INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-03 13:38 UTC

TeamPCP Supply Chain Attacks Draw ShinyHunters and Lapsus$ Into Attribution Dispute, Complicating Enterprise Breach Scoping

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0143
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Not specified in available source data, enterprise software supply chain targets implied; specific vendors/versions not confirmed from accessible content
Published	2026-04-03T11:11:16
Discovery Source	Rss

Executive Summary

TeamPCP is conducting an active supply chain attack campaign in which malicious code is embedded into or delivered through software components that enterprise organizations trust and consume. ShinyHunters and Lapsus$ have each publicly claimed involvement in overlapping breaches, creating an attribution dispute that prevents defenders from applying a single indicator set across affected environments. The business risk is two-fold: organizations may be scoping breach investigations against the wrong actor's known tooling, and the expanding victim footprint suggests ongoing compromise of additional targets. Note: Source coverage for this campaign is limited to a single secondary-tier publication. All technical claims require independent verification before operational deployment.

Technical Analysis

This campaign aligns with supply chain compromise tradecraft across three primary weakness categories: CWE-506 (Embedded Malicious Code), CWE-494 (Download of Code Without Integrity Check), and CWE-829 (Inclusion of Functionality from Untrusted Control Sphere). No CVE is associated with this campaign item; the attack surface is procedural and trust-based rather than a discrete software vulnerability. MITRE ATT&CK coverage spans the full compromise lifecycle: T1195.002 (Compromise Software Supply Chain), T1554 (Compromise Client Software Binary), T1078 (Valid Accounts), T1036 (Masquerading), T1584/T1583

(Compromise/Acquire Infrastructure), T1588.002 (Tool acquisition), and T1486 (Data Encrypted for Impact). The multi-actor attribution dispute between ShinyHunters and Lapsus$ has direct operational consequences: each group maintains distinct infrastructure, tooling preferences, and persistence mechanisms. Defenders cannot assume a single indicator set covers all implants present. Confidence in specific technical details is LOW; source material for this campaign originated from a single secondary-tier publication (Dark Reading) and was not independently verified during processing. Treat all technical specifics as requiring independent validation before operational use.

Action Checklist

1. Step 1: Containment, Identify all third-party software components, packages, and update mechanisms active in your environment. Isolate any recently updated components from vendors or repositories that cannot confirm their build pipeline integrity. Do not wait for vendor disclosure; initiate outreach to critical software suppliers now. Priority: systems with automated update ingestion and CI/CD pipelines pulling external dependencies.
2. Step 2: Detection, Query EDR and SIEM for indicators consistent with T1195.002 and T1554: unexpected binary modifications to trusted software, new scheduled tasks or services created by software installer processes, outbound connections initiated by software update agents to non-vendor infrastructure, and execution of code from temp or staging directories post-update. Cross-reference against MITRE ATT&CK T1036 (Masquerading) patterns: process names mimicking legitimate software with mismatched parent processes or unusual execution paths. Note: no confirmed IOCs are available from verified source material for this campaign. Do not treat the IOC section below as operationally confirmed.
3. Step 3: Eradication, For any confirmed or suspected compromise, remove the affected component and roll back to a known-good version verified through a hash-validated source. Revoke and reissue credentials (T1078) associated with any system where the compromised component had access. Review and rotate API keys, service account tokens, and OAuth grants used by affected software. Confirm software signing and integrity verification is enforced across all package managers and update channels.
4. Step 4: Recovery, Validate remediation by re-running integrity checks against all previously flagged binaries. Monitor for re-infection patterns: watch for the same process lineages and network destinations that triggered initial detection. Confirm that no persistence mechanisms (scheduled tasks, registry run keys, startup services) were introduced by the malicious component before declaring recovery. Retain forensic copies of affected systems for post-incident analysis.
5. Step 5: Post-Incident, Conduct a software bill of materials (SBOM) gap assessment: identify which components in production lack integrity verification. Evaluate CI/CD pipeline controls against NIST SP 800-218 (Secure Software Development Framework) and CISA's Secure by Design guidance. Implement or validate code signing verification at ingestion, not just at build time. Document which actor's TTPs were confirmed present to assist with future attribution scoping if this campaign resurfaces.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate immediately to senior IR leadership, legal counsel, and executive management if forensic evidence confirms data exfiltration from systems where the compromised component had access to PII, PHI, credentials, or source code — ShinyHunters' confirmed operational history of selling exfiltrated datasets and Lapsus$'s pattern of public data leaks create breach notification obligations under GDPR, CCPA, HIPAA, and SEC disclosure rules that require time-bounded legal review independent of technical remediation status.
Recovery Notes	Before returning any affected system to production, verify that all software components pulled from external registries during the incident window have been re-validated against vendor-published hashes from out-of-band sources (direct vendor contact or official release pages), not from the potentially compromised repository or pipeline. Maintain elevated monitoring on process lineages and outbound network connections from software update agents for a minimum of 30 days post-recovery, given Lapsus$'s documented practice of re-establishing access through dormant credentials or persistence mechanisms that survive initial eradication. If the SBOM gap assessment reveals that components lacking integrity verification remain in production due to operational constraints, treat those systems as conditionally recovered and document the residual risk with a formal risk acceptance and compensating control assignment.
Forensic Artifacts	Package manager install/update logs ('var/log/dpkg.log', '/var/log/yum.log', npm debug logs at '~/.npm/_logs/', pip install logs) with timestamps cross-referenced against the suspected compromise window — these establish the initial access artifact for T1195.002 (Supply Chain Compromise: Compromise Software Supply Chain) | Sysmon Event ID 7 (ImageLoad) logs showing DLLs or shared libraries loaded from non-standard or temp paths by processes that are children of software installer or update agents — primary forensic indicator of T1554 (Compromise Host Software Binary) | Windows Security Event ID 4698 (Scheduled Task Created) and registry snapshot of 'HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\' filtered to tasks created within 24 hours of any software update event — persistence artifact consistent with both ShinyHunters and Lapsus$ post-exploitation tooling | Identity provider authentication logs (Azure AD sign-in logs, Okta System Log, AWS CloudTrail 'AssumeRole' events) filtered on service accounts associated with the compromised software components, searched for anomalous token grants, impossible travel, or API calls to data exfiltration-relevant services (S3 GetObject, SharePoint bulk download, GitHub repository clone) — credential abuse artifact consistent with T1078 (Valid Accounts) and ShinyHunters' documented exfiltration methodology | CI/CD pipeline execution logs from the build and deployment system (GitHub Actions workflow run logs, Jenkins build console output, GitLab CI job traces) showing which external package registry endpoints were contacted, what artifact hashes were downloaded, and whether any hash verification steps were bypassed or absent — this is the supply chain entry point artifact and the primary evidence source for NIST SP 800-218 gap assessment

Per-Action IR Details

Step 1: Containment — Identify all third-party software components, packages, and update mechanisms active in your environment. Isolate any recently updated components from vendors or repositories that cannot confirm their build pipeline integrity. Do not wait for vendor disclosure; initiate outreach to critical software suppliers now. Priority: systems with automated update ingestion and CI/CD pipelines pulling external dependencies.

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy: isolate affected systems and prevent lateral spread while preserving evidence; for supply chain campaigns, containment scope is the software distribution surface, not individual hosts

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST CM-3 (Configuration Change Control) — prevent unvetted updates from entering the environment during active investigation, CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 2.3 (Address Unauthorized Software), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Without a CMDB or enterprise asset tool, run 'pip list', 'npm list --depth=0', 'gem list', and 'dpkg -l' (Linux) or 'Get-Package | Export-Csv' (PowerShell) on each host to enumerate installed packages and versions. Freeze automated updates immediately by disabling cron jobs referencing package managers ('crontab -l | grep -E apt|yum|pip|npm') and pausing CI/CD pipeline runs that pull external registries. For CI/CD pipeline freeze on GitHub Actions: set environment protection rules to require manual approval before workflow execution.

Evidence: Before isolating any component, capture: (1) package manager logs showing what was installed or updated and when — '/var/log/dpkg.log', '/var/log/yum.log', npm audit logs at '~/.npm/_logs/', pip install history via 'pip show '; (2) file system timestamps on recently modified binaries in install directories (e.g., '/usr/local/lib/', 'C:\Program Files\'); (3) network connection logs from software update agents captured via 'netstat -anb' or Sysmon Event ID 3 (Network Connection) filtered on processes associated with package managers or software updaters; (4) CI/CD pipeline execution logs showing which external registries (PyPI, npm, GitHub packages) were contacted and what artifact hashes were downloaded.

Step 2: Detection — Query EDR and SIEM for indicators consistent with T1195.002 and T1554: unexpected binary modifications to trusted software, new scheduled tasks or services created by software installer processes, outbound connections initiated by software update agents to non-vendor infrastructure, and execution of code from temp or staging directories post-update. Cross-reference against MITRE ATT&CK T1036 (Masquerading) patterns: process names mimicking legitimate software with mismatched parent processes or unusual execution paths. Note: no confirmed IOCs are available from verified source material for this campaign. Do not treat the IOC section below as operationally confirmed.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis: correlate behavioral indicators across log sources; when IOCs are unavailable or disputed (as in the TeamPCP/ShinyHunters/Lapsus$ attribution conflict), pivot to TTP-based detection using ATT&CK technique patterns rather than hash or IP blocklists

Controls: NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-12 (Audit Record Generation), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs), MITRE ATT&CK T1195.002 (Supply Chain Compromise: Compromise Software Supply Chain), MITRE ATT&CK T1554 (Compromise Host Software Binary), MITRE ATT&CK T1036 (Masquerading)

Compensating: Deploy Sysmon with SwiftOnSecurity's config (github.com/SwiftOnSecurity/sysmon-config) and hunt on: Event ID 11 (FileCreate) in temp/staging paths post-installer execution; Event ID 1 (Process Create) where parent is a package manager (msiexec.exe, python.exe, node.exe) and child is cmd.exe, powershell.exe, or wscript.exe; Event ID 3 (Network Connection) where the initiating process is a software update agent connecting to non-RFC-1918 IPs not matching the vendor's published update endpoint. Use the Sigma rule 'proc_creation_win_susp_parent_process.yml' (SigmaHQ repository) to detect masquerading via mismatched parent-child process pairs. For Linux, use auditd rules: '-a always,exit -F arch=b64 -S execve -F uid=0 -k supply_chain_exec' and review '/var/log/audit/audit.log' for post-install execution chains.

Evidence: Capture before beginning query sweeps: (1) Sysmon Event ID 7 (ImageLoad) logs showing DLLs loaded from non-standard paths by trusted software processes — this is the primary artifact for T1554 binary modification; (2) Windows Security Event Log Event ID 4688 (Process Creation) with full command-line logging enabled, filtered on installer parent processes (msiexec, setup.exe, npm.cmd) spawning shells; (3) Scheduled task creation artifacts — Windows Security Event ID 4698 (Scheduled Task Created) and registry key 'HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Schedule\TaskCache\Tasks\' for tasks created within the window of software updates; (4) Prefetch files in 'C:\Windows\Prefetch\' for evidence of binaries executed from temp directories post-update; (5) DNS query logs (Windows DNS debug log or Sysmon Event ID 22) for domains queried by software update processes that do not resolve to vendor-owned infrastructure.

Step 3: Eradication — For any confirmed or suspected compromise, remove the affected component and roll back to a known-good version verified through a hash-validated source. Revoke and reissue credentials

(T1078) associated with any system where the compromised component had access. Review and rotate API keys, service account tokens, and OAuth grants used by affected software. Confirm software signing and integrity verification is enforced across all package managers and update channels.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication: remove all components of the threat from the environment; in supply chain attacks, eradication scope must include credential material and API tokens that the malicious component may have exfiltrated, not just the binary artifact itself

Controls: NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST IA-5 (Authenticator Management) — rotate all credentials accessible to or used by the compromised component, NIST AC-2 (Account Management) — audit service accounts and OAuth grants tied to affected software, CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 7.2 (Establish and Maintain a Remediation Process), MITRE ATT&CK T1078 (Valid Accounts) — credential abuse is a primary post-exploitation path for both ShinyHunters and Lapsus$

Compensating: Verify rollback binary integrity using 'certutil -hashfile SHA256' (Windows) or 'sha256sum ' (Linux) against the vendor's published hash from their official release page — do not trust hashes published in the same repository or pipeline that may have been compromised. For credential rotation without a PAM tool: use 'net user /domain' for AD service accounts and document in a change record; for GitHub tokens, revoke via the GitHub Security Settings API ('gh auth token' revoke); for AWS keys, use 'aws iam delete-access-key --access-key-id ' followed by 'aws iam create-access-key'. Run 'Get-ScheduledTask | Where-Object {$_.TaskPath -notlike "\Microsoft*"} | Export-Csv' to enumerate and review non-Microsoft scheduled tasks introduced during the compromise window.

Evidence: Before removing any component: (1) Collect a full memory image using WinPmem or LiME to capture in-memory artifacts of the malicious component that will not survive disk removal — particularly relevant if ShinyHunters/Lapsus$ tooling included in-memory credential scrapers; (2) Export Windows Credential Manager vault ('cmdkey /list') and LSA secrets snapshot (requires offline registry analysis of SECURITY hive) to document what credential material was accessible; (3) Pull OAuth token grant logs from identity providers (Azure AD sign-in logs, Okta System Log) filtered on the service accounts associated with the compromised software, looking for token grants to unfamiliar applications or IPs in the 30 days preceding detection; (4) Capture a snapshot of all registry run keys and scheduled tasks before eradication: 'reg export HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run' and 'schtasks /query /fo CSV /v > schtasks_snapshot.csv'.

Step 4: Recovery — Validate remediation by re-running integrity checks against all previously flagged binaries. Monitor for re-infection patterns: watch for the same process lineages and network destinations that triggered initial detection. Confirm that no persistence mechanisms (scheduled tasks, registry run keys, startup services) were introduced by the malicious component before declaring recovery. Retain forensic copies of affected systems for post-incident analysis.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery: restore systems to normal operation only after verifying eradication completeness; for supply chain campaigns with disputed attribution between TeamPCP, ShinyHunters, and Lapsus$, extended monitoring is required because each actor has historically re-established access through different persistence mechanisms after initial remediation

Controls: NIST IR-4 (Incident Handling), NIST SI-7 (Software, Firmware, and Information Integrity), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST CP-10 (System Recovery and Reconstitution) — return to known-good state with verified integrity, CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Automate integrity re-checks using AIDE (Advanced Intrusion Detection Environment) on Linux — run 'aide --check' against a baseline built from a clean reference system, not the potentially compromised host. On Windows, use Sysinternals Sigcheck ('sigcheck -u -e C:\Windows\System32\') to identify unsigned or invalidly signed binaries in system directories. For persistence sweep without EDR: execute 'autoruns.exe -a * -c > autoruns_output.csv' (Sysinternals Autoruns) and diff against a pre-incident baseline, filtering for entries with no valid Microsoft or vendor signature. Monitor for re-infection for a minimum of 30 days given that Lapsus$ has historically used dormant access to re-establish footholds after victims believed remediation was complete.

Evidence: Before declaring recovery complete: (1) Re-run hash validation against all binaries in the software component's install directory and compare to the pre-eradication snapshot — document any discrepancies; (2) Pull network flow logs (NetFlow or Sysmon Event ID 3) for a 72-hour window post-remediation and verify no connections to the same external destinations observed during initial detection; (3) Confirm forensic disk images and memory captures taken during eradication are stored on isolated, write-protected media per NIST AU-9 (Protection of Audit Information) — these will be required if regulatory notification obligations arise given the PII/credential-focused TTPs associated with ShinyHunters; (4) Review Windows System Event Log Event ID 7045 (New Service Installed) and Event ID 4697 (Service Installed in System) for the 30-day pre-detection window to ensure no malicious services were installed and survived the eradication pass.

Step 5: Post-Incident — Conduct a software bill of materials (SBOM) gap assessment: identify which components in production lack integrity verification. Evaluate CI/CD pipeline controls against NIST SP 800-218 (Secure Software Development Framework) and CISA's Secure by Design guidance. Implement or validate code signing verification at ingestion, not just at build time. Document which actor's TTPs were confirmed present to assist with future attribution scoping if this campaign resurfaces.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity: lessons learned must produce durable improvements to detection and prevention; for TeamPCP's supply chain campaign, the post-incident output must include an SBOM-anchored control baseline and attribution documentation that accounts for the ShinyHunters/Lapsus$ overlap to prevent future scoping failures

Controls: NIST IR-4 (Incident Handling) — update IR plan with supply chain-specific procedures, NIST IR-8 (Incident Response Plan) — revise plan to include SBOM verification gates and multi-actor attribution workflows, NIST SI-2 (Flaw Remediation) — establish repeatable process for vetting third-party components before ingestion, NIST SI-7 (Software, Firmware, and Information Integrity) — enforce integrity verification at ingestion as a sustained control, not a one-time remediation step, NIST SA-12 (Supply Chain Protection) — assess and document supply chain risk for all critical software dependencies, CIS 2.1 (Establish and Maintain a Software Inventory), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Generate an SBOM using Syft ('syft -o spdx-json > sbom.json') for containerized workloads or CycloneDX's CLI tool for application directories — both are free and produce machine-readable output. Validate SBOM component integrity by cross-referencing hashes against the OSV (Open Source Vulnerability) database API ('curl https://api.osv.dev/v1/query' with package name and version). For CI/CD pipeline hardening without enterprise tooling: add a pre-merge GitHub Actions step using 'cosign verify' (Sigstore project) to enforce container image signing before any image is pulled into production. For attribution documentation: create a structured TTP matrix in a spreadsheet mapping confirmed MITRE ATT&CK techniques to each actor (TeamPCP, ShinyHunters, Lapsus$) with evidence source and confidence level — this becomes the scoping anchor for future incidents involving overlapping claims.

Evidence: Post-incident evidence to retain and document: (1) The complete SBOM diff between pre-incident and post-incident states — this establishes the ground truth for what changed and is the primary artifact for regulatory disclosure if required; (2) All collected Sysmon logs, EDR telemetry, and network captures from the detection and eradication phases, retained per AU-11 (Audit Record Retention) for a minimum period consistent with applicable regulations (GDPR 72-hour notification window, state breach laws, SEC 4-day disclosure rule if public company); (3) The attribution assessment document mapping confirmed TTPs to specific actor dossiers — given ShinyHunters' documented focus on credential theft and data exfiltration and Lapsus$'s use of social engineering and insider recruitment, document which of these behaviors were evidenced versus absent in your environment to support scoping decisions; (4) Timeline reconstruction artifact showing the full attack chain from initial software ingestion through detection, to support NIST SP 800-218 gap analysis against your actual SSDF implementation.

Detection Guidance

No confirmed IOCs are available from verified source material. Detection must rely on behavioral and structural indicators rather than hash or IP blocklists at this time. Focus on: (1) Software update processes spawning child

processes that write to non-standard directories or initiate outbound network connections, SIEM query target: process creation events where parent is a known updater or package manager and child performs network I/O; (2) Binary modification events on files owned by trusted software packages, EDR file integrity monitoring alerts on executables outside of expected patch windows; (3) Credential use anomalies following software updates, UEBA or authentication log review for service accounts that began accessing new resources after a component update; (4) T1486 ransomware precursor activity: volume shadow copy deletion, rapid file encryption events, or backup service termination commands. Given the multi-actor attribution dispute, do not anchor hunting to a single known IOC set. Treat ShinyHunters and Lapsus$ TTPs as separate hunt hypotheses running in parallel. MITRE ATT&CK Navigator layers for both groups are publicly available and should be used to structure independent hypothesis sets.

Indicators of Compromise

Type	Value	Context	Confidence
DOMAIN	`[not confirmed]`	No IOCs were extractable from verified source material for this campaign. The Dark Reading source URL was noted but content was not directly verified. Do not populate operational blocklists from this entry.	LOW

Framework Mappings

MITRE-ATTACK

- **T1078** — Valid Accounts
- **T1584** — Compromise Infrastructure
- **T1583** — Acquire Infrastructure
- **T1554** — Compromise Host Software Binary
- **T1588.002** — Tool
- **T1486** — Data Encrypted for Impact
- **T1036** — Masquerading
- **T1195.002** — Compromise Software Supply Chain

NIST-800-53R5

- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CP-9** — System Backup
- **CP-10** — System Recovery and Reconstitution
- **CM-7** — Least Functionality
- **SA-9** — External System Services

- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-3** — Configuration Change Control
- **SR-2** — Supply Chain Risk Management Plan

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **15.1** — Establish and Maintain an Inventory of Service Providers

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program

ISO-27001-2022

- **A.5.21** — Managing information security in the ICT supply chain

SOC2-TSC

- **CC9.2** — Manages risks associated with vendors and business partners

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1078	Valid Accounts	Defense-Evasion
T1584	Compromise Infrastructure	Resource-Development
T1583	Acquire Infrastructure	Resource-Development
T1554	Compromise Host Software Binary	Persistence
T1588.002	Tool	Resource-Development
T1486	Data Encrypted for Impact	Impact
T1036	Masquerading	Defense-Evasion
T1195.002	Compromise Software Supply Chain	Initial-Access

Sources

Source	URL	Tier
Security News	https://www.darkreading.com/threat-intelligence/teampcp-attacks-hac...	T3

Source	URL	Tier
"Common Text 1.9" vulnerability with CVE-2022-42889 patch fix	https://knowledge.broadcom.com/external/article/252807/siteminder-n...	T3
The vulnerability CVE 2022-42889 older commons-text- jar files ...	https://knowledge.informatica.com/s/article/000206280?language=en_US	T3
K24823443: Apache Commons Text vulnerability CVE-2022-42889	https://my.f5.com/manage/s/article/K24823443	T3
CVE-2022-42889 - Apache Commons Text Vulnerability	https://developer.harness.io/docs/continuous-delivery/armory/genera...	T3

Generated 2026-04-03 13:38 UTC by TJS Security Command Center