

Forum:[Hack Forums](#)

Thread: [Arch Linux AUR Attack](#)

Omniscient - August 3, 2026, 11:00 am

Arch Linux AUR Targeted in Major Supply-Chain Attack

Attackers hijacked trusted orphaned packages to distribute credential-stealing and rootkit-like malware

The Arch User Repository, better known as the AUR, has been targeted in a coordinated software supply-chain attack involving the takeover of legitimate but abandoned packages.

The campaign, dubbed Atomic Arch by security researchers, was publicly disclosed on June 11, 2026. Arch Linux issued its own warning the following day, confirming that the AUR was experiencing a high volume of malicious package adoptions and updates.

Unlike a traditional malware campaign that relies on tricking users into downloading an obviously suspicious program, the attackers reportedly adopted existing orphaned AUR packages that already had legitimate histories, recognizable names, and established users.

They then modified the packages' build instructions to retrieve and execute malicious dependencies.

What Is the AUR?

The Arch User Repository is a community-operated collection of package build instructions for software that is not available through Arch Linux's official repositories.

An AUR package usually contains a file called a PKGBUILD. This file tells the user's system where to download the software and how to compile and package it.

This distinction is important:

Official Arch packages are built and distributed through Arch Linux's official infrastructure.

AUR packages are community-maintained build recipes.

AUR helpers such as `yay` and `paru` automate downloading and building those recipes.

Commands contained inside a PKGBUILD or installation script may execute directly on the user's computer.

The AUR has always required a greater degree of user scrutiny than an official package repository.

This incident demonstrates why blindly accepting AUR updates can be dangerous.

How the Attack Worked

The attackers appear to have focused heavily on orphaned packages—packages whose previous maintainers had abandoned them.

The attack followed a relatively simple process:

Identify an abandoned package

The attackers searched for legitimate AUR projects that no longer had active maintainers.

Adopt the package

Through the normal AUR stewardship process, an attacker could take responsibility for an orphaned package while retaining its existing name, history, votes, comments, and reputation.

Modify the build instructions

The attackers changed PKGBUILD files or installation hooks so that building the package would install additional dependencies from the npm ecosystem.

Execute the malicious payload

When a user installed or updated the affected AUR package, the modified build process retrieved and executed attacker-controlled code.

One of the malicious dependencies was named `atomic-lockfile`. Later activity reportedly involved additional packages such as `js-digest` and `lockfile-js`, with some attack variations using Bun instead of npm.

The compromised AUR package itself did not necessarily contain the final malware. Instead, its build instructions downloaded the malicious component during installation.

What Did the Malware Target?

Static analysis of the payload found functionality consistent with credential theft, system reconnaissance, stealth, anti-debugging, and data exfiltration.

Researchers identified references to data associated with:
SSH private keys and SSH configuration

GitHub credentials and access tokens

npm authentication tokens

HashiCorp Vault tokens

Browser cookies and active login sessions

Cloud and development credentials

Slack, Discord, Microsoft Teams, and Telegram data

Developer and continuous-integration environments

The malware also reportedly contained references to eBPF functionality capable of hiding processes, files, network activity, or other system artifacts.

eBPF is a legitimate and powerful Linux technology, but malicious software can abuse it to obtain rootkit-like stealth when running with sufficient privileges.

How Many Packages Were Affected?

The exact number remains uncertain.

Early investigations identified several dozen compromised packages. Follow-up reporting described more than 400 maliciously modified AUR packages, while Sonatype later said preliminary analysis suggested that approximately 1,500 packages may have been affected across multiple waves.

Arch Linux has not published a definitive total. Its official advisory described the event as a high volume of malicious package adoptions and updates.

The package count should therefore be treated as an evolving estimate rather than a finalized number.

Who May Be Affected?

The incident primarily affects people or organizations that installed or updated AUR packages during the attack period.

Potentially affected systems include:
Arch Linux desktops and servers

Arch-based distributions using the AUR

EndeavourOS installations using AUR packages

Manjaro systems where the AUR was manually enabled

Arch installations running inside WSL

Self-hosted development or CI systems that build AUR packages

Users who only installed software from Arch Linux's official repositories are not directly affected by this specific AUR attack vector.

Simply running Arch Linux does not automatically mean a system was compromised.

What Arch Linux Users Should Do

Arch Linux has advised users to review all PKGBUILD and installation-script changes before installing AUR updates.

Users should take the following precautions:

Stop blindly accepting AUR updates

Inspect the PKGBUILD, associated installation scripts, and recent Git commits before building an update.

Look for unexpected dependency installation

Be suspicious of newly added commands involving `npm`, `bun`, `curl`, `wget`, encoded shell commands, unfamiliar domains, or executable downloads.

Review recently updated AUR packages

Check packages installed or upgraded around June 11 and June 12, 2026, along with any packages updated during the surrounding period.

Do not assume uninstalling the package is sufficient

If a malicious installation script executed, removing the original AUR package may not remove the downloaded payload or reverse credential theft.

Treat exposed credentials as compromised

Rotate SSH keys, GitHub tokens, npm tokens, cloud API keys, Vault tokens, and other credentials accessible from the affected machine.

Terminate active sessions

Invalidate browser sessions and sign out other active sessions for important accounts.

Reinstall severely affected systems

If the payload executed with root privileges or there is evidence of rootkit-like activity, the safest response is to reinstall the operating system from trusted media and restore only verified data.

Changing passwords from the potentially infected computer is not recommended. Credentials should be changed from a separate, trusted device.

How to Review an AUR Update

Before approving an update through an AUR helper, review the changes instead of automatically pressing Enter.

For a locally cloned AUR package, commands such as the following may help:

```
Code git log --oneline --decorate -10
git diff HEAD~1 HEAD
less PKGBUILD
Pay particular attention to changes inside:
```

```
Code prepare()
build()
check()
package()
post_install()
post_upgrade()
Also inspect any files ending in:
```

```
Code .install
.sh
.service
.timer
```

A valid checksum only proves that the downloaded file matches the checksum listed by the package maintainer. It does not prove that either the file or the PKGBUILD is trustworthy when the maintainer account itself is malicious.

Why This Attack Matters

The most significant part of Atomic Arch is that the attackers did not need to establish trust from scratch.

They inherited it.

By adopting abandoned packages, the attackers gained control of projects that users already

recognized and may have installed for years. A familiar package name, an old submission date, positive votes, or a long comment history did not protect users once control of the build instructions changed.

The incident highlights several weaknesses common throughout open-source software ecosystems: Abandoned projects can retain substantial downstream trust.

Package ownership transfers may receive limited scrutiny.

Build scripts frequently execute with powerful permissions.

Automated update tools can hide significant code changes.

A small modification can introduce malware from another package ecosystem.

Developer computers often contain credentials that provide access to larger systems.

A compromised developer workstation can become an entry point into source-code repositories, cloud infrastructure, CI pipelines, production servers, or other organizations.

Official Statement

Arch Linux confirmed the incident on June 12, 2026, stating that it was working to identify existing malicious commits and prevent additional malicious updates.

The project also warned that users could experience temporary restrictions when:
Creating new AUR accounts

Pushing package updates

Adopting packages

Creating new packages

Suspicious AUR commits should be reported to Arch Linux staff through the aur-general mailing list.

Conclusion

The Atomic Arch campaign is a reminder that community repositories should not be treated as equivalent to official, curated software repositories.

AUR helpers make package installation convenient, but convenience does not replace review. Users are ultimately executing build instructions supplied by community maintainers, and the trustworthiness of those instructions can change after a package is transferred to a new owner.

Anyone who installed suspicious AUR updates during the affected period should treat credential exposure seriously. If a malicious payload executed with elevated privileges, wiping and rebuilding the system may be safer than attempting to clean it manually.

Sources

[Arch Linux: Active AUR Malicious Packages Incident](#)

[Sonatype: Atomic Arch Campaign Analysis](#)

[StepSecurity: Atomic Arch Supply-Chain Analysis](#)

AUR users should always inspect PKGBUILD and installation-script changes before building or updating community packages.

Dennis - August 3, 2026, 11:16 am

It has always struck me as ironic that some people consider themselves highly advanced Linux users, yet install packages from the AUR without a second thought.

AUR is still a 3rd party repo. It carries many of the same supply-chain risks as npm, nuget, rpm, github projects, or any other source of community-maintained software. Being popular or convenient does not automatically make it trustworthy.

rd party code should always be reviewed and treated with caution, regardless of the ecosystem.

Omniscient - August 3, 2026, 12:24 pm

(Aug 3, 2026 11:16 AM)Dennis Wrote: [\[CITE\]](#) It has always struck me as ironic that some people consider themselves highly advanced Linux users, yet install packages from the AUR without a second thought.

AUR is still a 3rd party repo. It carries many of the same supply-chain risks as npm, nuget, rpm, github projects, or any other source of community-maintained software. Being popular or convenient does not automatically make it trustworthy.

rd party code should always be reviewed and treated with caution, regardless of the ecosystem. At this point doing any updates should only be done if it's older than 1 week unless it's a CRITICAL or SEVERE rated exploit. And only the packages that are under the exploit.

apt get update isn't suitable anymore imho

Dotexe - August 3, 2026, 3:28 pm

(Aug 3, 2026 11:16 AM)Dennis Wrote: [\[CITE\]](#) It has always struck me as ironic that some people consider themselves highly advanced Linux users, yet install packages from the AUR without a second thought.

AUR is still a 3rd party repo. It carries many of the same supply-chain risks as npm, nuget, rpm, github projects, or any other source of community-maintained software. Being popular or convenient does not automatically make it trustworthy.

rd party code should always be reviewed and treated with caution, regardless of the ecosystem.

That's why I intentionally leave most of my packages outdated, unless there is a patch that fixes Oday which then again is also manually reviewed.

Omniscient - August 4, 2026, 9:22 am

It looks like Arch has locked down the AUR now too.

My X feed full of info about this.

Dotexe - August 4, 2026, 10:02 am

(Aug 4, 2026 09:22 AM)Omniscient Wrote: [\[CITE\]](#) It looks like Arch has locked down the AUR now too.

My X feed full of info about this.

Sometimes I genuinely wonder how people can not give a damn about securing their fucking accounts when a compromise can eventually lead to a full blown supply chain attack. Why should everyone else have to worry about the consequences of someone's incompetence when they can't even secure their own repo? I've seen in the past some vendors getting hacked because they used 3rd party library that got compromised because dude is using 2008 password with no MFA

Omniscient - August 4, 2026, 10:18 am

(Aug 4, 2026 10:02 AM)Dotexe Wrote: [\[CITE\]](#) Sometimes I genuinely wonder how people can not give a damn about securing their fucking accounts when a compromise can eventually lead to a full blown supply chain attack. Why should everyone else have to worry about the consequences of someone's incompetence when they can't even secure their own repo? I've seen in the past some vendors getting hacked because they used 3rd party library that got compromised because dude is using 2008 password with no MFA

This is why some sites require 2FA for certain account permissions.

Betty Blue - August 5, 2026, 5:45 am

The vast majority of AUR users aren't installing orphaned packages. Arch devs have always maintained that the AUR is a third party repo, and to carefully pay attention to what is installed from the AUR.

If you got hit with this, you're not an "advanced user". This was also weeks ago lol, and easily solvable on a user machine.

Use hooks to check for an orphaned package before it's updated, use hooks to display pkgbuild changes. On top of that tons of new tools have come out for checking packages from the AUR.

(Aug 3, 2026 12:24 PM)Omniscient Wrote: [\[CITE\]](#) At this point doing any updates should only be done if it's older than 1 week unless it's a CRITICAL or SEVERE rated exploit. And only the packages that are under the exploit.

apt get update isn't suitable anymore imhoThat's not right. This started on June 9th, by the 12th I believe they had already stopped user accounts being created and orphaned packages being adopted. It's a 3 day window when Atomic Arch could have affected anyone from installing a malicious package.

List of affected packages: <https://md.archlinux.org/s/SxbqukK6IA>