Beyond SBOM: Integrating VEX into Open Source Workflows

Michael Winser
Munawar Hafiz
Piotr P. Karwasz

Alpha-Omega Mission

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

FOSDEM

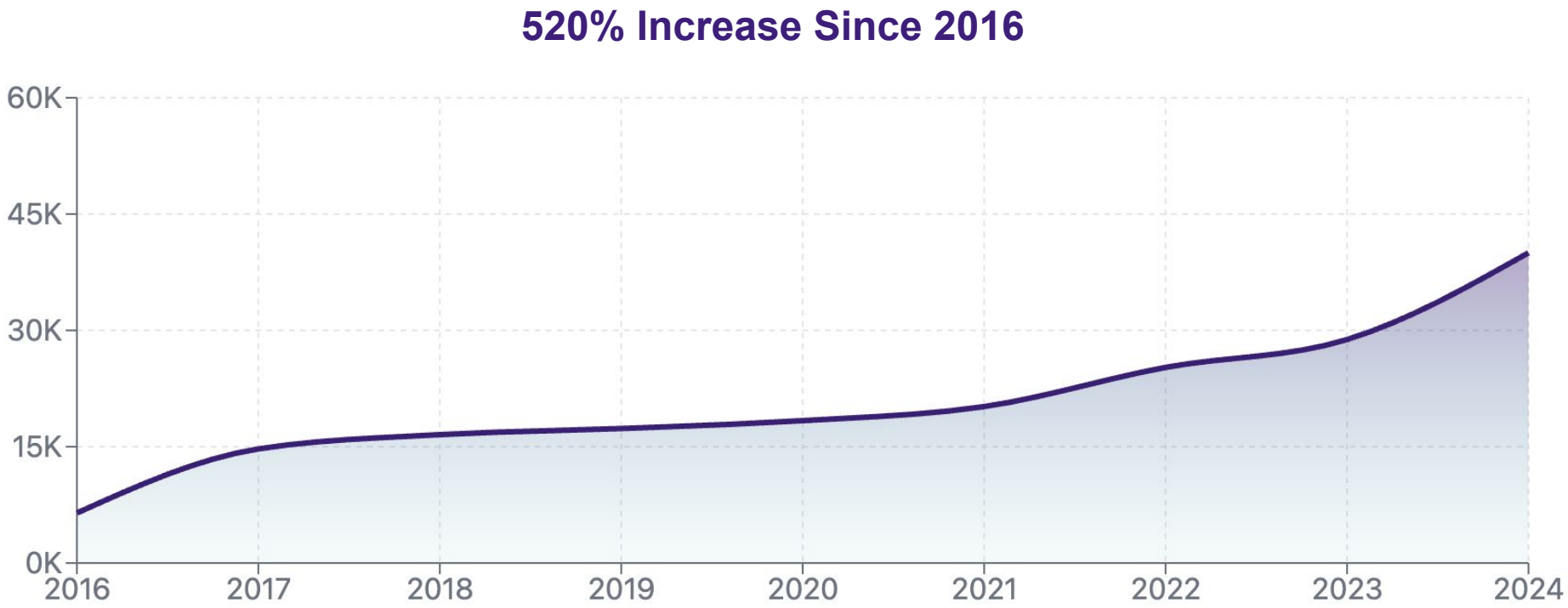
Alpha-Omega Explained

α → Leverage

Ω → Scale

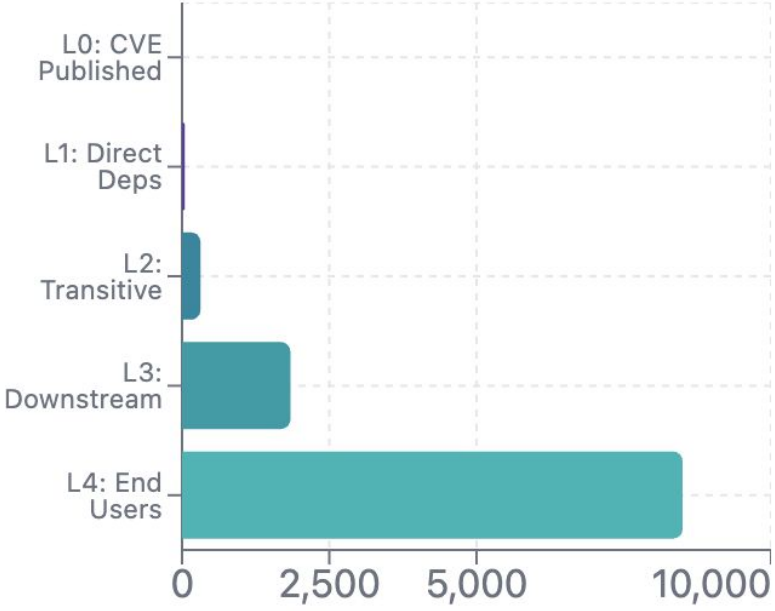
FOSDEM

Vulnerability Trends

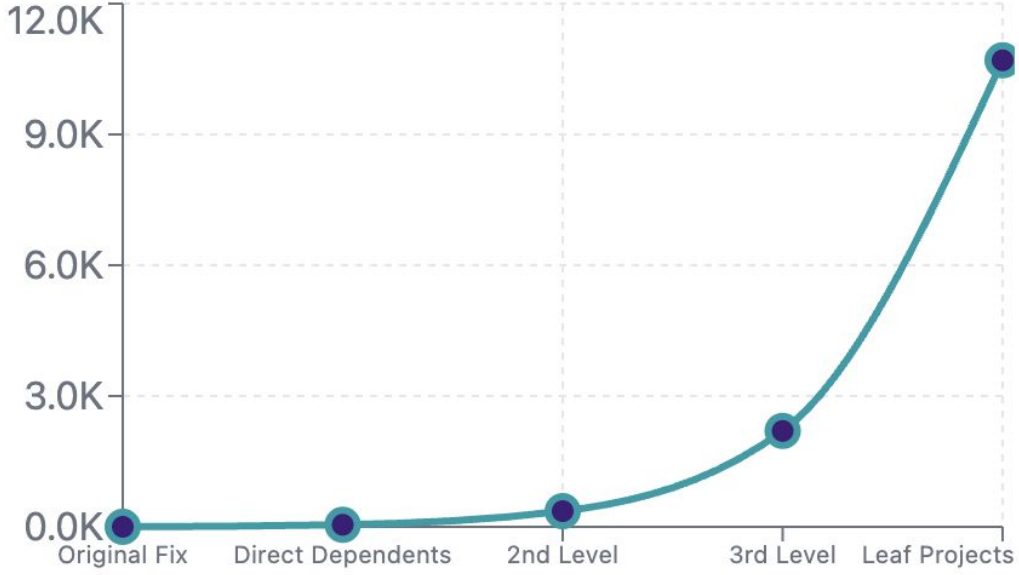


A Geometric Cascade

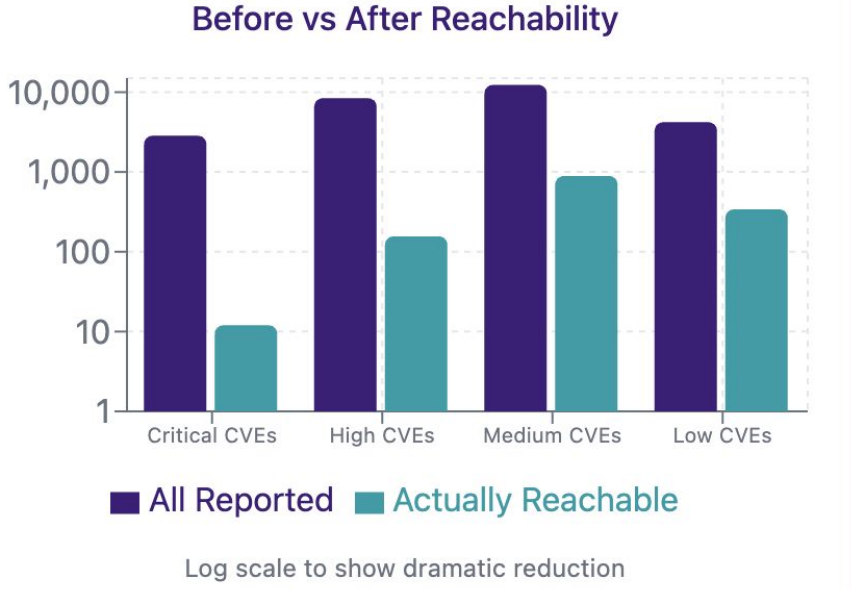
Cascade Effect: Single CVE Fix

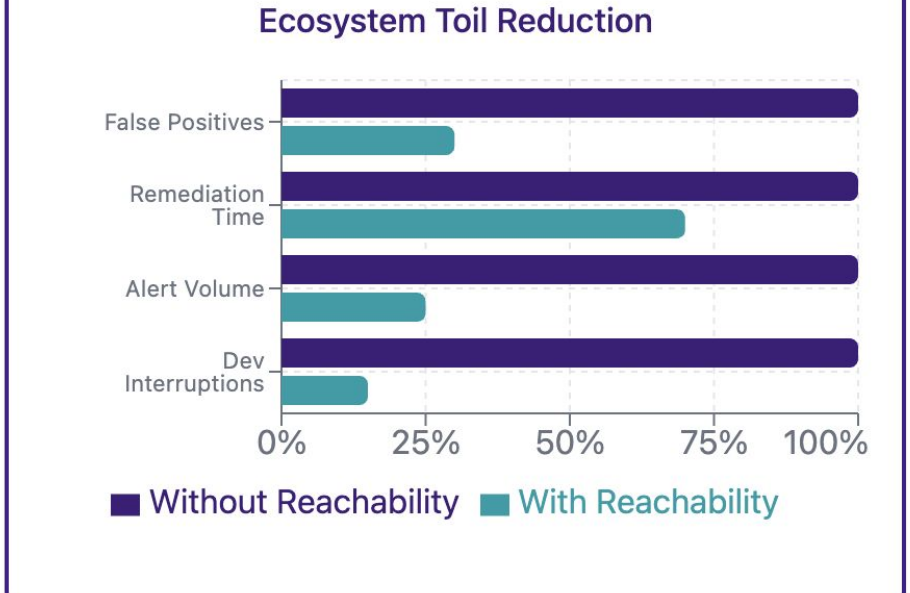


Cumulative Work Required



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:

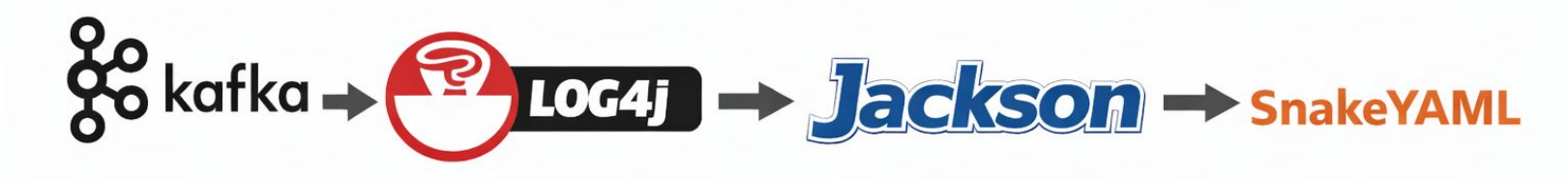
- Supports CRA requirement:
  *“without known **exploitable** vulnerabilities“*

FOSDEM

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?

The Cost Of Producing VEXes: Organizations

700,000 upgrade decisions per year

7M hours (10 hours for each VEX)

3,365 Person Year (2080 work hour/year)

$400M Per Year

The Cost Of Producing VEXes: Maintainers

Cost Of Generating VEX Documents

460 Dependencies

300 CVEs (approx. 1 per artifact)

3000 Hours Of Effort (10 hours per VEX)

1.5 Person Years (2080 work hour/year)

Consequences

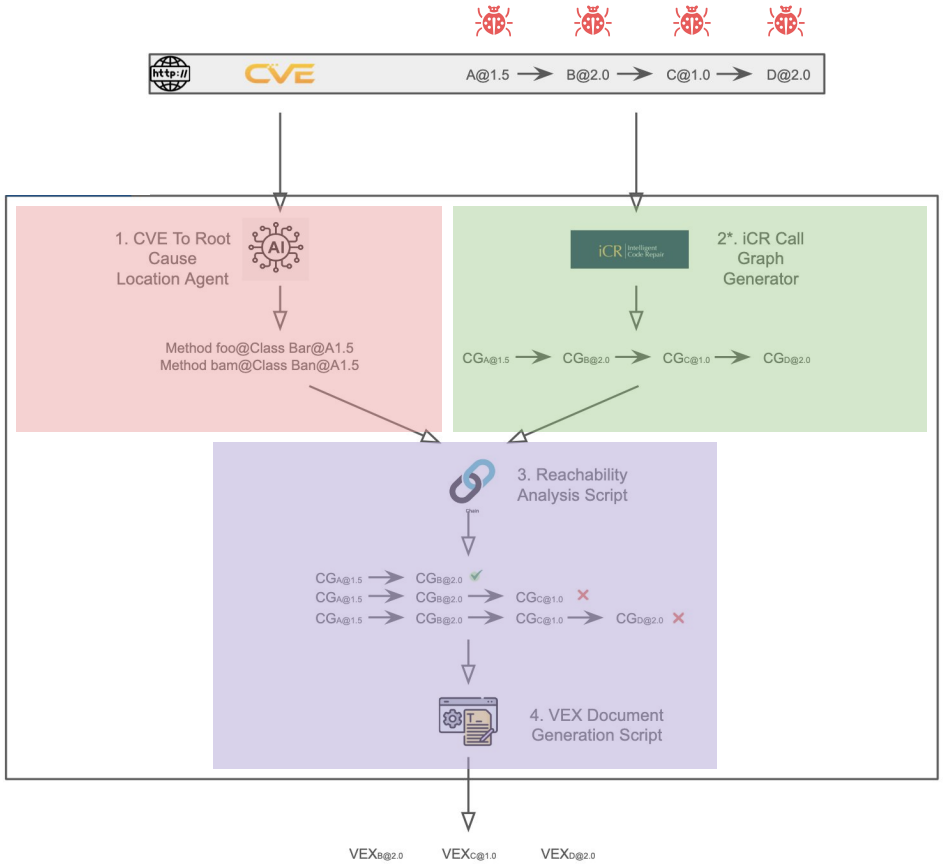
Really Low Adoption Of VEX Documents

97% Noise in SCA Reports

It Takes Months To Fix Vulnerabilities

FOSDEM

How Do We Generate VEX?



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

FOSDEM

How Does VEX Helps Maintainers?

CVE-2025-30065

Published on Apr 1, 2025

CVSS Score: 10

April 15, 2025

> I spent last week dealing with that Parquet CVE and I now know how trivial it is to instantiate any class with a string constructor from Parquet < 15.1 and from Avro < 1.11.4. You just declare it as the class to generate when iterating through an avro file/schema and then have the target app iterate through the code

> We've fixed trunk by moving to shaded avro 1.11.4, but the forthcoming 3.4.x release is on avro 1.9.2. Which is exposed. We use it internally, and expose some classes which others may use.
>
> We have not upgraded branch-3.4 because it appears to violate our compatibility rules.

VEX Document Generation For Apache Hadoop

CVE-2025-30065

Published on Apr 1, 2025

Apache Hadoop discussion on Apr 15, 2025

We started working on April 16, 2025

VEX evidence generated on April 18, 2025

Not Reachable

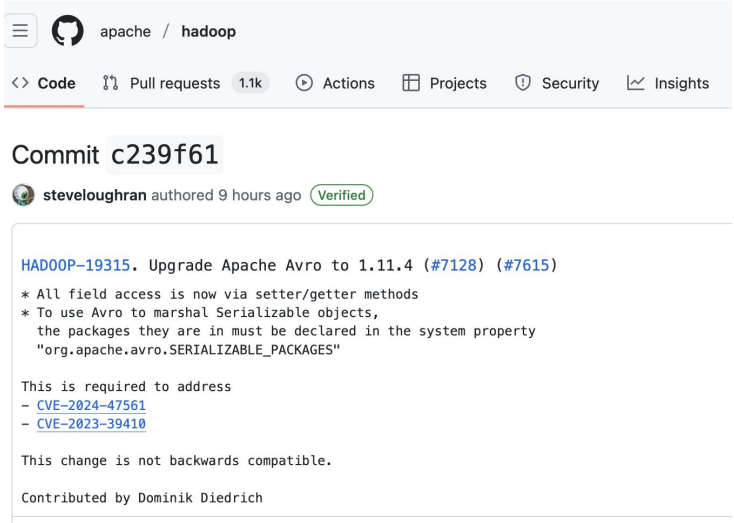
What Did The Maintainers Do?

April 21, 2025

Even if we aren't vulnerable there, by requiring avro 1.9 in some services, we may be forcing it on others, or at least getting into the hadoop common classpath, so making it hard for applications to upgrade -they may use the method.

I've upgraded branch-3.4; tagged in the release notes as incompatible.

Regarding the cve analysis, is it possible to run it against all the asf hadoop repos? or a set of private ones?



Generating VEX As A Part Of CI/CD Pipeline

Integrate VEX :

- 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

Demo Video
On YouTube

What Changes With VEX Automation?

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.

Link To This Slide Deck

Shoutout: What Else Do You Know Using Call Graphs?

Capslock analyzes your code to show/control privileged operations your dependencies can access: file I/O, network calls, code execution, etc.

Malicious code from a 2022 supply chain attack

```
CAPABILITY_EXEC: 1 callsons:commons-compress:1.24.0
CAPABILITY_FILES: 1 calls
CAPABILITY_MODIFY_SYSTEM_STATE: 2 calls
CAPABILITY_NETWORK: 1 callsle system operations
CAPABILITY_OPERATING_SYSTEM: 1 callstream.<init>
CAPABILITY_READ_SYSTEM_STATE: 1 calls
    • CAPABILITY_REFLECT - Reflection and dynamic code loading
      └─ ZstdUtils → Class.forName
```