INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-10 14:16 UTC

China-Nexus and DPRK Actors Lead Multi-Vector Assault on Technology Sector: Supply Chain Poisoning, Insider Threats, and Extortion Converge

THREAT CAMPAIGN | **HIGH** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0436
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	9.5
Affected Products	Axios npm package (v1.14.1, v0.30.4), GitHub repositories, mail infrastructure, private code repositories, North American tech organizations broadly; 572 technology organizations named on dedicated leak sites
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike's 2026 Technology Threat Landscape Report documents a converging three-front assault on the technology sector: China-nexus actors conducting sustained espionage against software development and cloud environments, DPRK-linked operatives embedding fraudulent IT workers inside target organizations while simultaneously poisoning the widely-used Axios npm package (v1.14.1 and v0.30.4) with a Remote Access Trojan, and financially motivated eCrime groups listing 572 technology firms on extortion leak sites. The Axios supply chain compromise is corroborated by Trend Micro, Endor Labs, Phoenix Security, and an official post-mortem on the Axios GitHub repository. Any organization with Axios in its software supply chain, remote workers hired without rigorous identity verification, or code repositories accessible to third parties faces espionage, insider threat, and ransomware risk across all three attack vectors.

Technical Analysis

Three distinct threat tracks are documented, each with separate technical characteristics. Track 1, China-nexus espionage: Actors targeted private code repositories, software development pipelines, and cloud infrastructure, consistent with MITRE T1213 (Data from Information Repositories) and T1589 (Gather Victim Identity Information). No specific CVEs are attributed to this track in the source material. Track 2, DPRK dual operations:

(a) Fraudulent IT worker placements exploit T1591.004 (Identify Roles) and T1078 (Valid Accounts) to establish persistent insider access; (b) The Axios npm supply chain compromise (CWE-494: Download of Code Without Integrity Check; CWE-1104: Use of Unmaintained Third-Party Components) involved a hijacked maintainer account used to publish malicious versions v1.14.1 and v0.30.4 of the axios npm package, embedding a Remote Access Trojan. Relevant techniques include T1195.001 (Compromise Software Dependencies and Development Tools), T1195.002 (Compromise Software Supply Chain), T1608.001 (Stage Capabilities: Upload Malware), T1547 (Boot or Logon Autostart Execution), and T1552 (Unsecured Credentials). CWE-521 (Weak Password Constraints) is implicated in the maintainer account hijack. Track 3, eCrime extortion: 572 technology organizations named on dedicated leak sites, consistent with T1486 (Data Encrypted for Impact) and T1650 (Acquire Access). Affected package versions: axios v1.14.1 and v0.30.4. Safe versions are those predating or postdating the malicious releases; organizations should verify via the official Axios GitHub post-mortem (issue #10636). The source data does not assign a CVE ID or CVSS vector string; severity is qualitatively rated as High based on the scope of targeting (572 organizations) and confirmed technical compromise (supply chain RAT injection).

Action Checklist

1. Step 1: Containment. Immediately audit all production and CI/CD environments for axios versions v1.14.1 and v0.30.4 using 'npm list axios' or equivalent dependency scanning. Isolate any build pipelines, containers, or deployed services confirmed to use these versions. Block outbound connections from affected systems pending eradication. Freeze new IT contractor onboarding pending identity re-verification (per DPRK insider threat track).
2. Step 2: Detection. Query dependency manifests (package.json, package-lock.json, yarn.lock) and SBOM artifacts across all repositories for axios v1.14.1 or v0.30.4. Review npm audit logs and artifact registry pull logs for these versions. For the RAT component: hunt for anomalous outbound network connections from Node.js processes, unexpected scheduled tasks or autostart entries (T1547), and credential access patterns (T1552) in SIEM. For insider threat track: review access logs to private code repositories (NIST AU-6) for accounts associated with recently onboarded remote IT contractors. Cross-reference contractor identities against known DPRK fraudulent worker indicators documented in CISA advisories. CIS 8.2 (Collect Audit Logs) should be verified as active across all relevant systems.
3. Step 3: Eradication. Upgrade axios to a verified clean version as confirmed by the official Axios GitHub post-mortem (issue #10636 at github.com/axios/axios). Remove v1.14.1 and v0.30.4 from all package caches, artifact registries, and container base images. Rotate all credentials and tokens accessible to systems that ran the compromised versions (including API keys, secrets, and service account credentials). Revoke and reissue all authentication material stored in affected environments. For suspected insider threat accounts: disable accounts under review, rotate all credentials those accounts had access to, and preserve forensic artifacts for investigation.
4. Step 4: Recovery. After upgrading axios, re-run dependency scans and SBOM generation to confirm no residual references to malicious versions. Validate CI/CD pipeline integrity: check that no malicious build steps, injected dependencies, or modified configuration files persist (NIST AU-9). Monitor network telemetry for RAT callback indicators for a minimum of 30 days post-remediation. Re-enable onboarding pipelines only after identity verification controls are documented and tested. Confirm audit logging is restored and baseline-compliant per CIS 8.2 and NIST AU-2.
5. Step 5: Post-Incident. Implement or enforce package integrity verification in CI/CD pipelines using cryptographic signing and checksum validation to address CWE-494. Enforce CIS 2.2 (Ensure Authorized

Software is Currently Supported) and CIS 2.3 (Address Unauthorized Software) to prevent unmaintained or unreviewed packages in production (addressing CWE-1104). Enforce MFA on all package registry maintainer accounts to prevent account hijack recurrence. Implement credential rotation on a defined schedule for all developer and service accounts with publish rights. Formalize an insider threat program with enhanced identity vetting for remote IT contractors, referencing CISA's published guidance on DPRK IT worker schemes.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to executive leadership, legal counsel, and external IR retainer immediately if: forensic evidence confirms the axios RAT executed and established outbound C2 from any environment with access to customer data, source code, or cloud credentials; any DPRK-linked contractor account is confirmed to have exfiltrated proprietary code or credentials; or your organization is among the 572 named on eCrime dedicated leak sites — all three conditions independently trigger breach notification assessment under applicable state, federal, or sector-specific regulations (CCPA, SEC material cybersecurity incident rules, HIPAA if PHI was in scope).
Recovery Notes	Re-enable production deployments only after SBOM validation confirms axios v1.14.1 and v0.30.4 are absent from all container images, package caches, and artifact registry proxies, and after all credentials with exposure to the compromised axios execution context have been rotated and confirmed active in downstream systems. Maintain 30-day continuous monitoring of outbound network connections from Node.js processes across all formerly affected hosts, specifically watching for periodic beacon patterns (regular intervals of DNS queries or TCP connections to non-business destinations) characteristic of the RAT's C2 callback behavior. Defer re-onboarding of remote IT contractors until the enhanced identity vetting process is documented, tested with at least one dry run, and signed off by HR and Security leadership — rushed re-enablement is the primary recurrence vector for the DPRK insider threat track.
Forensic Artifacts	npm and package manager transaction logs (~/.npm/_logs/, npm-debug.log, yarn-error.log) timestamped to the install window of axios v1.14.1 or v0.30.4 — establish which CI runner user accounts and build job IDs pulled the poisoned package from which registry endpoint | Sysmon Event ID 3 (Network Connection) and Event ID 22 (DNS Query) records from all hosts where Node.js processes executed against the compromised axios version — the RAT embedded in v1.14.1/v0.30.4 would produce anomalous outbound TCP connections and DNS resolutions from 'node.exe' or CI runner processes to non-business IP ranges | Container image layer history ('docker history --no-trunc') and artifact registry pull audit logs for every image built during the exposure window — identifies which production-deployed containers contain the malicious axios code and which downstream services are affected | GitHub, GitLab, or Bitbucket repository audit logs for contractor-associated accounts showing git clone, secret read, Actions secret access, or settings modification events — the DPRK insider threat track targets private source code and CI/CD secrets, leaving access event trails in platform audit APIs | Windows Task Scheduler XML exports ('C:\Windows\System32\Tasks\') and Linux persistence locations ('/etc/cron.d/', '/var/spool/cron/crontabs/', '~/.config/autostart/', systemd unit files under '/etc/systemd/system/') for entries created during the axios compromise window — the RAT's T1547 persistence mechanism would register a scheduled or autostart entry referencing a Node.js script or dropped binary

Per-Action IR Details

Step 1: Containment — Immediately audit all production and CI/CD environments for axios versions v1.14.1 and v0.30.4 using 'npm list axios' or equivalent dependency scanning. Isolate any build pipelines, containers, or deployed services confirmed to use these versions. Block outbound connections from affected systems pending eradication. Freeze new IT contractor onboarding pending identity re-verification (per DPRK insider threat track).

NIST Phase: Containment

Reference: NIST 800-61r3 §3.3 — Containment Strategy

Controls: NIST AC-4 (Information Flow Enforcement), NIST AC-6 (Least Privilege), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 4.4 (Implement and Manage a Firewall on Servers)

Compensating: Run 'npm list axios --all 2>/dev/null | grep -E "1\.14\.1|0\.30\.4"' recursively across all Node.js project roots. Use a bash one-liner: 'find / -name package-lock.json 2>/dev/null | xargs grep -l "1\.14\.1\|0\.30\.4"' to locate affected manifests. Block outbound C2 at the host firewall using 'iptables -I OUTPUT -p tcp -m owner --uid-owner node -j DROP' on Linux build agents. Freeze contractor VPN/SSO accounts via AD 'Disable-ADAccount' PowerShell cmdlet pending re-verification.

Evidence: Before isolating any container or build pipeline, snapshot the running process list ('ps aux' or 'Get-Process') and active network connections ('ss -tulnp' or 'netstat -ano') from each affected host to capture RAT beacon state. Preserve container image digests from the artifact registry pull logs to establish which builds consumed axios v1.14.1 or v0.30.4. Export CI/CD job execution histories (GitHub Actions logs, Jenkins build logs) showing which pipelines pulled the poisoned versions before isolation cuts log access.

Step 2: Detection — Query dependency manifests (package.json, package-lock.json, yarn.lock) and SBOM artifacts across all repositories for axios v1.14.1 or v0.30.4. Review npm audit logs and artifact registry pull logs for these versions. For the RAT component: hunt for anomalous outbound network connections from Node.js processes, unexpected scheduled tasks or autostart entries (T1547), and credential access patterns (T1552) in SIEM. For insider threat track: review access logs to private code repositories (NIST AU-6) for accounts associated with recently onboarded remote IT contractors. Cross-reference contractor identities against known DPRK fraudulent worker indicators documented in CISA advisories. CIS 8.2 (Collect Audit Logs) should be verified as active across all relevant systems.

NIST Phase: Detection Analysis

Reference: NIST 800-61r3 §3.2 — Detection and Analysis

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-2 (Event Logging), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Deploy Sysinternals Autoruns to enumerate T1547 persistence entries (scheduled tasks, Run keys, service registrations) on all hosts that executed Node.js processes linked to the compromised axios build. Run 'schtasks /query /fo LIST /v | findstr /i "node\|npm\|axios"' on Windows build agents. For network hunting without a SIEM, use Wireshark or tcpdump with filter 'tcp and (port 443 or port 80) and host ' correlated via Sysmon Event ID 3 (Network Connection) filtering on Image path containing 'node.exe'. Query GitHub audit log API ('GET /orgs/{org}/audit-log?phrase=actor:{contractor_username}') to surface repository access events for flagged contractor accounts.

Evidence: Capture Sysmon Event ID 1 (Process Creation) for 'node.exe' and 'npm.exe' with command-line arguments referencing axios install or execution, and Sysmon Event ID 3 (Network Connection) from those same processes to identify RAT C2 beacon destinations. Extract GitHub/GitLab repository access audit logs for contractor accounts showing clone, fork, or secret-read events against private repos. Pull npm registry pull logs from your artifact registry (Nexus, Artifactory, or npm audit log at ~/.npm/_logs/) for timestamps when v1.14.1 or v0.30.4 were fetched. Enumerate Windows Task Scheduler XML entries under 'C:\Windows\System32\Tasks\' and Linux cron entries in '/etc/cron.d/' and '/var/spool/cron/' for tasks created during the window the poisoned axios version was active.

Step 3: Eradication — Upgrade axios to a verified clean version as confirmed by the official Axios GitHub post-mortem (issue #10636 at github.com/axios/axios). Remove v1.14.1 and v0.30.4 from all package caches, artifact registries, and container base images. Rotate all credentials and tokens accessible to systems that

ran the compromised versions (T1552 exposure). Revoke and reissue API keys, secrets, and service account credentials stored in affected environments. For suspected insider threat accounts: disable accounts under review, rotate all credentials those accounts had access to, and preserve forensic artifacts for investigation.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-2 (Account Management), NIST IA-4 — no mapped control, NIST CM-6 — no mapped control, CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 7.3 (Perform Automated Operating System Patch Management), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Purge poisoned versions from local npm cache with 'npm cache clean --force' and delete cached tarballs under '~/.npm/axios/' matching v1.14.1 or v0.30.4. In Nexus or Artifactory, use the REST API to delete specific component versions: 'curl -X DELETE "http:///service/rest/v1/components/"'. Rotate secrets stored in .env files, GitHub Actions secrets, and CI environment variables by auditing 'git log --all --full-history -- "*.env"' and 'git grep -i "API_KEY\|SECRET\|TOKEN"' across all affected repositories to enumerate what the RAT may have accessed. Disable insider threat contractor accounts immediately using 'Disable-ADAccount -Identity ' and revoke all associated OAuth tokens via your identity provider's admin console.

Evidence: Before rotating credentials, forensically preserve a list of all secrets, tokens, and API keys accessible to the compromised axios execution context: capture environment variable dumps ('printenv' on Linux, '[System.Environment]::GetEnvironmentVariables()' on Windows), .env file contents, and CI/CD secret names (not values) from pipeline configuration. Image or snapshot any container that ran the poisoned axios version prior to teardown to preserve the RAT's file-system footprint, including any dropped binaries under '/tmp/', '/var/tmp/', or Node.js module cache directories. Preserve the full npm install transaction log from 'npm-debug.log' and package manager output to establish the exact install timestamp for chain-of-custody.

Step 4: Recovery — After upgrading axios, re-run dependency scans and SBOM generation to confirm no residual references to malicious versions. Validate CI/CD pipeline integrity: check that no malicious build steps, injected dependencies, or modified configuration files persist (NIST AU-9, D3-SFA). Monitor network telemetry for RAT callback indicators for a minimum of 30 days post-remediation. Re-enable onboarding pipelines only after identity verification controls are documented and tested. Confirm audit logging is restored and baseline-compliant per CIS 8.2 and NIST AU-2.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-9 (Protection Of Audit Information), NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, And Reporting), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

Compensating: Re-run SBOM generation using Syft ('syft -o cyclonedx-json') and validate output with Grype ('grype sbom:./sbom.json') to confirm axios v1.14.1 and v0.30.4 are absent from all rebuilt container images. Validate CI/CD YAML integrity using 'git diff ..HEAD -- .github/workflows/ .gitlab-ci.yml Jenkinsfile' to detect injected build steps. For 30-day RAT callback monitoring without SIEM, configure Sysmon Event ID 3 log forwarding to a central syslog server and write a daily Sigma-compatible grep: 'grep -E "DestinationIp|DestinationHostname" /var/log/sysmon.log | grep node' to surface anomalous Node.js outbound connections.

Evidence: During the recovery validation window, continuously collect Sysmon Event ID 3 (Network Connection) and Event ID 22 (DNS Query) logs from all formerly affected hosts, filtering on processes spawned by 'node.exe' or CI runner agents, to detect dormant RAT callbacks that activate post-remediation. Preserve git commit hash records for all CI/CD configuration files (.github/workflows/, Jenkinsfile, .gitlab-ci.yml) at the point of re-enablement to establish a clean baseline for future integrity comparisons. Archive the SBOM artifacts generated post-remediation with cryptographic checksums as evidence of the verified clean state.

Step 5: Post-Incident — Implement or enforce package integrity verification in CI/CD pipelines using cryptographic signing and checksum validation to address CWE-494 (no mapped control in the provided knowledge base for npm-specific signing enforcement beyond CWE-494 guidance). Enforce CIS 2.2 (Ensure

Authorized Software is Currently Supported) and CIS 2.3 (Address Unauthorized Software) to prevent unmaintained or unreviewed packages in production (addressing CWE-1104). Enforce MFA on all package registry maintainer accounts (CIS 6.5, D3-MFA) to prevent account hijack recurrence. Implement D3-CRO (Credential Rotation) on a defined schedule for all developer and service accounts with publish rights. Formalize an insider threat program with enhanced identity vetting for remote IT contractors, referencing CISA's published guidance on DPRK IT worker schemes.

NIST Phase: Post Incident

Reference: NIST 800-61r3 §4 — Post-Incident Activity

Controls: CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 2.3 (Address Unauthorized Software), CIS 6.5 (Require MFA for Administrative Access), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), NIST AC-2 (Account Management), NIST AU-6 (Audit Record Review, Analysis, And Reporting)

Compensating: Enforce npm package integrity by adding 'npm config set ignore-scripts true' and pinning exact package versions with SHA-512 integrity hashes in package-lock.json — validate these are present by running 'cat package-lock.json | python3 -c "import sys,json; pkgs=json.load(sys.stdin)[\"packages\"]; print([k for k,v in pkgs.items() if \"integrity\" not in v])"'. For DPRK insider threat vetting without a commercial background check platform, implement a structured video interview protocol cross-referencing CISA Advisory AA23-347A indicators: inconsistent timezone/accent signals, reluctance to appear on camera, use of name-forwarding services, and multiple personas sharing the same payment destination. Enable npm 2FA for all maintainer accounts at no cost via 'npm profile enable-2fa auth-and-writes'.

Evidence: Document all lessons-learned artifacts for the post-incident review: the full timeline of axios v1.14.1/v0.30.4 presence in each environment (first install to remediation), the complete list of credentials rotated with rotation timestamps, the contractor account access audit trail from repository logs, and any RAT network indicators (C2 IPs/domains, beacon intervals) observed during the 30-day monitoring window. These artifacts support both internal process improvement and any required regulatory breach notification assessment. Retain all forensic images and log archives per your documented retention schedule under NIST AU-11 — no mapped control in the provided knowledge base for the specific retention period assignment.

Detection Guidance

For the Axios supply chain compromise: scan all dependency manifests (package.json, package-lock.json, yarn.lock) and container image layers for axios v1.14.1 or v0.30.4. Use SBOM tooling (Syft, Grype, or equivalent) across repositories and deployed artifacts. In SIEM, alert on Node.js process trees spawning unexpected child processes or making outbound connections to non-whitelisted external IPs, which may indicate RAT callback activity (T1547, T1552). Check npm audit logs and private artifact registry pull records for these specific version strings. For the insider threat track: audit access logs on private code repositories and cloud environments (NIST AU-6) for accounts with anomalous access patterns, high-volume data reads, off-hours access, or bulk downloads from source code repositories (T1213). Cross-reference recently onboarded remote IT contractor accounts against CISA's DPRK IT worker indicators. For the eCrime/ransomware track: monitor for T1486 indicators including mass file encryption events, VSS deletion commands, and ransom note file creation via endpoint detection tooling.

Indicators of Compromise

Type	Value	Context	Confidence
HASH	`not available in source material`	Malicious axios v1.14.1 and v0.30.4 package hashes not published in reviewed sources — consult Trend Micro and Endor Labs advisories for artifact hashes	LOW
URL	`https://github.com/axios/axios/issues/10636`	Official Axios post-mortem issue documenting the supply chain compromise — use to identify clean version guidance	HIGH
DOMAIN	`not available in source material`	RAT C2 infrastructure not published in reviewed sources — consult Trend Micro advisory for network IOCs	LOW

Framework Mappings

MITRE-ATTACK

- **T1589** — Gather Victim Identity Information
- **T1213** — Data from Information Repositories
- **T1078** — Valid Accounts
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1195.002** — Compromise Software Supply Chain
- **T1552** — Unsecured Credentials
- **T1547** — Boot or Logon Autostart Execution
- **T1566** — Phishing
- **T1591.004** — Identify Roles
- **T1486** — Data Encrypted for Impact
- **T1110.003** — Password Spraying
- **T1650** — Acquire Access
- **T1608.001** — Upload Malware

NIST-800-53R5

- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **CM-7** — Least Functionality
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **SI-3** — Malicious Code Protection

- **SI-4** — System Monitoring
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-8** — Spam Protection
- **CP-9** — System Backup
- **CP-10** — System Recovery and Reconstitution
- **CM-3** — Configuration Change Control
- **SA-4** — Acquisition Process
- **IR-4** — Incident Handling
- **SR-2** — Supply Chain Risk Management Plan

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures
- **A06:2021** — Vulnerable and Outdated Components

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **16.4** — Establish and Manage an Inventory of Third-Party Software Components
- **15.1** — Establish and Maintain an Inventory of Service Providers

NIST-CSF-2

- **RS.MI-01** — Incidents are contained
- **GV.SC-01** — Cybersecurity supply chain risk management program

HIPAA-SECURITY

- **164.308(a)(7)(ii)(A)** — Data Backup Plan

ISO-27001-2022

- **A.5.29** — Information security during disruption
- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

SOC2-TSC

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

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1589	Gather Victim Identity Information	Reconnaissance

Technique ID	Technique Name	Tactic
T1213	Data from Information Repositories	Collection
T1078	Valid Accounts	Defense-Evasion
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1195.002	Compromise Software Supply Chain	Initial-Access
T1552	Unsecured Credentials	Credential-Access
T1547	Boot or Logon Autostart Execution	Persistence
T1566	Phishing	Initial-Access
T1591.004	Identify Roles	Reconnaissance
T1486	Data Encrypted for Impact	Impact
T1110.003	Password Spraying	Credential-Access
T1650	Acquire Access	Resource-Development
T1608.001	Upload Malware	Resource-Development

Sources

Source	URL	Tier
Blog	https://www.crowdstrike.com/en-us/blog/crowdstrike-2026-technology-...	T3
Axios NPM Package Compromised: Supply Chain Attack Hits ...	https://www.trendmicro.com/en_us/research/26/c/axios-npm-package-co...	T3
Post Mortem: axios npm supply chain compromise #10636 - GitHub	https://github.com/axios/axios/issues/10636	T3
axios npm Compromised: RAT in v1.14.1 & v0.30.4 (2026)	https://phoenix.security/axios-supply-chain-compromise-npm-rat-2026/	T3
Axios compromised: hijacked maintainer account pushes malicious ...	https://www.endorlabs.com/learn/npm-axios-compromise	T3

Generated 2026-06-10 14:16 UTC by TJS Security Command Center