INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-20 18:47 UTC

Technology Sector Under Siege: China and DPRK Drive State-Sponsored Intrusions While eCrime Extortion Surges

THREAT CAMPAIGN | **HIGH** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0526
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	9.5
Affected Products	GitHub repositories, Axios npm package (v1.14.1, v0.30.4), macOS platforms, mail infrastructure, private code repositories (unnamed software development company), broad technology sector organizations
Discovery Source	Rss:T1 Threatintel

Executive Summary

CrowdStrike's 2026 Technology Threat Landscape Report documents a coordinated, multi-vector assault on the technology sector: China-nexus adversaries drove more than 58% of state-sponsored intrusions targeting intellectual property and software development infrastructure, while DPRK's FAMOUS CHOLLIMA group accounted for 47% of all hands-on-keyboard state-sponsored operations, consistent with insider threat placement. A concrete embedded incident illustrates the software supply chain risk: the widely used Axios npm package (v1.14.1 and v0.30.4) was compromised through a hijacked maintainer account, delivering a remote access trojan to downstream build pipelines. Simultaneously, eCrime extortion groups named 572 technology organizations on data leak sites, and the volume of IAB-purchased access listings targeting the tech sector rose nearly 30% year-over-year, signaling a commoditized, scalable threat market that compounds state-sponsored pressure.

Technical Analysis

The report documents three converging threat vectors. First, China-nexus clusters pursued persistent access to software development infrastructure and IP repositories, consistent with MITRE T1213 (Data from Information Repositories) and T1195.001 (Compromise Software Dependencies and Development Tools). Second, FAMOUS CHOLLIMA (DPRK) conducted hands-on-keyboard intrusions consistent with T1586.002 (Compromise Accounts: Email Accounts), T1059 (Command and Scripting Interpreter), and T1078 (Valid Accounts), reflecting documented insider threat placement operations. Third, the Axios npm supply chain

compromise (v1.14.1 and v0.30.4) involved a hijacked maintainer account (T1586.002) used to publish malicious package versions containing a RAT (T1219, Remote Access Software; T1195.002, Compromise Software Supply Chain). The malicious payload aligns with CWE-494 (Download of Code Without Integrity Check) and CWE-829 (Inclusion of Functionality from Untrusted Control Sphere). Across multiple intrusion clusters, initial access was enabled by authentication weaknesses (CWE-287) and weak credential practices (CWE-521), mapped to T1110.003 (Password Spraying) and T1078 (Valid Accounts). eCrime actors exploited IAB-purchased access (T1588.001, Obtain Capabilities: Malware; T1608.003, Stage Capabilities: Install Digital Certificate) to pursue extortion (T1657) and data exfiltration (T1567.001, Exfiltration to Code Repository). No CVE ID is assigned to the Axios compromise; the attack is characterized by CWE-494 and CWE-829. Affected Axios versions are v1.14.1 and v0.30.4; remediation requires upgrading to a clean release confirmed by the axios maintainers via the GitHub post-mortem (issue #10636).

Action Checklist

1. Step 1: Containment. Immediately audit all package.json, package-lock.json, and yarn.lock files across build pipelines and developer workstations for Axios versions v1.14.1 and v0.30.4. Block these versions at your artifact registry (e.g., Artifactory, Nexus) and in CI/CD pipeline dependency resolution. For FAMOUS CHOLLIMA and China-nexus exposure, query EDR and authentication logs for anomalous interactive sessions on code repositories or development infrastructure (e.g., off-hours logins, new device enrollments, access from unexpected geographies). Isolate flagged accounts pending investigation (NIST AC-2, Account Management; NIST AC-3, Access Enforcement).
2. Step 2: Detection. Query SIEM and EDR for outbound connections originating from build agents or developer endpoints following npm install events; flag any unexpected C2 beaconing post-build. Search package audit logs for installation of Axios v1.14.1 or v0.30.4 (npm audit log, registry pull logs). For insider threat vectors, review authentication logs for FAMOUS CHOLLIMA TTPs: off-hours VPN logins, new device enrollments, and access to private code repositories by recently onboarded contractors or employees (NIST AU-6, Audit Record Review, Analysis, and Reporting; NIST AU-12, Audit Record Generation; CIS 8.2, Collect Audit Logs). Behavioral indicator: RAT activity from Axios compromise may manifest as persistent low-volume outbound sessions from CI/CD runners to non-CDN IPs. Note: RAT beaconing signatures depend on the specific malware variant. Consult the axios post-mortem (issue #10636) or vendor threat reports for known C2 indicators and beacon patterns. Apply NIST SI-7 (Software, Firmware, and Information Integrity) to detect modifications to build artifacts post-install.
3. Step 3: Eradication. Upgrade Axios to a clean version confirmed safe in the axios GitHub security advisory or official release notes (see https://github.com/axios/axios/releases); do not remain on v1.14.1 or v0.30.4. Re-run full builds from clean dependency resolution after version pinning is updated. If your organization publishes npm packages, rotate all npm publish credentials and maintainer tokens. For accounts suspected of FAMOUS CHOLLIMA compromise or China-nexus intrusion, disable, re-credential, and audit all associated access tokens, SSH keys, and OAuth grants (NIST AC-2, Account Management; NIST IA-5, Authentication and Authorization). Enforce MFA on all npm package maintainer accounts and code repository admin roles (CIS 6.5, Require MFA for Administrative Access; NIST IA-2, Authentication and Authorization).
4. Step 4: Recovery. Validate that clean Axios versions are deployed across all environments by re-running npm audit and confirming no flagged versions remain in lock files. Monitor build pipeline outputs for anomalous network behavior for at least 30 days post-remediation using EDR and network flow analysis. Restore any affected CI/CD runners from verified clean images rather than patching in place.

Confirm audit logging is continuous and log integrity is protected (NIST AU-9, Protection of Audit Information; NIST AU-11, Audit Record Retention). For accounts involved in insider threat scenarios, enforce re-authentication and privileged access reviews before restoring elevated permissions (NIST AC-6, Least Privilege).

5. Step 5: Post-Incident. Conduct a software supply chain dependency audit across all internal and third-party packages; implement package integrity verification (e.g., npm provenance, Sigstore) to address CWE-494 and CWE-829 at the process level. Establish or review insider threat detection procedures given FAMOUS CHOLLIMA's documented pattern of personnel placement; include identity verification steps for new engineering hires and contractors with access to source code (NIST AC-5, Separation of Duties; CIS 5.1, Establish and Maintain an Inventory of Accounts; CIS 5.3, Disable Dormant Accounts). Review IAB exposure by searching dark web sources for your organization's credentials or access listings; implement a formal vulnerability management process covering third-party dependencies (CIS 7.1, Establish and Maintain a Vulnerability Management Process; CIS 7.2, Establish and Maintain a Remediation Process).

IR / Forensic Enrichment

Triage Priority	IMMEDIATE
Escalation Criteria	Escalate to executive leadership, legal counsel, and your sector ISAC immediately if npm audit or registry pull logs confirm that Axios v1.14.1 or v0.30.4 was installed in a pipeline that built and published packages consumed by external customers or partners, or if forensic review of a flagged FAMOUS CHOLLIMA or China-nexus account reveals access to repositories containing proprietary source code, credentials, or data subject to breach notification requirements under applicable regulations.
Recovery Notes	Rebuild all CI/CD runners from verified clean base images rather than patching in place, as the malicious Axios install lifecycle hook may have persisted implants outside the node_modules directory. Validate every downstream package published from affected pipelines during the compromise window by comparing published checksums against pre-compromise build manifests — any discrepancy indicates potential supply chain poisoning of your own packages requiring customer notification. Maintain elevated network monitoring on restored runner environments for a minimum of 30 days, specifically watching for low-volume persistent outbound sessions from node.exe to non-CDN IP ranges, which is the documented behavioral signature of the RAT implanted via the compromised Axios versions.

Forensic Artifacts	npm debug and timing logs (~/.npm/_logs/ on Linux, %APPDATA%\npm-cache_logs\ on Windows) capturing the full install lifecycle hook execution sequence for Axios v1.14.1 or v0.30.4, including any preinstall/postinstall script invocations that would reveal the RAT dropper execution | Network flow records (Zeek conn.log, NetFlow, or pcap) timestamped to within 60 seconds of the malicious npm install event, filtered for outbound TCP sessions from node.exe or CI/CD runner processes to non-Cloudflare, non-npmjs.com IP addresses — the primary artifact of the embedded RAT's C2 beacon | GitHub or GitLab audit log exports (JSON format via API) for all accounts matching FAMOUS CHOLLIMA behavioral indicators: contractor or recently onboarded employee accounts performing off-hours repository clones, new SSH key registrations, or bulk private repository access within 30 days of hire | Memory image (via WinPmem or LiME kernel module) of any CI/CD runner process showing active network connections at time of discovery, preserving in-memory C2 configuration, encryption keys, and staging data that the RAT would have loaded from the malicious Axios package before volatile state is destroyed by isolation | Cryptographic hashes (SHA-256) and provenance metadata for all packages published from affected pipelines during the compromise window, cross-referenced against npmjs.com registry metadata to detect whether the malicious Axios version introduced unauthorized modifications into any packages your organization publishes to external consumers

Per-Action IR Details

Step 1: Containment — Immediately audit all package.json, package-lock.json, and yarn.lock files across build pipelines and developer workstations for Axios versions v1.14.1 and v0.30.4. Block these versions at your artifact registry (e.g., Artifactory, Nexus) and in CI/CD pipeline dependency resolution. For FAMOUS CHOLLIMA and China-nexus exposure, isolate any employee or contractor accounts flagged for anomalous interactive sessions on code repositories or development infrastructure (NIST AC-2 — Account Management; NIST AC-3 — Access Enforcement).

NIST Phase: Containment

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

Controls: NIST AC-2 (Account Management), NIST AC-3 (Access Enforcement), NIST AC-4 (Information Flow Enforcement)

Compensating: Use 'grep -r "axios" --include="*.json" /path/to/repos' across all developer workstations and CI/CD runner file systems to locate pinned versions in package.json and lock files. Block v1.14.1 and v0.30.4 in Nexus OSS or Artifactory CE using repository-level blacklist rules (both are free tiers). For account isolation without enterprise IAM, disable the GitHub or GitLab user account via CLI ('gh api -X PATCH /users/{username} --field suspended=true') and revoke all active personal access tokens from the admin console immediately.

Evidence: Before isolating flagged accounts or blocking registry access, capture: (1) active session tokens and OAuth grants currently associated with the flagged npm maintainer or developer account (export via 'npm token list' and GitHub API '/user/installations'); (2) full network connection state on CI/CD runners at time of discovery using 'netstat -ano' (Windows) or 'ss -tunap' (Linux) to document any live outbound connections to non-CDN IPs established post-npm-install; (3) process tree on runner hosts via 'ps auxf' (Linux) or 'Get-WmiObject Win32_Process' (Windows) to identify any child processes spawned by the malicious Axios package during install lifecycle hooks; (4) memory image of any runner process showing anomalous network activity before session revocation destroys volatile state.

Step 2: Detection — Query SIEM and EDR for outbound connections originating from build agents or developer endpoints following npm install events; flag any unexpected C2 beaconing post-build. Search package audit logs for installation of Axios v1.14.1 or v0.30.4 (npm audit log, registry pull logs). For insider threat vectors, review authentication logs for FAMOUS CHOLLIMA TTPs: off-hours VPN logins, new device enrollments, and access to private code repositories by recently onboarded contractors or employees (NIST AU-6 — Audit Record Review, Analysis, and Reporting; NIST AU-12 — Audit Record Generation; CIS 8.2 —

Collect Audit Logs). Behavioral indicator: RAT activity from Axios compromise may manifest as persistent low-volume outbound sessions from CI/CD runners to non-CDN IPs. Apply D3-SFA (System File Analysis) to detect modifications to build artifacts post-install.

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-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Without a SIEM, run Zeek or tcpdump on the CI/CD network segment and parse for low-volume persistent outbound TCP sessions to non-CDN IP ranges initiated within 60 seconds of an npm install event — this is the RAT beacon timing window for supply chain implants. Use 'npm audit --json' and cross-reference registry pull logs in ~/.npm/_logs/ or /var/lib/jenkins/.npm/ for Axios v1.14.1 or v0.30.4 pull timestamps. For FAMOUS CHOLLIMA insider detection without EDR, deploy Sysmon with EventID 3 (Network Connection) and EventID 1 (Process Create) filtering on node.exe and npm.cmd as parent processes; export to Windows Event Forwarding for centralized review. Query GitHub audit log API ('/orgs/{org}/audit-log?phrase=action:git.clone&include=git') for off-hours clone activity by contractor accounts onboarded within the past 90 days.

Evidence: Preserve before any remediation action: (1) npm debug and timing logs from affected runner at ~/.npm/_logs/ containing the exact install sequence and lifecycle hook execution for Axios v1.14.1 or v0.30.4; (2) network flow records (NetFlow, pcap, or Zeek conn.log) timestamped to the npm install event showing any outbound session to a non-npmjs.com, non-Cloudflare IP; (3) GitHub or GitLab audit log exports filtered for the flagged contractor/employee account showing repository access, clone events, and branch activity — specifically targeting private repository access outside normal working hours; (4) VPN authentication logs (RADIUS or ZTNA) for the flagged FAMOUS CHOLLIMA-pattern account showing source IP geolocation, device fingerprint, and session duration anomalies; (5) Sysmon EventID 1 records showing node.exe or npm.cmd spawning unexpected child processes (cmd.exe, powershell.exe, curl, or bash) during the install lifecycle hook phase.

Step 3: Eradication — Upgrade Axios to a clean version confirmed safe in the axios GitHub post-mortem (issue #10636); do not remain on v1.14.1 or v0.30.4. Re-run full builds from clean dependency resolution after version pinning is updated. Rotate all npm publish credentials and maintainer tokens for your organization's packages (NIST AC-2 — Account Management; D3-CRO — Credential Rotation). For accounts suspected of FAMOUS CHOLLIMA compromise or China-nexus intrusion, disable, re-credential, and audit all associated access tokens, SSH keys, and OAuth grants (NIST AC-2; D3-CH — Credential Hardening). Enforce MFA on all npm package maintainer accounts and code repository admin roles (CIS 6.5 — Require MFA for Administrative Access; D3-MFA — Multi-factor Authentication).

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-2 (Account Management), CIS 6.5 (Require MFA for Administrative Access), CIS 7.4 (Perform Automated Application Patch Management)

Compensating: Pin Axios to the confirmed clean version in all package.json files using 'npm install axios@ --save-exact' and commit updated lock files. Use 'npm token revoke ' to invalidate all existing npm publish tokens; regenerate with scope limited to specific package names only. For SSH key audit across GitHub, use the GitHub API ('GET /users/{username}/keys') to enumerate all registered public keys for flagged accounts and revoke any unrecognized entries. Enforce TOTP-based MFA on npm accounts (npmjs.com account settings) and GitHub admin roles at no cost using GitHub's built-in MFA enforcement policy at the organization level ('Settings > Authentication security > Require two-factor authentication').

Evidence: Before rotating credentials and revoking tokens — which destroys the forensic record of what was authorized — capture: (1) full export of all active npm tokens and their creation timestamps, scopes, and last-used dates via 'npm token list --json' for every maintainer account; (2) complete list of SSH keys, OAuth app authorizations, and GitHub Personal Access Tokens (PATs) associated with the flagged FAMOUS CHOLLIMA or China-nexus account via GitHub API '/user/keys', '/user/installations', and '/authorizations'; (3) git reflog and commit history for any private repositories the flagged account had write access to, to identify whether unauthorized commits, tags, or branch

modifications were made; (4) any build artifact checksums (SHA-256) from pipelines that executed with the malicious Axios version, preserving evidence of what was built and potentially distributed downstream before eradication.

Step 4: Recovery — Validate that clean Axios versions are deployed across all environments by re-running npm audit and confirming no flagged versions remain in lock files. Monitor build pipeline outputs for anomalous network behavior for at least 30 days post-remediation using EDR and network flow analysis. Restore any affected CI/CD runners from verified clean images rather than patching in place. Confirm audit logging is continuous and log integrity is protected (NIST AU-9 — Protection of Audit Information; NIST AU-11 — Audit Record Retention). For accounts involved in insider threat scenarios, enforce re-authentication and privileged access reviews before restoring elevated permissions (NIST AC-6 — Least Privilege).

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-9 (Protection of Audit Information), NIST AU-11 (Audit Record Retention), NIST AC-6 (Least Privilege), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: Rebuild CI/CD runners from a known-good base image stored in a version-controlled, immutable registry (e.g., Docker Hub private repo or a local Harbor instance) rather than attempting in-place remediation on potentially compromised runner environments. Validate clean Axios deployment by running 'npm audit --audit-level=critical' and 'cat package-lock.json | grep -A2 "axios"' across all environment lock files. For post-recovery monitoring without EDR, configure Sysmon EventID 3 on runner hosts to alert on any outbound connection from node.exe to IP ranges outside known npm CDN CIDR blocks (104.16.0.0/12 for Cloudflare-backed npmjs.com). Store audit logs to a write-once, off-host destination (e.g., AWS S3 with Object Lock or a syslog server with append-only configuration) to satisfy AU-9 integrity requirements.

Evidence: Before restoring elevated permissions to accounts involved in the FAMOUS CHOLLIMA or China-nexus insider scenario, capture and retain: (1) the complete access review record documenting which repositories, secrets, and pipeline configurations the account accessed during the compromise window — this is required for regulatory notification assessments; (2) network flow baselines from the first 72 hours post-recovery on restored CI/CD runners to establish a clean behavioral baseline for anomaly detection during the 30-day monitoring window; (3) cryptographic hashes (SHA-256) of all build artifacts produced from the first clean build post-eradication, stored in an integrity-protected manifest, to provide a verified reference point for downstream consumers of your packages.

Step 5: Post-Incident — Conduct a software supply chain dependency audit across all internal and third-party packages; implement package integrity verification (e.g., npm provenance, Sigstore) to address CWE-494 and CWE-829 at the process level. Establish or review insider threat detection procedures given FAMOUS CHOLLIMA's documented pattern of personnel placement; include identity verification steps for new engineering hires and contractors with access to source code (NIST AC-5 — Separation of Duties; CIS 5.1 — Establish and Maintain an Inventory of Accounts; CIS 5.3 — Disable Dormant Accounts). Review IAB exposure by searching dark web sources for your organization's credentials or access listings; implement a formal vulnerability management process covering third-party dependencies (CIS 7.1 — Establish and Maintain a Vulnerability Management Process; CIS 7.2 — Establish and Maintain a Remediation Process).

NIST Phase: Post Incident

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

Controls: NIST AC-5 (Separation of Duties), CIS 5.1 (Establish and Maintain an Inventory of Accounts), CIS 5.3 (Disable Dormant Accounts), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Implement npm provenance attestation at zero cost by enabling it in GitHub Actions with 'npm publish --provenance' — this cryptographically links published packages to the source commit and CI workflow, directly addressing the Axios supply chain injection vector. Use the free SBOM generation tool 'syft' to produce a CycloneDX or SPDX software bill of materials for all internal packages and scan against OSV.dev (free, Google-maintained) for known vulnerabilities in transitive dependencies. For FAMOUS CHOLLIMA insider threat detection procedures, implement a documented 'two-person integrity' rule requiring that any new contractor granted access to private source

code repositories must have their GitHub account and associated SSH keys co-verified by both a hiring manager and a security team member before provisioning — achievable with a simple documented checklist and enforced via a GitHub branch protection rule requiring two approvers for repository access grants.

Evidence: Post-incident artifacts to preserve for lessons-learned, regulatory review, and threat intelligence sharing: (1) complete timeline of the Axios v1.14.1/v0.30.4 installation events across all affected pipelines, reconstructed from npm debug logs and registry pull records, documenting the full blast radius; (2) final access review report for all accounts involved in the FAMOUS CHOLLIMA or China-nexus scenario, including a list of repositories, secrets managers, and pipeline configurations touched during the compromise window — required if PII or regulated data was accessible; (3) all IOCs extracted during detection_analysis (C2 IP addresses, beacon intervals, process ancestry chains from node.exe) formatted as STIX 2.1 for sharing with sector ISACs (e.g., IT-ISAC) per NIST 800-61r3 §4 intelligence sharing guidance.

Detection Guidance

Priority detection surfaces: (1) Package registry pull logs, query for any installation of axios@1.14.1 or axios@0.30.4 across all CI/CD pipelines, developer workstations, and artifact caches. (2) Network telemetry from build agents, flag outbound connections to non-CDN, non-registry endpoints initiated within 60 seconds of a npm install event; RAT beaconing from compromised Axios versions may present as low-interval polling to adversary-controlled IPs. Note: RAT beaconing signatures depend on the specific malware variant. Consult the axios post-mortem (issue #10636) or vendor threat reports for known C2 indicators and beacon patterns. (3) Authentication anomalies consistent with FAMOUS CHOLLIMA TTPs (T1078, T1586.002): new MFA device enrollments paired with off-hours logins, access to private GitHub repositories or internal code stores from geolocations inconsistent with employee history, and service account activity outside normal job function. (4) Password spraying indicators (T1110.003, CWE-521): multiple failed authentications across different accounts from shared source IPs, particularly targeting developer portals, npm registries, and VPN endpoints. (5) Data staging and exfiltration (T1567.001): large or unusual commits to external repositories, or file transfers from internal code stores to personal or unmanaged cloud storage. Apply NIST AU-12 (Audit Record Generation) for interactive sessions on build infrastructure. Enable NIST AU-2 (Event Logging) across CI/CD tooling if not already active, ensuring log coverage for package install events, authentication events, and outbound network connections from build runners.

Indicators of Compromise

Type	Value	Context	Confidence
HASH	`not published in source material`	Malicious Axios v1.14.1 and v0.30.4 package hashes were not disclosed in the referenced sources; obtain from npm registry integrity records and the axios GitHub post-mortem (issue #10636)	LOW
URL	`https://github.com/axios/axios/issues/10636`	Official axios post-mortem for the npm supply chain compromise; use to obtain clean version confirmation and incident timeline	HIGH

Framework Mappings

MITRE-ATTACK

- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1219** — Remote Access Tools
- **T1213** — Data from Information Repositories
- **T1195.002** — Compromise Software Supply Chain
- **T1110.003** — Password Spraying
- **T1586.002** — Email Accounts
- **T1566** — Phishing
- **T1588.001** — Malware
- **T1059** — Command and Scripting Interpreter
- **T1657** — Financial Theft
- **T1608.003** — Install Digital Certificate
- **T1586.003** — Cloud Accounts
- **T1567.001** — Exfiltration to Code Repository
- **T1078** — Valid Accounts

NIST-800-53R5

- **CM-7** — Least Functionality
- **SA-9** — External System Services
- **SR-3** — Supply Chain Controls and Processes
- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness
- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **AC-2** — Account Management
- **AC-6** — Least Privilege
- **IA-2** — Identification and Authentication (Organizational Users)
- **IA-5** — Authenticator Management
- **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
- **164.308(a)(6)(ii)** — Response and Reporting

NIST-CSF-2

- **RS.CO-03** — Recovery activities and progress communicated
- **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
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1219	Remote Access Tools	Command-And-Control
T1213	Data from Information Repositories	Collection
T1195.002	Compromise Software Supply Chain	Initial-Access
T1110.003	Password Spraying	Credential-Access
T1586.002	Email Accounts	Resource-Development
T1566	Phishing	Initial-Access
T1588.001	Malware	Resource-Development
T1059	Command and Scripting Interpreter	Execution
T1657	Financial Theft	Impact

Technique ID	Technique Name	Tactic
T1608.003	Install Digital Certificate	Resource-Development
T1586.003	Cloud Accounts	Resource-Development
T1567.001	Exfiltration to Code Repository	Exfiltration
T1078	Valid Accounts	Defense-Evasion

Sources

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

Generated 2026-06-20 18:47 UTC by TJS Security Command Center