

Remediating 1000s of untracked vulnerabilities in nixpkgs

delroth — FOSDEM 2024

CVE-2023-4863

“A **buffer overflow in libwebp** which allows a malicious actor to potentially get code execution in software that displays a specially crafted image file. This impacts pretty much all web browsers, as well as other software which might process or display untrusted images (image editing software, email clients, chat clients, social media clients, etc.). Chrome has rated this vulnerability as **critical severity** and has indicated that they have evidence **some actors are already exploiting it in the wild.**”

● libwebp: cherry-pick suspected upstream fix for CVE-2023-4863

This CVE is critical severity and has been exploited in the wild. It was reported as being a Chromium vulnerability, but it seems to in fact impact libwebp (and thus all its downstream users). There is however no official confirmation of this yet.

The upstream fix patch ([webmproject/libwebp@902bc919](#)) does not cleanly apply onto 1.3.1, so we vendor a very slightly modified version which does cleanly apply. This is my original work, so YMMV on whether you trust it or not, reviews very much welcomed :-)



master (#254775)



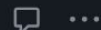
24.05-pre ... 23.11-beta



delroth committed on Sep 12, 2023 Verified

commit 0f11042876c07f1abbe172d9c8fe41feedd0be9c

> 361 ■■■■■ pkgs/development/libraries/libwebp/CVE-2023-4863.patch



> 8 ■■■■■ pkgs/development/libraries/libwebp/default.nix



Problem solved!



delroth commented on Sep 12, 2023 • edited by vcunat ▾

Member



Filing this issue to track [CVE-2023-4863](#) related actions in nixpkgs. Feel free to send questions my way and/or contribute via comments in this issue!

Current status

Firefox and Chromium are not vulnerable anymore as of 2023-09-16 in `unstable` and `23.05`. Direct dependents of the system `libwebp` are also not vulnerable anymore. **Some applications bundle their own version of libwebp** instead of using the system version (including some other web browsers in nixpkgs: Brave, Tor Browser, etc.). **Each of these need to be updated separately by nixpkgs maintainers.** See below for a list of all the known applications that need an update and their status.

[nixpkgs#254798](#)

Vendoring



delroth commented on Oct 2, 2023

Member

Author



All the packages I evaluated as being high-risk have now been taken care of (and most of them did actually get updates, woo! only marked 2 or 3 as insecure). Right now I don't think anyone has the bandwidth to try and track the rest. I'm going to close this bug - if someone does want to take over the remediation for the rest of the impacted packages, feel free to reopen and assign yourself.

(Note however that there is a large overlap between vulnerable to this libwebp vuln and vulnerable to the recent libvpx vuln... so maybe just go help over there instead of reusing this bug.)



1

Packages containing...

libwebp copies in nixpkgs? 116

libpng copies in nixpkgs? 237

libjpeg copies in nixpkgs? 253

zlib copies in nixpkgs? 761

on nixpkgs-unstable, as measured on 2024-02-02,
unfree and insecure packages excluded

Is this a problem?

libpng per version

2 1.2.7
2 1.5.23
2 1.6.18
4 1.6.2
4 1.6.22
4 1.6.23
4 1.6.25
4 1.6.28
4 1.6.35
6 1.2.57
7 1.6.36
12 1.6.29

release date: 2004-09-12

release date: 2013-04-25

12 1.6.34
14 1.5.10
16 1.2.56
16 1.5.26
27 1.6.38
28 1.2.59
33 1.7.0
100 1.6.40
104 1.6.39
196 1.6.37

release date: 2012-03-29

Rust software analysis in nixpkgs

1844 Rust packages (has cargoDeps)

1149 locked to vulnerable dependencies (62%)

744 with high or critical severity vulnerabilities in dependencies (40%)

Some of it is nixpkgs's fault, most of it is upstream's fault...

What is causing vendoring?

- We don't try to prevent it.
- Newer language ecosystems encourage it.
- We don't have the tooling to detect and measure it.

Policies & Documentation

Packages fetched from AppImages: 58 (excl. unfree)

Packages fetched from .deb files: 66 (excl. unfree)

Many of these could be built from source, but it's harder!

nixpkgs does not currently document a preference for building from source. Some other distros do, famously Debian.

Rust, Go, NPM, Java, .NET

Lockfiles sound great, except upstream doesn't keep them up to date.

The shift towards lockfile-based language ecosystems mean distros have limited ways to fix vulnerable dependency. Upstreams don't understand this, or don't care.

- nixpkgs is special: huge package set, containing software that would in many distros be relegated to community / unofficial repos.
- Users should be made more aware of the risks: `knownVulnerable`, etc.

Tooling

Until recently, no tooling to detect or measure vendoring in nixpkgs.

In the wake of CVE-2023-4863: [github:delroth/grep-nixos-cache](https://github.com/delroth/grep-nixos-cache)

- Via simple signatures (currently, strings), find vendoring of common libs.

WIP: [github:delroth/nixpkgs-vendored-vulns-scan](https://github.com/delroth/nixpkgs-vendored-vulns-scan)

- Focusing on language specific ecosystems and lockfiles.

Conclusion

Conclusion

With new tooling, we have a better idea of the scale of vendoring in nixpkgs.

It's not great.

This talk does not come with any immediate solutions that can be applied. But a combination of policy changes, tooling improvement, and better support for informing users of the maintenance status of software will likely be necessary.

Questions & Contact Info

Github: [delroth](#)

Matrix: [@delroth:delroth.net](#)

Mastodon: [@delroth@delroth.net](#)

Email: delroth@delroth.net

[github:delroth/grep-nixos-cache](#)

[github:delroth/nixpkgs-vendored-vulns-scan](#)

Thank you!