INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-15 06:39 UTC

AUR Supply Chain Compromise Deploys eBPF Rootkit and Infostealer Across 400+ Arch Linux Packages

THREAT CAMPAIGN | **CRITICAL** | CVSS 9.5

SCC Item ID	SCC-CAM-2026-0461
Type	Threat Campaign
Severity	CRITICAL
CVSS Base Score	9.5
Affected Products	Arch Linux / Arch User Repository (AUR), npm (atomic-lockfile package), GitHub, SSH, HashiCorp Vault, Docker, Podman, Slack, Microsoft Teams, Discord, Telegram, browser credential stores, Electron applications
Published	2026-06-12T13:03:55
Discovery Source	Rss

Executive Summary

A coordinated supply chain attack compromised more than 400 packages in the Arch Linux User Repository (AUR) and introduced a malicious npm package, delivering credential-stealing malware paired with a kernel-level rootkit that hides itself from standard security tools. Developer workstations and CI/CD build environments are the primary targets, with confirmed exfiltration of GitHub credentials, SSH private keys, HashiCorp Vault tokens, and secrets from Slack, Teams, Discord, and browser stores. Any organization whose developers use Arch Linux or install packages from AUR must assume compromise of all secrets accessible on affected machines and treat full system reinstallation as the minimum remediation threshold.

Technical Analysis

The campaign compromised 400+ AUR packages and the npm package 'atomic-lockfile' as part of a cross-ecosystem supply chain strategy (T1195.001, T1554). The dual-payload delivers: (1) an infostealer targeting GitHub credentials, SSH keys (T1552.004), HashiCorp Vault tokens, browser-stored credentials (T1555.003), and secrets from Electron-based collaboration apps including Slack, Teams, Discord, and Telegram (T1555); and (2) an eBPF-based rootkit (T1014) operating at kernel level that hides processes, files, and network connections from userspace inspection tools, including ps, ls, netstat, and standard EDR agents. The rootkit achieves persistence via system initialization hooks (T1547, T1543) and is loaded through bash/shell

script execution (T1059.004). Stolen data is archived (T1560) and exfiltrated over network channels (T1041). File discovery (T1083) and system information collection (T1082) support targeting of high-value secrets. Local account credentials (T1078.003) and credentials from files (T1552.001) round out the credential collection scope. Because the eBPF rootkit operates below userspace, kernel-level compromise renders standard forensic tools unreliable. Vendor advisory for AUR package scope is not yet published as of 2026-06-12. No CVE is assigned to this campaign. Relevant CWEs: CWE-506 (Embedded Malicious Code), CWE-494 (Download of Code Without Integrity Check), CWE-693 (Protection Mechanism Failure). Threat actor is unattributed as of 2026-06-12. Sources: BleepingComputer, The Hacker News, Sonatype.

Action Checklist

1. Step 1: Containment. Immediately isolate any Arch Linux developer workstations and build servers from the network. Revoke and rotate all secrets potentially accessible on those systems: GitHub personal access tokens and deploy keys, SSH private keys, HashiCorp Vault tokens, and API credentials stored in Slack, Teams, Discord, Telegram, and browser credential stores. Treat every secret on an affected machine as compromised regardless of observed exfiltration evidence, because the eBPF rootkit suppresses visibility. Disable the 'atomic-lockfile' npm package in any pipeline dependency manifest and block its installation at the package registry or firewall level (CIS 2.3, Address Unauthorized Software).
2. Step 2: Detection. Standard process-listing and file-inspection tools are unreliable on rootkit-compromised hosts; prioritize out-of-band detection. Review AUR package installation history against the list of 400+ confirmed compromised packages published by BleepingComputer and Sonatype (retrieve from their official security advisories). Audit npm lock files and CI/CD dependency manifests for 'atomic-lockfile'. Check network egress logs for anomalous outbound connections from developer endpoints to unknown external IPs, particularly following AUR package installation events (NIST AU-6, Audit Record Review, Analysis, and Reporting; CIS 8.2, Collect Audit Logs). On hosts where integrity is uncertain, use a trusted live-boot environment to inspect filesystem and running eBPF programs ('bpftool prog list' from trusted media). Monitor GitHub, SSH, and Vault audit logs for credential use from unfamiliar IPs or geographic locations following the suspected compromise window.
3. Step 3: Eradication. Because the eBPF rootkit operates at kernel level and hides itself from userspace tools, in-place remediation is not reliable. Perform full system reinstallation from trusted installation media on all affected Arch Linux hosts. Do not attempt to remove the rootkit in situ. Before rebuilding, capture disk images for forensic preservation if regulatory or legal requirements apply. Remove 'atomic-lockfile' from all npm dependency trees and regenerate lock files from a clean environment. Audit all AUR helper configurations (yay, paru, etc.) and disable or restrict AUR access at the organizational level until a vetted package allowlist is established (NIST CM-6, Configuration Management; CIS 2.1, Establish and Maintain a Software Inventory; CIS 2.3, Address Unauthorized Software).
4. Step 4: Recovery. After rebuilding systems from trusted media, reissue all rotated credentials to fresh systems only. Verify GitHub, SSH, Vault, and collaboration platform access logs show no continued unauthorized use of previously rotated secrets. Enforce code signing and integrity verification for all packages before installation (NIST SA-7, Software, Firmware, and Information Integrity; CIS 2.3, Address Unauthorized Software). Restore CI/CD pipelines only after dependency trees are audited and rebuilt clean. Monitor rebuilt systems for anomalous eBPF program loading ('bpftool prog list') as a post-recovery indicator (NIST AU-6; CIS 8.2).
5. Step 5: Post-Incident. This attack exploited the AUR's community-maintained, unvetted nature and the npm ecosystem's dependency resolution without integrity enforcement (CWE-494, CWE-693). Control

gaps to address: implement a software supply chain policy that restricts AUR use to organizationally approved and reviewed packages (CIS 7.1, Establish and Maintain a Vulnerability Management Process); enforce dependency pinning with hash verification in all CI/CD pipelines; deploy secrets management practices that store credentials in hardware-backed vaults rather than on developer workstation filesystems (NIST AC-6, Least Privilege; D3-CRO, Credential Rotation; D3-CH, Credential Hardening); establish a regular secrets rotation schedule independent of suspected compromise events. Evaluate whether developer workstations should be restricted from accessing production secrets directly.

Detection Guidance

Standard userspace tools (ps, ls, netstat, lsof, most EDR agents) are unreliable on hosts where the eBPF rootkit has loaded. Use a trusted live-boot environment or out-of-band hypervisor/cloud console inspection for host forensics. Key detection signals: (1) Run 'bpftool prog list' from trusted bootable media to enumerate loaded eBPF programs; presence of unrecognized programs not tied to known legitimate tools (e.g., systemd, network monitoring agents) is a strong indicator. (2) Audit AUR package installation logs (typically in pacman.log at /var/log/pacman.log) for any of the 400+ confirmed compromised package names published by BleepingComputer and Sonatype (retrieve from their official security advisories). (3) Search npm lock files (package-lock.json, yarn.lock) for 'atomic-lockfile' across all repositories and CI environments. (4) Review GitHub audit logs, SSH server logs, and HashiCorp Vault audit logs for access events from unrecognized IPs or user agents occurring after the compromise window (approximate start date: pre-2026-06-12 per discovery reporting). (5) Inspect network egress logs for outbound connections from developer endpoints to unknown external hosts, particularly following AUR or npm installation events. (6) Check browser credential store access logs where available and review Slack, Teams, Discord, and Telegram access logs for sessions from unrecognized devices or locations. Behavioral indicator: the rootkit hides its own processes and network connections, so a host appearing unusually quiet in monitoring tools despite active developer use may indicate rootkit presence rather than absence of activity. Confidence in any single signal is limited by the rootkit's concealment capability; treat corroborating signals across multiple data sources as the confirmation threshold.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://www.npmjs.com/package/atomic-lockfile`	Malicious npm package identified as part of the cross-ecosystem supply chain campaign; remove from all dependency trees	HIGH

Framework Mappings

MITRE-ATTACK

- **T1560** — Archive Collected Data
- **T1041** — Exfiltration Over C2 Channel
- **T1083** — File and Directory Discovery
- **T1014** — Rootkit

- **T1078.003** — Local Accounts
- **T1195.001** — Compromise Software Dependencies and Development Tools
- **T1082** — System Information Discovery
- **T1543** — Create or Modify System Process
- **T1555** — Credentials from Password Stores
- **T1059.004** — Unix Shell
- **T1555.003** — Credentials from Web Browsers
- **T1552.004** — Private Keys
- **T1547** — Boot or Logon Autostart Execution
- **T1552.001** — Credentials In Files
- **T1554** — Compromise Host Software Binary

NIST-800-53R5

- **CA-7** — Continuous Monitoring
- **SC-7** — Boundary Protection
- **SI-4** — System Monitoring
- **CM-7** — Least Functionality
- **SI-3** — Malicious Code Protection
- **SI-7** — Software, Firmware, and Information Integrity
- **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

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

ISO-27001-2022

- **A.5.21** — Managing information security in the ICT supply chain

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1560	Archive Collected Data	Collection
T1041	Exfiltration Over C2 Channel	Exfiltration
T1083	File and Directory Discovery	Discovery
T1014	Rootkit	Defense-Evasion
T1078.003	Local Accounts	Defense-Evasion
T1195.001	Compromise Software Dependencies and Development Tools	Initial-Access
T1082	System Information Discovery	Discovery
T1543	Create or Modify System Process	Persistence
T1555	Credentials from Password Stores	Credential-Access
T1059.004	Unix Shell	Execution
T1555.003	Credentials from Web Browsers	Credential-Access
T1552.004	Private Keys	Credential-Access
T1547	Boot or Logon Autostart Execution	Persistence
T1552.001	Credentials In Files	Credential-Access
T1554	Compromise Host Software Binary	Persistence

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/over-400-arch-linux-...	T3
Atomic Arch npm Campaign Adds Malicious Dependency - Sonatype	https://www.sonatype.com/blog/atomic-arch-npm-campaign-adds-malicio...	T3
Over 400 Arch Linux AUR Packages Hijacked to Deploy Infostealer ...	https://thehackernews.com/2026/06/over-400-arch-linux-aur-packages....	T3

Source	URL	Tier
Over 400 Arch Linux packages compromised to push rootkit ... - Reddit	https://www.reddit.com/r/cybersecurity/comments/1u41u02/over_400_ar...	T3
Staying Safe On Arch Linux - YouTube	https://www.youtube.com/watch?v=bTKATeeQlvM	T3

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-06-15 06:39 UTC by TJS Security Command Center