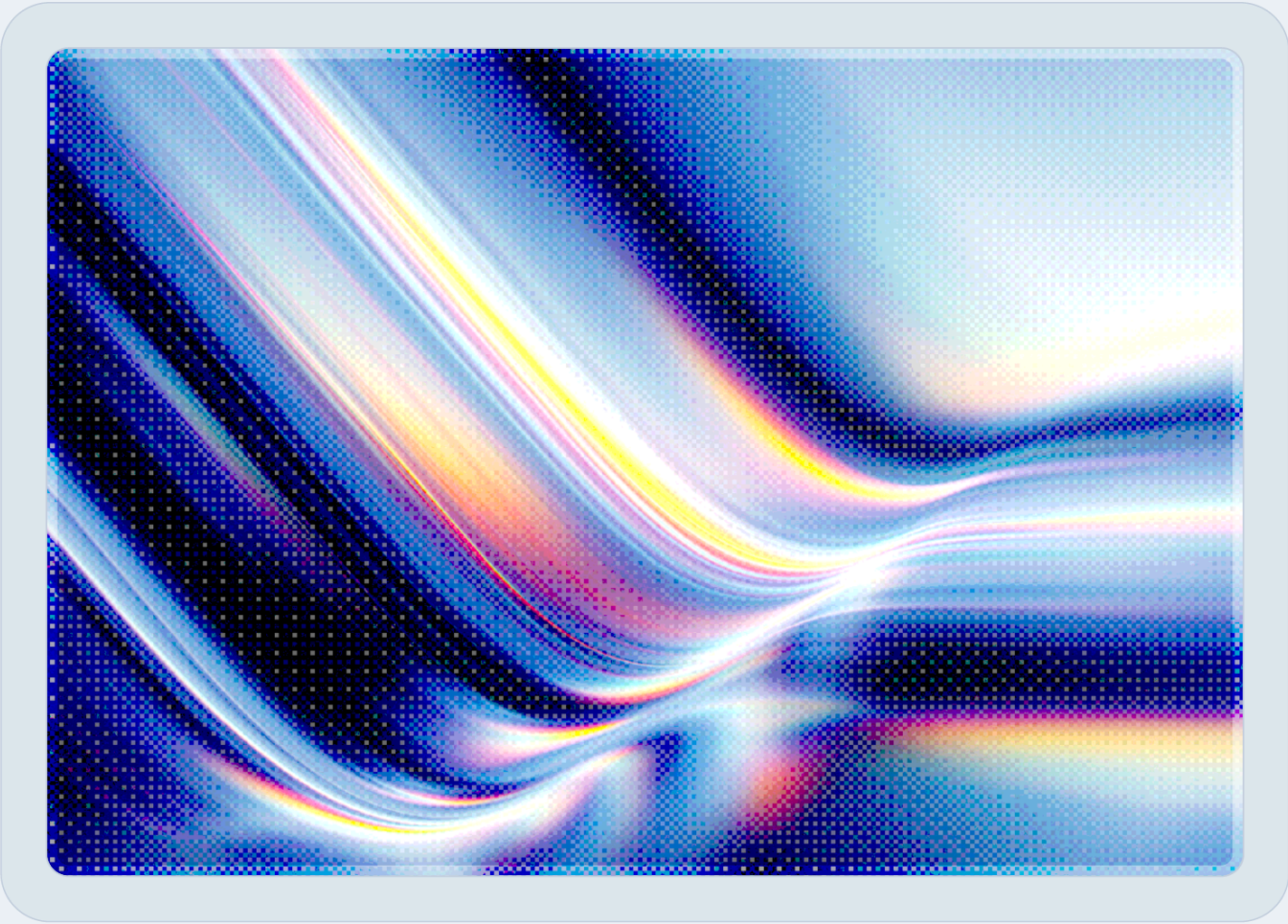
CSAI Foundation | Cloud Security Alliance

RedC2 4.0: AI-Assisted C2 Hidden in npm Packages

2026-08-22

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

Key Takeaways

Researchers at TrendAI, Trend Micro's cybersecurity research arm, identified 14 trojanized npm packages, disguised as calendar-streak and math-utility helpers, that silently install RedC2 4.0, a commercially sold Linux implant, the moment any package in the dependency chain is imported [1]. No install hook or exported function call is required; the packages' entry file re-exports legitimate date-helper logic while also marking a bundled binary executable and launching it as a detached background process, so the malicious behavior activates even through a transitive dependency [1]. RedC2 4.0 is notable less for its delivery mechanism, which follows a well-established pattern of publishing plausible-sounding new packages, than for the AI component bundled with it: a natural-language command layer called Red Agent that translates an operator's plain-language intent into the beacon commands needed to execute multi-stage intrusions, which, on its face, would lower the skill threshold for running a full-featured command-and-control operation [1]. The framework was advertised on Hack Forums by an actor using the handle "MarlboroMan" in June 2026 as a cross-platform toolkit "built for evasion," and the Linux variant adds SSH and browser credential theft, SOCKS5 proxying, and in-memory execution of shellcode and .NET assemblies to its capability set [1]. The discovery lands amid a broader surge in npm-targeted campaigns: a separate ReversingLabs investigation published in the same window documented packages that patch persistent backdoors directly into already-installed dependencies [4], and Sonatype's 2026 supply chain research puts the cumulative count of known malicious open-source packages above 1.2 million, a 75 percent year-over-year increase [2].

Background

npm's dependency model has long made it an efficient distribution channel for malware, because a single published package can reach every downstream project that lists it, directly or transitively, without any additional action from the attacker. The RedC2 campaign exploits this structurally: the 14 packages Trend Micro catalogued present themselves as small, genuinely functional utilities for tracking habit streaks, formatting calendar data, and accelerating date-math operations, categories that are common, low-scrutiny dependencies a developer might pull in without close review — a selection that plausibly reflects an intent to minimize scrutiny, though TrendAI's writeup does not state the attacker's reasoning directly [1]. Security researcher Aliakbar Zahravi, who led the analysis, described the loader mechanism directly: the package entry file, `dist/index.mjs`, "acts as a trojan loader" that "re-exports the date

helpers and launches the bundled implant as soon as the module loads, with no install hook and no exported function required" [1]. That design choice matters because many supply chain defenses focus on scanning `postinstall` and `preinstall` lifecycle scripts, the mechanism used by earlier high-profile npm attacks; a payload that fires on ordinary module evaluation instead sidesteps that specific control.

Package name	Version(s) observed
streak-metrics-math	1.0.0, 1.0.1
kit-map-vim	1.0.0
streak-map-cache	1.0.0
streak-map-kit	1.0.0
map-streak-kit	1.0.0
streak-cache-map	1.0.0
streak-calc-metrics	1.0.0
streak-calc-math	1.0.0
streak-math-abz	1.0.0
streak-metricsaz	1.0.0
streak-math-metrics	1.0.0
streak-metricazbd	1.0.0
streak-metricsazb	1.0.0
streak-kit-map	1.0.0

Table 1: The 14 trojanized npm packages identified by Trend Micro, all published under permuted names built around "streak," "map," "metrics," and "math" [1].

Once the bundled binary is marked executable and launched as a detached process, it functions as an endpoint for RedC2 4.0, the fourth generation of a command-and-control framework that has been sold on underground forums since at least August 2025, when version 2.0 first appeared; version 3.0 followed in January 2026 [1]. The framework is marketed as a legitimate-looking red-team toolkit for Windows, macOS, and Linux, a positioning that echoes how other dual-use offensive-security platforms have marketed themselves, straddling the line between authorized penetration-testing tooling and criminal infrastructure. MarlboroMan's Hack Forums listing pitched RedC2 4.0 explicitly on its evasion properties, and the feature set Trend Micro observed included staged payload delivery, multi-beacon management, network visualization, host-to-host tunneling, and in-memory execution of Beacon Object Files, shellcode, and .NET assemblies [1]. While TrendAI's writeup does not draw the comparison itself, that feature set reads to this analyst as a close analog to established commercial C2 frameworks such as Cobalt Strike and Sliver, adapted for a criminal buyer base rather than an authorized testing engagement.

Security Analysis

What distinguishes RedC2 4.0 from the broader field of commodity C2 tooling is Red Agent, the framework's built-in large language model interface. Rather than requiring an operator to know the specific beacon syntax for a given task, such as establishing a SOCKS5 tunnel or harvesting browser-stored credentials, Red Agent accepts natural-language prompts and translates them into the corresponding command sequences [1]. This is functionally the same abstraction that legitimate AI coding assistants provide for benign software development, applied instead to intrusion operations. The practical effect, at least in principle, is to compress the skill curve for running a multi-stage compromise: an operator no longer needs deep familiarity with the framework's command reference to chain together credential theft, lateral movement, and persistence steps, since the AI layer handles translation from intent to execution. This fits a broader pattern this note's authors have observed of legitimate AI tooling paradigms being repurposed for criminal operations, and appears likely to become a standard feature of commercial malware-as-a-service offerings rather than a one-off novelty.

On Linux specifically, the implant Trend Micro examined provides interactive shell access, SSH and browser credential theft, multiple persistence mechanisms, and SOCKS5 proxying for pivoting deeper into a compromised network [1]. None of this individually represents a novel capability; interactive shells and credential harvesting are baseline features of essentially every mature C2 framework. The risk this campaign poses comes instead from the delivery vector. Because the 14 packages perform their advertised function, a developer who installs one, or who inherits it as a transitive dependency of some other package, has no obvious signal that anything is wrong: the calendar or streak-tracking code works exactly as documented, while the implant runs silently in the background. This is consistent with a

broader shift Sonatype's 2026 State of the Software Supply Chain report describes across the npm ecosystem, in which open-source malware has moved from "spam and stunts" toward "sustained, industrialized campaigns" targeting developers and their tooling directly, with more than 454,600 new malicious packages identified across major registries in 2025 alone [3].

The RedC2 campaign is also notable for what it is not: it does not rely on account takeover, dependency confusion against an internal package name, or a compromised maintainer credential, the mechanisms behind other prominent npm incidents, such as the account-hijack campaign that compromised the @mastra package scope in June 2026 [8]. It is a straightforward typosquat-adjacent publication of new, standalone packages under plausible names, which means standard maintainer-account security controls, such as mandatory two-factor authentication or provenance attestation for existing publishers, would not have prevented it. Defenders should treat this as a reminder that account-security hardening, while necessary, addresses only one class of npm supply chain risk; a parallel and equally important control is scrutiny of any new, low-reputation package before it is added to a dependency tree, regardless of how legitimate its account history looks.

The timing of this disclosure alongside a separate ReversingLabs investigation reinforces that npm-focused attackers are actively diversifying technique. That campaign, involving packages named ethers-provider2, ethers-providerz, reproduction-hardhat, and an npm-scoped package impersonating a provider library, took a different approach: rather than shipping a self-contained payload, it monitored the local filesystem for an already-installed legitimate package and patched a reverse shell directly into that package's source files, so the backdoor persisted even after the malicious package that introduced it was removed [4]. Read together, the two campaigns illustrate that attackers are experimenting simultaneously with delivery innovation, such as loaders that avoid lifecycle-script scanning, and persistence innovation, such as infecting packages that were never themselves malicious. Separately, an August 4, 2026 compromise of packages in the keyv and cacheable npm namespaces used a malicious preinstall hook to harvest cloud and CI/CD credentials and specifically planted autostart hooks in `.claude/settings.json` and `.vscode/tasks.json` so that the payload would execute when either a developer or an AI coding agent opened a cloned repository, without requiring `npm install` at all [5]. That detail is directly relevant to organizations adopting AI coding assistants and agentic development tools, since it shows attackers are already targeting the specific configuration files those tools read on startup.

Recommendations

Immediate Actions

Organizations should search their dependency trees, including transitive dependencies, for any of the 14 package names Trend Micro identified and remove them immediately if found, regardless of the version installed [1]. Because the implant launches as a detached background process rather than through a documented service, incident responders should also check for unexpected executable files or running processes in `node_modules` directories and review outbound network connections originating from build servers, CI runners, and developer workstations for unfamiliar destinations consistent with C2 beacon traffic. Any system where one of the flagged packages was installed should be treated as potentially compromised until credential rotation, particularly for SSH keys and browser-stored secrets, has been completed, given RedC2's documented credential-theft capability [1].

Short-Term Mitigations

Development teams should adopt dependency-scanning tooling capable of flagging packages that bundle native or compiled binaries, since legitimate calendar, date-math, and streak-tracking utilities have no functional reason to ship an executable, and this pattern is a stronger signal than package name or download count alone. Organizations should also extend supply chain monitoring beyond lifecycle-script analysis, since this campaign's loader executes on ordinary module import rather than through `postinstall` or `preinstall` hooks that many existing scanners prioritize. Given the ReversingLabs findings on packages that patch backdoors into other, already-installed dependencies, integrity verification, such as periodically re-hashing critical dependencies against known-good checksums, provides a defense that a one-time install-scan does not [4]. Teams using AI coding assistants or agentic development tools should specifically audit `.claude/settings.json`, `.vscode/tasks.json`, and equivalent configuration files for unauthorized autostart entries, since recent campaigns have targeted these files precisely because AI agents and IDEs read them without an explicit `npm install` step [5].

Strategic Considerations

The Red Agent component of RedC2 4.0 is an early but likely representative example of AI-assisted tooling being integrated directly into commercial malware-as-a-service offerings, and security teams should expect the operational sophistication gap between less-skilled and more-skilled threat actors to continue narrowing as similar natural-language interfaces are added to other C2 frameworks. Enterprises

should incorporate detection logic for known C2 frameworks' network and behavioral signatures into their security operations independent of how the initial access occurred, since a framework like RedC2 will recur across multiple, unrelated delivery campaigns over its commercial lifetime. At a program level, this incident supports the case for treating npm dependency vetting as a continuous control rather than a one-time review at package adoption, since the packages here were newly published under innocuous names rather than compromises of already-trusted, high-download packages, meaning reputation-based allowlisting alone would not have caught them.

CSA Resource Alignment

CSA's Top Threats to Cloud Computing Survey Report 2026 identifies third-party and API risk, including software dependency exposure, alongside AI-enhanced attacks as two of the leading threat categories organizations face this year, and the RedC2 campaign sits squarely at the intersection of both [6]. The report defines AI-enhanced attacks as "the various ways adversaries use AI to improve and automate their attacks," a definition that directly anticipates the Red Agent pattern this note describes: a natural-language interface that lowers the skill floor for executing a multi-stage intrusion is a concrete instance of the AI-enhanced-attack category the survey flags as an emerging concern for 2026, rather than a hypothetical one [6]. Organizations using the report to prioritize their threat model should treat this campaign as evidence that AI-enhanced attacker tooling is already commercially available, not confined to nation-state actors.

CSA's AI Controls Matrix (AICM) v1.1 provides the control structure best suited to translating this incident into concrete organizational requirements, particularly through its Threat and Vulnerability Management (TVM) and Application and Interface Security (AIS) domains [7]. The TVM domain's expectations around dependency inventory, continuous vulnerability identification, and risk-based remediation map directly onto the package-auditing and integrity-verification guidance above, while the AIS domain's requirements for secure software development practices speak to the underlying need for pre-adoption scrutiny of third-party packages rather than reliance on post-hoc scanning alone. CSA's own research on AI-framework npm compromises, including its analysis of the Sapphire Sleet campaign that backdoored the @mastra package scope in June 2026, has separately documented how npm's account-security model and lifecycle-script execution create structural exposure for AI-development toolchains specifically [8]; the RedC2 and keyv/cacheable campaigns described here extend that same underlying exposure to a wider set of delivery techniques beyond account takeover.

References

[1] The Hacker News. "14 Trojanized npm Packages Drop RedC2 4.0 Linux Backdoor With AI-Assisted C2." The Hacker News, August 2026.

[2] Sonatype. "Sonatype Research Reveals Open Source Malware Grows 75%." Sonatype, January 2026.

[3] Sonatype. "2026 State of the Software Supply Chain: Open Source Malware." Sonatype, 2026.

[4] BleepingComputer. "New npm attack poisons local packages with backdoors." BleepingComputer, 2026.

[5] Socket. "Popular npm Packages in the keyv and Cacheable Namespaces Compromised in Active Supply Chain Attack." Socket, August 2026.

[6] Cloud Security Alliance. "Top Threats to Cloud Computing Survey Report 2026." Cloud Security Alliance, 2026.

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

[8] Cloud Security Alliance. "Sapphire Sleet Poisons Mastra AI npm Supply Chain." CSA Lab Space, June 2026.

