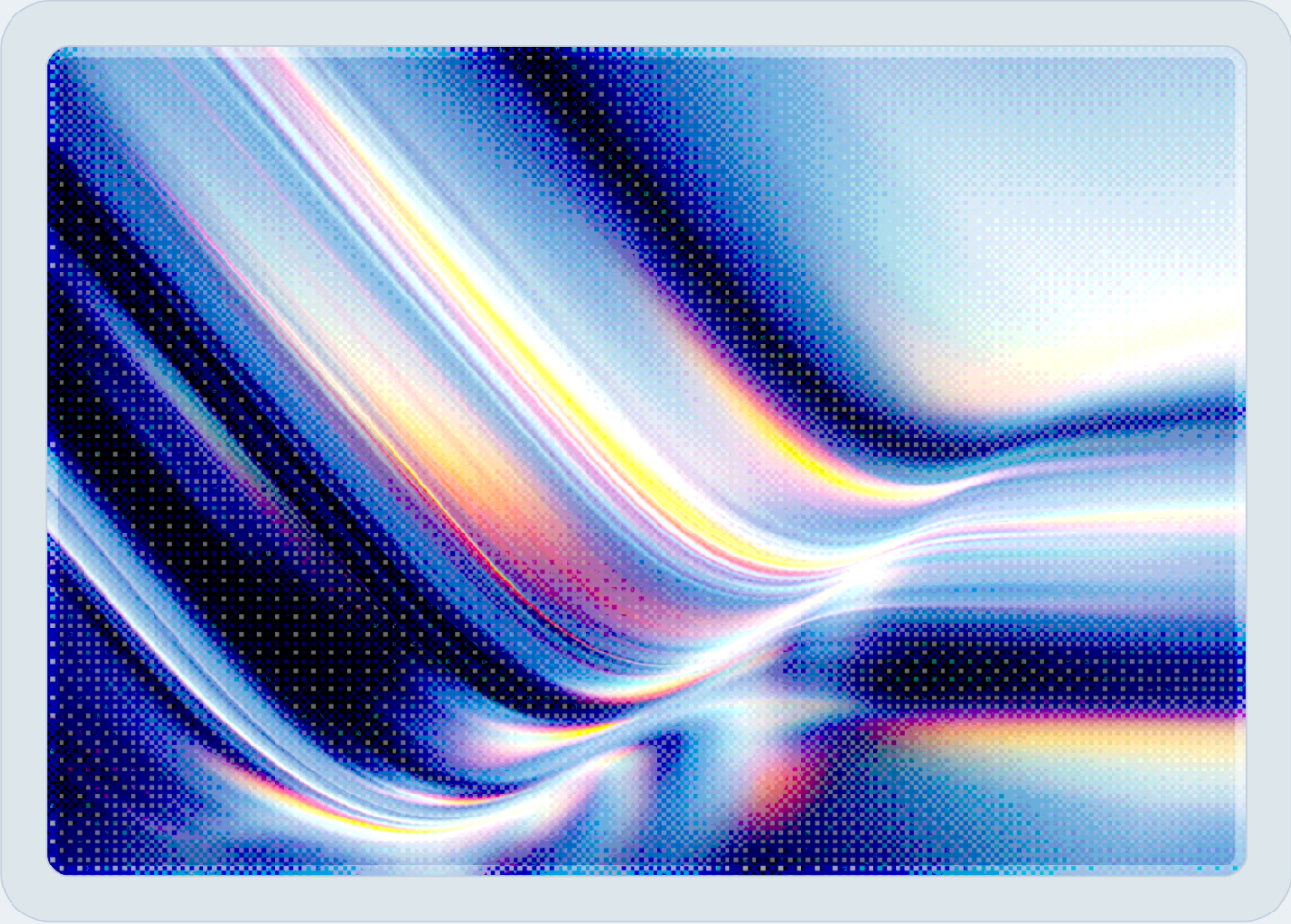
CSAI Foundation | Cloud Security Alliance

JINX-0164: Developer Targeting in CI/CD and Crypto Pipelines

2026-05-29

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- JINX-0164 is a financially motivated threat actor, named and disclosed on May 27, 2026 by Wiz Research, that targets cryptocurrency organizations by impersonating job recruiters on LinkedIn to deliver macOS malware and pivot from developer workstations into CI/CD pipelines [1].
- The actor operates two custom malware families — AUDIOFIX, a Python-based infostealer and remote access trojan targeting 51 cryptocurrency wallet browser extensions, and MINIRAT, a Go-based backdoor — and exploited the `@velora-dex/sdk` npm package on April 7, 2026, demonstrating the convergence of social engineering and supply chain compromise [1] [5].
- Lateral movement from compromised developer identities into GitHub Actions pipelines was facilitated by the open-source `nord-stream` tool, which extracts CI/CD secrets via stolen GitHub tokens, illustrating how a single developer compromise can escalate to code distribution infrastructure [1][3].
- JINX-0164 operates within a substantially larger threat landscape: North Korean state-linked actors stole $2.02 billion in cryptocurrency in 2025 alone, and Sonatype documented a 156% year-over-year increase in malicious package uploads, with cumulative totals exceeding 1.233 million packages [16][29].
- The pattern JINX-0164 represents — targeting developers as a privileged entry point into the software supply chain rather than attacking end-user systems directly — has become an increasingly dominant attack model, particularly for actors targeting cryptocurrency infrastructure, and warrants defensive postures that treat developer identities as high-value targets [1][13].

Background

For much of the preceding decade, the dominant model of cryptocurrency theft involved attacking exchange infrastructure directly: compromising hot wallets, exploiting smart contract logic, or targeting custodial platform credentials. That model has not disappeared, but it has been joined by a more structurally insidious alternative. Developers who build and maintain cryptocurrency infrastructure hold a uniquely privileged position in the attack surface: their workstations contain signing keys, cloud provider

credentials, SSH certificates, and GitHub access tokens that collectively represent the ability to modify the software that millions of users run. A threat actor who successfully compromises a developer gains not merely access to that individual's assets but a potential pivot point into the full production environment and every downstream consumer of the packages they maintain.

The conditions that have made developer targeting so attractive in 2025 and 2026 are structural rather than incidental. The open-source software supply chain aggregates enormous leverage into individual maintainer accounts: a single compromised npm token can be used to publish malicious code to packages with hundreds of millions of weekly downloads, as demonstrated when phishing of a single maintainer exposed the chalk, debug, and supports-color packages — among 18 compromised in the operation — collectively accounting for approximately two billion weekly downloads in September 2025 [12]. GitHub Actions workflows create a further attack surface, because the CI/CD runner environments in which these workflows execute are routinely seeded with cloud provider keys, deployment secrets, and package registry credentials that have no equivalent protection to the production systems they control. The tj-actions/changed-files supply chain attack in March 2025, which exposed CI/CD secrets from over 23,000 repositories through manipulation of a single GitHub Action tag, illustrated how far a single point of leverage can propagate [13][27]. Against this backdrop, JINX-0164 represents a threat actor that has assembled a complete attack chain specifically engineered for this environment.

The cryptocurrency sector has absorbed disproportionate losses from this class of attack. North Korean state-linked actors alone stole $2.02 billion in cryptocurrency in 2025 — a 51% year-over-year increase — representing 76% of all cryptocurrency service compromise value despite executing significantly fewer attacks than in prior periods, indicating dramatically higher per-incident yield [16]. The $1.5 billion Bybit theft in February 2025 and the $285 million Drift hack in April 2026, which was traced to a six-month social engineering campaign targeting developers, anchor the upper bounds of what these operations can achieve when developer access translates into control of production infrastructure [16][22]. JINX-0164 does not currently appear to operate at that scale, but it employs identical foundational techniques and has demonstrated the capability to move from a developer's laptop to code distribution systems within weeks.

Threat Actor Profile: JINX-0164

JINX-0164 was named and publicly disclosed on May 27, 2026, in a comprehensive report by Wiz Research and Wiz CIRT, authored by Shira Ayal, Eden Abergil, Andre Maccarone, Yuval Dan, and Benjamin Read [1]. The actor's observed activity extends back to at least mid-2025, and a documented landmark intrusion in early 2026 lasted approximately two weeks before detection. JINX-0164 is assessed as financially motivated with a specific focus on cryptocurrency organizations and the developers who build

their software. No confirmed nation-state attribution has been established; while certain tactical similarities exist with North Korean clusters including BlueNoroff, Contagious Interview, and UNC1069, Wiz researchers confirmed no infrastructure overlaps with those groups. The absence of state attribution does not diminish the actor's technical sophistication — JINX-0164 compiled architecture-aware binaries for both Intel and Apple Silicon, built a Go-based backdoor with AES-CBC encrypted command-and-control, and demonstrated the ability to sustain recruitment-themed social engineering operations over weeks before deploying malware.

The actor's target selection reflects a deliberate theory of access. Rather than attacking cryptocurrency exchanges or wallets directly, JINX-0164 focuses on the developers who build the tooling those platforms depend on. The `@velora-dex/sdk` package that JINX-0164 compromised on April 7, 2026, is a decentralized exchange development toolkit with approximately 2,000 weekly downloads; its users are primarily developers building DeFi applications [1][5]. This choice prioritizes code execution in developer environments — where CI/CD credentials, cloud access keys, and signing certificates are routinely present — over broader consumer-facing exposure. The MINIRAT backdoor's module path (`alibaba.xyz/minirat`) and its Go 1.25.8 compilation timestamp, approximately March 20, 2026, indicate that infrastructure was staged roughly 2.5 weeks before the npm compromise, suggesting deliberate operational planning rather than opportunistic exploitation [4].

JINX-0164 uses commercial VPN services — Astrill VPN, Mullvad VPN, and ExpressVPN — to obscure attribution and operates C2 infrastructure hosted across multiple jurisdictions, with the primary C2 domain `datahub.ink` resolving to a Portland, Oregon host and the npm supply chain staging server located at AS9009 infrastructure in Romania [1][5]. The TLS certificate for `datahub.ink` was issued by Let's Encrypt on April 7, 2026 — the same day as the npm compromise — indicating that operational infrastructure was stood up immediately prior to use, a technique that minimizes the window during which indicators of compromise are exposed to proactive detection.

Attack Mechanics

Social Engineering Lures

JINX-0164's initial access relies entirely on social engineering, with no exploitation of known software vulnerabilities. The actor constructs LinkedIn profiles crafted to appear professionally credible and approaches software developers under the guise of job recruitment. Targets receive an invitation to a virtual meeting, and the meeting link redirects to a spoofed teleconference domain; observed examples include `teams.live.us.org`, `teams.cam`, `teamicrosoft.com`, `bitget-`

`meeting.com`, `us03-slack.online`, and `slktest.live` [1]. Once the target attempts to join the meeting, they are presented with a fabricated technical error — a fake audio or video driver failure — and instructed to download a "fix" from actor-controlled payload delivery domains including `apple.driver-store.com`, `apple.driver-hub.net`, and `apple.driver-update.io` [1].

This technique is not unique to JINX-0164, but the actor has refined it for macOS developer environments specifically. A February 2026 Reddit post described a recruitment impersonation attempt consistent with JINX-0164's methodology, indicating that some targets are suspicious enough to report rather than comply — but the attack's effectiveness depends only on a fraction of targets completing the download [1][3]. The payload, a bash shell script dropper, is architecture-aware: it selects the appropriate binary for Intel x86_64 or Apple Silicon ARM64 systems, installs it via the macOS LaunchAgent persistence mechanism, and establishes communication with the actor's command-and-control infrastructure. The use of a LaunchAgent ensures that the malware survives reboots without requiring privileged access to system directories.

CI/CD Pipeline Compromise

Once AUDIOFIX or MINIRAT is installed on a developer's workstation, the actor begins a systematic credential harvest that targets every identity used in the software development lifecycle. AUDIOFIX collects iCloud Keychain contents, credentials from more than ten browser families, SSH private keys, and cloud provider access credentials spanning AWS, GCP, and Azure [1]. These credentials collectively constitute a complete mapping of the developer's infrastructure access. With a stolen GitHub personal access token in hand, JINX-0164 uses `nord-stream`, a publicly available open-source tool designed to demonstrate GitHub Actions secret exfiltration, to extract secrets stored in CI/CD pipeline environments via the GitHub API [1]. Because the technique requires no modification of workflow files, it avoids generating artifacts in workflow execution logs — though the underlying API calls may appear in the GitHub organization's audit log for accounts that have enabled it.

The documented intrusion in early 2026 illustrates how quickly this access can translate into code distribution capability. Within a two-week window, the actor achieved lateral movement from a developer's compromised laptop to the organization's CI/CD systems and code distribution infrastructure [1]. Direct commits to main branches, branch hijacking, and developer impersonation via commit manipulation were all observed. GitHub's Vigilant Mode feature, which flags commits signed with keys that do not match the committing identity, assisted in detecting the impersonated commits during this intrusion — though the feature must be enabled explicitly by repository administrators [3].

Cryptocurrency Theft Mechanisms

AUDIOFIX's credential theft capabilities are targeted at the cryptocurrency ecosystem, specifically enumerating 51 browser-based wallet extensions covering the breadth of browser-based self-custody solutions used by the developers and users of DeFi applications [1]. Combined with clipboard monitoring — which captures wallet addresses at the moment a user would copy them for a transaction — and Discord, Slack, and Telegram session hijacking, AUDIOFIX provides the actor with multiple simultaneous pathways to intercept or redirect cryptocurrency transfers. The malware also implements TCC (Transparency, Consent, and Control) bypass techniques and deploys password phishing dialogs with live system password validation, confirming that entered passwords are correct before forwarding them to the C2 server [1].

The npm supply chain compromise on April 7, 2026, represents a second, distinct theft pathway. By prepending three lines of malicious code to the `dist/index.js` file of `@velora-dex/sdk@4.9.1`, JINX-0164 ensured that any Node.js environment that `require()`d the package would automatically execute a base64-encoded shell payload, download MINIRAT from the staging server at 89.36.224.5, and register persistence via `launchctl` [5]. The malicious version was available for approximately three hours before a clean version 4.9.2 was released, limiting but not eliminating the exposure window. This vector is particularly dangerous because it targets developers who have no reason to be suspicious: they are simply using a dependency they already trust.

Security Analysis

The risk profile that JINX-0164 presents to enterprises is materially different from the risk presented by commodity malware campaigns targeting end-user populations. Developers are not merely end users of enterprise systems; they are administrators of code distribution pipelines that directly determine what software runs in production environments, what secrets are embedded in CI/CD runners, and what authentication material is used to deploy to cloud infrastructure. A single successful compromise of a developer holding a privileged npm token, GitHub token with repository write access, or cloud provider key with deployment permissions can propagate malicious code to every user of every package that developer maintains. The September 2025 attack on the chalk ecosystem — where a single phishing email sent to one maintainer resulted in 18 packages being compromised within 16 minutes — illustrates how few downstream controls can reliably intercept malicious code once a trusted upstream identity has been captured [12].

The specific techniques JINX-0164 employs exploit systemic weaknesses that are not easily mitigated in isolation. The social engineering lure works because job recruitment outreach to developers on LinkedIn is a routine and expected activity that developers have little basis to treat as inherently suspicious, particularly when the initial contact appears to come from a credible professional profile. The LaunchAgent persistence mechanism requires no administrator privileges and is not unusual enough to generate alerts in most endpoint security configurations. The use of `nord-stream` for CI/CD secret exfiltration exploits GitHub's own APIs and, because it requires no modification of workflow files, avoids generating artifacts in workflow execution logs — though the underlying API calls may appear in the GitHub organization's audit log for accounts that have enabled it. The npm supply chain vector exploits the implicit trust that package consumers extend to the maintainers of dependencies they have evaluated and adopted. Each of these techniques, individually, is hard to detect; in combination, they constitute an attack chain with very few natural detection points.

The broader supply chain context amplifies this risk significantly. JINX-0164 operates in an environment where over 1.233 million malicious packages have been catalogued cumulatively across package registries, with Sonatype documenting 454,600 new malicious packages in 2025 alone [29]. The TrapDoor campaign in May 2026 spread 34 malicious packages across npm, PyPI, and Crates.io; the Megalodon campaign placed 5,718 malicious commits across 5,561 GitHub repositories in a single six-hour window [9][19]. In this environment, the question for enterprises is not whether their developers will encounter malicious packages or social engineering lures, but whether their security controls are designed to minimize the blast radius when a developer is successfully compromised.

For enterprises with development teams working on cryptocurrency, DeFi, or Web3 projects, this threat is directly applicable. For enterprises whose developers use npm packages in any capacity, the supply chain dimension of JINX-0164's attack model is equally relevant: the actor's willingness to compromise even relatively low-download packages such as `@velora-dex/sdk` indicates that targeting is based on the developer population that consumes a package, not merely its download volume.

Recommendations

Immediate Actions

Security teams should immediately distribute the JINX-0164 indicators of compromise published by Wiz Research to endpoint detection and network monitoring teams [1][5]. The C2 domains `datahub.ink`, `cloud-sync.online`, and `byte-io.us`, the staging server at 89.36.224.5, and the spoofed meeting domains listed in the Wiz report should be blocked at the DNS and network

layer. Endpoint teams should query for the MINIRAT LaunchAgent persistence label `com.apple.Terminal.profiler` and configuration cache path `~/Library/Application Support/com.apple.Terminal/.cache` across macOS developer systems [4]. The SHA256 hashes for both MINIRAT binaries should be submitted to endpoint security platforms as indicators: `0a8ab3d16b12d3a453ee5a3208fe04744ad54514ef8ea27bb8fe32679efad270` (ARM64) and `0b028b781950641818800fee2b4bf68e4ef2bcee53fe71a21755275ba108783d` (x86_64) [4][5].

Short-Term Mitigations

Enterprises should enable GitHub Vigilant Mode across all repositories where developers have push access, as this feature was documented as assisting in the detection of developer impersonation during the JINX-0164 intrusion [3]. Repository administrators should enforce branch protection rules requiring signed commits and requiring pull request reviews before merging to main branches. CI/CD pipeline secrets should be audited and rotated, with particular attention to secrets that have been accessible to any developer who has worked on the relevant repositories in the past six months. GitHub Actions workflows should be reviewed for use of the `pull_request_target` trigger in combination with code checkout from forked branches, which has been the root cause of multiple independent supply chain attacks [13][32][34]. Where possible, GitHub Actions tags should be pinned to specific commit SHAs rather than mutable version tags, a mitigation that would have significantly limited the exposure from the tj-actions/changed-files class of attack [30].

Developer security awareness programs should be updated to address the specific social engineering lure pattern JINX-0164 uses: unsolicited LinkedIn job recruitment followed by a virtual meeting invitation and a request to download a "fix" for a fabricated technical problem. Developers should be instructed to verify the legitimacy of meeting platform domains before joining any call associated with an unsolicited recruitment contact. Organizations with cryptocurrency or DeFi development teams should also consider restricting the cryptocurrency wallet browser extensions permitted in developer browser profiles, given that AUDIOFIX specifically targets 51 such extensions [1].

Strategic Considerations

The JINX-0164 campaign illustrates a structural challenge that cannot be solved by tactical controls alone. Developer identities — their SSH keys, GitHub tokens, npm credentials, and cloud access keys — are increasingly the highest-value targets in enterprise environments because they provide access to code distribution pipelines that downstream controls cannot effectively monitor. Enterprises should

consider adopting a developer identity security program that treats these credentials with the same rigor applied to privileged access management for system administrators, including regular rotation, hardware security key enforcement for code signing and registry publishing, and out-of-band verification for publishing actions above defined thresholds.

The supply chain dimension argues for adoption of software composition analysis tools that provide continuous monitoring of dependency trees rather than point-in-time scans, combined with lockfile integrity verification that detects unexpected changes between the version of a package a developer evaluated and the version that installs at build time. The signed artifact problem — where malicious packages can carry valid SLSA and Sigstore attestations because the CI/CD pipeline that built them was compromised before signing — means that cryptographic provenance alone is insufficient; behavioral monitoring of what packages do at install time provides a complementary signal [32][33]. Dependency review policies that require internal approval before developers can add new npm dependencies to repositories reduce the attack surface from both typosquatting and supply chain compromise vectors.

CSA Resource Alignment

JINX-0164's attack model touches several domains addressed by CSA frameworks and research programs. The AI Controls Matrix (AICM), CSA's comprehensive controls framework for AI-native environments, addresses Insecure Supply Chains as one of nine critical threat categories, encompassing the specific risks that JINX-0164's npm compromise and CI/CD pivot represent. The AICM's Software Supply Chain domain covers dependency integrity verification, maintainer identity assurance, and artifact provenance — the exact controls that, if fully implemented, would limit the blast radius of the attack chain JINX-0164 has demonstrated. The AICM's AI-CAIQ provides a standardized questionnaire that enterprises can use to assess the supply chain security posture of third-party AI-adjacent development tools and package ecosystems.

The MAESTRO framework (Multi-Agent Environment Security, Trust, Risk, and Observability) addresses the agentic dimension of CI/CD security that is increasingly relevant as development organizations adopt AI coding assistants. The SANDWORM_MODE campaign disclosed in February 2026 specifically targeted MCP server configurations used by AI coding assistants including Claude Code and Cursor, injecting malicious tool definitions that would execute with the permissions of the AI agent during subsequent coding sessions [14]. MINIRAT's documented persistence mechanism, which writes to `~/.zshrc` in addition to installing a LaunchAgent, would affect AI coding assistant sessions that inherit the compromised shell environment. MAESTRO's Layer 4 controls (Tool and Resource

Management) and Layer 5 controls (Infrastructure and Dependency Security) map directly to these risks, providing a structured evaluation framework for organizations assessing the security of their AI-assisted development environments.

CSA's Cloud Controls Matrix (CCM) and the Zero Trust principles articulated in CSA Zero Trust research are relevant to the lateral movement phase of JINX-0164's attack chain. The documented two-week intrusion that moved from a developer's laptop to CI/CD and code distribution systems reflects an environment where implicit trust — in the developer's workstation, in their GitHub token, in their ability to commit to main branches — was not constrained by Zero Trust network segmentation or just-in-time access provisioning. CCM control domains covering Identity and Access Management, Supply Chain Management and Transparency, and Security Incident Management, Identification, and Response all map to defensive controls that would have limited the intrusion's scope. The STAR for AI certification pathway provides enterprises with an independent assurance mechanism for validating that their development infrastructure meets these control objectives, and the CSA AI Risk Observatory — operating with CNA authority for agentic AI vulnerabilities — represents the coordination infrastructure that translates threat intelligence about actors like JINX-0164 into structured, actionable disclosure.

References

[1] Shira Ayal, Eden Abergil, Andre Maccarone, Yuval Dan, Benjamin Read (Wiz Research). "Commit to Compromise: A New Threat Actor Targeting the Cryptocurrency Industry's Software Development Infrastructure." Wiz Research Blog, May 27, 2026.

[2] The Hacker News. "JINX-0164 Targets Cryptocurrency Firms with Fake Recruiter Lures and macOS Malware." The Hacker News, May 27, 2026.

[3] Infosecurity Magazine. "New Threat Actor Jinx-0164 Targets Crypto Developers on macOS." Infosecurity Magazine, May 27, 2026.

[4] iru security. "MiniRAT: A Go-based macOS RAT delivered via malicious npm package." iru security blog, April 28, 2026.

[5] SafeDep. "Malicious @velora-dex/sdk Delivers Go RAT via npm." SafeDep, April 2026.

[6] Google Cloud Blog. "UNC1069 Targets Cryptocurrency Sector with New Tooling and AI-Enabled Social Engineering." Google Cloud, 2026.

[7] The Hacker News. "Lazarus Campaign Plants Malicious Packages in npm and PyPI Ecosystems." The Hacker News, February 2026.

[8] The Hacker News. "N. Korean Hackers Spread 1,700 Malicious Packages Across npm, PyPI, Go, Rust." The Hacker News, April 2026.

[9] Cyber Kendra. "Malicious Packages on npm, PyPI, and Crates.io Steal Crypto Wallets, SSH Keys, and Cloud Credentials." Cyber Kendra, May 2026.

[10] ReversingLabs. "Fake recruiter campaign targets crypto developers with RAT." ReversingLabs, 2025.

[11] Silobreaker. "12 Months That Changed Supply Chain Security." Silobreaker, 2026.

[12] Sygnia. "16 Minutes to Impact: npm Supply Chain Abuse Deploys crypto-draining malware." Sygnia, September 2025.

[13] Unit 42 (Palo Alto Networks). "GitHub Actions Supply Chain Attack: Targeted Attack on Coinbase Expanded to tj-actions/changed-files Incident." Palo Alto Networks, March 2025.

[14] The Hacker News. "Malicious npm Packages Harvest Crypto Keys, CI Secrets, and API Tokens." The Hacker News, February 2026.

[15] SecurityScorecard. "Lazarus Group Targets Developers Through NPM Packages and Supply Chain Attacks." SecurityScorecard, 2026.

[16] Chainalysis. "2025 Crypto Theft Reaches $3.4 Billion." Chainalysis 2026 Crypto Crime Report, 2026.

[17] TechCrunch. "North Korea's hijack of one of the web's most used open source projects was likely weeks in the making." TechCrunch, April 2026.

[18] CNN. "North Korean hackers bug software used by thousands of US companies in potential crypto heist attempt." CNN, March 31, 2026.

[19] SecurityWeek. "Over 5,500 GitHub Repositories Infected in 'Megalodon' Supply Chain Attack." SecurityWeek, May 2026.

[20] The Hacker News. "Popular GitHub Action Tags Redirected to Imposter Commit to Steal CI/CD Credentials." The Hacker News, May 2026.

[21] Zscaler ThreatLabz. "Malicious NPM Packages Deliver NodeCordRAT." Zscaler, November 2025.

[22] The Hacker News. "$285 Million Drift Hack Traced to Six-Month DPRK Social Engineering Operation." The Hacker News, April 2026.

[23] Trend Micro. "Analyzing TeamPCP's Supply Chain Attacks: Checkmarx KICS and elementary-data in CI/CD Credential Theft." Trend Micro, 2026.

[24] Dark Reading. "Supply Chain Attacks Targeting GitHub Actions Increased in 2025." Dark Reading, 2025.

[25] TRM Labs. "North Korea Stole 76% of All Crypto Hack Value in 2026 — With Just Two Attacks." TRM Labs, 2026.

[26] Microsoft Security Blog. "Contagious Interview: Malware delivered through fake developer job interviews." Microsoft, March 11, 2026.

[27] CISA. "Supply Chain Compromise of tj-actions/changed-files (CVE-2025-30066)." CISA Alert, March 18, 2025.

[28] Google Cloud Blog. "North Korea-Nexus Threat Actor Compromises Axios NPM Package." Google Cloud Threat Intelligence, 2026.

[29] Sonatype. "Sonatype 2026 Software Supply Chain Report — Open Source Malware." Sonatype, 2026.

[30] GitHub Advisory Database. "CVE-2025-30066: tj-actions/changed-files Supply Chain Attack." GitHub Advisories, March 2025.

[31] StepSecurity. "Megalodon: Mass GitHub Actions Secret Exfiltration Across 5,500+ Public Repositories." StepSecurity, May 2026.

[32] InfoQ. "TanStack Details Sophisticated npm Supply Chain Attack That Compromised 42 Packages." InfoQ, May 2026.

[33] BleepingComputer. "Shai-Hulud Attack Ships Signed Malicious TanStack, Mistral npm Packages." BleepingComputer, 2026.

[34] Orca Security. "pull_request_target Nightmare Part 2: Exploiting GitHub Actions for RCE and Supply Chain." Orca Security, 2025.