

Fitness for Purpose

Comparing and Contrasting the CVE List with OSV.dev

Andrew Pollock

I'm back!

- FYI I just finished up at Google
 - Thank you OpenSSF 🙏
- Expanding on last year's presentation

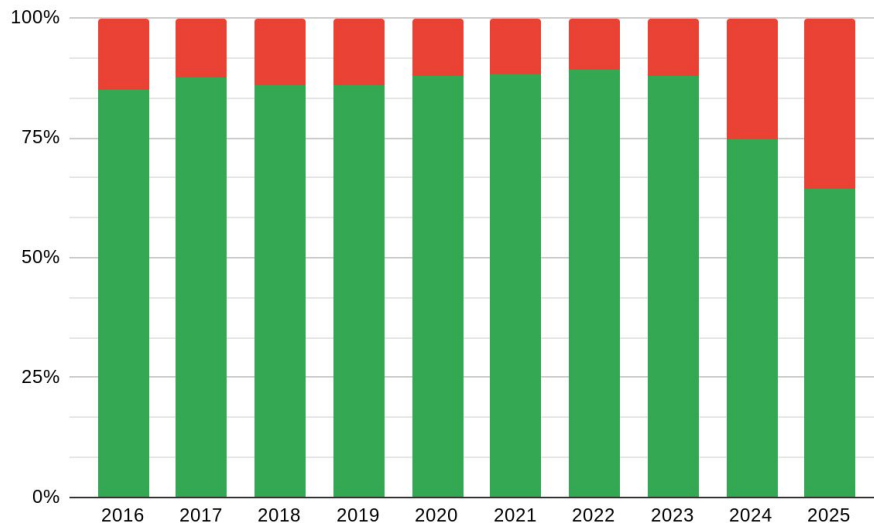


My work in a nutshell

1. Take CVEs from the NVD
2. In conjunction with the CPE Dictionary
 - a. Try to figure out the Git repository for the CPE(s) involved
 - b. Try and map the versions involved to Git commits
 - c. Generate an OSV record
3. Profit!
 - a. Enable commit-level scanning for source-code vulnerabilities (in C/C++ in particular)

Read more at [OSV.dev/blog](https://osv.dev/blog):

- <https://osv.dev/blog/posts/introducing-broad-c-c++-support/>
- <https://osv.dev/blog/posts/using-the-determineversion-api/>



Last year's call to action

- **CNAs**
 - Think about the CVEs you're authoring and their fitness for purpose, in aggregate
- **CVE Program**
 - Make it easy for CNAs to do the right thing, and harder for them to do the wrong things

Fitness for purpose

- Why do we have vulnerability metadata?
 - Detection
 - Remediation

Compare and Contrast

OSV in a nutshell

- Precise identification of vulnerabilities in open source software
 - To enable prioritized vulnerability remediation
- It all began in 2021 with OSV 1.0
 - CVE 4.0 (and CVE 3.3) couldn't express findings
 - CVE allocation + manual analysis
 - Automated record creation + submission was... tedious
- OSV schema donated to the OpenSSF
 - github.com/ossf/osv-schema
- OSV.dev, OSV-Scanner and OSV-SCALIBR are Google-sponsored infrastructure and open source projects
 - github.com/google/osv.dev
 - github.com/google/osv-scanner
 - github.com/google/osv-scalibr

Languages

- C/C++
- CRAN (R)
- Crates.io (Rust)
- Go
- Hackage (Haskell)
- Hex (Erlang)
- Maven (Java)
- npm (JavaScript)
- NuGet (.NET)
- Packagist (PHP)
- Pub (Dart and Flutter)
- PyPI (Python)

Distributions

- AlmaLinux
- Alpine
- Android
- Bitnami
- Chainguard/Wolfi
- Debian
- Mageia
- openSUSE/SUSE
- RHEL
- Rocky Linux
- Ubuntu

Mission statements



Identify, define, and catalog publicly disclosed cybersecurity vulnerabilities

- No specific purpose
- No specific target audience
- Broad scope



Enable developers to reduce security risk arising from known vulnerabilities in open source components they use

- Specific purpose
- Specific target audience
- Narrower scope

Scale: records



- 2024
 - 37,381
 - 307 distinct CNAs



- 2024
 - 82,968
 - 9,816 directly alias a CVE
 - 58,419 cross-reference with a CVE
 - 21 distinct home databases

Scale: record publishers



CNAs

- 447
 - 125 open source



Home databases

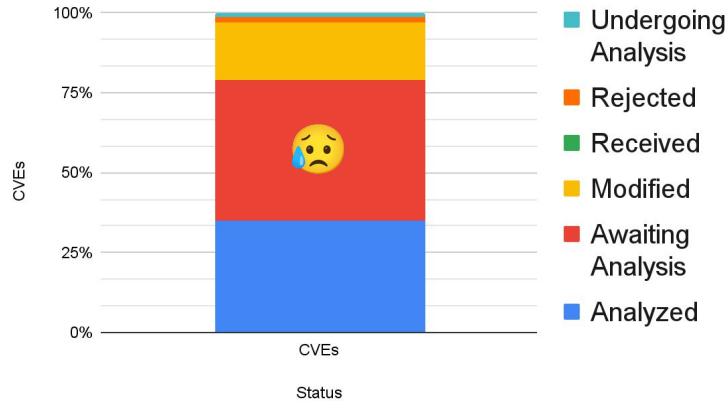
- 22

Let's talk about scale

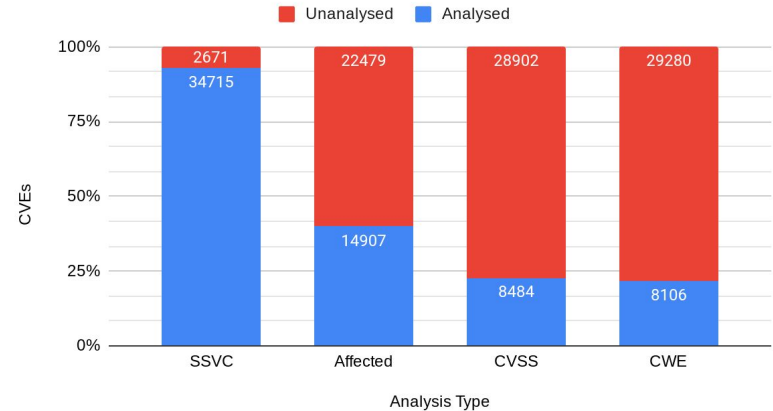
Problem: we're drowning in vulnerabilities

- Hard for vendors
- Hard for defenders
- Hard for analysts

2024 NVD CVEs by Status



2024 Vulnrichment CVE Analysis Performed



Solution: send in the machines!

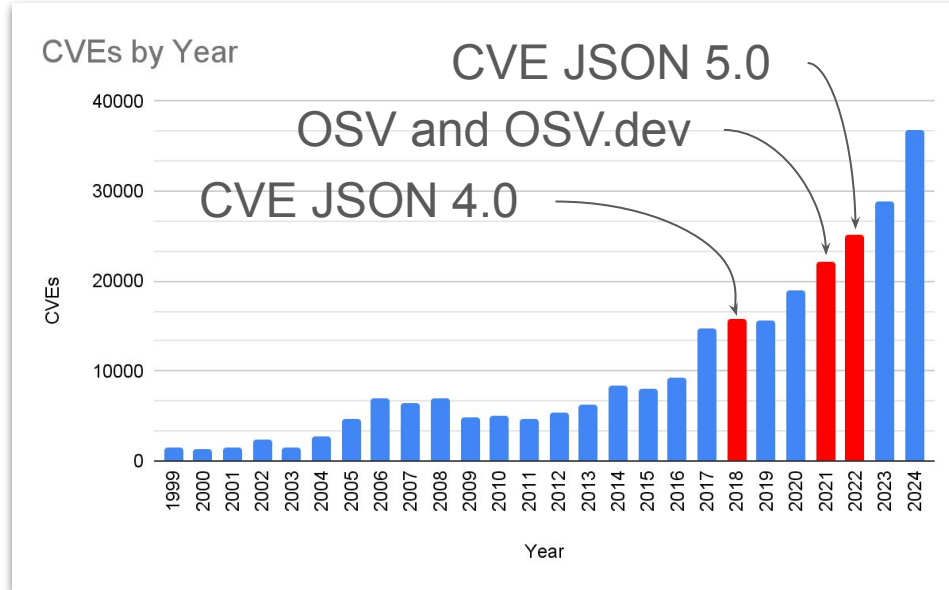
- Machine readable records enable



Programmatic analysis



Programmatic detection



CVE JSON 5.0

“CVE Record creation and publication **could** now be automated, meaning more quality CVE Records **could** be produced at a faster pace.”

– *CVE Program 25th Anniversary Report*

But...

- 15% (5,665) CVEs from 2024 have no usable `.affected` field
- Plenty more with
 - invalid SemVer versions
 - other invalid version strings
 - other ambiguities
- **Inconsistent usage of CVE 5.x undermines this potential**


```
{  
  "vendor": "n/a",  
  "product": "n/a",  
  "versions": [  
    {  
      "version": "n/a",  
      "status": "affected"  
    }  
  ]  
}
```

Fitness for purpose

- Why do we have vulnerability metadata?

- **Detection** (at scale)
- Remediation (at scale)

Identify, define, and catalog publicly disclosed cybersecurity vulnerabilities



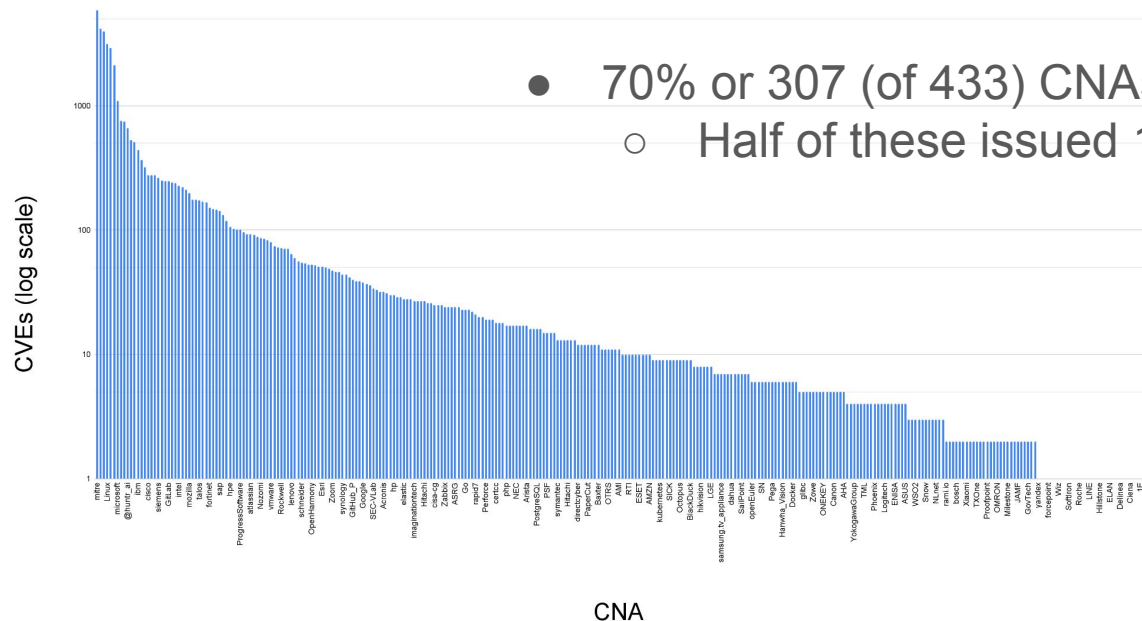
Must haves

- Automation friendly
- Accurate

Federation without consistency is pointless

An ever growing number of unique snowflakes

CVEs by CNA for 2024



- 70% or 307 (of 433) CNAs issued CVEs in 2024
- Half of these issued 10 CVEs or less

Solutions?



- CISA Secure by Design Pledge
 - Voluntary
 - Up to a 1 year lag
 - Include CPE and CWE
 - 16% of CVEs (5,997) from 2024
- CNA Enrichment Recognition List (ERL)
- **An attempt at upleveling quality**



- Import-time validations
 - JSON Schema
 - Properties of a High Quality OSV Record
- **Prevent backsliding in quality**

Properties of a High Quality OSV Record

Properties of a High Quality OSV Record

Valid

- Passes JSON Schema validation

Precise

- version and commit ranges
 - Have an **introduced** version (and it exists)
 - Prefer a **fixed** version over **last_affected** (and it exists) and post-dates the **introduced** version
 - Distinct
- Package ecosystem and purl (if present) is valid
- Package exists in ecosystem
- References work at publication time

Identifiable

- Links back to a CVE where applicable

But what about the CVEs?!

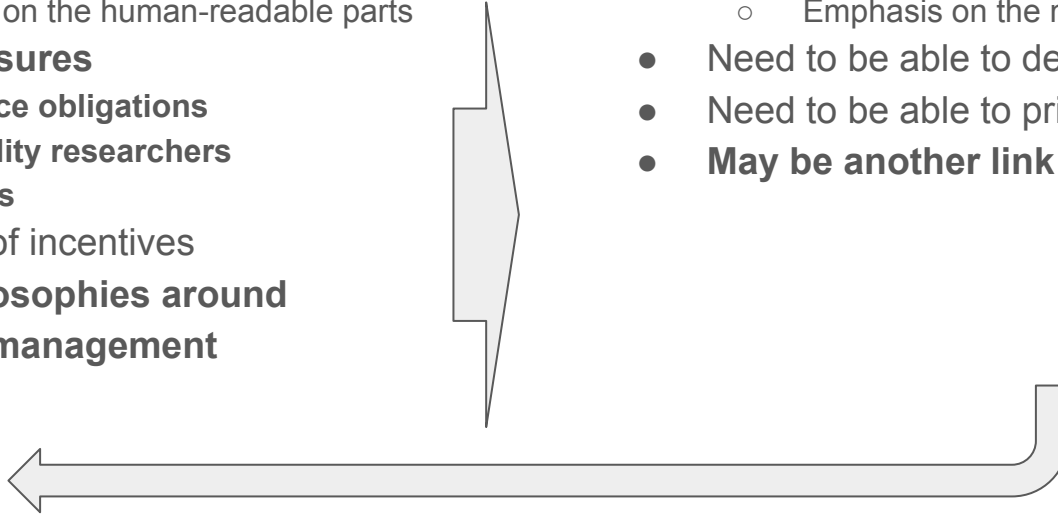
But what about the CVEs?!

CNA (Vendor)

- Individually
 - Emphasis on the human-readable parts
- **External pressures**
 - **compliance obligations**
 - **vulnerability researchers**
 - **customers**
- Misalignment of incentives
- **Differing philosophies around vulnerability management**

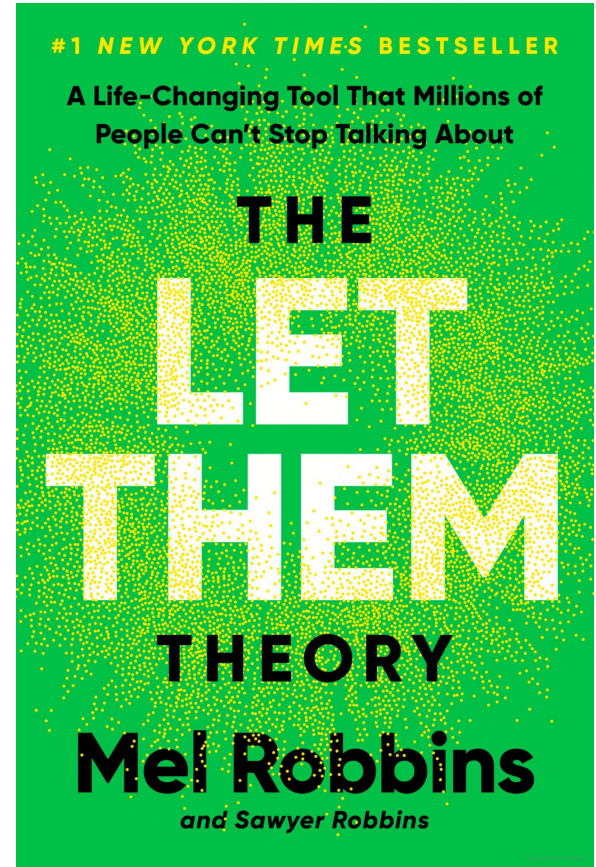
Downstream consumer (customer)

- In aggregate
 - Emphasis on the machine-readable parts
- Need to be able to determine applicability
- Need to be able to prioritise
- **May be another link in the chain**



But what about the CVEs?!

- External pressures
 - compliance obligations
 - vulnerability researchers
 - customer contractual obligations and evolved expectations



But what about the CVEs?!

- Differing philosophies around vulnerability management
 - “Just upgrade!”



Chromium Docs

[Home](#) [Sitemap](#) [Getting Started](#) [Testing](#)
[Design Docs](#) [Contact](#) [Bugs](#) [Style Guide](#)
[Markdown Syntax](#) [Old Docs](#) [Search](#)

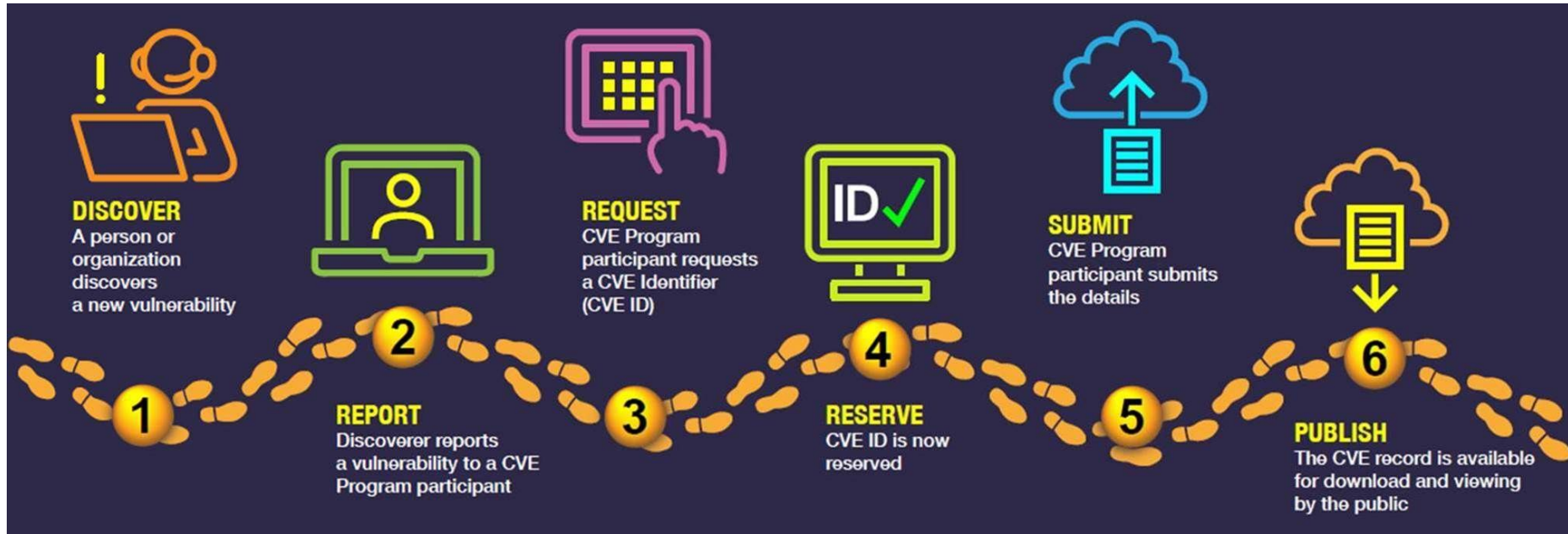
Chrome Security Update FAQ

Bookmark this page as <https://g.co/chrome/security-update-faq>

TL:DR

Almost all Chrome updates contain security fixes, and should be prioritized equally. The most secure option is to automatically update Chrome as soon as any update is available, independent of the specific details of any security fixes included in the update.

But what about the CVEs?!



Personas in the vulnerability lifecycle

Finder



Faith

CNA



Cathy

Vulnerability
Subject
Owner



Veronica

Vulnerability
Impacted
User



Lucky

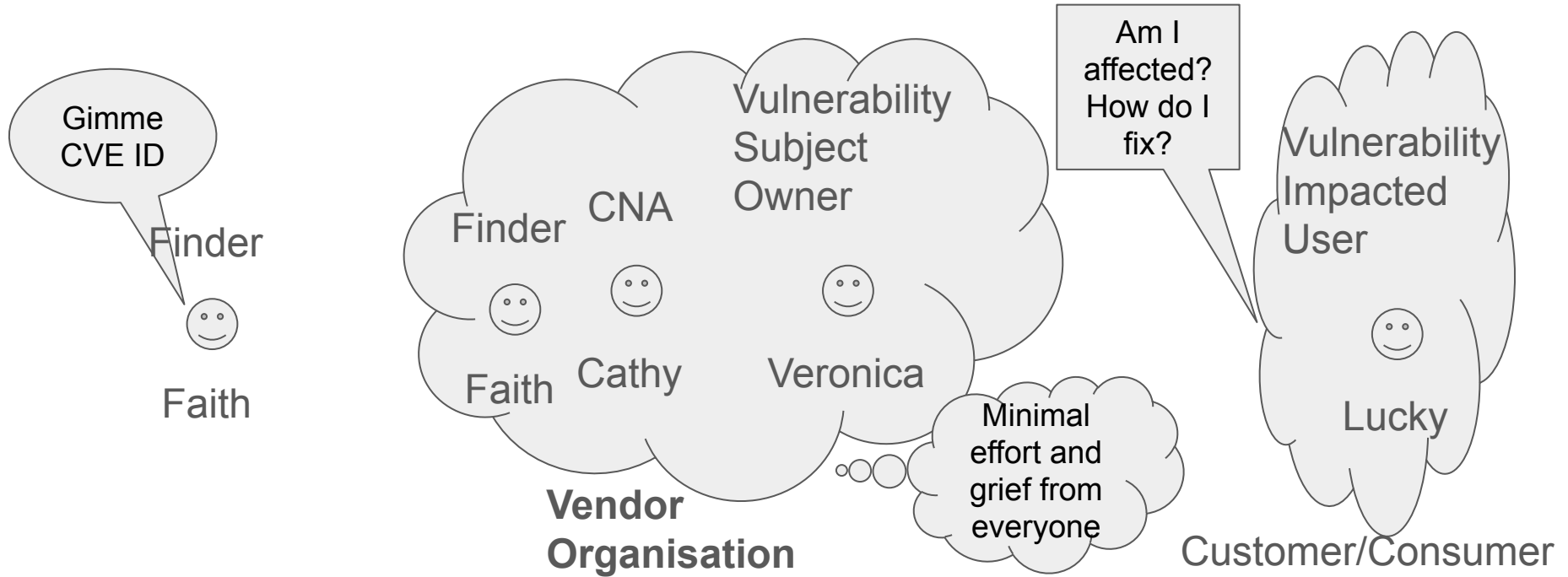
CNA types

- Vendor (236)
- Open Source (85)
- Researcher (60)
- CERT (15)
- Hosted Service (14)
- Bug Bounty Provider (7)
- Consortium (1)

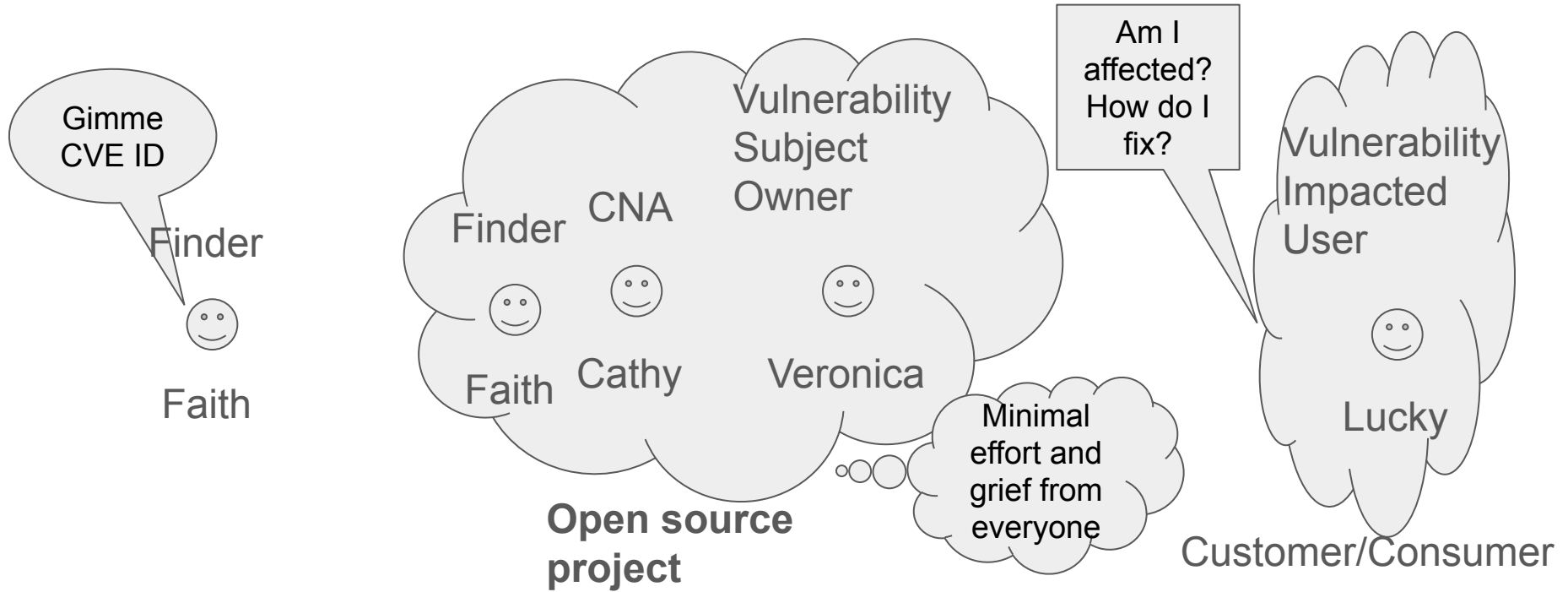
- Vendor (236)
- Open Source (85)
- Researcher (60)
- Bug Bounty Provider (7)

Contributed 29,870 (~80%) of the CVEs in 2024

Personas: Vendor CNAs



Personas: Open Source CNAs



Personas: Open Source CNAs

4.1.12 The act of updating Product dependencies MUST NOT be determined to be a Vulnerability, regardless of whether the dependencies have Vulnerabilities. For example, updating a library to address a Vulnerability in that library MUST NOT be determined to be a new Vulnerability in a Product that uses the library, and a Vulnerability advisory for the Product SHOULD reference the CVE ID for the Vulnerability in the library. See 4.2.13.

4.2.13 If multiple Products are affected by the same Independently Fixable Vulnerability, then the CNA:

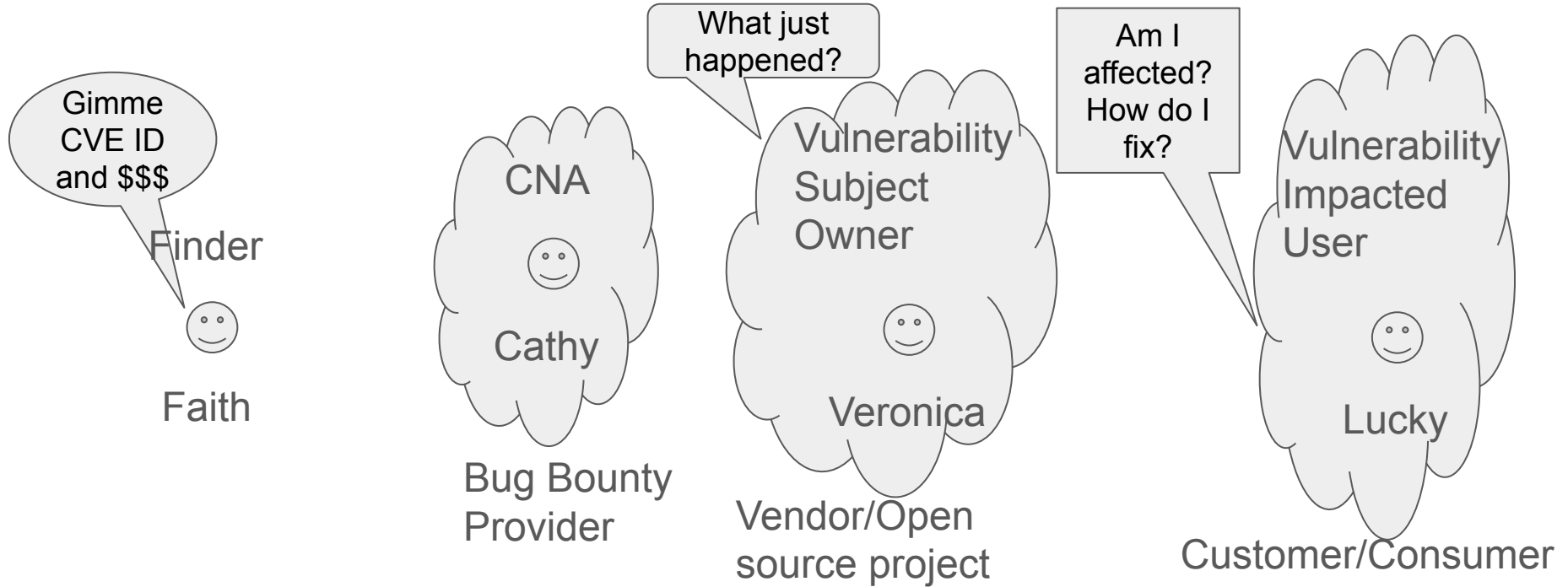
1. MUST NOT assign more than one CVE ID if the Products are vulnerable because they share the vulnerable code. The assigned CVE ID will be shared by the vulnerable Products.
2. SHOULD assign different CVE IDs if the Products do not share vulnerable code.
3. SHOULD assign different CVE IDs if the CNA is uncertain whether the Products share vulnerable code.

4.2.14 If a Product is affected by a Vulnerability because it uses the functionality or specification of another Product, then a CNA:

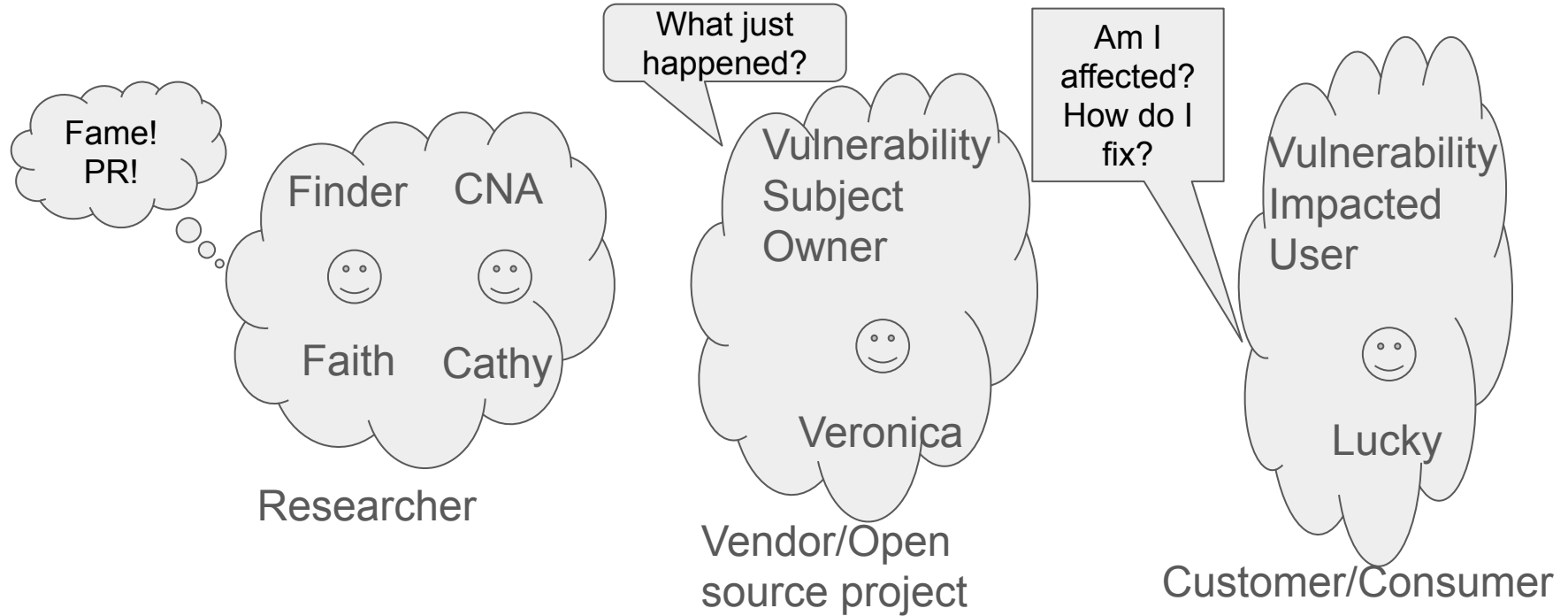
1. MUST assign a CVE ID to each known vulnerable implementation if there is a secure way of using the functionality or specification.
2. MUST assign a single CVE ID if there is no option to use the functionality or specification in a secure way.
3. SHOULD assign different CVE IDs to each known vulnerable implementation if the CNA is uncertain whether there is a secure way.

4.2.15 CNAs MUST NOT assign a different CVE ID to a Vulnerability that is fully interdependent with another Vulnerability. The Vulnerabilities are effectively the same single Vulnerability and MUST use one CVE ID.

Personas: Bug Bounty Provider CNAs



Personas: Researcher CNAs



“unless covered by the
scope of another
CNA”

Possible steps forward

- Mission statement
 - Formalise the interpretation for the next 25 years
- Incrementally raise the minimum standard
 - In the JSON Schema
 - Start new CNAs at the desired standard
 - Bring existing CNAs up to this standard
- Work with the top 15 CNAs to model the desired behaviour
 - Covers ~75% of CVEs from 2024
 - **Especially the CNA-LRs**

Incrementally raise the minimum standard

Use cases

1. Automatable Detection
 - Affected product information
 - CPEs
 - Purls
2. Remediation Prioritisation
 - CVSS
3. Secondary Prioritisation and Retrospective Analysis
 - CWE

Strong leadership necessary

- CISA
 - As the funding body
- CVE Board
 - As the program governing body
- CNAs and CNA-LRs
 - Require enough information from vulnerability researchers and requesters
- Downstream consumers
 - Demand better from your vendor CNAs
- Vulnerability researchers
 - Demand better from your CVE issuers
 - Help them to do better

Let's see how quickly we can do this