INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-29 19:06 UTC

vpmdhaj npm Campaign: Dual-Stage Credential Harvester Targets AWS, Vault, and CI/CD Pipelines via Typosquatted Packages

THREAT CAMPAIGN | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0381
Type	Threat Campaign
Severity	CRITICAL
CVSS Base Score	9.5
Affected Products	npm ecosystem (14 malicious typosquatted packages), AWS (IMDS, STS, Secrets Manager, ECS), HashiCorp Vault, GitHub Actions, Bun runtime (abused as evasion vehicle), OpenSearch, ElasticSearch, Microsoft Defender XDR, Microsoft Defender Antivirus
Published	2026-05-29T03:04:52+00:00
Discovery Source	Rss:T1 Threatintel

Executive Summary

On May 28, 2026, an unidentified threat actor published 14 malicious npm packages designed to impersonate legitimate libraries and silently steal cloud credentials during software installation. Any development or CI/CD environment that ran npm install against these packages must treat AWS credentials, HashiCorp Vault tokens, GitHub Actions secrets, and npm publish tokens as fully compromised. The theft of npm publish tokens creates a downstream supply chain risk: packages published from compromised accounts could distribute malware to every organization that installs them, multiplying the blast radius well beyond the initial 14 packages.

Technical Analysis

The vpmdhaj campaign deployed 14 typosquatted npm packages that execute a dual-stage credential harvester on npm install, requiring no post-install interaction. Stage one drops a stager that invokes the Bun JavaScript runtime to bypass endpoint detection tools, including Microsoft Defender XDR and Microsoft Defender Antivirus (MITRE T1036.001, Masquerading: Invalid Code Signature; T1027, Obfuscated Files or Information; T1059.007, Command and Scripting Interpreter: JavaScript). Stage two exfiltrates credentials via HTTP to attacker-controlled infrastructure (T1071.001): AWS credentials via the Instance Metadata Service (IMDS) and STS endpoints (T1552.005, Cloud Instance Metadata API), HashiCorp Vault tokens (T1552.007, Container API), GitHub Actions CI/CD secrets (T1552.001, Credentials In Files), and npm publish tokens enabling

downstream supply chain compromise (T1195.001, Compromise Software Supply Chain). Automated collection runs without user interaction (T1119, Automated Collection). The attack abuses cloud valid accounts after credential theft (T1078.004, Valid Accounts: Cloud Accounts). Relevant CWEs: CWE-312 (Cleartext Storage of Sensitive Information), CWE-829 (Inclusion of Functionality from Untrusted Control Sphere), CWE-494 (Download of Code Without Integrity Check), CWE-506 (Embedded Malicious Code). Microsoft confirmed package removal from the npm registry, but no CVE is assigned. Any environment that executed npm install against any of the 14 packages between May 28, 2026 and takedown must be treated as compromised.

Action Checklist

1. Step 1: Containment. Immediately audit npm install logs and CI/CD pipeline execution logs for any install activity on or after May 28, 2026 involving packages matching the vpmdhaj campaign package list. Isolate any build runner, developer workstation, or container that executed an install against flagged packages. Revoke and rotate all AWS IAM credentials, STS session tokens, HashiCorp Vault tokens, GitHub Actions secrets, and npm publish tokens accessible from affected environments. Enforce NIST AC-2 (Account Management) by disabling suspected compromised accounts pending rotation confirmation.
2. Step 2: Detection. Query npm install logs, CI/CD pipeline logs (GitHub Actions workflow logs, Jenkins build logs), and AWS CloudTrail for API calls to IMDS (169.254.169.254) and STS endpoints originating from build environments on or after May 28, 2026. In AWS CloudTrail, look for GetCallerIdentity, AssumeRole, and GetSecretValue calls from unexpected source IPs or IAM entities tied to build infrastructure. Search SIEM for Bun runtime execution events on build hosts (NIST AU-6, Audit Record Review, Analysis, and Reporting; CIS 8.2, Collect Audit Logs). Cross-reference endpoint telemetry for bun process execution initiated by npm scripts. Flag any outbound HTTP/S connections from build runners to non-standard destinations during install phases (NIST SI-4, System Monitoring).
3. Step 3: Eradication. Remove all 14 malicious packages from any local npm caches, package-lock.json files, and container image layers. Purge and rebuild any container images built from affected environments. Rotate all credentials confirmed or suspected exposed (AWS IAM keys, Vault tokens, GitHub secrets, npm tokens) using credential rotation processes. Audit all npm packages published using tokens that were present in affected environments after May 28, 2026; any published package must be treated as a potential trojan until verified clean. Enforce CIS 2.1 (Establish and Maintain a Software Inventory) and CIS 2.3 (Address Unauthorized Software) to validate only authorized packages are present in build pipelines.
4. Step 4: Recovery. After credential rotation, validate AWS CloudTrail shows no continued unauthorized API calls from compromised credential sets. Re-enable CI/CD pipelines only after confirming clean package inventories and rotated secrets. Monitor HashiCorp Vault audit logs for any residual token usage post-rotation. Validate GitHub Actions workflow runs post-incident for unexpected steps or external network calls. Apply NIST AU-9 (Protection of Audit Information) to ensure audit logs from the exposure window are preserved for forensic review (NIST AU-11, Audit Record Retention).
5. Step 5: Post-Incident. Implement package integrity verification in CI/CD pipelines (CIS 7.1, Establish and Maintain a Vulnerability Management Process; NIST CM controls), enforce package allowlisting. Enforce file integrity verification (magic byte checks) and package hash pinning in package-lock.json or equivalent lockfiles. Restrict build environment access to AWS IMDS using IMDSv2 token-required policy and block IMDS access from non-EC2 workloads. Apply least-privilege IAM policies to build runners (NIST AC-6, Least Privilege; CIS 5.4, Restrict Administrator Privileges). Establish monitoring for Bun runtime

execution in build environments where it is not an authorized tool (system file analysis, local account monitoring). Review and tighten npm publish token scopes and rotation schedules.

Detection Guidance

Primary detection surface is build environment logs and cloud API audit trails. Key signals: (1) Bun runtime process (bun or bun.exe) spawned as a child of npm or node on any build host - this is the stage-one evasion indicator. (2) Outbound HTTP/S connections from build runners to non-registry, non-CDN destinations during or immediately after npm install. (3) AWS CloudTrail events: GetCallerIdentity, GetIAMInstanceProfileAssociations, AssumeRole, GetSecretValue, or ListSecrets calls originating from build infrastructure IP ranges, particularly if the IAM entity is a build runner role not normally calling STS or Secrets Manager. (4) IMDS access (HTTP GET to 169.254.169.254/latest/meta-data/) from non-expected processes - flag in host-based telemetry. (5) HashiCorp Vault audit log entries showing token reads from build host IPs outside normal deployment windows. (6) GitHub Actions: workflow runs with unexpected external HTTP calls in steps that should only perform package installs. (7) npm audit log entries for the 14 flagged package names - match against published campaign indicators from the Microsoft Security Blog source. Relevant NIST controls for detection architecture: AU-2 (Event Logging), AU-6 (Audit Record Review), SI-4 (System Monitoring). CIS 8.2 (Collect Audit Logs) validates baseline log coverage. System file analysis and local account monitoring support host-level behavioral detection.

Indicators of Compromise

Type	Value	Context	Confidence
DOMAIN	`vpmdhaj campaign infrastructure — specific domains not publicly confirmed in available sources`	Attacker-controlled exfiltration endpoint used by stage-two payload; verify against Microsoft Security Blog advisory for confirmed IOCs	LOW
HASH	`Not publicly confirmed in available sources — refer to Microsoft Security Blog advisory for package hashes`	Hashes for the 14 malicious npm packages	LOW
URL	`http://169.254.169.254/latest/meta-data/`	AWS IMDS endpoint queried by stage-two payload to harvest instance credentials	HIGH
URL	`https://sts.amazonaws.com (GetCallerIdentity)`	AWS STS endpoint queried by stage-two payload to validate and exfiltrate credential context	HIGH

Framework Mappings

MITRE-ATTACK

- **T1036.001** — Invalid Code Signature

- **T1027** — Obfuscated Files or Information
- **T1020** — Automated Exfiltration
- **T1552.005** — Cloud Instance Metadata API
- **T1059.007** — JavaScript
- **T1190** — Exploit Public-Facing Application
- **T1552.001** — Credentials In Files
- **T1543** — Create or Modify System Process
- **T1552.004** — Private Keys
- **T1119** — Automated Collection
- **T1078.004** — Cloud Accounts
- **T1071.001** — Web Protocols
- **T1552.007** — Container API
- **T1195.001** — Compromise Software Dependencies and Development Tools

NIST-800-53R5

- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **CA-8** — Penetration Testing
- **RA-5** — Vulnerability Monitoring and Scanning
- **SC-7** — Boundary Protection
- **SI-2** — Flaw Remediation
- **SI-7** — Software, Firmware, and Information Integrity
- **AC-6** — Least Privilege
- **CM-3** — Configuration Change Control
- **SR-2** — Supply Chain Risk Management Plan

OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **6.3** — Require MFA for Externally-Exposed Applications
- **15.1** — Establish and Maintain an Inventory of Service Providers
- **8.2** — Collect Audit Logs

HIPAA-SECURITY

- **164.312(d)** — Person or Entity Authentication

SOC2-TSC

- **CC6.1** — Logical access security software, infrastructure, and architectures
- **CC9.2** — Manages risks associated with vendors and business partners

NIST-CSF-2

- **GV.SC-01** — Cybersecurity supply chain risk management program
- **DE.CM-01** — Networks and network services are monitored

ISO-27001-2022

- **A.5.21** — Managing information security in the ICT supply chain
- **A.5.23** — Information security for use of cloud services

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1036.001	Invalid Code Signature	Defense-Evasion
T1027	Obfuscated Files or Information	Defense-Evasion
T1020	Automated Exfiltration	Exfiltration
T1552.005	Cloud Instance Metadata API	Credential-Access
T1059.007	JavaScript	Execution
T1190	Exploit Public-Facing Application	Initial-Access
T1552.001	Credentials In Files	Credential-Access
T1543	Create or Modify System Process	Persistence
T1552.004	Private Keys	Credential-Access
T1119	Automated Collection	Collection
T1078.004	Cloud Accounts	Defense-Evasion
T1071.001	Web Protocols	Command-And-Control
T1552.007	Container API	Credential-Access
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access

Sources

Source	URL	Tier
Microsoft Security Blog	https://www.microsoft.com/en-us/security/blog/2026/05/28/typosquatt...	T1
	https://www.microsoft.com/en-us/security/blog/2026/05/28/typosquatt...	T1

Source	URL	Tier
	https://gbhackers.com/typosquatted-npm-packages/	T3
	https://thehackernews.com/2026/05/malicious-sicoob-nuget-steals-ban...	T3
Safeguard Your Container Supply Chain with Microsoft Defender for ...	https://www.youtube.com/watch?v=gTOm3vo4REE	T3

Generated 2026-05-29 19:06 UTC by TJS Security Command Center