{code & compliance}

29 January 2026

FOSDEM EDITION

CRA-ready: Integrating VEX into Open Source Workflows

Michael Winser, Munawar Hafiz and Piotr P. Karwasz

Alpha-Omega Mission

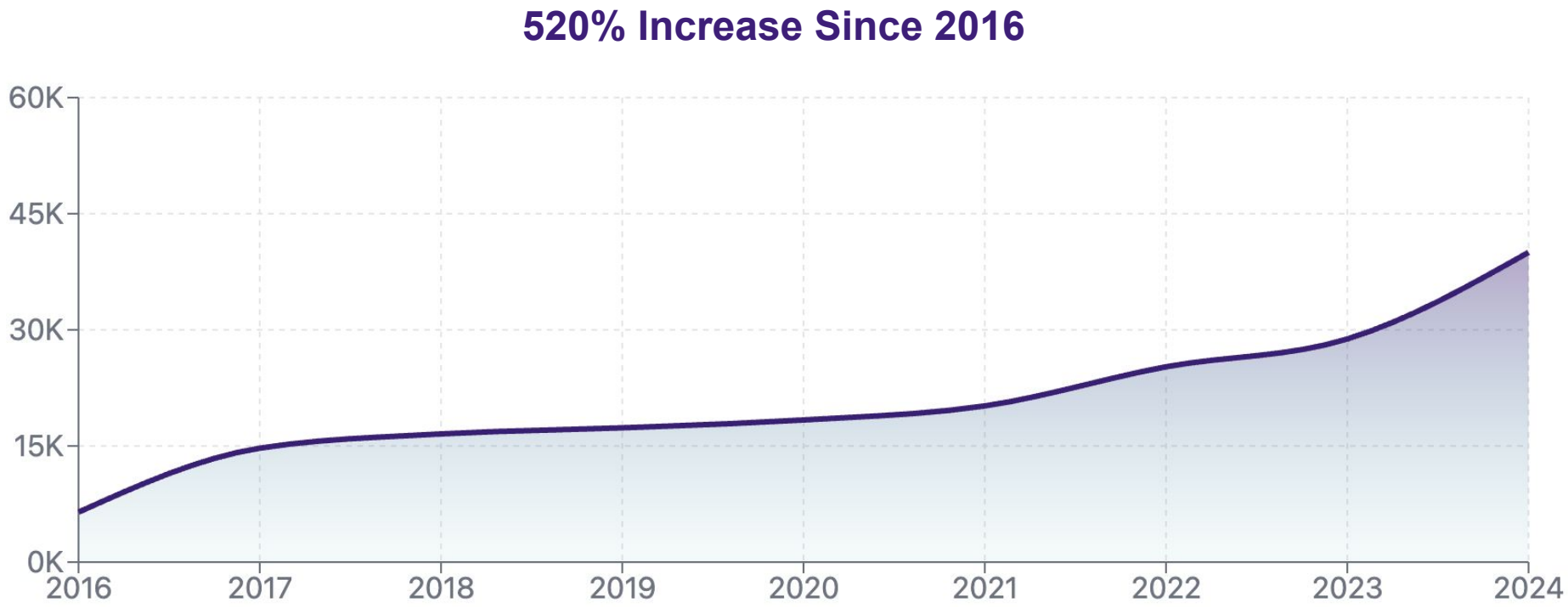
Catalyze sustainable security improvements within the most critical open source projects and ecosystems.

Alpha-Omega Explained

α → Leverage

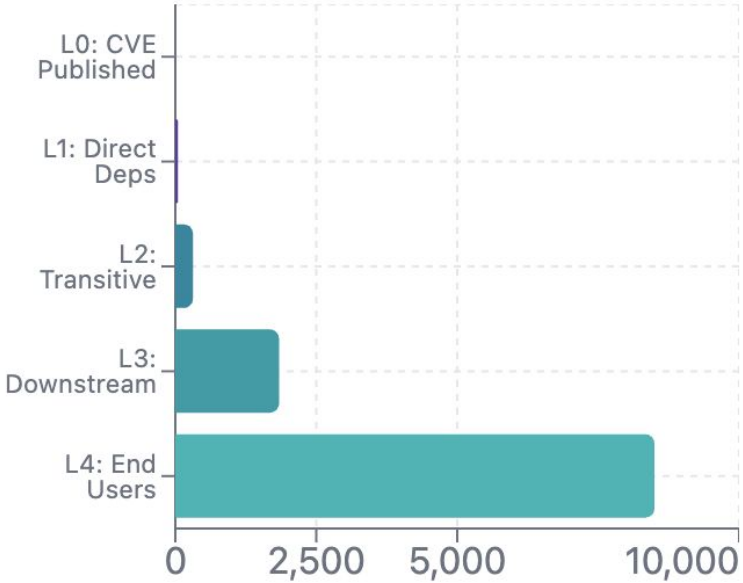
Ω → Scale

Vulnerability Trends

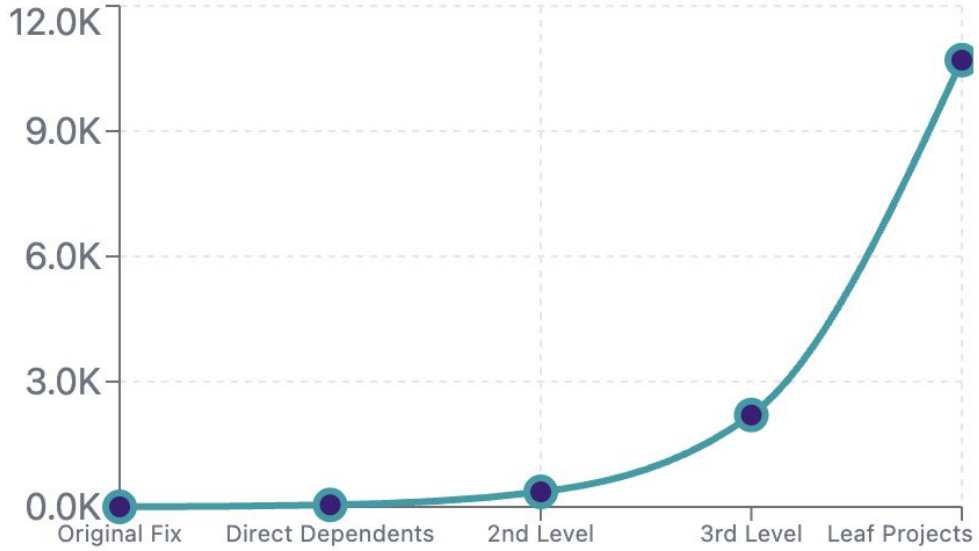


A Geometric Cascade

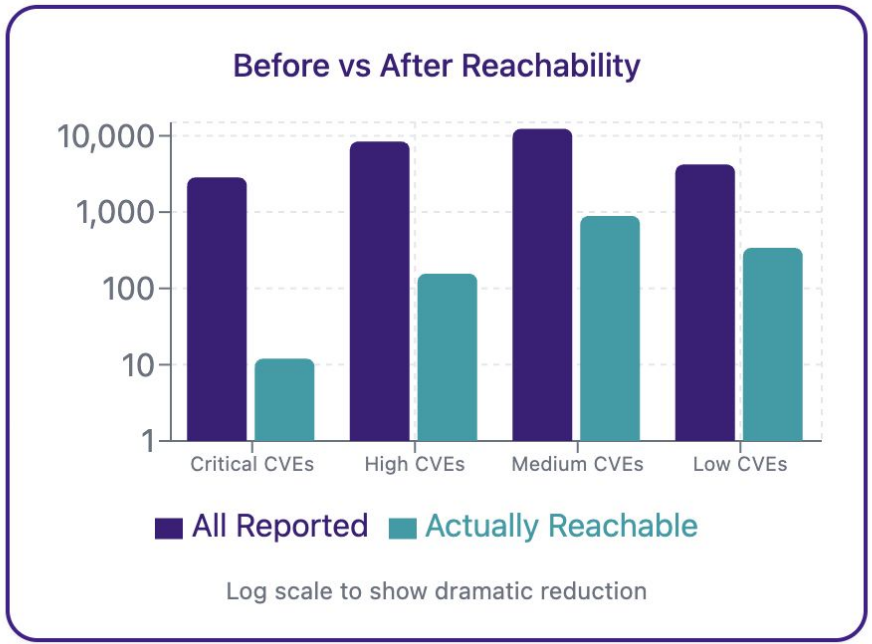
Cascade Effect: Single CVE Fix

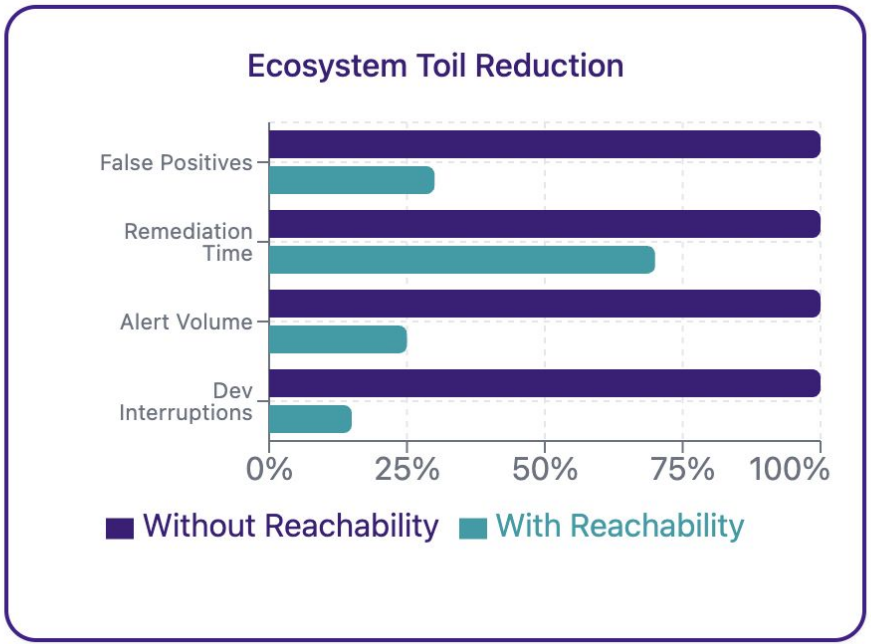


Cumulative Work Required



The Cascade Effect





What is a VEX?

VEX (Vulnerability Exploitability eXchange) is:

- Machine-readable statement about **exploitability**
- Answers: “Is this vulnerability actually exploitable here?“

In use by:

- Microsoft, Red Hat, OpenSUSE, Cisco, ServiceNow, …

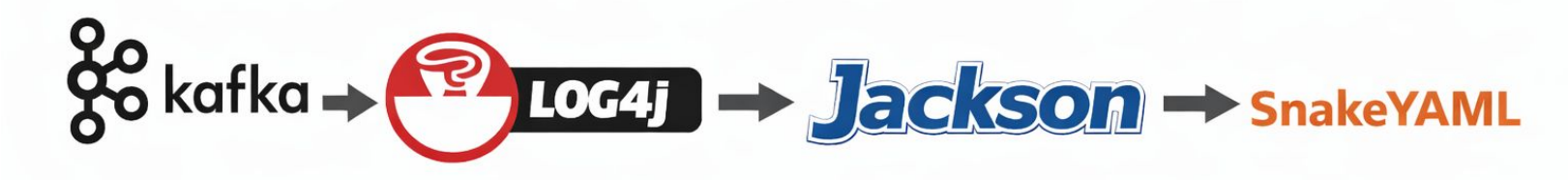
Why it matters:

- Supports CRA requirement:
  *“without known **exploitable** vulnerabilities“*

Day in the life of a security engineer

100+ new CVEs every day.
For each one:

1. Check if the component is present (SBOM)
2. Understand the CVE
3. Trace the dependency path (SBOM?)
4. Assess exploitability at each step



5. Repeat for every application and version

Should OSS Projects Produce VEXes?

Benefits:

- Builds trust in the project
- Saves time for downstream users
- Many downstream users are OSS too

Challenges:

- Not required by regulations
- Consumes scarce volunteer time

At **FOSDEM 2025**, Munawar and Piotr brought this challenge to Michael:

Can we make VEX generation scalable and realistic for OSS?

High-Level Architecture

CVEs

Root Cause Service

- AI agent
- Vulnrichment
- Finds vulnerable method

SBOMs

Call Graph Service

- Per component call graphs
- Capslock-compatible format
- Source-based (shading-resistant)

Metadata GitHub Repository

Project-specific VEX workflows

- Determines maximum reachable path
- Generates enriched VEX statement (VEXplanation)

Maintainers

Repo Org in GitHub

Root cause service: the problem

- Vulnerability disclosure quality varies widely
- Precise vulnerable methods are rarely specified
- References are mostly human-readable text
- Structured data exists (e.g. `programRoutines`), but is underused

GVIP talk

Root cause service: the approach

- Triggered when new CVEs appear (in monitored components)
- AI agent locates:
  - Fixing commit
  - Vulnerable method
- Results stored in a **public GitHub repo** (`callgraph-metadata`)

Results (sample of 140 CVEs):

  - Vulnerable method found: **73%**
  - Correct method identified: **95% accuracy**

Call graph service: why it’s hard

- Polymorphism & dynamic dispatch
- Frameworks have hidden calls
- Shaded Java artifacts

Call graphs are **hard**, but essential.

Call graph service: the solution

- Apache-2.0 license
- Source-based
- Shading-resistant
- Built on Eclipse IDE components
- Lightweight: most Solr dependencies analyzed on a GH runner

Trade-offs

- Git metadata often missing in POM
- Git-based storage doesn’t scale long-term

Real-World OSS Contraints (Apache Solr)

- ~400 dependencies
- Tens of thousands of users
- Each dependency upgrade is risky
- Fully volunteer-driven

Before

- Manual VEX for high-severity CVEs
- Limited action on **transitive** dependencies

What Changes with VEX Automation

The project is evaluating workflows that:

- Create a PR per CVE
- Work across **multiple** versions
- **Objective** data for exploitability vs upgrade risk
- Better answers to user security questions

In parallel:

- Human-friendly HTML security pages from VEX data

Where this is going

The VEX Generation Toolset has shown a need to:

- Improve VEX structure and standards with more structured data
- Easier VEX exchange (Transparency Exchange API)
- Automate interpretation of **exploitability** from **reachability**, by sharing statements with upstream and downstream
- Reduce the pain of evaluating **transitive** dependency upgrades

SBOMs took years to mature: VEX will too.

Thank you!

VEX Generation Toolset: https://github.com/vex-generation-toolset

Michael Winser

Munawar Hafiz

Piotr P. Karwasz

Alpha-Omega

Open Refactory

THE ASF