalous API activity [12]. CERT-EU was notified on March 25, and the compromised credentials were revoked. Public disclosure by the European Commission followed on March 27. The stolen dataset — 91.7 GB compressed, approximately 340 GB uncompressed, including at least 51,992 email-related files totaling 2.22 GB, confidential documents, contracts, and database contents — was published on dark web forums by ShinyHunters on March 28. In total, 71 EU entities were affected: 42 internal Commission clients plus 29 other EU entities relying on the Commission's Europa web hosting service [11][12].

CERT-EU's April 3, 2026 advisory confirmed that initial access "occurred through the Trivy supply-chain compromise, publicly attributed to a threat actor known as TeamPCP," and found no evidence of lateral movement to other Commission AWS accounts or any website tampering [12]. While the technical footprint was contained once detected, the exfiltration window of approximately five days — from credential theft to detection — reflects the challenge of identifying supply chain compromise when attackers move quickly from initial access to data theft.

## Attack Surface and Risk Factors

Several structural characteristics of the Trivy ecosystem amplified the attack's reach. The use of mutable tags — resolved by name to whichever commit the tag currently points to — is a pipeline practice that trades convenience for integrity verification. Any consumer of `aquasecurity/trivy-action@0.24.0` or similar version-pinned references had no built-in indication that the commit backing that tag had changed, absent third-party SHA-pinning or monitoring tooling [3]. The `pull_request_target` misconfiguration that enabled the initial credential theft is a well-

documented pitfall specific to GitHub Actions; it is distinct from `pull_request` in that it runs with write repository permissions and access to secrets even for external pull requests, a distinction that is easy to overlook during workflow authoring and not enforced by GitHub by default [3].

The use of self-hosted runners within CI/CD pipelines substantially increases the blast radius of payload execution. GitHub-hosted runners are ephemeral and isolated, limiting the credential exposure to secrets present in that specific pipeline run. Self-hosted runners may accumulate credentials across many runs and may be machines with access to internal networks, cloud provider credentials, and developer filesystems — exactly the environment the TeamPCP payload was designed to exploit comprehensively [15].

---

# Recommendations

## Immediate Actions

Organizations that ran `aquasecurity/trivy-action` or `aquasecurity/setup-trivy` using version tags (rather than full commit SHAs) at any point between March 19 and approximately March 24, 2026, should treat their CI/CD environment as fully compromised. Immediate response steps include revoking and rotating all secrets accessible to those pipeline runs — cloud provider access keys, service account credentials, registry tokens, SSH keys, API keys — and auditing IAM activity logs from those accounts for anomalous activity, particularly new access key creation on existing IAM users and reconnaissance API calls such as TruffleHog-consistent patterns. Threat hunting should look for the consistent IOC patterns across this campaign: files named `tpcp.tar.gz`, the path `~/.config/sysmon.py`, the systemd unit `sysmon.service`, and Python `.pth` files named `litellm_init.pth` if LiteLLM was in use.

Network telemetry should be reviewed for outbound connections to `scan.aquasecurtiy[.]org` (45.148.10.212), `checkmarx[.]zone` (83.142.209.11), `models.litellm[.]cloud`, and DNS queries for `tdtqy-oyaaa-aaaae-af2dq-cai.raw.icp0.io`. The ICP blockchain C2 should be blocked at the network perimeter and flagged for DNS monitoring, as it cannot be taken down through conventional means and should be treated as a persistent indicator for the foreseeable future.

## Short-Term Mitigations

SHA pinning is among the most effective structural changes available for this threat class, as it cryptographically binds a pipeline reference to a specific commit and prevents silent tag redirection. A reference such as `uses: aquasecurity/trivy-action@sha-abc1234` cannot be silently redirected by tag force-push, because SHA references are cryptographically bound to a specific commit object. Tools including Dependabot, Renovate, and StepSecurity's Harden-Runner can automate SHA pinning and flag when referenced SHAs change [16]. Supplementing this with workflow-level permissions minimization — requesting only the specific permissions each job requires, avoiding `write-all` workflow permissions — limits the scope of any future credential compromise [3][13].

The `pull_request_target` trigger should be avoided in workflows that access secrets or have write permissions unless the security implications are fully understood and additional safeguards are in place, such as requiring manual approval for first-time contributors or using label-gated workflows. GitHub's own documentation on this trigger recommends treating code from external pull requests as untrusted regardless of the trigger used [3].

Organizations using LiteLLM v1.82.7 or v1.82.8 should audit Python environments for `litellm_init.pth` files across all machines and containers where those versions were installed; `.pth` files in Python's site-packages directory execute at interpreter startup, meaning the backdoor activates before any application code and persists across application restarts and updates.

## Strategic Considerations

This campaign demonstrates that security tooling is itself a high-value supply chain target. The properties that make a security scanner attractive — broad deployment, privileged execution, access to secrets during scans, presence in every pipeline — are the same properties that make it a force multiplier for attackers. Supply chain security programs should explicitly include the security tools themselves in their software composition analysis scope, not only the application dependencies those tools are used to scan.

The use of ICP blockchain infrastructure for command and control represents a meaningful operational shift. Traditional incident response relies on the ability to disrupt attacker infrastructure through domain seizure, IP blocking, and hosting provider engagement. A C2 channel hosted on decentralized blockchain infrastructure cannot be disrupted through these mechanisms; response must focus on endpoint detection and network-layer blocking rather than infrastructure takedown. Security teams should evaluate whether their monitoring and response playbooks account for this class of persistent infrastructure.

The Commission's experience — where the supply chain vector predated detection by approximately five days — illustrates how organizations without cross-correlation between CI/CD security events and cloud account monitoring may have limited visibility into downstream IAM activity until it manifests as cloud account anomalies. Detection coverage should span the supply chain vector through to the cloud and infrastructure impact.

---

## CSA Resource Alignment

The TeamPCP campaign surfaces risks and mitigations mapped directly to several CSA frameworks. Within the **AI Controls Matrix (AICM)**, this incident is relevant to the Supply Chain Security domain, which requires controls for the integrity of third-party components used in development and operational pipelines, and to the Infrastructure Security domain with respect to CI/CD pipeline hardening and secrets management. AICM's identity and access management controls speak directly to the IAM misuse observed in the European Commission breach — specifically the creation of new access keys on existing users to evade detection.

**MAESTRO** (the CSA Agentic AI Threat Modeling framework) identifies supply chain compromise as a primary threat vector for AI systems, and the LiteLLM compromise in this campaign is a direct instance. LiteLLM is widely used as the AI model routing layer in agentic workflows [9], reporting approximately 95 million monthly downloads [8], and backdooring it at the Python importer level represents an attack against the AI application layer via its infrastructure dependencies. MAESTRO's guidance on vetting AI infrastructure components before deployment is reinforced by this incident.

The **CSA STAR** (Security Trust Assurance and Risk) program provides a framework within which cloud-hosted organizations can assess and communicate supply chain security posture. EU entities using cloud-hosted services that rely on the affected tooling should review their STAR assessments to ensure that CI/CD pipeline integrity controls are explicitly scoped and that incident response procedures account for supply chain vectors.

The **CSA Zero Trust guidance** is applicable to the lateral movement phase of this campaign. The European Commission's IAM environment, once the initial credential was operationalized, allowed the attacker to create additional access keys and expand access — a pattern that Zero Trust approaches to cloud IAM, including just-in-time access, session-bound credentials, and anomaly-based access monitoring, are designed to interrupt.

# References


[1] Flare. "TeamPCP Cloud-Native Ransomware Profile." Flare Security, March 2026.

[2] CrowdStrike. "From Scanner to Stealer: Inside the Trivy Action Supply Chain Compromise." CrowdStrike Blog, March 2026.

[3] ARMO. "Trivy Supply Chain Attacks Explained: CI/CD Security Lessons." ARMO Security, March 2026.

[4] Legit Security. "The Trivy Supply Chain Compromise: What Happened and Playbooks to Respond." Legit Security, March 2026.

[5] Wiz. "Trivy Compromised by 'TeamPCP': A Supply Chain Attack Against DevSecOps." Wiz Blog, March 2026.

[6] GitGuardian. "Trivy's March Supply Chain Attack Shows Where Secret Exposure Hurts Most." GitGuardian Blog, March 2026.

[7] MrCloudBook. "TeamPCP Supply Chain Attack: Telnyx CanisterWorm Full Analysis (CVE-2026-33634)." MrCloudBook, March–April 2026.

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

[9] Datadog Security Labs. "LiteLLM and Telnyx Compromised: TeamPCP's PyPI Supply Chain Campaign." Datadog Security Labs, March 2026.

[10] SANS Internet Storm Center. "TeamPCP Supply Chain Campaign Update 005: First Confirmed Victim Disclosure, Post-Compromise Cloud Enumeration Documented, and Axios Attribution Narrows." SANS ISC Diary, April 2026.

[11] Help Net Security. "European Commission Cloud Breach: 71 EU Entities Affected via Trivy Supply Chain." Help Net Security, April 2026.

[12] CERT-EU. "European Commission Cloud Breach: Trivy Supply Chain." CERT-EU Advisory, April 3, 2026.

[13] Wiz. "Tracking TeamPCP: Post-Compromise Attacks Seen in the Wild." Wiz Blog, March 2026.


 

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

[15] Palo Alto Networks. "When Security Scanners Become the Weapon: The Trivy Supply Chain Attack." Unit 42 Blog, March 2026.

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

[17] Aqua Security. "Trivy Supply Chain Attack: What You Need to Know." Aqua Security Blog, March 2026.

[18] BleepingComputer. "CERT-EU: European Commission Hack Exposes Data of 30+ EU Entities." BleepingComputer, March 2026. *Note: The title figure of "30+ EU entities" refers to the external Union entities only; CERT-EU's advisory [12] confirms 71 total entities (42 internal Commission clients + 29 external).*