INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-15 06:40 UTC

TeamPCP Turns Security Tooling Against Defenders: Active Supply Chain Campaign Expands Across Ecosystems

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0462
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Security scanners and CI/CD pipeline infrastructure, specific vendor products not identified in available source material
Published	2026-06-08T13:07:37
Discovery Source	Rss

Executive Summary

TeamPCP is an active threat campaign, documented by SANS researchers, that weaponizes the security scanning tools and CI/CD pipeline infrastructure organizations rely on for defense. By compromising the tooling itself, attackers gain access to the software supply chain at build time, meaning malicious code can reach production systems before defenders have any opportunity to detect it. Organizations that run automated vulnerability scanners or build pipelines face direct risk of supply chain compromise, code injection, and credential theft across any system those pipelines touch.

Technical Analysis

TeamPCP exploits security scanners and CI/CD infrastructure as primary attack vectors, inverting the typical threat model by targeting defensive tooling rather than production targets directly. Confirmed CWEs include CWE-494 (Download of Code Without Integrity Check), CWE-829 (Inclusion of Functionality from Untrusted Control Sphere), CWE-798 (Use of Hard-coded Credentials), and CWE-284 (Improper Access Control). Mapped MITRE ATT&CK techniques include T1195/T1195.001/T1195.002 (Supply Chain Compromise), T1554 (Compromise Client Software Binary), T1072 (Software Deployment Tools), T1609/T1610 (Container execution abuse), T1059 (Command and Scripting Interpreter), T1552.001 (Credentials in Files), and T1588.001 (Obtain Capabilities: Malware). No CVE identifiers are assigned. Specific affected scanner products and CI/CD platforms are not identified in available source material; the primary reference is a SANS white paper. The campaign was confirmed active through at least mid-2026, with documented expansion to new targets and

ecosystems. No patch or vendor advisory is available from current source material; full technical detail requires direct access to the SANS publication.

Action Checklist

1. Step 1: Containment, Audit all security scanner integrations and CI/CD pipeline service accounts immediately. Isolate any pipeline that pulls dependencies or scanner plugins from external registries without integrity verification. Map which pipelines have elevated permissions or write access to production artifact stores. Reference: NIST AC-4 (Information Flow Enforcement), CIS 4.6 (Securely Manage Enterprise Assets and Software).
2. Step 2: Detection, Query pipeline execution logs for unsigned or unverified code downloads (CWE-494 pattern: outbound fetch at build time from unexpected domains). Search CI/CD system logs for new or modified runner configurations, injected environment variables, and unexpected script execution (MITRE T1059, T1554). Review credential stores and environment variable files for hard-coded secrets (CWE-798 pattern). D3FEND: D3-SFA (System File Analysis) on pipeline configuration files and runner init configs; D3-SICA (System Init Config Analysis) for modified startup behavior in scanner agents.
3. Step 3: Eradication, Remove or revoke any credentials exposed in pipeline environment variables or scanner configuration files (CWE-798). Enforce dependency integrity checks (checksums, signed artifacts) for all packages pulled during build (CWE-494 remediation). Restrict scanner and CI/CD service account permissions to least privilege; remove standing write access to production systems (NIST AC-6, CIS 5.4). Replace any flagged pipeline components with verified, clean versions from authoritative sources. Reference: NIST CM controls for configuration baseline restoration.
4. Step 4: Recovery, Validate all production artifacts built during the suspected exposure window by comparing against known-good hashes. Re-run pipelines from a clean, verified state with integrity checks enforced before promoting any artifact. Monitor pipeline execution logs continuously for re-emergence of unexpected external calls or config changes (NIST AU-6, CIS 8.2). Rotate all service account credentials and API tokens used by affected scanners and runners (D3-CRO: Credential Rotation).
5. Step 5: Post-Incident, Conduct a gap assessment against NIST AC-6 (Least Privilege) and AC-4 (Information Flow Enforcement) for all pipeline service accounts. Implement code signing and integrity verification for all scanner plugins and build dependencies (CWE-494, CWE-829 control gap). Establish a software bill of materials (SBOM) process for pipeline tooling. Review third-party scanner vendor security posture; this campaign exposes the risk of implicitly trusting security tooling. Current NIST AC and CIS frameworks do not include a control family dedicated to scanner/tooling weaponization; recommend escalating to GRC to develop a defensive supply chain policy covering security tooling procurement, configuration, and monitoring.

Detection Guidance

Focus detection on CI/CD pipeline execution logs and scanner agent behavior. Key behavioral indicators: (1) outbound network calls from build runners to domains not in an approved allowlist, particularly during dependency resolution or scanner plugin updates, CWE-494/T1195 pattern; (2) new or modified runner configuration files, init scripts, or environment variable overrides not traceable to an approved commit, T1554/T1609 pattern (D3-SICA, D3-SFA); (3) hard-coded credentials or tokens appearing in pipeline logs, artifact metadata, or environment variable dumps, CWE-798/T1552.001; (4) unexpected container spawning or

privileged execution within pipeline stages, T1610; (5) scanner agent processes initiating script execution outside their documented scan scope, T1059. Log sources to prioritize: CI/CD platform audit logs (GitHub Actions, GitLab CI, Jenkins, etc.), container runtime logs, artifact registry access logs, and network egress logs from build infrastructure. Alert on any pipeline stage that fetches and executes code without a verified checksum. D3FEND countermeasures to implement: D3-SFA (System File Analysis) on pipeline config files; D3-SICA (System Init Config Analysis) on scanner agent startup; D3-LAM (Local Account Monitoring) on CI/CD service accounts; D3-UAP (User Account Permissions) audit for pipeline roles. Full IOC disclosure (IPs, domains, file hashes) requires direct access to the SANS white paper; behavioral and pattern-based detection guidance is provided here to enable immediate defensive action.

Framework Mappings

MITRE-ATTACK

- **T1552.001** — Credentials In Files
- **T1059** — Command and Scripting Interpreter
- **T1588.001** — Malware
- **T1610** — Deploy Container
- **T1609** — Container Administration Command
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1195** — Supply Chain Compromise
- **T1554** — Compromise Host Software Binary
- **T1195.002** — Compromise Software Supply Chain
- **T1072** — Software Deployment Tools

NIST-800-53R5

- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **SA-9** — External System Services
- **SR-2** — Supply Chain Risk Management Plan
- **SR-3** — Supply Chain Controls and Processes
- **IA-5** — Authenticator Management
- **CM-3** — Configuration Change Control
- **AC-3** — Access Enforcement

OWASP-TOP10-2021

- **A07:2021** — Identification and Authentication Failures
- **A08:2021** — Software and Data Integrity Failures
- **A01:2021** — Broken Access Control

CIS-V8

- **16.10** — Apply Secure Design Principles in Application Architectures
- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **6.1** — Establish an Access Granting Process
- **6.2** — Establish an Access Revoking Process
- **6.3** — Require MFA for Externally-Exposed Applications
- **15.1** — Establish and Maintain an Inventory of Service Providers
- **8.2** — Collect Audit Logs

ISO-27001-2022

- **A.8.28** — Secure coding
- **A.8.8** — Management of technical vulnerabilities
- **A.5.21** — Managing information security in the ICT supply chain

SOC2-TSC

- **CC6.1** — The entity implements logical access security software, infrastructure, and architectures over protected information assets
- **CC9.2** — Manages risks associated with vendors and business partners

HIPAA-SECURITY

- **164.312(a)(1)** — Access Control
- **164.312(d)** — Person or Entity Authentication

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program
- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1552.001	Credentials In Files	Credential-Access
T1059	Command and Scripting Interpreter	Execution
T1588.001	Malware	Resource-Development
T1610	Deploy Container	Defense-Evasion
T1609	Container Administration Command	Execution
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1195	Supply Chain Compromise	Initial-Access

Technique ID	Technique Name	Tactic
T1554	Compromise Host Software Binary	Persistence
T1195.002	Compromise Software Supply Chain	Initial-Access
T1072	Software Deployment Tools	Execution

Sources

Source	URL	Tier
Security News	https://www.sans.org/white-papers/when-security-scanner-became-weapon	T3
What is CI/CD Vulnerability Scanning? - VikingCloud	https://www.vikingcloud.com/blog/what-is-ci-cd-vulnerability-scanning	T3
Best Vulnerability Scanning Tools for CI/CD Pipelines: 8 Platforms ...	https://www.kiuwan.com/blog/vulnerability-scanning-tools/	T3
How to implement CI/CD security scanning: Best practices - Wiz	https://www.wiz.io/academy/application-security/ci-cd-security-scan...	T3
Integrating Vulnerability Scanning with Continuous ... - ResearchGate	https://www.researchgate.net/publication/383414498_Integrating_Vuln...	T3

Generated 2026-06-15 06:40 UTC by TJS Security Command Center