



KubeCon



CloudNativeCon

Europe 2026

#KubeCon #CloudNativeCon

Why Isn't the Fix in My Container?

Tracking CVE Propagation Across 10,000 Projects





Mor Weinberger

Software Architect

Echo

 morwn



Lior Kaplan

Open Source Expert

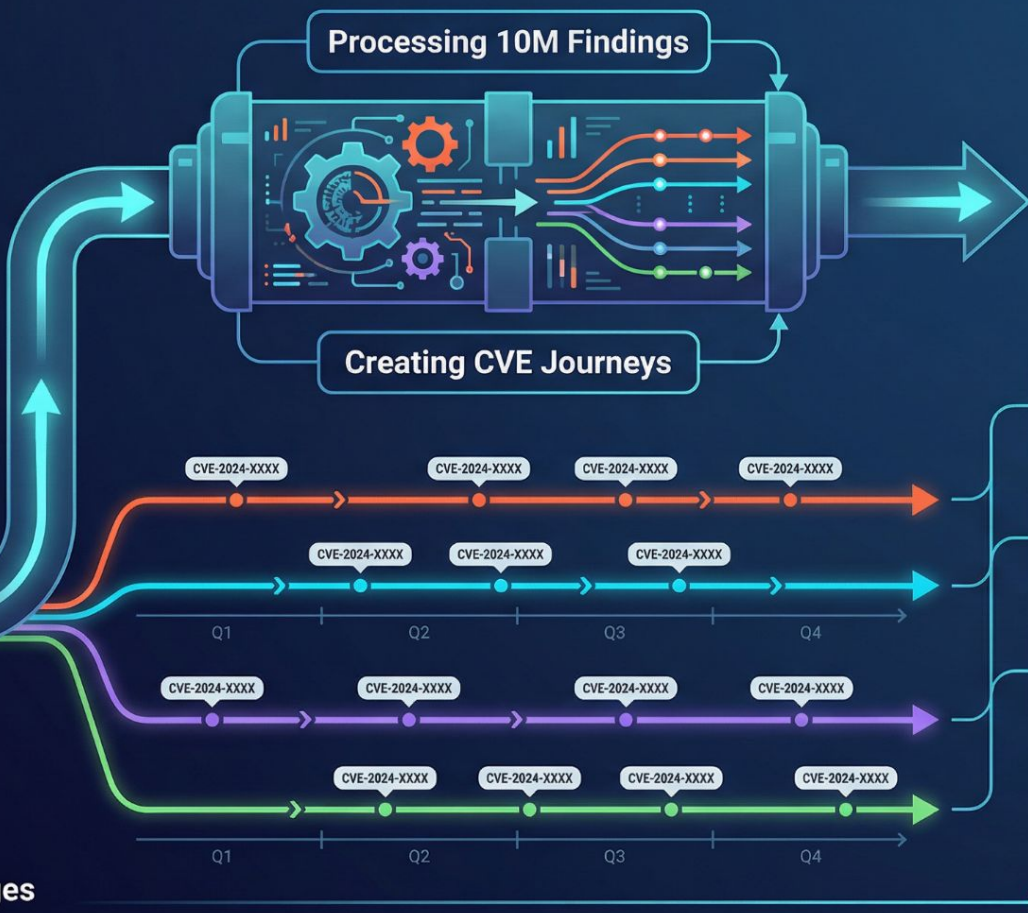
Kaplan Open Source

 liorkaplan



365

1 Year Daily Scans
Hundreds of OSS Images



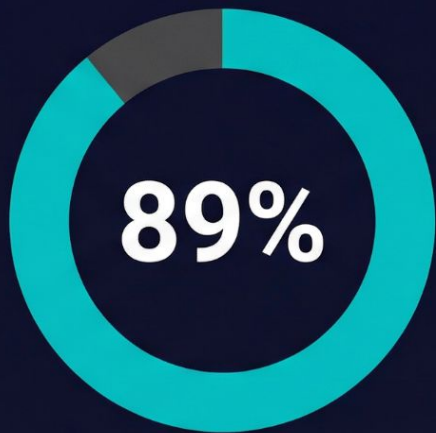
40K
CVE
Journeys

MTTR
Analysis

Fix %

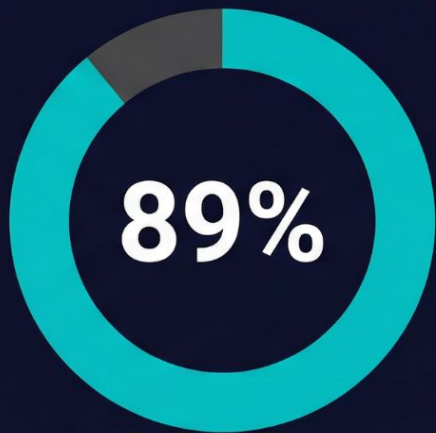
Propagation
Lag





of CVEs Have a Fix

Remediable with existing patches

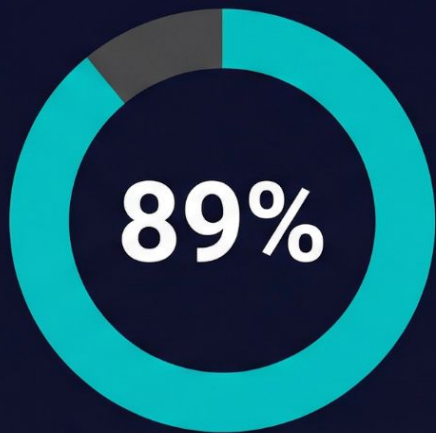


of CVEs Have a Fix

Remediable with existing patches



The Fix is Ready



of CVEs Have a Fix

Remediable with existing patches



The Fix is Ready

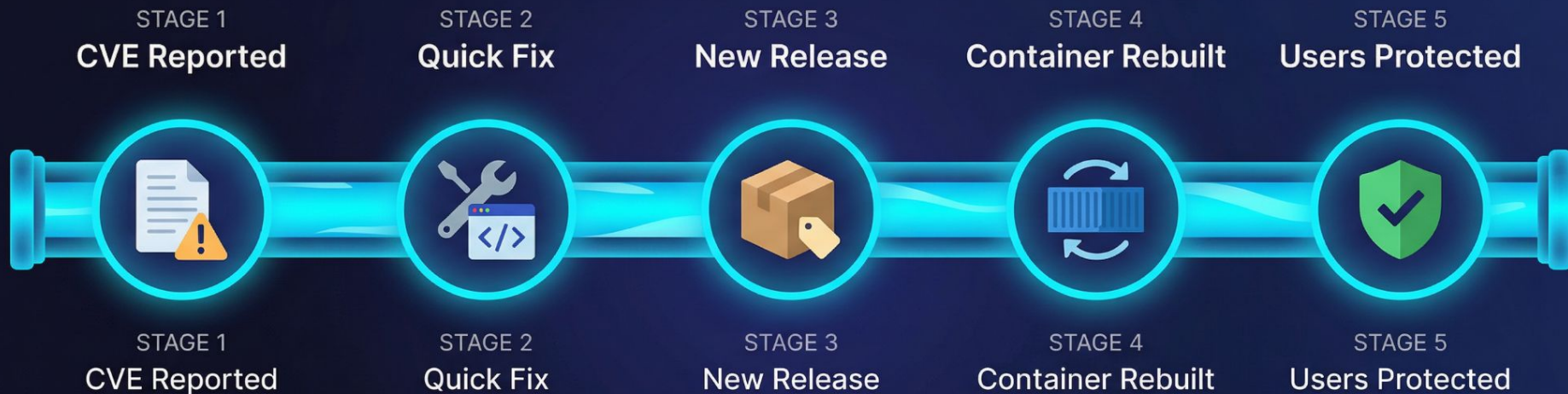
40%

take 180+ days

to remediate

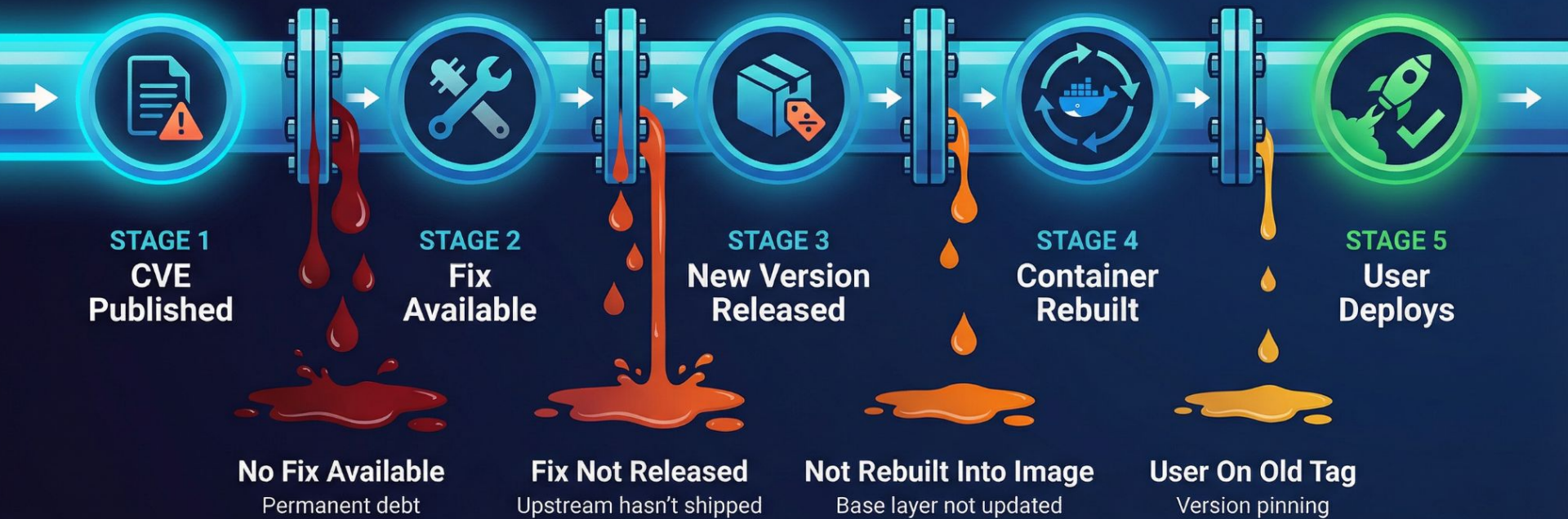


The Happy Path



In an ideal world:
1-2 days from CVE to patched container

Works for upstream-maintained images (e.g., curlimages/curl)



Journey 1: Discovery to Fix

Journey 2: Fix to Deployment

! Patch Available $\neq$ Risk Gone



Vulnerability
Discovered



CVE
Assigned



Fix Developed
& Accepted



New Version
Released



Container
Image Rebuilt



User Deploys
Updated Image

From problem discovery to it being fixed

From fix accepted to users getting the fixed version

Journey 1: Discovery to Fix



KubeCon



CloudNativeCon

Europe 2026



Issue Found



CVE Being Reported



Maintainer Fixes the Issue



Everyone Wins

🙋 Raise Your Hand if You Heard About 🙋

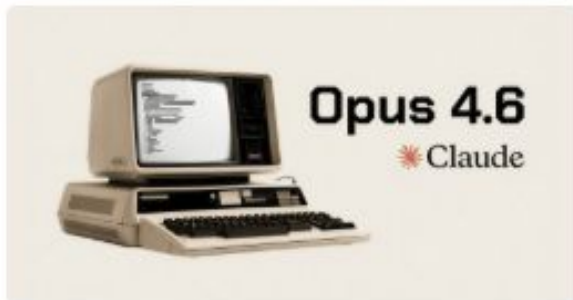


KubeCon



CloudNativeCon

Europe 2026



Claude Opus 4.6 Finds 500+ High-Severity Flaws Across Major Open-Source Libraries

📅 Feb 06, 2026 Artificial Intelligence / Vulnerability

Artificial intelligence (AI) company Anthropic revealed that its latest large language model (LLM), Claude Opus 4.6, has found more than 500 previously...

👋 Raise Your Hand if You Heard About 👋



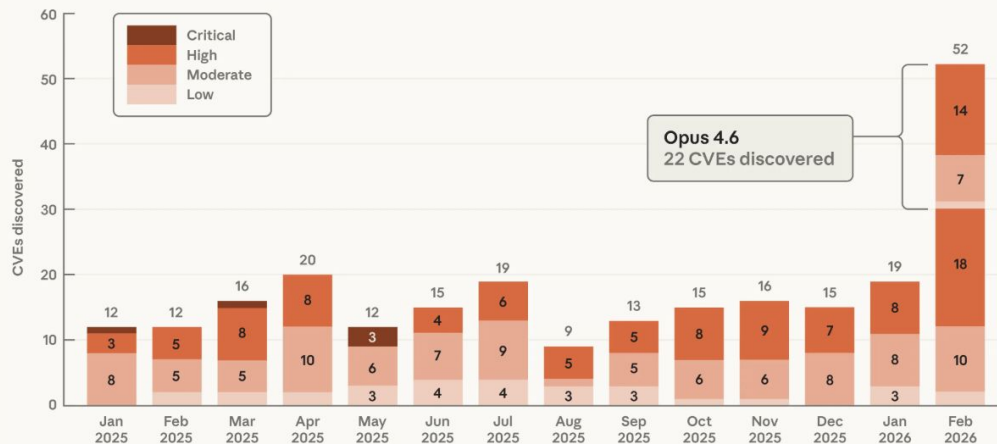
KubeCon



CloudNativeCon

Europe 2026

Firefox Security Vulnerabilities by Month

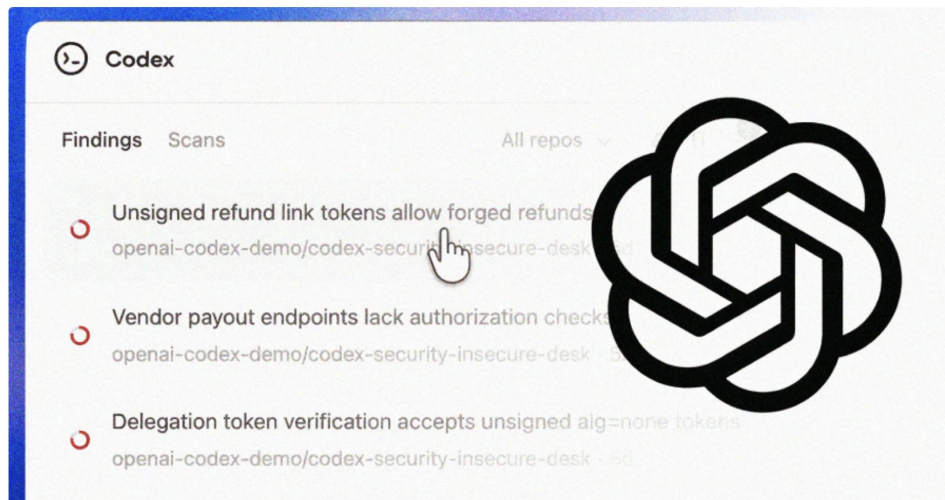


Firefox security vulnerabilities reported from all sources, by month. Claude Opus 4.6 found 22 vulnerabilities in February 2026, more than were reported in any single month in 2025.

🙌 Raise Your Hand if You Heard About 🙌

OpenAI Codex Security Scanned 1.2 Million Commits and Found 10,561 High-Severity Issues

👤 Ravie Lakshmanan 📅 Mar 07, 2026



Journey 1: Discovery to Fix



KubeCon



CloudNativeCon

Europe 2026



~~Issue Found~~ - Use AI



~~CVE Being Reported~~ - Use AI



~~Maintainer Fixes the Issue~~ - Use AI



Everyone Wins - goes Skiing



KubeCon



CloudNativeCon

Europe 2026

Solved Problem, Right?





KubeCon



CloudNativeCon

Europe 2026

What About the maintainers side?



Reports Quality over Quantity



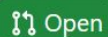
KubeCon



CloudNativeCon

Europe 2026

BUG-BOUNTY.md: we stop the bug-bounty end of Jan 2026



Open

bagder wants to merge 2 commits into `master` from `bagder/rm-bug-bounty`



Conversation 24



Commits 2



Checks 194



Files changed 11



bagder commented 2 weeks ago • edited ▾

Member

Remove mentions of the bounty and hackerone.

There will be more mentions, blog posts, timings etc in the coming weeks.



148



105



17

<https://github.com/curl/curl/pull/20312>

Reports Quality over Quantity



KubeCon



CloudNativeCon

Europe 2026

Triaging security issues reported by third parties

 Closed  Issue created 9 months ago by Nick Wellnhofer

I have to spend several hours each week dealing with security issues reported by third parties. Most of these issues aren't critical but it's still a lot of work. In the long term, this is unsustainable for an unpaid volunteer like me. I'm thinking about some changes that allow me to continue working on libxml2. The basic idea is to treat security issues like any other bug. They will be made public immediately and fixed whenever maintainers have the time. There will be no deadlines. This policy will probably make some downstream users nervous, but maybe it encourages them to contribute a little more.

The more I think about it, the more I realize that this is the only way forward. I've been doing this long enough to know that most of the secrecy around security issues is just theater. All the "best practices" like OpenSSF Scorecards are just an attempt by big tech companies to guilt trip OSS maintainers and make them work for free. My one-man company recently tried to become a OpenSSF member. You have to become a Linux Foundation member first which costs at least \$10,000/year. These organizations are very exclusive clubs and anything but open. It's about time to call them and their backers out.

In the long run, putting such demands on OSS maintainers without compensating them is detrimental. I just stepped down as libxslt maintainer and it's unlikely that this project will ever be maintained again. It's even more unlikely with Google Project Zero, the best white-hat security researchers money can buy, breathing down the necks of volunteers.

Edited 9 months ago by Nick Wellnhofer



27



0



3



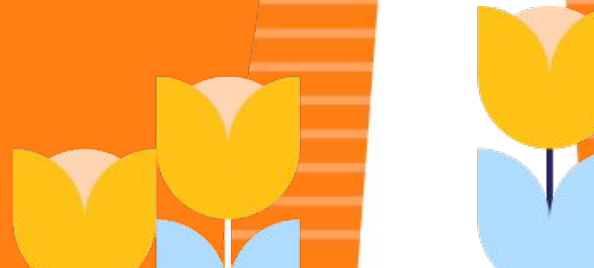
KubeCon



CloudNativeCon

Europe 2026

But wait.. Is this just a security issue?



Contributions Quality over Quantity



KubeCon



CloudNativeCon

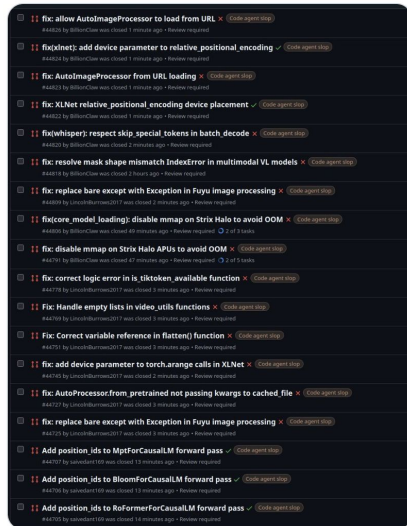
Europe 2026



clem 🤖
@ClementDelangue

Our biggest open-source repos are getting overwhelmed by AI slop which literally makes Github unusable (~a new pull request every 3 minutes).

Fun new challenges in an agentic world!



5:43 PM · Mar 18, 2026 · 147.9K Views



GitHub ponders kill switch for pull requests to stop AI slop

Code community site begins to see that AI could drive people away

 [Thomas Claburn](#)

Tue 3 Feb 2026 // 20:32 UTC

UPDATED GitHub, the Microsoft code-hosting shop that popularized AI-assisted software development, is having some regrets about its Copilot infatuation.

Last week, product manager Camilla Moraes opened [a GitHub community discussion](#) to address "a critical issue affecting the open source community: the increasing volume of low-quality contributions that is creating significant operational challenges for maintainers."

AI slop has come home to roost and GitHub wants help from its community of software developers to figure out how to manage the mess.

"We've been hearing from you that you're dedicating substantial time to reviewing contributions that do not meet project quality standards for a number of reasons – they fail to follow project guidelines, are frequently abandoned shortly after submission, and are often AI-generated," Moraes wrote.

Can this be replaced with AI?



KubeCon



CloudNativeCon

Europe 2026

☐ AI is powerful and can improve OSS

☐ Importance of Quality (over quantity)

☐ Have AI to help fix security reports

☐ Work with maintainers, reduce fatigue

Steps of Security Reports



KubeCon



CloudNativeCon

Europe 2026



Responsible Disclosure



Discuss CVE assignment



Help with fix / verification



Work with maintainers, reduce fatigue

The Wrong Way



KubeCon



CloudNativeCon

Europe 2026



Ask for CVE (for your resume)



Post issue online



Let maintainers find the CVE themselves



Complain issues are not fixed

 Closed  LibXML2 xmlcatalog Infinite Recursion

doubt that anyone is still using SGML catalogs at all.

**Alan Coopersmith** @alanc 6 months ago 
VulDB assigned CVE-2025-8732 to this bug.

**Michael Catanzaro** @mcatanzaro 5 months ago Developer 
Thanks for the heads-up, Alan. I appreciate that, since otherwise GNOME Security would not know about the existence of the CVE.

I'm pretty sure libxml2 should be robust to invalid input, regardless of whether it makes sense to use the command in this way or not. Low-severity CVEs might be annoying, but we'd need pretty strong justification to claim that it's not a valid availability issue.

That said, I wasn't able to reproduce the problem myself. Notably, the POC file is just `<?xm` which is strange; is that really correct and not corrupted? Running the provided `xmlcatalog` command fails with `could not open /etc/sgml/catalog for saving`, which seems like a good failure as I certainly wasn't expecting that to try to modify my /etc.

Lior: *“I was wondering if you had a chance to look at <https://nvd.nist.gov/vuln/detail/CVE-2025-45582>”*

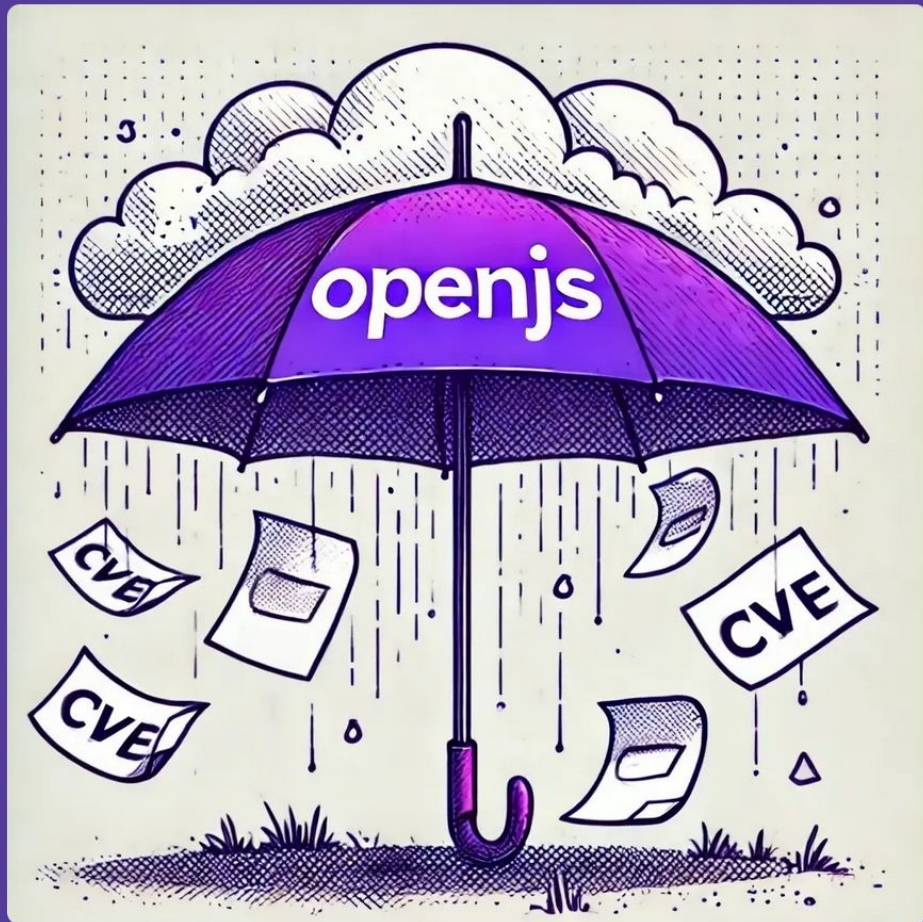
Maintainer: *“First I've heard of it. Thanks for mentioning it.”*

<https://lists.gnu.org/archive/html/bug-tar/2025-08/msg00001.html>

SECURITY NEWS

OpenJS Foundation Is Now a CNA for 40+ JavaScript Projects Under Its Umbrella

New CNA status enables OpenJS Foundation to assign CVEs for security vulnerabilities in projects like ESLint, Fastify, Electron, and others, while leaving disclosure responsibility with individual maintainers.



From a problem to a Fix



KubeCon



CloudNativeCon

Europe 2026



Issue Found



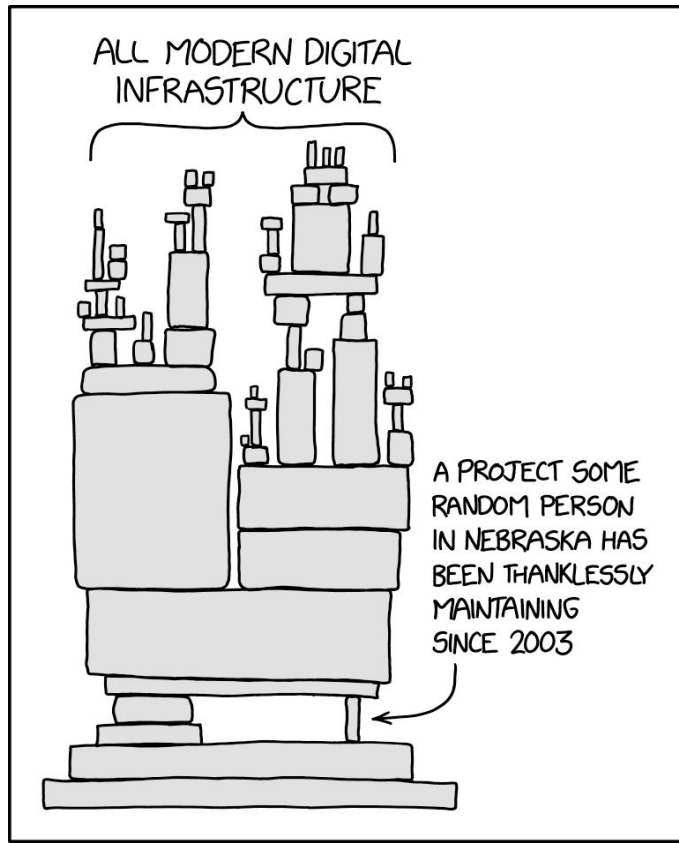
CVE Being Reported



Maintainer Fixes the Issue



Everyone Wins



Maintainers are volunteers



KubeCon



CloudNativeCon

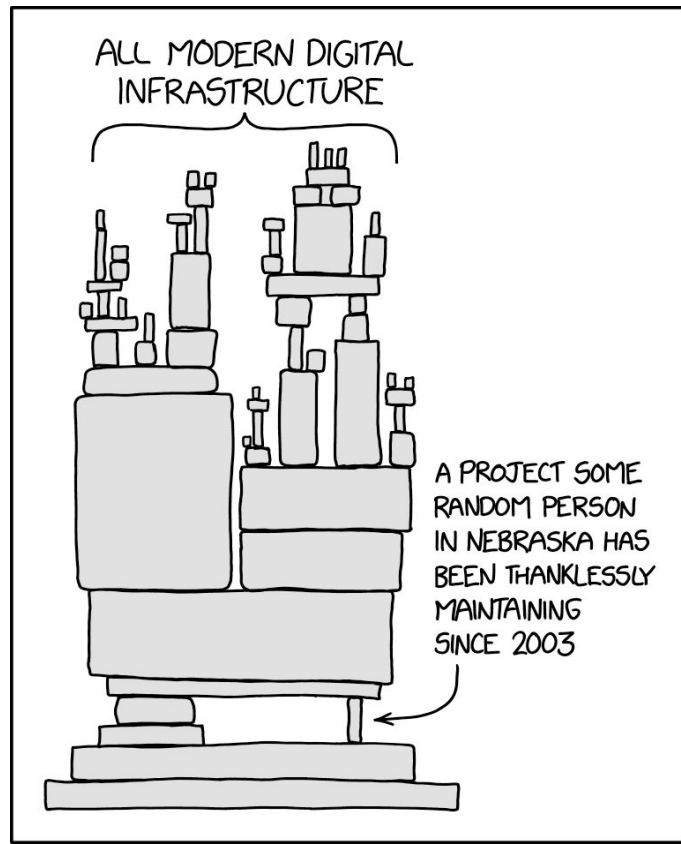
Europe 2026

☐ No SLA

☐ Be respectful

☐ Offer to help

☐ Contribute / Sponsor



From Fix to User

A patch is merged. Now what? Each step is a waiting game.



Patch committed & merged

Release that includes the fix

- - Immediate release (critical severity)
- - Wait for next release cycle (could be weeks/months)

Scanners detect the fixed version

Requires changelog / CVE advisory to be updated

Direct dependencies updated

Your package manager picks up the new version

Indirect dependencies updated

Longer cycle — transitive deps may lag behind

Container image rebuilt with the fix

Image maintainer must trigger a new build



Days



Weeks



Months

Why Can't we just use the new version?

The fix exists, but upgrading isn't always straightforward

The Problem

- ⚠ Fixed version might be "too new" - contains breaking changes
- ⚠ API changes, dependency conflicts, behavior differences

Can't Upgrade → Wait for Backport

Patches might get backported to your version, but...

Longer Process

Backports take weeks/months.
Patch must be adapted to
older codebase.

Less Interest

Maintainers focus on latest.
Older branches get lower
priority (if any).

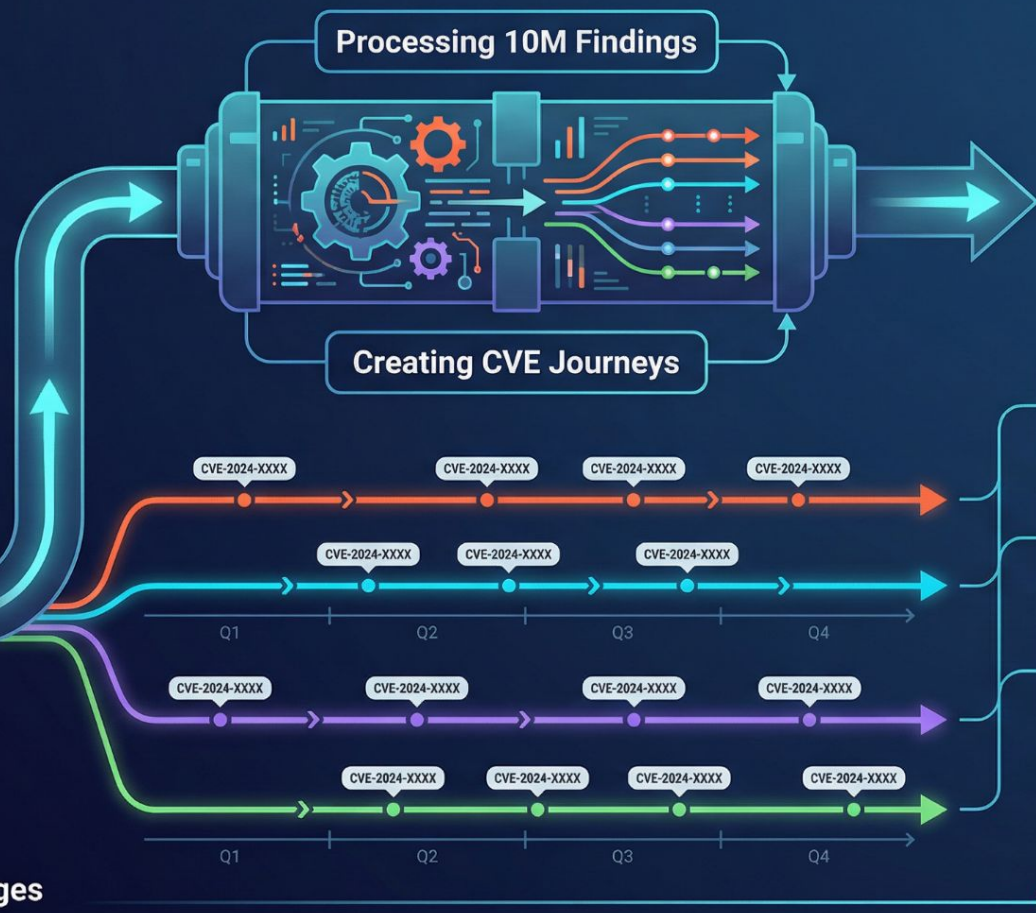
Third-party Dependency

Upstream may not support
your version. Distro maintainers
(Debian, Red Hat) must step in.



365

1 Year Daily Scans
Hundreds of OSS Images



40K
CVE
Journeys

MTTR
Analysis

Fix %

Propagation
Lag



Research Scope

A year of Continuous scanning of the most popular open source container images

250

OSS Container Images
Monitored

7,852

Unique CVEs
Discovered

40K

CVEs Journey

10M

CVEs Findings

Many containers are bloated

The default container of many projects is a largest container they have

56%

of CVEs can be removed
by eliminating bloated packages

Most CVEs come from packages the application never uses:

- Optional dependencies
- Unused OS utilities
- Kernel headers
- Dev tools left in production images.

89% of Remaining CVEs Have a Fix Available

The open source community delivers patches — the problem is propagation, not availability

89%

Have a known fix version

11%

No fix available

of that 11%:

~12% are disputed

Vendors disagree it's
a real vulnerability

Fix availability by severity:



The Patching Gap

CVEs with known fixes persist in containers far longer than they should

36%

of non-bloat CVEs are 180+ days old

~40%

overall take
180+ days to remediate

What You Can Do About it?

Pull Latest Container

78%

of images have fewer CVEs
in their latest tag

The other **22%**

introduce new packages
→ new CVEs

41%

average reduction just by being on the latest

The simplest action: start by trying upgrade to the latest tag

Choose Your Container Flavor

Full Image (Default Tag)

The kitchen sink — everything included

What's Inside

- ✓ Full Operating System
- ✓ Build tools (gcc, make, autoconf)
- ✓ System libraries (libssl, imagemagick, libxml2)
- ✓ Utilities (git, curl, wget, ssh)
- ✓ Package manager (apt)

When to Use

- ▶ Build stages in multi-stage Dockerfiles
- ▶ Compiling native extensions at runtime
- ▶ Quick prototyping & development

Size	~1 GB
Shell	bash
CVEs	1,000 – 2,000
Migration	None

922 - 1,890 CVEs in Python

**NOT recommended
for production**

Minimal Image

The production default — 95% fewer CVEs

What's Inside

- ✓ Minimal base image (essential only)
- ✓ Language runtime
- ✓ bash, basic coreutils
- ✓ package manager
- ✓ SSL certificates, tzdata

What's Removed

- ✗ Build tools (gcc, make)
- ✗ Development headers (-dev)
- ✗ Optional dependencies
- ✗ Common utilities (vim, git, etc.)

Size **~150 MB**

Shell **bash**

CVEs **25 – 70**

Effort **Low**

python:slim 25
node:slim 42

Tip

Use Multi-stage to keep the production container minimal

**RECOMMENDED for
production**

Automate the Tedium

When to use

- Ongoing maintenance of application dependencies (npm, pip, go modules)
- Base image version updates (FROM node:20 → node:22)
- Keeping up with the ecosystem without manual tracking

How it works

- Bot scans your repo for dependency files
- Checks registries for newer versions
- Opens PRs automatically with the update
- Runs your CI tests on the PR

Limitations

- ⚠ PRs still need CI time and (sometimes) human review
- ⚠ Major version updates often need manual intervention
- ⚠ Doesn't fix CVEs in packages you don't directly depend on (transitive deps)

Dependabot
Renovate

Patch Without Rebuilding

When to use

- Emergency patches (1-day drops on Friday)
- Base image CVEs you don't control
- Bridging the gap until upstream rebuilds

How it works

- Reads vulnerability report from Trivy / Gripe
- Fetches updated packages
- Patches the image layer
- Preserves your application code untouched

Limitations

- ⚠ App-level patching is experimental
- ⚠ Scanner might scan layer by layer



github.com/project-copacetic/copacetic

Key Takeaways

1

56% of CVEs are bloat

Minimal base images eliminate the majority of vulnerabilities before they appear

2

89% of remaining CVEs have fixes

The open source maintainers delivers — the gap is in propagation process, not patches

3

~40% of fixable CVEs persist 180+ days

Containers aren't being rebuilt frequently enough to pick up available patches

Feedbacks are welcome

Q & A

Feel free to reach out:

lior@kaplanopensource.co.il
mor.weinberger@echo.ai

