Hive Pro

Threat Level

Red

Hiveforce Labs

THREAT ADVISORY

ATTACK REPORT

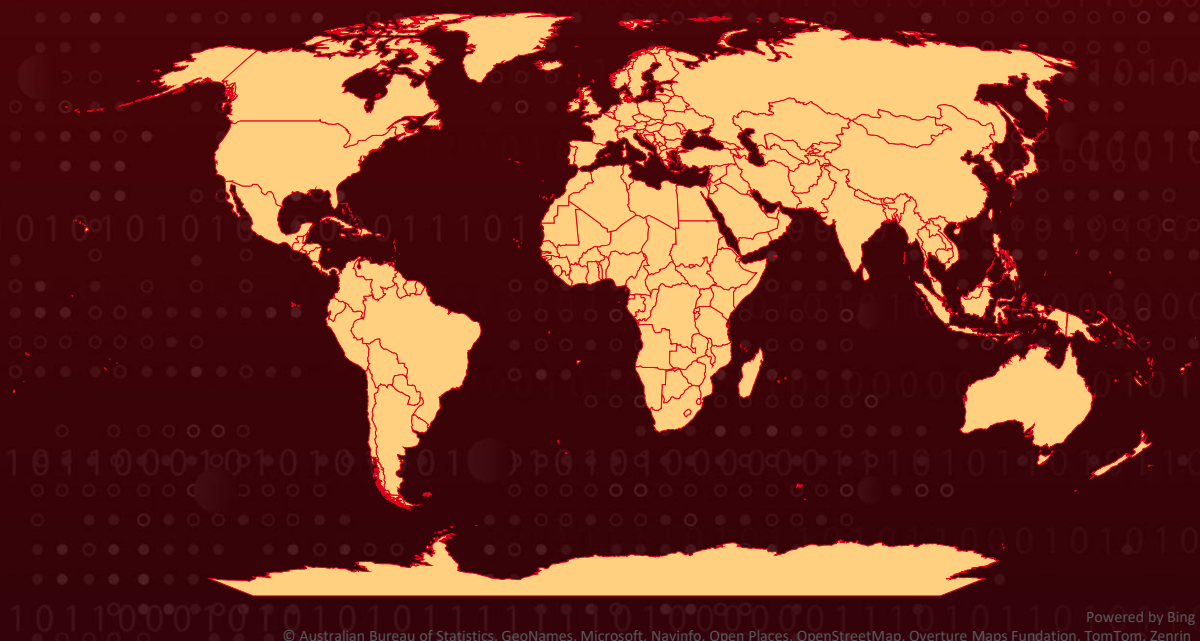
PhantomRaven: Multi-Wave npm Campaign Stealing CI/CD and Developer Credentials

Date of Publication	Admiralty Code	TA Number
March 12, 2026	A1	TA2026069

Summary

First Seen: August 2025
Targeted Regions: Global
Targeted Platforms: npm (Node.js package) developer environments across Windows, Linux, and macOS, especially CI/CD pipelines and developer workstations
Campaign Name: PhantomRaven
Attack: PhantomRaven is an ongoing npm supply chain campaign that has distributed over 200 malicious packages across four waves since August 2025, targeting JavaScript developers worldwide. The attack leverages Remote Dynamic Dependencies (RDD) to fetch credential-stealing payloads from external servers during npm install, completely bypassing static security scanners. The malware harvests CI/CD tokens, developer credentials, and system information, exfiltrating data through redundant channels to attacker-controlled infrastructure. With 81 of the most recent 88 packages still live on npm and active C2 servers, organizations using npm dependencies should immediately audit their environments for indicators of compromise and verify the authenticity of all installed packages.

Attack Regions



Targeted Non-Targeted

Attack Details

#1 PhantomRaven is an ongoing npm supply chain campaign targeting JavaScript developers since August 2025. First exposed in October 2025, the initial wave included 126 malicious packages with over 86,000 downloads. Despite public disclosure, the attacker continued operations, pushing the total malicious packages to over 200 across four waves.

#2 The campaign uses a technique called Remote Dynamic Dependencies (RDD). Instead of placing malicious code directly in the package, the attacker adds an HTTP URL in the package.json dependencies field. When a developer runs npm install, npm itself downloads the payload from attacker-controlled servers. The published packages contain only a harmless "Hello World" script with zero visible dependencies, allowing them to slip past static analysis tools entirely. The payload runs automatically through a preinstall hook with all console output hidden, leaving victims unaware of the compromise.

#3 Three new waves (2, 3, and 4) were uncovered between November 2025 and February 2026, adding 88 malicious packages published through over 50 throwaway npm accounts. Wave-2 targeted GraphQL Codegen plugins, Wave-3 shifted to Babel plugin names, and Wave-4 focused on import/export utilities. Despite changing domains, accounts, and metadata, the payload remains 257 out of 259 lines identical across all waves. The infrastructure consistently uses AWS-registered domains containing "artifact" with no TLS certificates, and the tarball author field reads "JPD" throughout, confirming a single operator. At discovery, 81 of 88 packages were still live and two of three C2 servers remained active.

#4 Once installed, the malware collects emails from .gitconfig, .npmrc, and environment variables, steals CI/CD tokens from GitHub Actions, GitLab CI, Jenkins, CircleCI, and npm, and gathers system details including IP, hostname, OS, and Node.js version to tell apart individual developers from corporate build environments. Stolen data is sent out using a fallback chain — first HTTP GET, then POST, then WebSocket — and the C2 runs a PHP application with a searchable database that lets the attacker link victim data across different packages.

#5 The attackers also use "slopsquatting," registering package names that AI coding assistants are likely to falsely suggest as real, taking advantage of common typos and missing scope prefixes. Claims that this may be security research have been dismissed, pointing to excessive data collection far beyond simple check-in signals, no disclosure in README files, and deliberate identity rotation. PhantomRaven remains an active threat requiring developers to verify package authenticity and organizations to adopt detection tools that track external dependencies beyond package contents alone.

Recommendations

Audit Dependencies for Remote Dynamic Dependencies (RDD): Review all package.json files across projects for dependencies specified as HTTP/HTTPS URLs instead of standard version ranges. This is the primary indicator of the PhantomRaven technique. Automated scripts or registry proxies can flag such entries before they reach production environments.

Disable Automatic Script Execution During Installation: Use npm install --ignore-scripts to prevent automatic execution of preinstall and postinstall hooks, which PhantomRaven relies on to trigger its payload silently. Review lifecycle scripts manually before enabling them, and enforce this practice across all CI/CD pipelines and developer workstations.

Verify Package Authenticity Before Installation: Never install packages based solely on AI-generated suggestions or unverified sources. Confirm the package exists under its correct scoped namespace on npmjs.com, check publisher history and download counts, and enforce the use of scoped packages (@org/package-name) in internal projects to reduce slopsquatting risk.

Rotate Compromised Credentials Immediately: If any suspicious or unverified npm package was recently installed, assume compromise and rotate all CI/CD tokens, npm publish tokens, GitHub/GitLab credentials, and Jenkins/CircleCI secrets. Review .gitconfig, .npmrc, and environment variables on affected machines for signs of unauthorized access.

Monitor Build Environment Network Activity: Deploy behavioral monitoring on developer machines and CI/CD runners to detect unexpected outbound connections during package installation. Flag traffic to unknown AWS-hosted domains, particularly those containing "artifact" in the domain name or lacking TLS certificates, as these align with PhantomRaven's known infrastructure patterns.

Potential MITRE ATT&CK TTPs

Tactic	Technique	Sub-technique
Initial Access	T1195: Supply Chain Compromise	T1195.001: Compromise Software Dependencies and Development Tools
	T1199: Trusted Relationship	
Execution	T1204: User Execution	T1204.002: Malicious File
	T1059: Command and Scripting Interpreter	T1059.007: JavaScript
Persistence	T1546: Event Triggered Execution	
Defense Evasion	T1036: Masquerading	
	T1027: Obfuscated Files or Information	
	T1564: Hide Artifacts	
Credential Access	T1552: Unsecured Credentials	T1552.001: Credentials in Files
	T1528: Steal Application Access Token	
Discovery	T1082: System Information Discovery	
	T1016: System Network Configuration Discovery	
Collection	T1119: Automated Collection	
Exfiltration	T1041: Exfiltration Over C2 Channel	
	T1048: Exfiltration Over Alternative Protocol	
Command and Control	T1071: Application Layer Protocol	T1071.001: Web Protocols
	T1105: Ingress Tool Transfer	

Indicators of Compromise (IOCs)

TYPE	VALUE
URLs	hxxp[:]//54[.]173[.]15[.]59[:]8080/jpd[.]php, hxxp[:]//packages[.]storeartifact[.]com/jpd[.]php, hxxp[:]//npm[.]jpartifacts[.]com/jpd[.]php, hxxp[:]//package[.]storeartifacts[.]com/npm[.]php, hxxp[:]//npm[.]artifactsnpm[.]com/npm[.]php
IPv4	54[.]173[.]15[.]59, 100[.]26[.]42[.]247, 13[.]219[.]250[.]107, 54[.]227[.]45[.]171
Domains	packages[.]storeartifact[.]com, npm[.]jpartifacts[.]com, package[.]storeartifacts[.]com, npm[.]artifactsnpm[.]com
RDD Dependency Name	ui-styles-pkg, js-pkg, ts-pkg
Email Address	jpdtester01[@]hotmail[.]com, jpdtester13[@]gmail[.]com, jjjpd01[@]outlook[.]com, jjjpd20[@]outlook[.]com, dharshanjp*[@]outlook[.]com, jpddharsh*[@]outlook[.]com, jpdplugin*[@]outlook[.]com

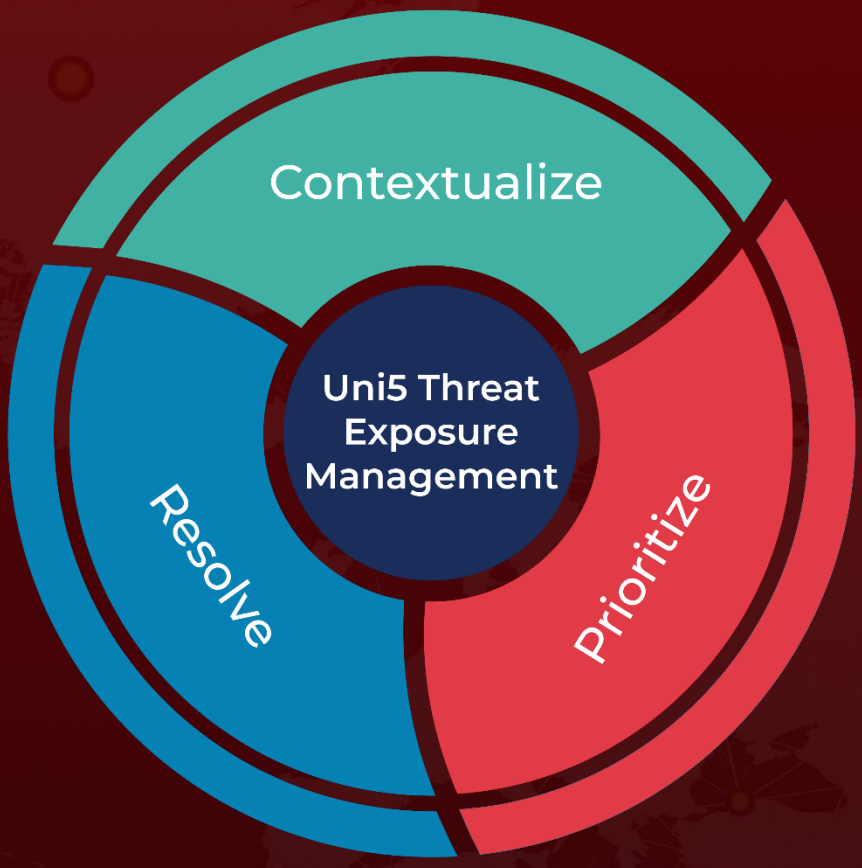
References

https://www.endorlabs.com/learn/return-of-phantomraven#indicators-of-compromise-iocs

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 free demo with **HivePro Uni5**: Threat Exposure Management Platform.



REPORT GENERATED ON

March 12, 2026 • 07:30 AM

More at www.hivepro.com