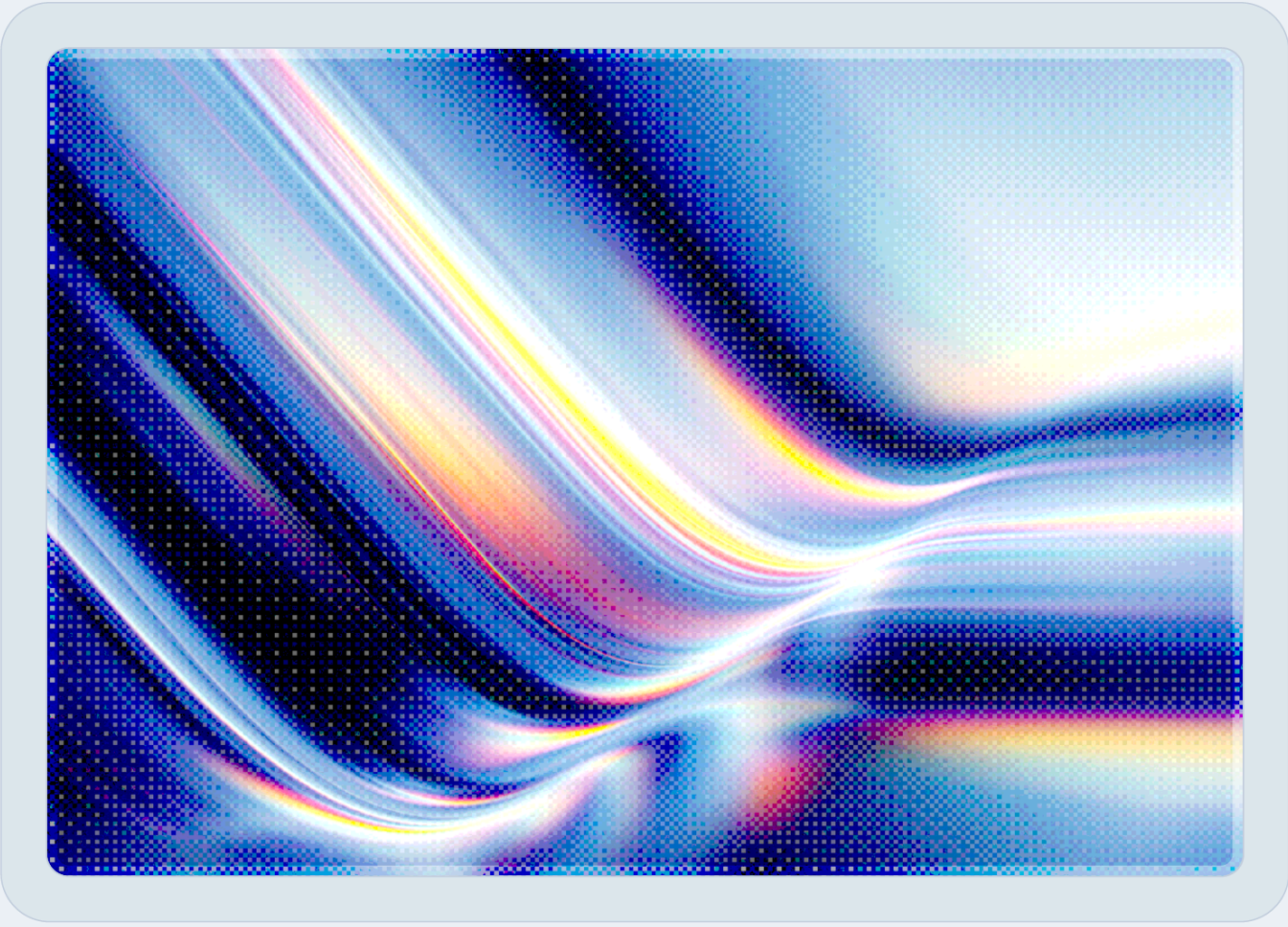
CSAI Foundation | Cloud Security Alliance

Amazon Links Debug, Chalk, and Axios npm Attacks to Sapphire Sleet

2026-08-02

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

- Amazon Threat Intelligence has attributed, with medium confidence, four separate npm package compromises — typo-crypto (March 2025), debug and chalk (September 2025), and axios (March 2026) — to a single North Korean threat actor tracked as Sapphire Sleet (also known as BlueNoroff, Stardust Chollima, CageyChameleon, and Alluring Pisces) [1][2][3].
- The debug and chalk compromise reached an estimated 10% of cloud environments within two hours of publication, consistent with the packages' combined weekly download count exceeding two billion [2][3], yet the browser-side wallet-address hijacking payload netted the attacker only about $600, according to Socket's analysis [2].
- Sapphire Sleet is systematically refining its tradecraft across campaigns: later operations split malicious functionality across multiple packages, decoupled malware from the packages themselves via externally hosted scripts, layered stronger encryption and multi-stage payloads, and built sandbox-evasion logic to delay detonation until a real developer or CI environment is detected [1][3].
- The same actor's tactics, techniques, and command-and-control infrastructure connect this campaign to Sapphire Sleet's June 2026 compromise of roughly 144 npm packages in the Mastra AI framework ecosystem, which CSA covered in prior rapid-research reporting, indicating a sustained, multi-ecosystem campaign rather than isolated incidents [4].
- AWS has jointly invested $12.5 million alongside other partners in the Akrites initiative, a collaboration with the Linux Foundation aimed at hardening critical open-source infrastructure against AI-accelerated supply chain attacks, while also urging organizations to evaluate packages by their behavior and dependency relationships rather than by isolated point-in-time scans [1][3].

Background

On July 30, 2026, Amazon's threat intelligence team published research formally connecting four npm package compromises that had previously been treated as unrelated incidents [1]. The disclosure matters because it reframes more than a year of npm ecosystem attacks — spanning a niche cryptocurrency

utility, two of the platform's most widely depended-upon utility libraries, and one of the most downloaded HTTP client libraries in the JavaScript ecosystem — as a single, patient campaign run by a state-sponsored actor rather than a series of disconnected opportunistic breaches.

The actor behind the campaign, which Amazon tracks as Sapphire Sleet, is also known across the security industry as BlueNoroff, Stardust Chollima, CageyChameleon, and Alluring Pisces, and is associated with North Korea's Reconnaissance General Bureau [1][2]. The group has a long operational history of financially motivated cryptocurrency theft, including prior links to the 2016 Bangladesh Bank heist and subsequent attacks on cryptocurrency exchanges [5]. Its pivot toward the open-source software supply chain appears to reflect a broader trend among North Korean state-linked units: rather than attacking a single target directly, compromising a widely depended-upon package gives the attacker indirect access to an enormous pool of downstream victims through a single point of compromise [3].

The four packages identified in Amazon's analysis illustrate both the campaign's patience and its escalating scale, as summarized below.

Package	Compromise Date	Reach	Primary Technique	Observed Outcome
typo-crypto	March 2025	Small, niche user base	Trojanized `core.js` distributed via postinstall hook	Believed to be a limited-scale testing ground for later tradecraft [1] [2]
debug and chalk	September 2025	2+ billion combined weekly downloads	Browser-side interceptor hooking `fetch`, `XMLHttpRequest`, and wallet APIs to rewrite cryptocurrency transaction addresses before signing	Reached an estimated 10% of cloud environments within two hours [2][3]; approximately $600 in crypto theft, per Socket's analysis [2]
axios	March 2026	100+ million	Malicious update following maintainer compromise	High-visibility incident that first

Package	Compromise Date	Reach	Primary Technique	Observed Outcome
		weekly downloads		drew scrutiny to the group's broader activity [1] [2]

Notably, the debug and chalk compromise left no persistent artifacts on infected machines and relied entirely on social engineering to obtain a trusted maintainer's publishing credentials, rather than on the postinstall-script execution pattern that characterized the earlier typo-crypto incident and Sapphire Sleet's later Mastra AI framework campaign [2][4]. This variation in method across otherwise attributed-together incidents is itself informative: it suggests an actor iterating deliberately on delivery mechanisms rather than reusing a single fixed toolkit, which complicates detection strategies built around any one signature technique.

Security Analysis

Amazon's attribution rests on overlapping tactics, techniques, and procedures rather than a single smoking-gun artifact, and the company has been explicit that its confidence level is "medium" [1][2]. The evidentiary basis includes shared command-and-control infrastructure across campaigns, code reuse patterns in obfuscation routines, and consistent operational tradecraft such as multi-layer obfuscation that combines base64 encoding with XOR ciphers, environment-awareness checks designed to avoid detonating payloads inside sandboxed analysis environments, and AES-GCM encryption used to protect payload contents and command-and-control communications [1]. For the earliest identified compromise, typo-crypto, Amazon published concrete indicators of compromise, including a command-and-control domain (npmjs[.]store), an associated IP address (216.74.123.126), and a file hash for the trojanized payload, which has since been catalogued in the Open Source Vulnerabilities database as MAL-2026-3400 [1].

Reporting on the disclosure has itself noted that Amazon's public writeup does not fully enumerate which specific indicators tie the September 2025 debug and chalk compromise to North Korea as opposed to another financially motivated actor mimicking similar tradecraft [2]. Aikido Security, the one outside researcher named in that coverage, raised a related but distinct objection: that the finding was largely already known to the security community rather than a new discovery [2]. This gap is worth flagging directly: attribution of supply chain attacks to nation-state actors carries significant weight for how organizations prioritize response and for diplomatic and policy consequences, and readers should treat

the "medium confidence" label as a genuine qualifier rather than a formality. Even with that caveat, the pattern of escalating sophistication across the four incidents — from a single trojanized file, to a browser-side interceptor with no persistence, to campaigns explicitly linked by shared infrastructure to the group's Mastra AI framework operation — is consistent with a coherent and maturing capability regardless of the precise attribution confidence level.

Amazon's analysis further identifies six emerging tradecraft patterns that organizations should expect to recur in future incidents from this or similar actors. The group has begun distributing malicious functionality across multiple packages so that no single package contains a complete, easily signature-able attack chain. It has invested months of legitimate contribution activity to build trust with maintainers and package communities before introducing malicious code, mirroring the credential-and-trust-building approach documented in the Mastra compromise, where a stale but valid publishing credential from a genuine former contributor was hijacked via a LinkedIn-based social engineering lure [4]. Newer campaigns decouple the malicious payload from the package's published contents entirely, instead fetching it from an external, attacker-controlled location at install or runtime — a technique that defeats static analysis of the package as published to the registry. Payloads increasingly use stronger encryption and multi-stage delivery, and are built with environment-awareness logic that can distinguish a real developer workstation or CI runner from an automated sandbox, delaying or suppressing malicious behavior until the payload is confident it has reached a genuine target. Finally, Amazon flags a technique it terms "slopsquatting" — publishing malicious packages under names that AI coding assistants are prone to hallucinate as dependencies, positioning the attacker to benefit from a developer's AI tool suggesting a plausible-sounding but nonexistent or malicious package name [3].

Recommendations

Immediate Actions

Organizations that consume any of the four affected packages, even in older or pinned versions, should audit dependency lockfiles for the specific compromised version ranges and rotate any credentials, API tokens, or cryptocurrency wallet keys that were accessible on systems where those versions were installed. Given that the debug and chalk compromise operated through a browser-side interceptor rather than a filesystem-persistent implant, security teams should also review outbound network logs and browser extension activity on developer workstations for signs of transaction-address rewriting rather than relying solely on endpoint file-based indicators of compromise [2][3]. Teams should cross-

reference the typo-crypto indicators of compromise — the npmjs[.]store domain, the associated IP address, and the published file hash — against internal telemetry, since the OSV entry for this malware (MAL-2026-3400) provides a stable reference point for automated scanning tools [1].

Short-Term Mitigations

Because Sapphire Sleet has demonstrated a pattern of using post-install hooks to execute malicious code automatically upon package installation, organizations should move toward disabling postinstall script execution by default in CI/CD pipelines and developer environments, allowlisting only the specific packages that have a legitimate operational need for install-time scripts. Software composition analysis and dependency scanning tools should be integrated directly into the install pipeline rather than run as a periodic, after-the-fact review, since the debug and chalk compromise demonstrated that malicious updates can reach roughly 10% of cloud environments within two hours of publication — a window far shorter than most scheduled scanning cadences [1][3]. Amazon specifically recommends its own Inspector and GuardDuty services for detecting suspicious package behavior and related threat activity within AWS environments, and organizations on other cloud platforms should identify equivalent dependency-behavior monitoring and runtime threat detection capabilities [1].

Strategic Considerations

The recurrence of Sapphire Sleet's tradecraft across the npm ecosystem and, separately, the Mastra AI framework compromise indicates that this is a sustained campaign against open-source infrastructure rather than a series of one-off events, and organizations should treat open-source dependency risk as an ongoing threat rather than a single incident to remediate and close out [4]. A package that shows no malicious behavior today is not the same as a package that is safe by design, which argues for continuous monitoring of dependency behavior over time rather than one-time vetting at adoption [1]. Organizations should also account for the "slopsquatting" risk that AI coding assistants introduce by verifying that any package name suggested by an AI tool actually exists and is legitimate before adding it as a dependency, since this vector specifically exploits growing developer reliance on AI-generated code suggestions [3]. Finally, security and open-source program office leaders should track industry-level efforts such as the Akrites initiative, which represents a broader recognition that the security of foundational open-source infrastructure is now a shared responsibility between commercial cloud providers, foundations, and the maintainer community rather than something any single organization can fully own [1][3].

CSA Resource Alignment

This incident extends directly from Sapphire Sleet's June 2026 compromise of the Mastra AI npm ecosystem, which CSA's AI Safety Initiative analyzed in "Sapphire Sleet Poisons Mastra AI npm Supply Chain" [4]. That report documented the same actor's use of hijacked maintainer credentials, typosquatted transitive dependencies, and multi-stage cross-platform payloads to compromise roughly 144 packages in a single AI development framework, and mapped mitigations to CSA's AI Controls Matrix (AICM) and MAESTRO Agentic AI Threat Modeling Framework. The debug, chalk, typo-crypto, and axios attribution described in this note confirms that the Mastra compromise was not an isolated event but one campaign within a broader, multi-ecosystem pattern by the same actor, reinforcing that report's recommendation that organizations treat AI and JavaScript developer supply chains as a persistent, evolving threat surface rather than a single remediated incident.

The AI Controls Matrix v1.1 [6], as the superset successor to CSA's Cloud Controls Matrix, remains the most directly applicable framework for the dependency-integrity, software composition analysis, and non-human identity controls this campaign implicates, particularly its supply chain and identity-and-access-management domains covering credential scoping, publisher verification, and third-party component governance. Organizations implementing the immediate and short-term mitigations described above — postinstall script restrictions, lockfile-based version pinning, and continuous dependency behavior monitoring — should map those controls to AICM's supply chain risk management domain to ensure the response is durable rather than tied to this single set of package names.

References

[1] Amazon Web Services. "Amazon identifies North Korean hacker group behind open-source supply cha in attacks." AWS Security Blog, July 30, 2026.

[2] The Hacker News. "Amazon Links Debug and Chalk npm Hijack to North Korea's Sapphire Sleet." The Hacker News, July 2026.

[3] BleepingComputer. "Amazon links Debug, Chalk NPM supply-chain attacks to North Korean hackers." BleepingComputer, July 30, 2026.

[4] Cloud Security Alliance. "Sapphire Sleet Poisons Mastra AI npm Supply Chain." CSA AI Safety Initiative, June 22, 2026.

[5] MITRE ATT&CK. "APT38, Group G0082." MITRE ATT&CK, updated November 13, 2025.

[6] Cloud Security Alliance. "AI Controls Matrix (AICM) v1.1." Cloud Security Alliance, 2026.

