Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

ATTACK REPORT

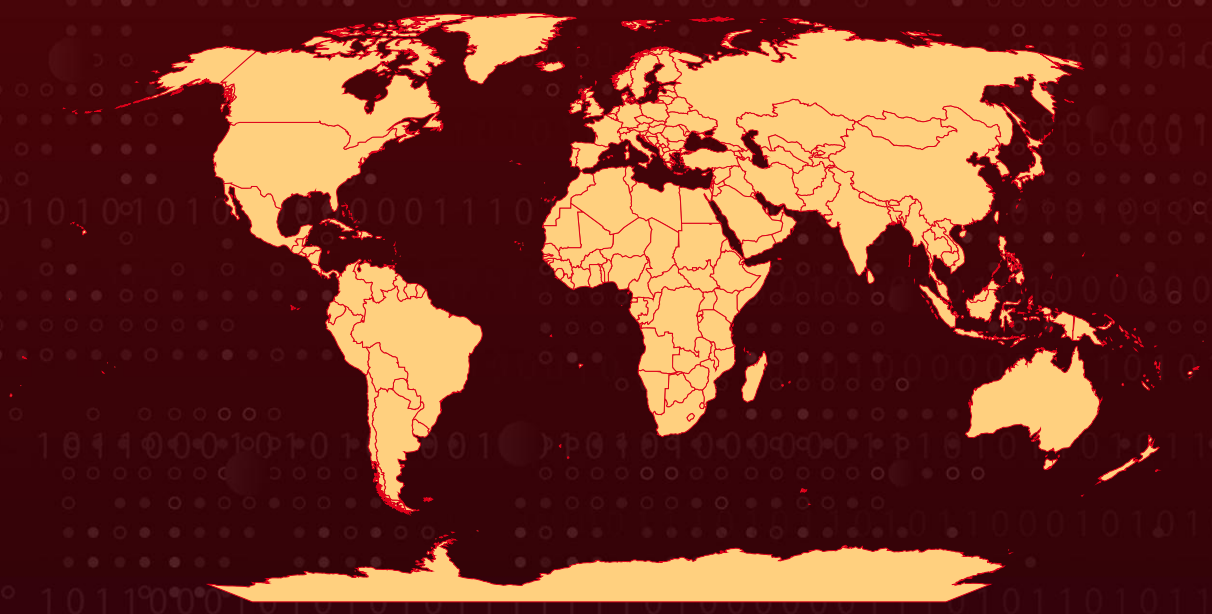
ChainDrop: Shai-Hulud npm Supply Chain Worm Compromises keyv Ecosystem

Date of Publication	Admiralty Code	TA Number
August 06, 2026	A1	TA2026221

Summary

First Seen: August 04, 2026
Targeted Regions: Global
Malware: ChainDrop (Shai-Hulud variant)
Attack: ChainDrop, a self-propagating Shai-Hulud npm worm, compromised the keyv maintainer's GitHub account on August 4, 2026, poisoning its caching package family and, within hours, hundreds of npm packages across thousands of versions representing over two billion monthly installs. A malicious preinstall hook silently runs an obfuscated stealer that harvests npm, GitHub, cloud, Kubernetes, and Vault credentials, then reuses stolen tokens to self-propagate across packages and repositories. Exfiltrated data is RSA-encrypted and dropped to attacker-created public GitHub repositories, with fallback infrastructure dynamically resolved via an Ethereum smart contract (EtherHiding), enabling C2 rotation without payload modification. Attribution to a named threat actor remains unresolved.

Attack Regions



Attack Details

#1 On August 4, 2026, an attacker hijacked the GitHub account of the maintainer behind keyv, a key-value library with approximately 155 million weekly downloads, and its sibling caching packages (flat-cache, file-entry-cache, cacheable, cacheable-request, cache-manager). Malicious files were pushed directly to the main branch and new releases were cut immediately, so the poisoned versions reached npm carrying valid GitHub Actions provenance and appeared authentic. Every affected package gained two files, setup.mjs and Math_Symbol.js, plus a preinstall lifecycle hook that executed the payload automatically during npm install, before installation completed or tests ran.

#2 The initial vector was compromise of the maintainer's GitHub account; the exact method of account takeover has not been publicly confirmed. setup.mjs is an obfuscated dropper that silently downloads the Bun runtime and launches a 728 KB obfuscated credential stealer. It checks its environment, terminates on Russian-language systems, and distinguishes developer workstations from CI runners so it can reach workflow secrets and OIDC publishing permissions. It then harvests npm, GitHub, AWS, Kubernetes, HashiCorp Vault, Stripe, and Slack credentials, reading files, environment variables, cloud metadata endpoints, and GitHub Actions runner memory, and sweeps the filesystem with roughly 200 glob patterns for keys, tokens, and .env data.

#3 The recovered identities are weaponized for self-propagation. Using a stolen npm token, the worm lists every package the token can publish, injects itself, bumps the patch version, and republishes, turning one stolen credential into malicious releases across an entire publisher. Stolen GitHub tokens are used to inject hooks into .claude and .vscode configuration files across repository branches, creating persistence and a developer-to-developer infection path that survives beyond the original install.

#4 Collected data is compressed, RSA-encrypted, and exfiltrated to attacker-created public GitHub repositories carrying the description "Shai-Hulud: Here We Go Again," with a fallback HTTPS endpoint (npm-cache[.]com:443/router) resolved dynamically via an Ethereum smart contract, a technique known as EtherHiding, allowing infrastructure rotation without modifying the payload. Security researchers additionally report a destructive capability that triggers if the monitored GitHub token is revoked.

#5 Within hours the campaign spread past the original maintainer to hundreds of packages across thousands of versions, representing over two billion monthly installs across organizations in multiple industries. Attribution to a named threat actor remains unconfirmed at this time.

Recommendations

Remove and Roll Back Poisoned Package Versions: Identify every compromised keyv, flat-cache, file-entry-cache, cacheable-family, cache-manager, ecto, and downstream package version in dependency trees, lockfiles, artifact repositories, and CI caches, including transitive references, and pin or roll back to known-good releases published before the compromise window.

Rotate All Exposed Credentials From a Clean Host: Treat any workstation or build runner that ran an install after the compromise as exposed, and rotate npm tokens, GitHub personal-access and OIDC/Actions credentials, AWS keys, Kubernetes secrets, Vault tokens, and Stripe and Slack API keys from a known-clean environment, invalidating active sessions where appropriate.

Purge npm, Yarn, and pnpm Caches: Clear package-manager caches on affected developer endpoints and shared build hosts, since poisoned tarballs written into shared CI caches can silently reinfect later jobs.

Block Malicious Indicators: Block the identified exfiltration domains and URL and alert on the payload file hashes and filenames (setup.mjs, Math_Symbol.js, math_init.js) across endpoint and network controls, while deliberately excluding legitimate infrastructure abused by the malware from blocklists.

Hunt for Preinstall and Bun Execution: Hunt endpoint and cloud telemetry for node executing setup.mjs, for the Bun runtime being downloaded and executed from node_modules or temporary bun download directories, and for the known payload hashes.

Audit Repositories for Injected Persistence: Review repositories for unauthorized additions to .vscode/tasks.json and .claude/settings.json, unexpected commits under routine configuration-update messages, and any public repositories created with the campaign's self-attribution description.

Enforce Phishing-Resistant MFA and Least Privilege: Require phishing-resistant MFA and hardware security keys on maintainer, source-repository, and package-publishing accounts, and apply least-privilege scoping to publishing tokens to limit the blast radius of a single stolen credential.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	T1195: Supply Chain Compromise	T1195.002: Compromise Software Supply Chain
	T1078: Valid Accounts	
Execution	T1059: Command and Scripting Interpreter	T1059.007: JavaScript
Persistence	T1546: Event Triggered Execution	
Defense Evasion	T1027: Obfuscated Files or Information	
	T1480: Execution Guardrails	
	T1036: Masquerading	T1036.005: Match Legitimate Name or Location
Credential Access	T1552: Unsecured Credentials	T1552.001: Credentials In Files
		T1552.005: Cloud Instance Metadata API
		T1552.007: Container API
	T1528: Steal Application Access Token	
	T1555: Credentials from Password Stores	T1555.005: Password Managers

Tactic	Technique	Sub-technique
Discovery	T1526: Cloud Service Discovery	
Collection	T1119: Automated Collection	
	T1005: Data from Local System	
Command and Control	T1071: Application Layer Protocol	T1071.001: Web Protocols
	T1102: Web Service	T1102.001: Dead Drop Resolver
	T1132: Data Encoding	T1132.001: Standard Encoding
Exfiltration	T1567: Exfiltration Over Web Service	T1567.001: Exfiltration to Code Repository
Lateral Movement	T1080: Taint Shared Content	
Impact	T1485: Data Destruction	

Indicators of Compromise (IOCs)

TYPE	VALUE
SHA256	54dc7ea54a1317cca0e890a2770630cf7fa6c97813e0cb9d2caa93012b350668, fd3ca4007b225fdf8de7af4345a19179d5efa8c4bb9205f88cda806e5684b1eb, 9fc2570b7cef51c1b8df116d144d11ff4096357be7d2c4c6367cfc2509cf1bcc
Filenames	setup.mjs, Math_Symbol.js, math_init.js, router_runtime.js
Domains	npm-cache[.]com, pypi-get[.]com, js-mirror[.]com, eth-mainnet[.]nodereal[.]io
URL	hxxps[:]//npm-cache[.]com[:]443/router
Ethereum Contract Address	0xE1f2395ee43e45A1556EC6438a88c31B83493103
Git Reference	github:opensearch-project/opensearch-js#d446803f4c3bc116263faa3499a1d3f95b2825de
String Indicator	Shai-Hulud: Here We Go Again, thebeautifulmarchoftime, IfYouBlockThisAPIKeyItWillCrashTheLiveProductionServersOfAllThirdPartyClients

References

https://www.aikido.dev/blog/keyv-and-friends-compromised-in-npm-supply-chain-attack

https://www.microsoft.com/en-us/security/blog/2026/08/04/chaindrop-supply-chain-compromise-anatomy-self-propagating-worm/

https://www.ox.security/blog/a-new-infostealer-worm-hits-npm-affecting-keyv-and-cacheable/

https://www.esecurityplanet.com/threats/github-account-breach-fuels-shai-hulud-npm-supply-chain-attack/

What Next?

At **Hive Pro**, it is our mission to detect the most likely threats to your organization and to help you prevent them from happening.

Book a Demo of **HivePro**.

REPORT GENERATED ON

August 06, 2026 • 07:30 AM

More at www.hivepro.com