INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-10 07:22 UTC

# Technology Sector Faces Converging State-Sponsored and eCrime Threats: China, DPRK, and Criminal Extortion Dominate 2026 Landscape

**THREAT CAMPAIGN** | **HIGH** | CVSS 7.5

| | |
|---|---|
| **SCC Item ID** | SCC-CAM-2026-0434 |
| **Type** | Threat Campaign |
| **Severity** | HIGH |
| **CVSS Base Score** | 7.5 |
| **Affected Products** | Technology sector broadly; Axios npm package (supply chain, STARDUST CHOLLIMA); GitHub repositories (Glassworm campaign); macOS systems (OpenClaw-lure infostealer); CrowdStrike Falcon Platform customers |
| **Discovery Source** | Rss:T1 Threatintel |

## Executive Summary

CrowdStrike's 2026 Technology Threat Landscape Report identifies the technology sector as the most targeted industry globally, facing simultaneous pressure from Chinese state-sponsored espionage targeting AI intellectual property, DPRK financial operations including a confirmed supply chain compromise of the Axios npm package, and criminal extortion groups that named 572 technology organizations on leak sites. Initial access broker listings for technology sector network access rose approximately 30% year-over-year, indicating an active criminal ecosystem brokering entry into technology environments. Organizations depending on open-source dependencies, GitHub-hosted repositories, or macOS endpoints face elevated and converging risk across supply chain, identity, and endpoint vectors.

## Technical Analysis

CrowdStrike's report (April 2025-March 2026) documents a multi-vector threat environment across the technology sector. China-nexus adversaries account for over 58% of state-sponsored intrusions, with AI system intelligence collection and IP theft as primary objectives. STARDUST CHOLLIMA (DPRK-nexus) compromised the widely-used Axios npm package, introducing malicious code into downstream consumers via supply chain poisoning, mapped to CWE-494 (Download of Code Without Integrity Check) and MITRE T1195.002 (Compromise Software Supply Chain). The Glassworm campaign targeted GitHub repositories. macOS-targeting infostealers using OpenClaw lures expand the endpoint threat surface. DPRK insider threat

operations via fraudulent IT worker infiltration continue as a parallel financial vector. eCrime extortion groups named 572 technology organizations on dedicated leak sites. Active MITRE techniques include T1199 (Trusted Relationship), T1133 (External Remote Services), T1586 (Compromise Accounts), T1657 (Financial Theft), T1059 (Command and Scripting Interpreter), T1486 (Data Encrypted for Impact), T1213 (Data from Information Repositories), T1608.001 (Stage Capabilities: Upload Malware), T1110.003 (Password Spraying), T1078 (Valid Accounts), T1566 (Phishing), T1588.001 (Obtain Capabilities: Malware), and T1083 (File and Directory Discovery). CWEs: CWE-693 (Protection Mechanism Failure), CWE-494 (Download of Code Without Integrity Check), CWE-287 (Improper Authentication). No single CVE is attributed; this is a campaign-level finding across multiple intrusion sets. Primary sourcing is CrowdStrike vendor reporting (T3), typical for threat landscape surveys.

## Action Checklist

1. Step 1, Audit: Containment. Audit all production and CI/CD dependencies on the Axios npm package immediately. Per CrowdStrike and Sonatype reporting on the STARDUST CHOLLIMA compromise, identify any Axios versions pulled during the compromise window and isolate affected build pipelines. Lock package versions in package-lock.json or yarn.lock to prevent silent upstream updates. Apply CIS 2.1 (Establish and Maintain a Software Inventory) to enumerate all npm package consumers across your environment.
2. Step 2, Detection: Query SIEM and EDR for anomalous outbound connections from Node.js processes, unexpected child process spawning from npm scripts, and file writes to sensitive directories from build tooling. Review GitHub repository access logs for the Glassworm campaign indicators: unauthorized commits, new collaborator additions, or unexpected Actions workflow modifications. On macOS endpoints, hunt for OpenClaw lure artifacts; look for unsigned binaries in user download directories and unexpected persistence entries in LaunchAgents or LaunchDaemons. Enable AU-2 (Event Logging) across CI/CD, endpoint, and identity planes if not already active.
3. Step 3, Eradication: Replace any compromised Axios package versions with a verified clean version confirmed through official npm registry integrity checks and subresource integrity validation (CWE-494 mitigation). Remove unauthorized GitHub repository access or modified workflow files identified during detection. For macOS infostealers, quarantine and reimage affected endpoints. Revoke and rotate any credentials that may have been exposed to compromised build environments per CIS 5.1 (Establish and Maintain an Inventory of Accounts). Audit and disable dormant accounts per CIS 5.3.
4. Step 4, Recovery: Validate clean dependency state by running software composition analysis (SCA) tooling against all production builds before redeployment. Confirm audit logging is intact and has not been tampered with per AU-9 (Protection of Audit Information). Re-verify MFA enforcement on all CI/CD pipeline service accounts and developer accounts per CIS 6.3 and CIS 6.5. Monitor post-fix for re-compromise indicators: unexpected package version changes, new outbound connections from build systems, or credential reuse alerts.
5. Step 5, Post-Incident: Conduct a supply chain dependency review to enforce signed package verification and implement a software bill of materials (SBOM) process aligned to CIS 2.1. Review insider threat controls: validate that contractor and IT worker onboarding includes identity verification steps that can detect DPRK fraudulent worker infiltration (AC-2, Account Management). Document gaps in dependency integrity verification and prioritize remediation in the vulnerability management process per CIS 7.1 (Establish and Maintain a Vulnerability Management Process).

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | IMMEDIATE |
| **Escalation Criteria** | Escalate to CISO and legal counsel immediately if forensic evidence confirms a trojanized Axios tarball was executed in a production build pipeline, any macOS endpoint with access to source code or credentials shows OpenClaw indicators of compromise, a GitHub repository with access to production secrets shows Glassworm unauthorized access, or contractor identity verification flags a potential DPRK fraudulent worker with access to sensitive IP or infrastructure — each condition represents active threat actor presence with data exfiltration potential triggering breach notification assessment under applicable state and sector regulations. |
| **Recovery Notes** | Before redeploying any production build that consumed Axios during the STARDUST CHOLLIMA compromise window, require a full clean build from a verified source state with SCA validation — do not patch in place. Monitor all CI/CD pipelines and npm dependency resolution logs continuously for a minimum of 30 days post-recovery, specifically watching for Axios version drift, unexpected postinstall script execution, and outbound connections from build agents to non-registry endpoints. Given the confirmed DPRK financial motivation and Chinese AI IP targeting identified in the 2026 CrowdStrike report, treat any re-compromise indicator as a potential persistent access attempt and re-engage the full IR lifecycle rather than treating it as a routine security event. |
| **Forensic Artifacts** | npm cache and lockfile integrity records — `~/.npm/_cacache/` tarballs and `package-lock.json` integrity hashes from every build that resolved Axios during the STARDUST CHOLLIMA compromise window, which will contain the SHA-512 hash of the trojanized tarball if pulled \| CI/CD pipeline execution logs — GitHub Actions run logs, Jenkins build console output, or equivalent, timestamped within the compromise window, showing which Axios version was resolved, whether postinstall scripts executed, and any spawned child processes or outbound network calls from the build agent \| macOS LaunchAgent and LaunchDaemon plist files — from `/Users/*/Library/LaunchAgents/`, `/Library/LaunchAgents/`, and `/Library/LaunchDaemons/` on endpoints targeted by OpenClaw lure delivery, which would contain the persistence entry dropped by the infostealer \| macOS Unified Log archive — collected via `log collect --last 14d` from OpenClaw-affected endpoints, capturing process execution chains, network connections from unsigned binaries, and file system writes to browser credential store paths indicative of infostealer data collection activity \| GitHub repository audit log export — full audit log from affected repositories covering the Glassworm campaign window, including actor, IP address, action type (PushEvent, MemberEvent, CreateEvent, WorkflowJobEvent), and modified file paths for any `.github/workflows/*.yml` files altered to establish Actions-based persistence or exfiltration |

### Per-Action IR Details

**Step 1: Containment — Audit all production and CI/CD dependencies on the Axios npm package immediately. Per CrowdStrike and Sonatype reporting on the STARDUST CHOLLIMA compromise, identify any Axios versions pulled during the compromise window and isolate affected build pipelines. Lock package versions in package-lock.json or yarn.lock to prevent silent upstream updates. Apply CIS 2.1 (Establish and Maintain a Software Inventory) to enumerate all npm package consumers across your environment.**

**NIST Phase:** Containment

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

**Controls:** CIS 2.1 (Establish and Maintain a Software Inventory), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 2.3 (Address Unauthorized Software)

**Compensating:** Run `npm ls axios --all --depth=99 2>/dev/null | tee axios-dep-tree.txt` in each repository root to enumerate all direct and transitive Axios consumers without a SIEM. Use `find . -name 'package-lock.json' -o -name 'yarn.lock' | xargs grep -l 'axios'` to identify every lockfile referencing Axios across a monorepo. For CI/CD pipeline isolation, disable the affected pipeline job in GitHub Actions by commenting out the trigger block and committing — no enterprise tooling required. Cross-reference installed Axios versions against the STARDUST CHOLLIMA compromise window using `npm view axios time --json` to map publish timestamps.

**Evidence:** Before locking or replacing any package versions, snapshot the current state: copy all package.json, package-lock.json, and yarn.lock files from every affected repository to a read-only evidence archive. Capture npm cache contents at `~/.npm/_cacache/` and any local node_modules/.package-lock.json files — these record the exact resolved version hashes pulled at install time and can confirm whether a trojanized Axios tarball was fetched. Also preserve CI/CD pipeline run logs (GitHub Actions logs, Jenkins build logs) timestamped within the STARDUST CHOLLIMA compromise window, as they will show which Axios version was resolved and whether any postinstall scripts executed.

**Step 2: Detection — Query SIEM and EDR for anomalous outbound connections from Node.js processes, unexpected child process spawning from npm scripts, and file writes to sensitive directories from build tooling. Review GitHub repository access logs for the Glassworm campaign indicators: unauthorized commits, new collaborator additions, or unexpected Actions workflow modifications. On macOS endpoints, hunt for OpenClaw lure artifacts using D3-SFA (System File Analysis) — look for unsigned binaries in user download directories and unexpected persistence entries in LaunchAgents or LaunchDaemons (D3-SICA — System Init Config Analysis). Enable AU-2 (Event Logging) across CI/CD, endpoint, and identity planes if not already active.**

**NIST Phase:** Detection Analysis

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

**Controls:** AU-2 (Event Logging), AU-6 (Audit Record Review, Analysis, And Reporting), AU-12 (Audit Record Generation)

**Compensating:** Without SIEM/EDR: deploy Sysmon with SwiftOnSecurity's config (https://github.com/SwiftOnSecurity/sysmon-config — search-retrieved, recommend human validation) and hunt Event ID 3 (Network Connection) filtered on `node.exe` or `npm.exe` as the initiating process making outbound connections to non-registry hosts. For macOS OpenClaw hunting, run `sudo find /Users/*/Downloads /Users/*/Library/LaunchAgents /Library/LaunchDaemons -type f -newer /var/log/install.log 2>/dev/null` to surface recently dropped binaries and persistence plists. For Glassworm GitHub exposure without enterprise tooling, use the GitHub REST API (`gh api /repos/{owner}/{repo}/events --paginate`) to pull repository event logs and filter for `PushEvent`, `MemberEvent`, and `CreateEvent` types from unfamiliar actors. Use `codesign -vv --deep ` on flagged macOS binaries to confirm unsigned status consistent with OpenClaw delivery.

**Evidence:** Capture before any remediation: on macOS endpoints, image volatile memory with osxpmem or use `sudo osascript -e 'tell application "Activity Monitor"'` to record live process trees, then collect LaunchAgent and LaunchDaemon plist files from `/Users/*/Library/LaunchAgents/`, `/Library/LaunchAgents/`, and `/Library/LaunchDaemons/` — OpenClaw persistence would register here. Collect the macOS Unified Log stream with `log collect --last 7d --output /tmp/openClaw-unified.logarchive` to capture process execution, network, and file events. For the Glassworm GitHub campaign, export the full audit log from GitHub Enterprise (Settings → Audit Log → Export) or use the API to capture actor, action, repository, and timestamp fields before any unauthorized collaborator is removed — removal destroys the access record. For STARDUST CHOLLIMA build pipeline compromise, preserve npm postinstall script execution logs and any spawned child process records from the CI runner.

**Step 3: Eradication — Replace any compromised Axios package versions with a verified clean version confirmed through official npm registry integrity checks and subresource integrity validation (CWE-494 mitigation). Remove unauthorized GitHub repository access or modified workflow files identified during detection. For macOS infostealers, quarantine and reimage affected endpoints. Revoke and rotate any**

**credentials that may have been exposed to compromised build environments per D3-CRO (Credential Rotation). Audit and disable dormant accounts per CIS 5.3.**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication

**Controls:** CIS 5.3 (Disable Dormant Accounts), CIS 5.2 (Use Unique Passwords), AC-2 (Account Management), AC-6 (Least Privilege)

**Compensating:** To verify Axios package integrity without an enterprise SCA tool, compare the SHA-512 integrity hash in package-lock.json against the hash published on the official npm registry: `npm view axios@ dist.integrity` — a mismatch confirms a tampered tarball. For credential rotation without a secrets management platform, enumerate CI/CD secrets exposure by reviewing GitHub Actions secrets (`gh secret list --repo owner/repo`), then rotate any token that had read/write access to the repository during the compromise window via the GitHub Settings → Developer Settings → Personal Access Tokens interface. On macOS, confirm full-disk reimage rather than AV-only remediation — OpenClaw infostealers targeting macOS have demonstrated persistence mechanisms (LaunchAgent/LaunchDaemon) that survive user-space AV removal. Use ClamAV with a YARA rule targeting the OpenClaw binary signature for pre-reimage triage scanning: `clamscan --recursive --detect-pua=yes /Users/`.

**Evidence:** Before reimaging any macOS endpoint, acquire a forensic image using `dd` or target-disk mode, and collect the following OpenClaw-specific artifacts: browser credential stores at `/Users/*/Library/Application Support/Google/Chrome/Default/Login Data`, `/Users/*/Library/Application Support/Firefox/Profiles/*/logins.json`, and Safari keychain-linked credentials; cryptocurrency wallet files at `/Users/*/Library/Application Support/` for common wallet apps; and the specific plist file used by the OpenClaw persistence mechanism from LaunchAgents. For the Glassworm GitHub eradication, before removing unauthorized workflow files, copy the malicious `.github/workflows/*.yml` content and any modified Actions secrets — the workflow file itself is a forensic artifact revealing the attacker's exfiltration or persistence mechanism. Document the exact npm integrity hash of any replaced Axios version as evidence of the compromised tarball.

**Step 4: Recovery — Validate clean dependency state by running software composition analysis (SCA) tooling against all production builds before redeployment. Confirm audit logging is intact and has not been tampered with per AU-9 (Protection of Audit Information). Re-verify MFA enforcement on all CI/CD pipeline service accounts and developer accounts per CIS 6.3 and CIS 6.5. Monitor post-fix for re-compromise indicators: unexpected package version changes, new outbound connections from build systems, or credential reuse alerts.**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery

**Controls:** AU-9 (Protection Of Audit Information), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.5 (Require MFA for Administrative Access), AU-11 (Audit Record Retention)

**Compensating:** Without enterprise SCA, run `npm audit --audit-level=moderate` and supplement with the free OWASP Dependency-Check CLI (`dependency-check.sh --project --scan ./`) against each production build artifact before redeployment. To verify audit log integrity without a SIEM, compute SHA-256 hashes of CI/CD log archives immediately after collection (`sha256sum *.log > log-manifest.sha256`) and store off-system — this provides a tamper-evidence baseline consistent with AU-9. For MFA enforcement verification on GitHub, use `gh api /orgs/{org}/members --paginate | jq '.[].login'` combined with `gh api /orgs/{org}/members/{username}` to identify accounts without two_factor_authentication enabled. Set a cron job on build servers: `*/15 * * * * npm ls axios --json | sha256sum >> /var/log/axios-integrity-check.log` to detect silent package version drift post-recovery.

**Evidence:** Before declaring recovery complete, verify that GitHub repository webhook configurations have not been silently modified (Glassworm persistence vector): `gh api /repos/{owner}/{repo}/hooks` should return only known, authorized webhooks. Capture a post-recovery baseline of all npm lockfile hashes and store them as a signed manifest — any future deviation is a re-compromise indicator. On recovered macOS endpoints, run `sudo log show --predicate 'eventMessage contains "LaunchAgent"' --last 24h` to confirm no new OpenClaw-style persistence has re-registered. Document the exact timestamp of recovery completion and the first clean CI/CD build hash for each affected pipeline, as these serve as the forensic baseline for post-incident monitoring.

**Step 5: Post-Incident — Conduct a supply chain dependency review to enforce signed package verification and implement a software bill of materials (SBOM) process aligned to CIS 2.1 and NIST SI-7 (no mapped control from verified KB for SI-7 — note: use only KB-verified control IDs; CIS 2.1 is confirmed). Review insider threat controls: validate that contractor and IT worker onboarding includes identity verification steps that can detect DPRK fraudulent worker infiltration (AC-2 — Account Management). Document gaps in dependency integrity verification and prioritize remediation in the vulnerability management process per CIS 7.1 (Establish and Maintain a Vulnerability Management Process).**

**NIST Phase:** Post Incident

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

**Controls:** CIS 2.1 (Establish and Maintain a Software Inventory), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), AC-2 (Account Management), AC-5 (Separation Of Duties)

**Compensating:** To implement SBOM generation without enterprise tooling, integrate the free Syft CLI (`syft packages dir:. -o spdx-json > sbom.spdx.json`) into each CI/CD pipeline's build step — this produces a CycloneDX or SPDX-format SBOM for every build artifact at no cost. For DPRK fraudulent IT worker detection in contractor onboarding — a confirmed STARDUST CHOLLIMA TTPs vector — implement video-verified identity checks cross-referenced against government-issued ID, flag mismatches between stated location and VPN/IP geolocation, and require live keyboard interaction during technical screening (DPRK operators have used remote-desktop proxies to pass asynchronous screenings). Document the SBOM baseline in a version-controlled repository so any dependency drift is auditable via `git diff` without a dedicated SCA platform.

**Evidence:** Post-incident, preserve the complete lessons-learned record including: the exact Axios versions confirmed compromised and the npm publish timestamps from the STARDUST CHOLLIMA window; the full list of GitHub repositories touched by Glassworm unauthorized access, including the diff of any modified Actions workflow files; macOS endpoint forensic images from OpenClaw-affected machines with chain-of-custody documentation; and any identity records (access logs, onboarding documents, VPN connection metadata) associated with contractor accounts flagged during the DPRK fraudulent worker review under AC-2. This artifact package supports both internal lessons-learned review and potential CISA or sector ISAC intelligence sharing under NIST 800-61r3 §4 recommendations for post-incident coordination.

## Detection Guidance

Supply chain: Run software composition analysis against all Node.js/npm dependencies and flag any Axios package versions pulled during the compromise window identified in CrowdStrike and Sonatype reporting. Alert on package.json or package-lock.json modifications in monitored repositories outside approved change windows. CI/CD pipeline: Alert on unexpected outbound network connections from build agents, especially to unknown external IPs from npm postinstall scripts. GitHub: Review repository audit logs for unauthorized Actions workflow changes, new outside collaborator additions, or force-push events to protected branches (Glassworm campaign vector). macOS endpoints: Hunt for unsigned binaries in ~/Downloads or /tmp claiming to be legitimate tools (OpenClaw lure pattern); audit LaunchAgents and LaunchDaemons for new persistence entries. Apply system file analysis and system initialization configuration analysis countermeasures. Identity: Alert on password spray patterns per T1110.003, multiple failed logons across accounts in short windows (AC-7, Unsuccessful Logon Attempts). Monitor for Valid Account abuse (T1078) via anomalous login times, geolocation mismatches, or logins from unexpected ASNs. Audit local account activity for anomalies on developer workstations. Insider threat: Flag new IT contractor accounts with anomalous onboarding patterns, inconsistent identity documents, VPN routing through DPRK-adjacent infrastructure, or rapid access requests to sensitive repositories shortly after hire.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| DOMAIN | `no specific IOCs published in available sources at time of generation` | CrowdStrike and Sonatype reporting does not surface specific IOC values in the referenced blog posts as indexed; consult the full CrowdStrike intelligence portal and Sonatype advisory directly for current IOC lists | LOW |

## Framework Mappings

### MITRE-ATTACK

- **T1199** — Trusted Relationship
- **T1133** — External Remote Services
- **T1586** — Compromise Accounts
- **T1657** — Financial Theft
- **T1059** — Command and Scripting Interpreter
- **T1486** — Data Encrypted for Impact
- **T1213** — Data from Information Repositories
- **T1083** — File and Directory Discovery
- **T1608.001** — Upload Malware
- **T1110.003** — Password Spraying
- **T1078** — Valid Accounts
- **T1566** — Phishing
- **T1588.001** — Malware
- **T1195.002** — Compromise Software Supply Chain

### NIST-800-53R5

- **AC-17** — Remote Access
- **AC-20** — Use of External Systems
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **SC-7** — Boundary Protection
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-7** — Software, Firmware, and Information Integrity
- **CP-9** — System Backup
- **CP-10** — System Recovery and Reconstitution

- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SI-8** — Spam Protection
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **CM-3** — Configuration Change Control
- **IA-8** — Identification and Authentication (Non-Organizational Users)
- **SR-2** — Supply Chain Risk Management Plan

### OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures
- **A07:2021** — Identification and Authentication Failures

### CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **6.3** — Require MFA for Externally-Exposed Applications
- **6.4** — Require MFA for Remote Network Access
- **6.5** — Require MFA for Administrative Access
- **15.1** — Establish and Maintain an Inventory of Service Providers

### 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(d)** — Person or Entity Authentication

### 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

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1199 | Trusted Relationship | Initial-Access |
| T1133 | External Remote Services | Persistence |

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1586 | Compromise Accounts | Resource-Development |
| T1657 | Financial Theft | Impact |
| T1059 | Command and Scripting Interpreter | Execution |
| T1486 | Data Encrypted for Impact | Impact |
| T1213 | Data from Information Repositories | Collection |
| T1083 | File and Directory Discovery | Discovery |
| T1608.001 | Upload Malware | Resource-Development |
| T1110.003 | Password Spraying | Credential-Access |
| T1078 | Valid Accounts | Defense-Evasion |
| T1566 | Phishing | Initial-Access |
| T1588.001 | Malware | Resource-Development |
| T1195.002 | Compromise Software Supply Chain | Initial-Access |

## Sources

| Source | URL | Tier |
|---|---|---|
| **Blog** | https://www.crowdstrike.com/en-us/blog/crowdstrike-2026-technology-... | T3 |
| **STARDUST CHOLLIMA Likely Compromises Axios npm Package** | https://www.crowdstrike.com/en-us/blog/stardust-chollima-likely-com... | T3 |
| **Axios Compromise on npm Introduces Hidden Malicious Package** | https://www.sonatype.com/blog/axios-compromise-on-npm-introduces-hi... | T3 |
| **STARDUST CHOLLIMA Compromises Axios npm Package - LinkedIn** | https://www.linkedin.com/posts/steven-stover-64329414_stardust-chol... | T3 |
| **Examining the Blast Radius from the Axios npm Supply Chain ...** | https://www.esentire.com/blog/examining-the-blast-radius-from-the-a... | T3 |

Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-06-10 07:22 UTC by TJS Security Command Center