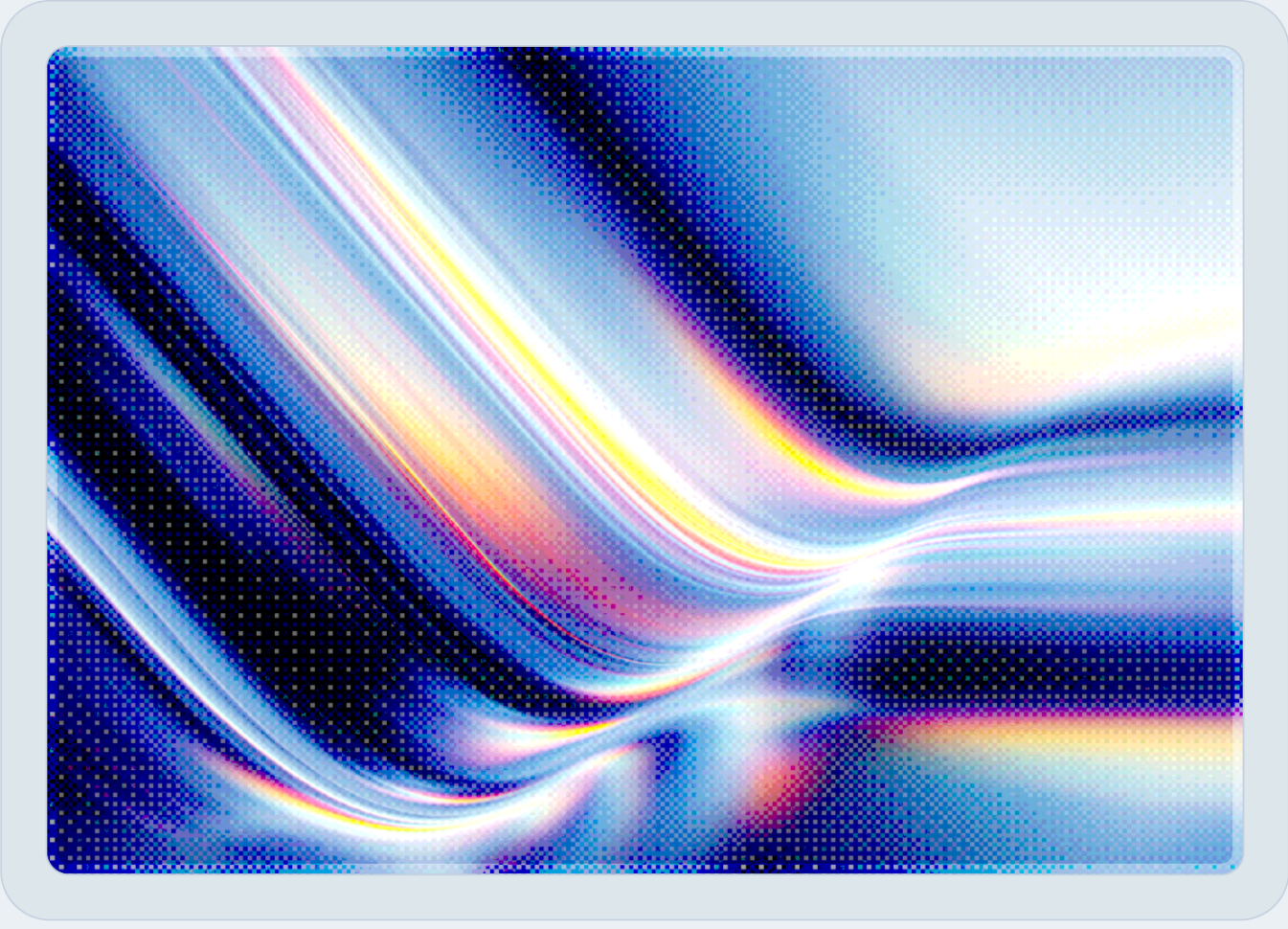
CSAI Foundation | Cloud Security Alliance

TeamPCP (UNC6780): AI Supply Chain's Most Active Threat Actor

Two Months of Cascading Compromises Across Developer Tooling and AI Infrastructure

2026-05-24

AI-assisted Rapid Research



CSAI | cloud security alliance®

Key Takeaways

- TeamPCP, formally designated UNC6780 by Google's Threat Intelligence Group, has executed at least three distinct supply chain campaign waves between March 19 and May 20, 2026, compromising security scanners, AI gateways, package managers, and — most recently — GitHub's own internal infrastructure. [1][2]
- The group's March campaign exploited mutable GitHub Actions version tags in Aqua Security's Trivy scanner (CVE-2026-33634, CVSS 9.4) to deploy the SANDCLOCK credential stealer across an estimated 10,000-plus CI/CD pipeline runs, subsequently cascading to Checkmarx KICS, LiteLLM, and the Telnyx Python SDK. [3][4][9]
- TeamPCP's CanisterWorm marked a significant C2 innovation: the first documented weaponization of an Internet Computer Protocol (ICP) blockchain canister as a resilient, decentralized command-and-control channel in a software supply chain attack, rendering conventional domain-based takedowns ineffective as a primary defensive response against this channel. [10][16]
- Stolen credentials from the March Trivy compromise were later used to breach Cisco's development environment, exfiltrating more than 300 internal repositories including source code for Cisco AI Assistant and AI Defense, demonstrating the multi-organization, cascading blast radius of a single supply chain entry point. [6]
- On May 18–20, 2026, TeamPCP breached GitHub's internal infrastructure via a trojanized Nx Console VS Code extension that was live on Visual Studio Marketplace for only 18 minutes, exfiltrating approximately 3,800 of GitHub's internal repositories. [7][8]
- AI-specific tooling — LLM gateways, model serving frameworks, and AI inference libraries — appears to be a primary target category for this actor rather than incidental collateral damage, based on the repeated, specific selection of AI-layer packages across all three campaign waves; AI operators should apply supply chain rigor equivalent to that applied to core infrastructure. [4][5][14]

Background

TeamPCP is a financially motivated threat actor whose campaigns since early 2026 have established it as the most prolific active adversary targeting the software development supply chain. Google's Threat Intelligence Group formally tracks the actor under the designation UNC6780; security vendors and incident responders have assigned additional aliases including DeadCatx3, PCPcat, ShellForce, and CanisterWorm to overlapping clusters of this activity. [1] The group's defining operational signature is indirect initial access: rather than attacking target organizations directly, TeamPCP compromises widely trusted open-source security and developer tooling that those organizations already run in their build environments, then harvests the elevated secrets that CI/CD pipelines must hold to function.

From the adversary's perspective, this methodology is economically rational and operationally efficient. A single compromised scanner or package manager can simultaneously yield credentials from thousands of downstream organizations without any of them being directly targeted. Those credentials — AWS keys, GitHub personal access tokens, Kubernetes configuration files, npm publishing tokens — are then monetized through criminal forum sales or through subsequent exploitation by ransomware and data extortion groups. [1][4] The group deliberately targets security scanning infrastructure first, exploiting the deep trust that organizations extend to tools that are granted broad access to source code and build secrets by design.

The AI development ecosystem has emerged as a specifically attractive target within TeamPCP's operational focus. Libraries and tools that serve as gateways to large language model APIs — notably LiteLLM, an open-source AI gateway with approximately 95 million monthly downloads at the time of its compromise — hold a category of credential that is both high-value and frequently under-protected relative to cloud infrastructure credentials. [5][14] LLM API keys provide access to paid inference capacity and, in enterprise deployments involving retrieval-augmented generation or agentic systems, can expose prompts, retrieved organizational data, and model outputs. TeamPCP's targeting of LiteLLM, the xinference LLM serving framework, and the Cisco AI Defense product line signals an adversary that has specifically mapped the AI infrastructure layer as a lucrative attack surface, not merely stumbled across AI tooling while pursuing generic developer credentials.

Security Analysis

The March Campaign: Trivy as a Beachhead (March 19–27, 2026)

The campaign's opening move demonstrated precise knowledge of GitHub Actions dependency chain mechanics. Aqua Security's Trivy vulnerability scanner is embedded in CI/CD pipelines across a broad population of organizations, from individual open-source projects to large enterprises. Its corresponding GitHub Action — `aquasecurity/trivy-action` — was referenced by downstream workflows using mutable version tags rather than pinned commit hashes, meaning that any attacker who could push a new commit and update those tags would instantly reach every pipeline that ran a Trivy scan on subsequent executions. [3][9]

TeamPCP gained access through a prior compromise that had not been fully remediated, then force-pushed malicious binaries beginning with Trivy v0.69.4 and simultaneously poisoned both `aquasecurity/trivy-action` and `setup-trivy`. [4] The injected payload was the SANDCLOCK credential stealer, a purpose-built tool that mimics legitimate telemetry traffic — disguising exfiltration as POST requests to domains resembling monitoring services — to evade network detection controls. [2] SANDCLOCK scraped secrets from CI runner memory and transmitted AWS keys, GitHub tokens, Kubernetes configurations, and SSH private keys to attacker-controlled infrastructure. SANS Institute estimated that more than 10,000 CI/CD workflow runs were directly exposed during the March 19–24 window before malicious artifacts were removed. [2]

The cascade from Trivy was rapid and crossed multiple ecosystems. Within days, TeamPCP had leveraged stolen credentials or parallel access techniques to compromise Checkmarx's KICS static analysis tool, two releases of LiteLLM on PyPI (versions 1.82.7 and 1.82.8), and the Telnyx Python SDK. [5] The LiteLLM compromise carried distinct consequences for AI operators beyond credential theft: the backdoored releases included not only a credential harvester but also a Kubernetes lateral movement toolkit and a persistent backdoor, giving the attacker ongoing access to the LLM-serving infrastructure of any organization that installed the affected packages without subsequently auditing its environment for persistence mechanisms. [5][14]

By March 20, the group had also deployed CanisterWorm: a self-propagating npm worm that spread across more than 47 npm packages spanning multiple publisher scopes. [10] The worm operated autonomously, stealing npm authentication tokens from compromised environments, resolving which packages each token could publish, incrementing patch version numbers to trigger routine update workflows, and republishing poisoned copies while preserving original READMEs to avoid raising maintainer suspicion. [10] Its command-and-control channel was hosted on an Internet Computer Protocol (ICP) canister — a compute unit on the Dfinity blockchain — which cannot be taken down

through registrar or hosting provider action, unlike conventional C2 infrastructure. Security researchers identified this as the first documented weaponization of decentralized blockchain infrastructure for command-and-control in a software supply chain attack. [16]

The Cisco Breach: Supply Chain as Force Multiplier

One major downstream consequence of the March Trivy compromise became public in April 2026, when Cisco confirmed that its internal development environment had been breached via credentials harvested during the March 19–24 window. [6] Attackers used stolen credentials to clone more than 300 of Cisco's internal GitHub repositories, exfiltrating source code for Cisco AI Assistant, Cisco AI Defense, and products that had not yet been publicly announced. [6] A portion of the exfiltrated repositories reportedly contained code belonging to Cisco's corporate customers, including financial institutions and U.S. government agencies. [6] The stolen data was subsequently offered for sale on criminal forums, with a portion independently claimed by the ShinyHunters group. [12]

The Cisco incident demonstrates the force-multiplier dynamic that supply chain entry points provide to adversaries. No Cisco employee was directly phished; no Cisco system was attacked by conventional means. The initial access came from a vulnerability scanner that a Cisco developer was running in a CI pipeline — weeks before any theft was confirmed. The organization that suffers the breach is not the organization that was compromised, and by the time downstream victims discover their exposure, credential rotation at the original entry point may not undo the data loss already sustained. For organizations that build or operate AI systems, the theft of Cisco AI Defense source code — a product designed to detect prompt injection and model misuse — carries additional strategic risk: adversaries with detailed knowledge of defensive AI system internals may be better positioned to engineer evasion techniques calibrated to bypass them.

The April Return: CanisterSprawl and Three Concurrent Compromises (April 21–22, 2026)

After a 26-day operational pause, TeamPCP resumed on April 21–22, 2026, executing three simultaneous package compromises across npm, PyPI, and Docker Hub. [2] Checkmarx's KICS container image was re-compromised, this time through an attacker authenticating to Docker Hub using valid Checkmarx publisher credentials likely obtained during or following the March campaign. Bitwarden's CLI npm package (`@bitwarden/cli@2026.4.0`) was briefly distributed with a malicious payload after a compromised GitHub Action in Bitwarden's CI/CD pipeline injected code into the npm packaging step, with the malicious version available for approximately 90 minutes. [11][19] Simultaneously, the xinference PyPI package — a framework for running open-source LLMs in research and production environments —

was poisoned with an expanded credential harvester that added cryptocurrency wallet data (MetaMask, Phantom, and Solana wallet files) to its exfiltration targets alongside standard developer credentials, suggesting the group's targeting intelligence on AI-ecosystem participants is becoming increasingly granular. [2]

The worm component evolved as well. Security researchers Socket and StepSecurity identified a new variant, CanisterSprawl, that retained the ICP blockchain C2 architecture while adding cross-ecosystem propagation logic. [2] If the infected environment contained PyPI publish credentials, CanisterSprawl would generate Python `.pth` file payloads — entries in the `site-packages` directory that execute automatically at Python interpreter startup — to infect Python packages via a mechanism the target organization might not associate with an npm-channel infection. A single compromise in the npm ecosystem can therefore produce PyPI propagation without any additional targeting decision by the attacker, compounding the blast radius significantly.

The GitHub Breach: Developer Trust Surface Exploited (May 18–20, 2026)

GitHub, as a central dependency of the global software supply chain, represents the highest-profile target in the TeamPCP campaign series; this breach also exploited an attack surface that merits considerably more defensive attention: the VS Code extension marketplace. On May 18, 2026, a trojanized build of Nx Console — a widely used Angular and React project management extension with approximately 2.2 million installs and a verified publisher badge — was live on Visual Studio Marketplace for 18 minutes, between 12:30 and 12:48 UTC. [7][8] A GitHub employee installed a routine extension update during that window. The trojanized extension harvested developer secrets and access tokens from the IDE's local environment, providing TeamPCP with credentials that proved sufficient to exfiltrate approximately 3,800 of GitHub's internal source code repositories. [7][8]

GitHub assessed that customer repositories, enterprise accounts, and user data were not affected by the compromise; the breach appears confined to GitHub's corporate estate. [7] The stolen data was subsequently listed for sale on a criminal forum at upwards of $50,000 USD. [18] The incident underscores how consequential the VS Code extension attack surface can be: extensions are routinely granted access to the local development environment, including file systems, terminal sessions, and stored credentials. The 18-minute exposure window demonstrates that even transient marketplace availability of a weaponized extension can produce significant, persistent access, because the harm from installation outlasts the window during which installation was possible.

AI Infrastructure as a Primary Target

Across all three campaign waves, TeamPCP has specifically and repeatedly targeted the tooling layer of AI development. LiteLLM serves as the API gateway layer for organizations running multi-provider LLM integrations; its compromise provides direct access to LLM API keys and can expose the prompts, retrieved documents, and model outputs flowing through those keys in real time. Xinference is used to serve open-source LLMs in research and production contexts; the expanded credential harvester in its compromised version suggests TeamPCP has profiled the types of credentials that AI developers and operators hold. The Cisco AI Defense theft is specifically concerning for AI security practitioners because it represents potential adversarial acquisition of detailed knowledge about an enterprise AI security product's detection capabilities. The pattern suggests deliberate adversarial attention to the AI infrastructure layer as a target category with distinct credential and intelligence value, rather than coincidental inclusion in a broader developer tooling sweep.

Recommendations

Immediate Actions

Security teams should immediately audit CI/CD pipeline configurations for mutable version tag references across GitHub Actions, Docker Hub images, npm packages, and PyPI dependencies. Any `@latest`, `@v2`, or floating tag reference in a workflow or Dockerfile represents a dependency whose content a supply chain attacker may have already replaced. Pin all references to immutable commit SHAs or content-addressed digests, which cannot be retroactively modified after pinning. Organizations that ran any of the affected versions of Trivy (v0.69.4–v0.69.6), LiteLLM (1.82.7 or 1.82.8), or `@bitwarden/cli@2026.4.0` during the relevant compromise windows should treat all secrets accessible from those build environments as compromised and initiate full credential rotation regardless of whether direct intrusion activity has been confirmed. [4][9]

For VS Code deployments across development teams, confirm that installed extensions match known-good hashes and inventory any extension installed or auto-updated on May 18, 2026 in the 12:30–12:48 UTC window. Security teams with endpoint telemetry should review outbound connections from developer workstations for POST requests to endpoints that mimic monitoring service domains but resolve to unfamiliar infrastructure, as SANDCLOCK specifically uses this evasion pattern. [2] Python environments should be audited for unexpected `.pth` files in `site-packages` directories; the

presence of `litellm_init.pth` or any `.pth` file not placed by a known package installation should be treated as a strong indicator of CanisterSprawl infection requiring immediate isolation and forensic review.

Short-Term Mitigations

Build environments should enforce minimum-privilege token scoping using short-lived OIDC-based workload identity federation rather than long-lived personal access tokens or static API keys wherever the CI/CD platform supports it. [9][13] Every secret that a CI runner holds — cloud credentials, container registry tokens, package registry publish credentials — should be scoped to the minimum necessary permission and rotated on a schedule shorter than the adversary's observed dwell window. The six-week gap between the March Trivy compromise and the Cisco breach demonstrates that harvested credentials may be held and acted upon well after the initial collection window; rotation cadences calibrated to convenience rather than threat tempo will not contain this pattern.

VS Code extension governance should receive the same structured treatment applied to npm or PyPI dependencies. Organizations should enumerate all VS Code extensions installed across development workstations, establish a reviewed allowlist, and consider policy controls that restrict installation to audited extensions from verified publishers. The Nx Console incident makes clear that a "Verified Publisher" badge indicates only that Microsoft has validated the publisher's account, not that each individual extension release has been inspected; a policy requiring internal approval before extensions are updated on developer workstations provides a meaningful additional control.

AI-specific package management deserves explicit governance attention. LLM API keys should be treated as high-value credentials subject to per-environment isolation, per-request audit logging, and the same rotation discipline applied to cloud root credentials — not as configuration values stored in `.env` files committed alongside application code. Organizations running open-source LLM serving frameworks should monitor the PyPI and Docker Hub release channels for those frameworks and establish a policy for controlled version advancement rather than automatic latest-version installation.

Strategic Considerations

The TeamPCP campaign sequence reveals a deliberate operational pattern: the group acquires initial access through compromised open-source tooling, harvests credentials across many downstream organizations simultaneously, and then monetizes selectively by identifying and targeting the highest-value organizations whose credentials were captured. The sequencing of events is inconsistent with an opportunistic spray-and-pray operation; both the Cisco and GitHub breaches required TeamPCP to identify the most valuable credentials from a large harvested pool and execute targeted intrusions weeks

after the initial collection event, suggesting deliberate selection rather than automated bulk exploitation. Defenders should assume that any credential exposure event affecting widely used open-source tooling carries a longer tail of downstream consequences than immediate incident response will surface. An organization may not learn of its exposure until a criminal forum lists its source code for sale.

The group's consistent use of blockchain-hosted C2 infrastructure raises a broader defensive consideration. Traditional incident response playbooks for supply chain attacks rely heavily on C2 domain blocking and infrastructure takedowns to limit the adversary's ongoing access. Against an actor using ICP canisters as C2 channels, traditional infrastructure takedowns are insufficient as a primary defensive response — they do not prevent the C2 channel from functioning even after discovery. Detection and response strategies must shift toward behavioral indicators — abnormal CI runner egress, unexpected process execution patterns, anomalous credential use — that do not depend on blocking known C2 infrastructure. Defenders who have not yet reviewed their logging and detection coverage for SANDCLOCK-style telemetry-mimicking egress should treat that gap as a priority.

CSA Resource Alignment

The TeamPCP campaign implicates several areas of CSA guidance. CSA's AI Incident and Vulnerability Management practices address the need to maintain software and AI bills of materials (SBOMs and AI-BOMs) covering model dependencies, inference libraries, and API gateway components; the LiteLLM and xinference compromises confirm that AI-layer dependencies must be included in this inventory and subjected to the same integrity monitoring applied to infrastructure packages. [15] Organizations operating under CSA's AI Controls Matrix (AICM) — which supersedes the CCM for AI workloads — should note that supply chain integrity controls map to the AICM's Development and Operations (DEV) and Infrastructure (INF) control domains, and that the TeamPCP incidents represent failures in mutable dependency reference management, developer endpoint security, and CI/CD token scoping that those control domains are designed to address.

CSA's MAESTRO threat model for agentic AI systems identifies infrastructure layer integrity — including the integrity of tools and services that AI agents invoke at runtime — as a first-class threat category. An AI agent that routes requests through LiteLLM, or that is built by a developer whose IDE was compromised by a trojanized extension, faces integrity risks that are not captured by model-layer threat models alone. The MAESTRO framework's treatment of trust boundaries between AI orchestration layers and the underlying software stack applies directly to organizations assessing their exposure from the TeamPCP incident series. The CI/CD pipeline is not outside the threat model for agentic AI systems; it is where those systems and their secrets are assembled.

CISA added CVE-2026-33634 (the Trivy mutable-tag exploitation) to its Known Exploited Vulnerabilities catalog following the March 2026 campaign. [13][17] Organizations using CSA's STAR self-assessment program to communicate supply chain posture to customers and partners should review whether their CI/CD governance controls, dependency pinning practices, and developer endpoint security policies accurately reflect their current state, given that the TeamPCP incidents exposed systemic gaps in these areas across a broad and varied population of organizations.

References

[1] SANS Internet Storm Center. "TeamPCP Supply Chain Campaign: Update 007 – Cisco Source Code Stolen via Trivy-Linked Breach, Google GTIG Tracks TeamPCP as UNC6780, and CISA KEV Deadline Arrives." SANS ISC, April 2026.

[2] SANS Internet Storm Center. "TeamPCP Supply Chain Campaign: Update 008 – 26-Day Pause Ends with Three Concurrent Compromises, CanisterSprawl npm Worm Identified." SANS ISC, April 27, 2026.

[3] NIST National Vulnerability Database. "CVE-2026-33634 Detail." NVD, March 2026.

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

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

[6] BleepingComputer. "Cisco Source Code Stolen in Trivy-Linked Dev Environment Breach." BleepingComputer, April 2026.

[7] Help Net Security. "TeamPCP Breached GitHub's Internal Codebase via Poisoned VS Code Extension." Help Net Security, May 20, 2026.

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

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

[10] SC Media. "Namastex npm Packages Compromised in 'CanisterWorm' Supply Chain Attack." SC Media, March 2026.

[11] Socket.dev. "Bitwarden CLI Compromised in Ongoing Checkmarx Supply Chain Campaign." Socket.dev, April 2026.

[12] SOCRadar. "Trivy-Linked Cisco Breach & ShinyHunters' Stolen Data Claim." SOCRadar, April 2026.

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

[14] Endor Labs. "TeamPCP Isn't Done: Threat Actor Behind Trivy and KICS Compromises Now Hits Lite LLM's 95 Million Monthly Downloads on PyPI." Endor Labs, 2026.

[15] Cloud Security Alliance. "TeamPCP: Cascading Supply Chain Attack on AI/ML Tooling." CSA Lab Space, March 30, 2026.

[16] Cloud Security Alliance. "CanisterWorm: Blockchain C2 in CI/CD Supply Chain Attack." CSA Lab Space, March 2026.

[17] Help Net Security. "CISA Sounds Alarm on Langflow RCE, Trivy Supply Chain Compromise After Rapid Exploitation." Help Net Security, March 27, 2026.

[18] Tom's Hardware. "Hacker Group Hits 3,800 Internal GitHub Repositories via Poisoned Developer Plugin — TeamPCP Claims Source Code Theft and Attempts $50,000 Sale." Tom's Hardware, May 2026.

[19] SecurityWeek. "Bitwarden npm Package Hit in Supply Chain Attack." SecurityWeek, April 2026.