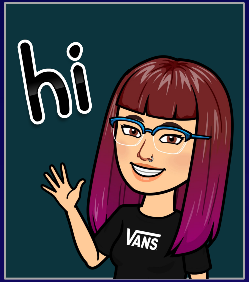
OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Crossing the Streams:
Helping downstream understand how upstream manages vulns

VulnCon 2024

OpenSSF
OPEN SOURCE SECURITY FOUNDATION



Madison Oliver, @taladrane

Pronunciation: TALA-drane

Security transparency & disclosure advocate

Cat fancier & WoW enthusiast

CRob, n, adj, and v

Pronunciation: U.S. (K-robe)

42nd level Dungeon Master

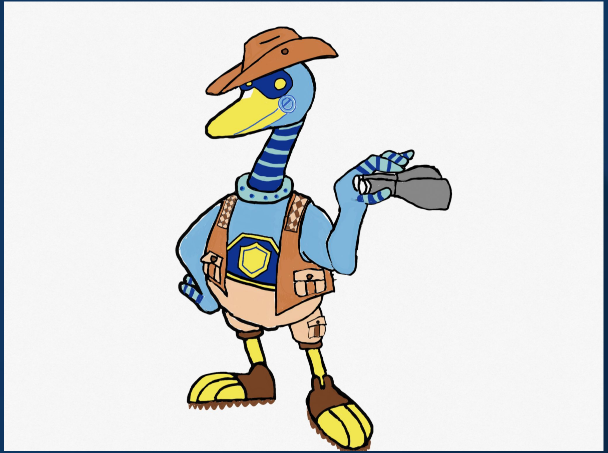
25th level Securityologist

Security Lorax, Cat-herder

Pirate-enthusiast & hat-owner

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Agenda



01 - Why CVD is important?

02 - What makes CVD hard?

03 - Who is helping

04 - How upstream does cvd/
how downstream can get integrated

05 - How can YOU help?

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Why CVD is important?

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Coordinated Vulnerability Disclosure (CVD)

The process of gathering information from vulnerability finders, **coordinating** the sharing of that information between relevant stakeholders, and disclosing the existence of software vulnerabilities and their mitigations to various stakeholders, including the public.

Principles of CVD

- Reduce Harm
- Presume Benevolence
- Avoid Surprise
- Incentivize Desired Behavior
- Ethical Considerations
- Process Improvement

https://resources.sei.cmu.edu/asset_files/SpecialReport/2017_003_001_503340.pdf

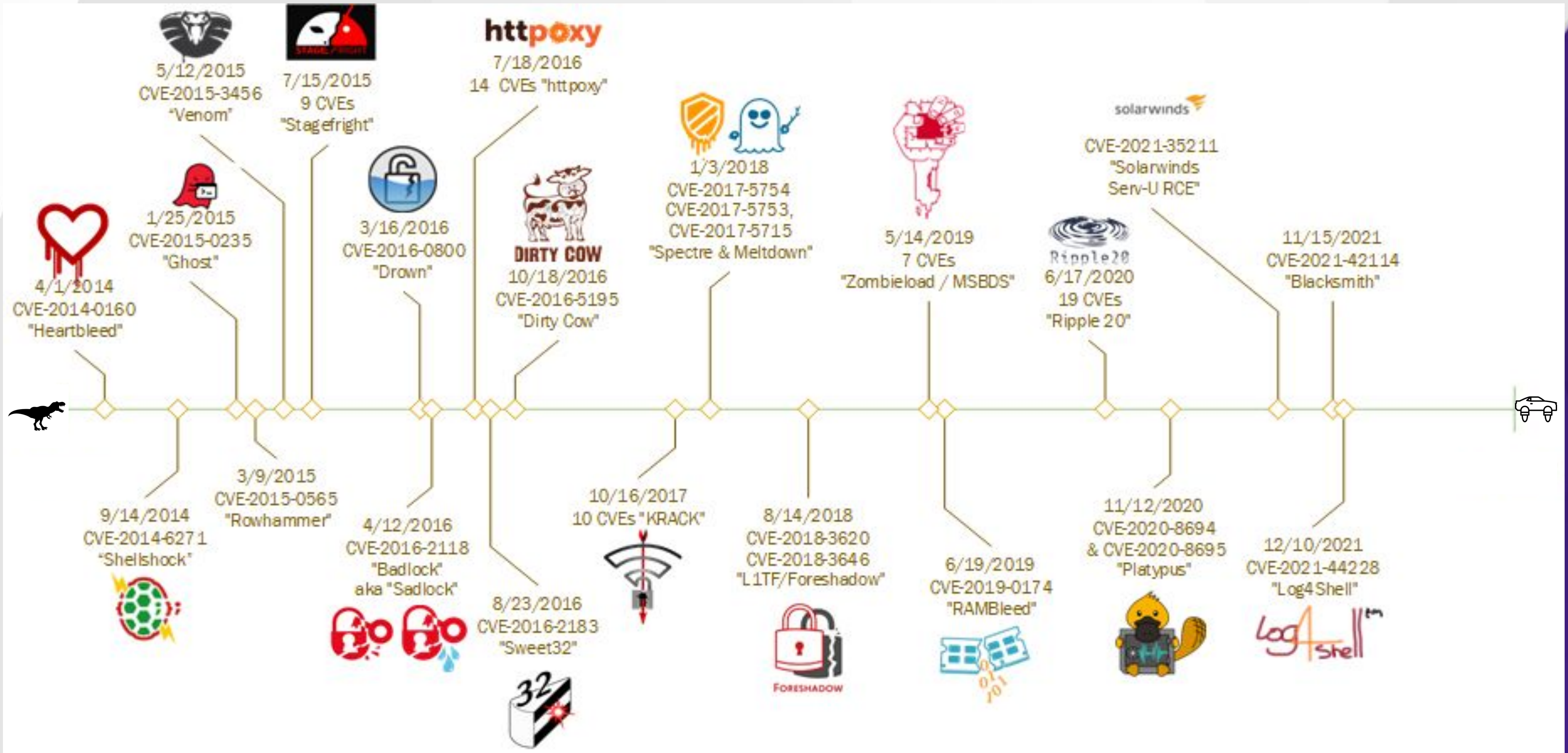
Why CVD?

CVD helps ensure that software maintainers have access to the resources they need to analyze, test, and fix a reported vulnerability.

As fixes are developed, authorized trusted parties can assist in testing and staging the patches for **public disclosure** (PD).

At PD, notifications go out to the public and impacted downstream consumers. **Everyone has access to the fixes at the same time so that no one group is put at risk more than others**. Conversely, no one has pre-access to the bits (so no preference or priority); ***everyone has equal access***.

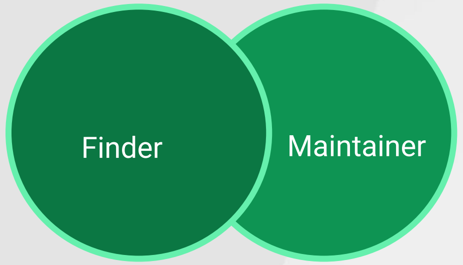
A Tour of Celebrity/Branded Vulnerabilities 2014 - present


4/1/2014 CVE-2014-0160 "Heartbleed"
9/14/2014 CVE-2014-6271 "Shellshock"
1/25/2015 CVE-2015-0235 "Ghost"
3/9/2015 CVE-2015-0565 "Rowhammer"
5/12/2015 CVE-2015-3456 "Venom"
7/15/2015 9 CVEs "Stagefright"
STAGEFRIGHT
3/16/2016 CVE-2016-0800 "Drown"
4/12/2016 CVE-2016-2118 "Badlock" aka "Sadlock"
httpoxy
7/18/2016 14 CVEs "httpoxy"
8/23/2016 CVE-2016-2183 "Sweet32"
32
DIRTY COW
10/18/2016 CVE-2016-5195 "Dirty Cow"
10/16/2017 10 CVEs "KRACK"
1/3/2018 CVE-2017-5754 CVE-2017-5753, CVE-2017-5715 "Spectre & Meltdown"
8/14/2018 CVE-2018-3620 CVE-2018-3646 "L1TF/Foreshadow"
FORESHADOW
5/14/2019 7 CVEs "Zombieload / MSBDS"
6/19/2019 CVE-2019-0174 "RAMBleed"
Ripple20
6/17/2020 19 CVEs "Ripple 20"
11/12/2020 CVE-2020-8694 & CVE-2020-8695 "Platypus"
solarwinds
CVE-2021-35211 "Solarwinds Serv-U RCE"
11/15/2021 CVE-2021-42114 "Blacksmith"
12/10/2021 CVE-2021-44228 "Log4Shell"
log4shell™

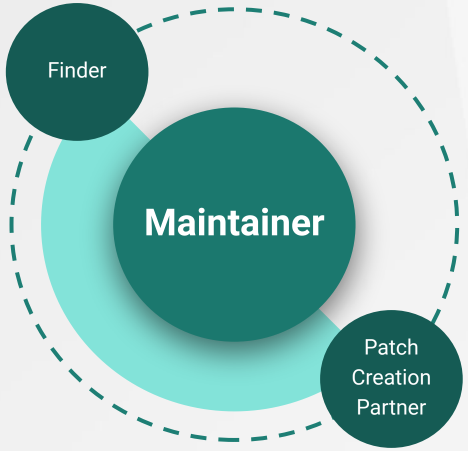
OpenSSF
OPEN SOURCE SECURITY FOUNDATION

VD can take MANY forms

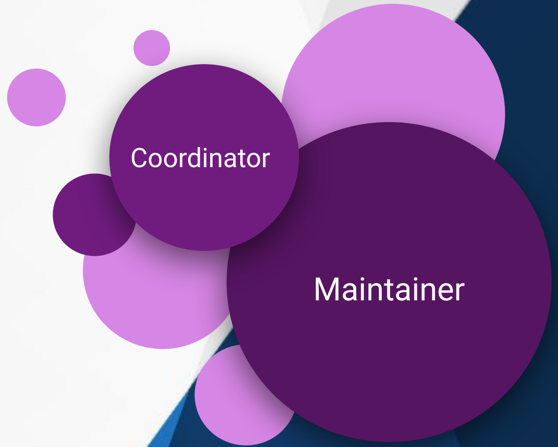
Two-party (bilateral)



Multi-party



Coordinated



https://www.first.org/global/sigs/vulnerability-coordination/multiparty/guidelines-v1.1

Benefits for OSS of CVD

Adding vital skills/capacity to the remediation process

Broader regression testing/patch review prior to PD

Ecosystem can prepare and stage patches and documentation prior to PD so that **ALL downstream consumers have access to fixes** **AT THE SAME TIME**

Downstream stakeholders get notification when patches are released

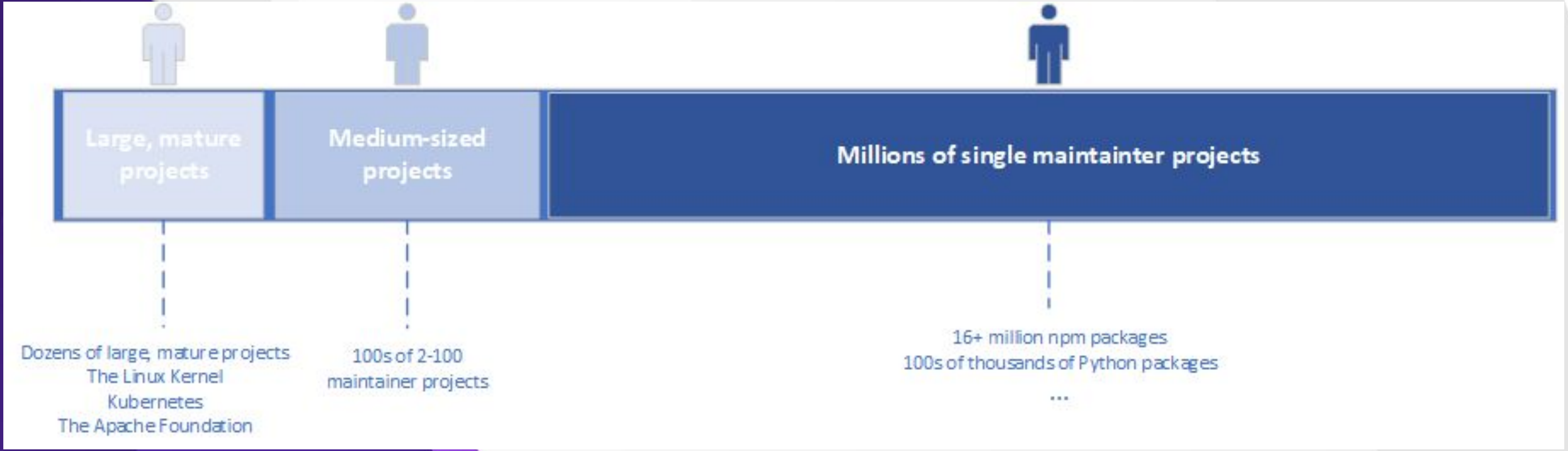
OpenSSF

OPEN SOURCE SECURITY FOUNDATION

What makes OSS CVD hard?

Some OSS "Fun Facts"

There is NO such thing as THE Open source


Large, mature projects
Medium-sized projects
Millions of single maintainter projects
Dozens of large, mature projects
The Linux Kernel
Kubernetes
The Apache Foundation
100s of 2-100 maintainer projects
16+ million npm packages
100s of thousands of Python packages
...

https://anchore.com/blog/open-source-is-bigger-than-you-imagine

MOAR OSS "Fun Facts"

OSS Projects have many different types of goals

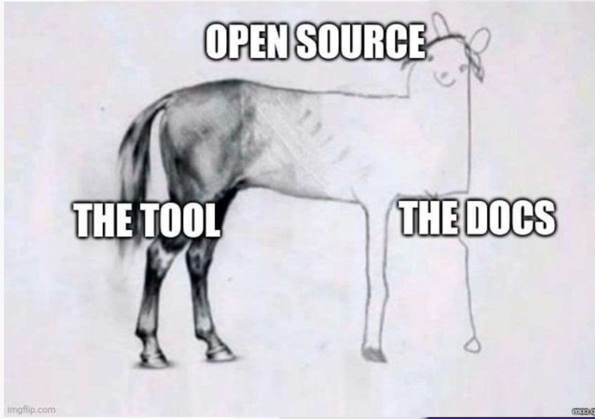


Image Source

Solving a problem

Building a community/helping others

Seeking recognition from peers

Academic project

Learning a new skill

Feel that contributions help their career

"Since I use this piece of FOSS, I feel I should contribute back."

"I believe in the mission of FOSS or the particular area I contribute to!"

Paid by some company for some reason to work on FOSS

https://www.linuxfoundation.org/blog/open-source-maintainers-what-they-need-and-how-to-support-ther

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Why is OSS CVD hard?

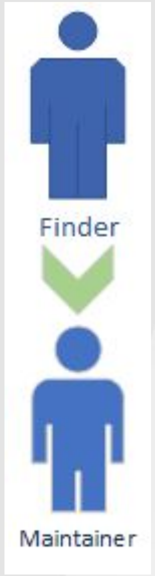
Determining how to contact a project is **complicated**

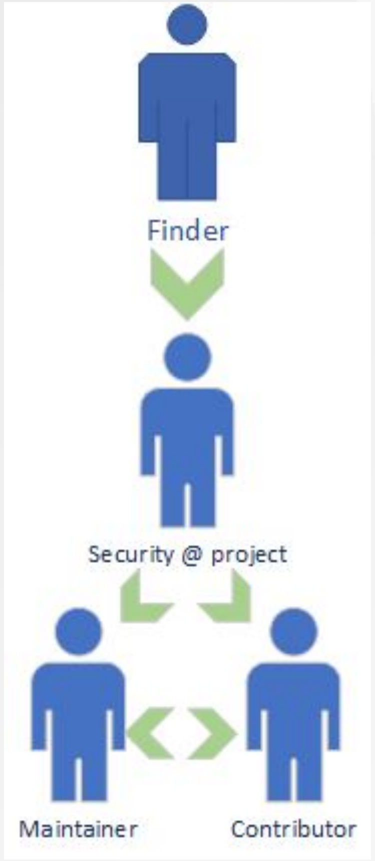
FINDING the appropriate maintainers can be challenging; sometimes upstream **no longer maintains** software

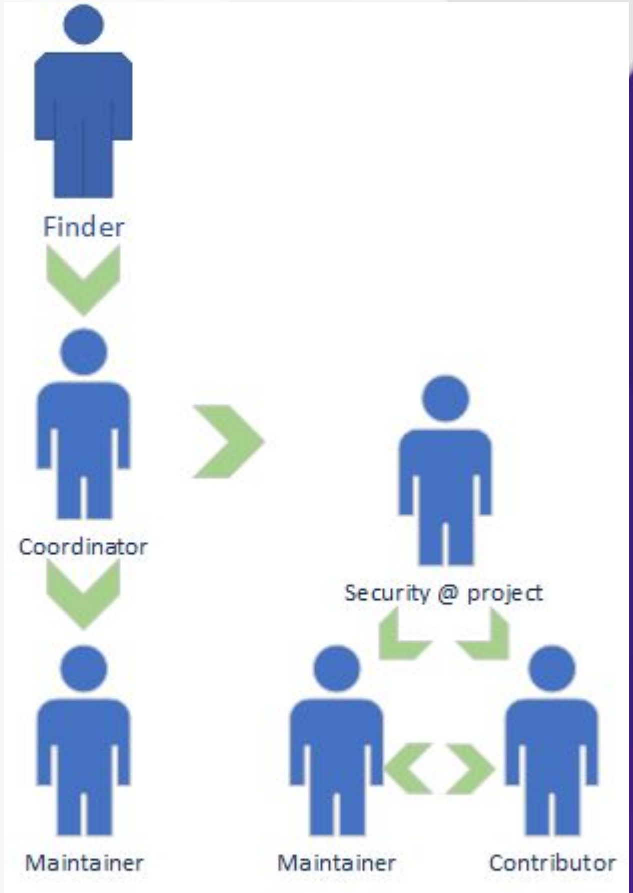
CVD can move **slower** than OSS devs are used to

Project IR **capabilities or processes** may be lacking

Reporting to upstream can be hard







OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Upstream, Downstream

Upstream

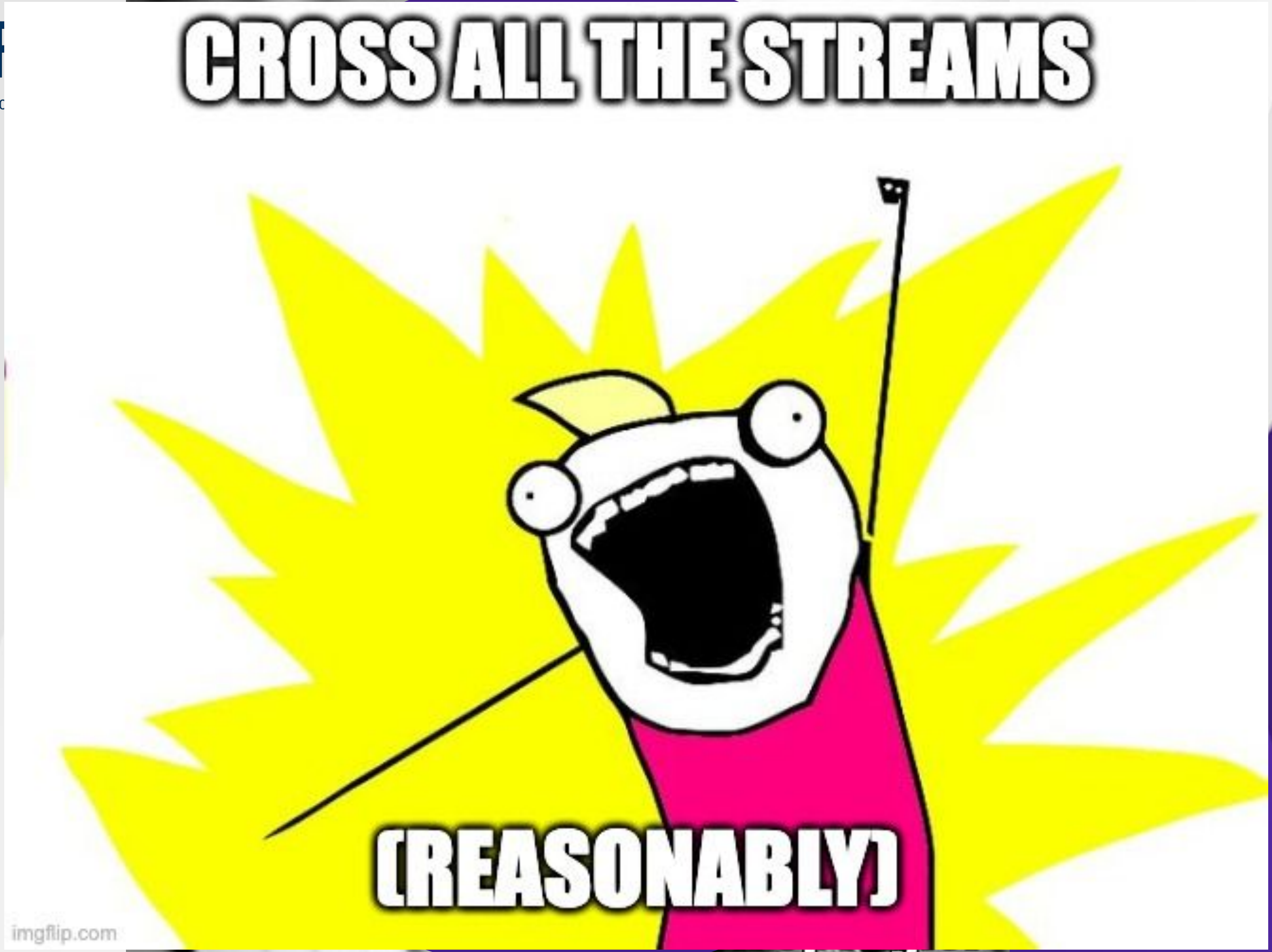
The creator or supplier of software, components or libraries

Downstream

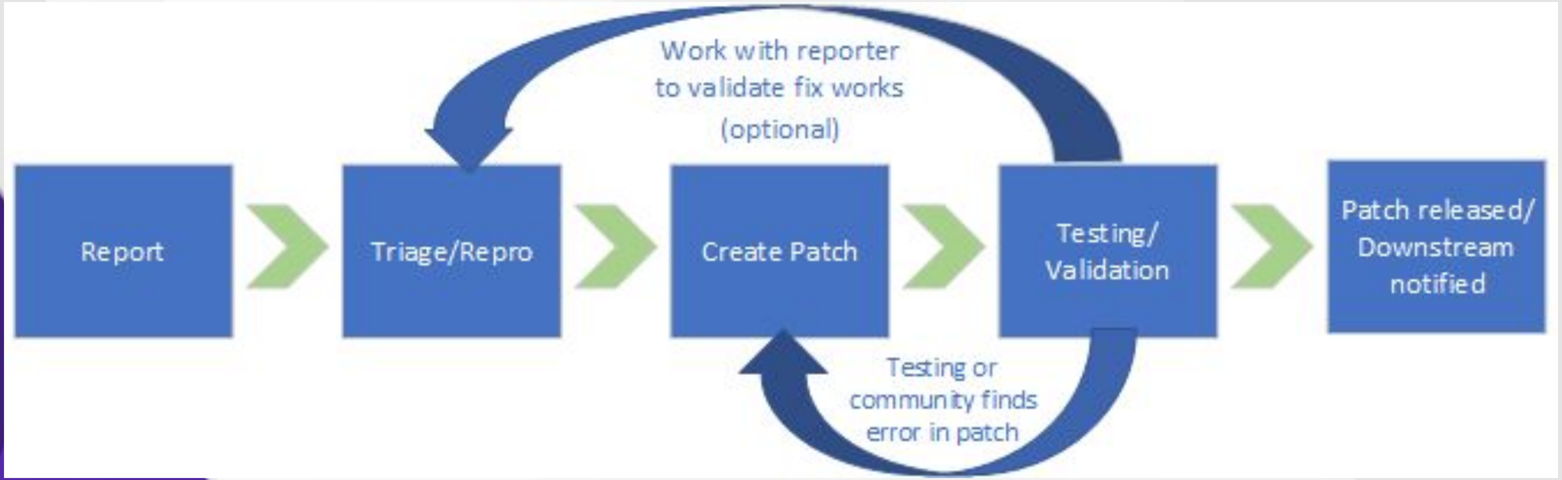
The consumer of software they did not create themselves

Image Source

OpenSS
OPEN SOURCE SECURITY FOUNDATIO



Generic OSS bug intake process


Work with reporter to validate fix works (optional)
Report
Triage/Repro
Create Patch
Testing/ Validation
Patch released/ Downstream notified
Testing or community finds error in patch

This is all done within systems & processes designed for developer velocity and public transparency

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Why is ~~OSS~~ CVD hard?

Vulnerability disclosure is a human process!

Disclosures can go awry for human-related reasons, like **unavailability** or **inability** to handle or **emotions**

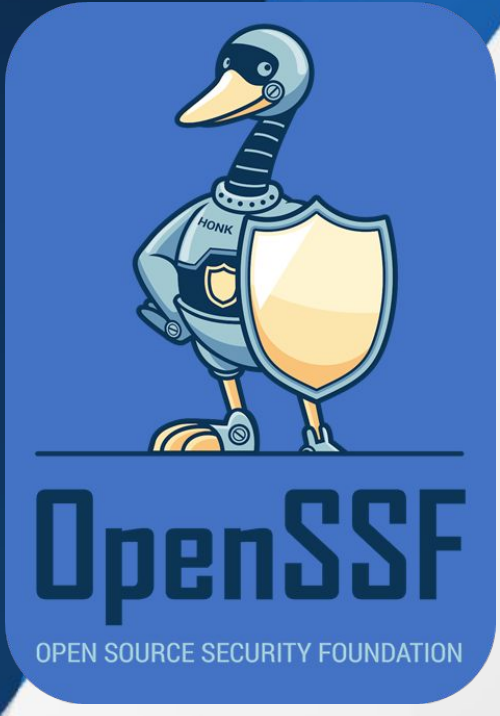
Understanding **motivations** can help drive CVD decisions

Is *anyone* trying to **help**?

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

The Rising Tide lifting all boats

- The OpenSSF is a cross-industry collaboration that brings together leaders to improve the security of open source software (OSS) by building a broader community, targeted initiatives, and best practices
- The OpenSSF brings together open source security initiatives under one foundation to accelerate work through cross-industry support. This is beginning with the Core Infrastructure Initiative and the Open Source Security Coalition, and will include new working groups that address vulnerability disclosures, security tooling and more.
- OpenSSF is committed to collaboration and working both upstream and with existing communities to advance open source security for all.



https://openssf.org/

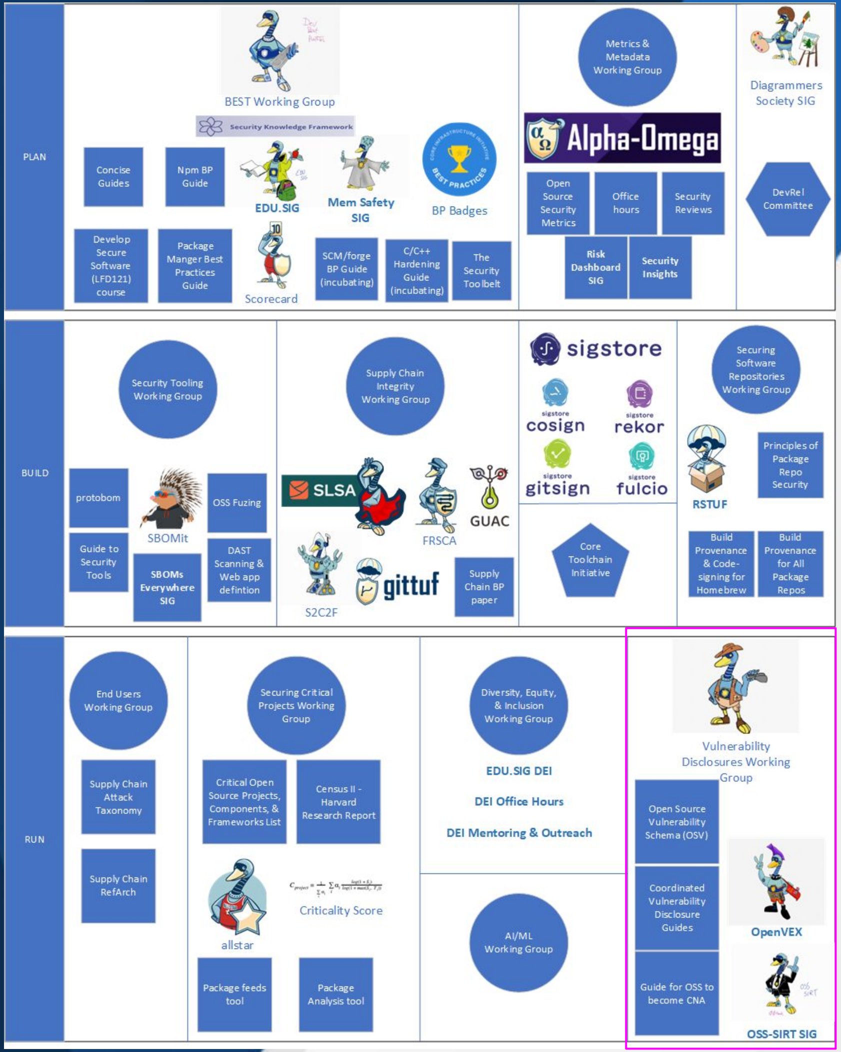
OpenSSF
OPEN SOURCE SECURITY FOUNDATION

A Gaggle of Geese!

We have numerous software projects, guidelines, specs, and experts to help both upstream OSS developers **AND** downstream OSS consumers

https://openssf.org/community/openssf-working-groups/

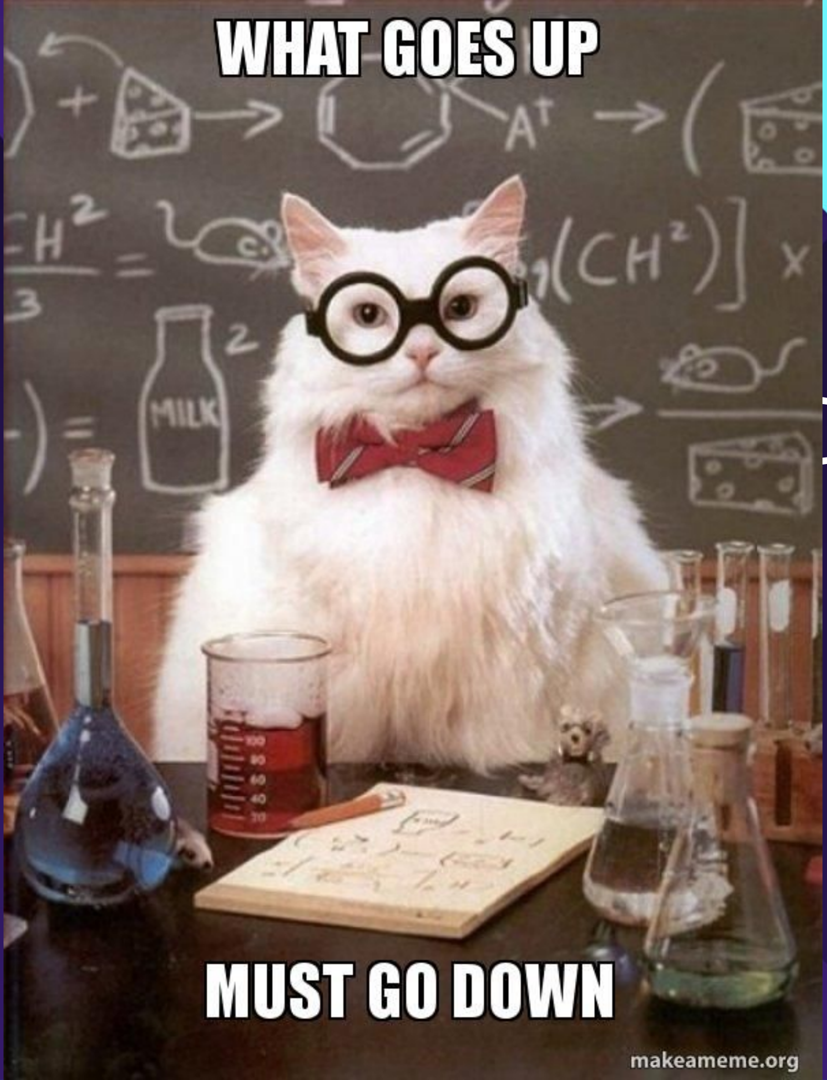
What's up with all the Geese?


PLAN
BEST Working Group
Security Knowledge Framework
Concise Guides
Npm BP Guide
EDU.SIG
Mem Safety SIG
BP Badges
Develop Secure Software (LFD121) course
Package Manger Best Practices Guide
Scorecard
SCM/forge BP Guide (incubating)
C/C++ Hardening Guide (incubating)
The Security Toolbelt
Metrics & Metadata Working Group
Alpha-Omega
Open Source Security Metrics
Office hours
Security Reviews
Risk Dashboard SIG
Security Insights
Diagrammers Society SIG
DevRel Committee
BUILD
Security Tooling Working Group
protobom
SBOMit
OSS Fuzing
Guide to Security Tools
SBOMs Everywhere SIG
DAST Scanning & Web app defintion
Supply Chain Integrity Working Group
SLSA
FRSCA
GUAC
S2C2F
gittuf
Supply Chain BP paper
sigstore
cosign
rekor
gitsign
fulcio
Core Toolchain Initiative
Securing Software Repositories Working Group
RSTUF
Principles of Package Repo Security
Build Provenance & Code-signing for Homebrew
Build Provenance for All Package Repos
RUN
End Users Working Group
Supply Chain Attack Taxonomy
Supply Chain RefArch
Securing Critical Projects Working Group
Critical Open Source Projects, Components, & Frameworks List
Census II - Harvard Research Report
allstar
Criticality Score
Package feeds tool
Package Analysis tool
Diversity, Equity, & Inclusion Working Group
EDU.SIG DEI
DEI Office Hours
DEI Mentoring & Outreach
AI/ML Working Group
Vulnerability Disclosures Working Group
Open Source Vulnerability Schema (OSV)
Coordinated Vulnerability Disclosure Guides
Guide for OSS to become CNA
OpenVEX
OSS-SIRT SIG

OpenSSF

OPEN SOURCE SECURITY FOUNDATION

How CVD?



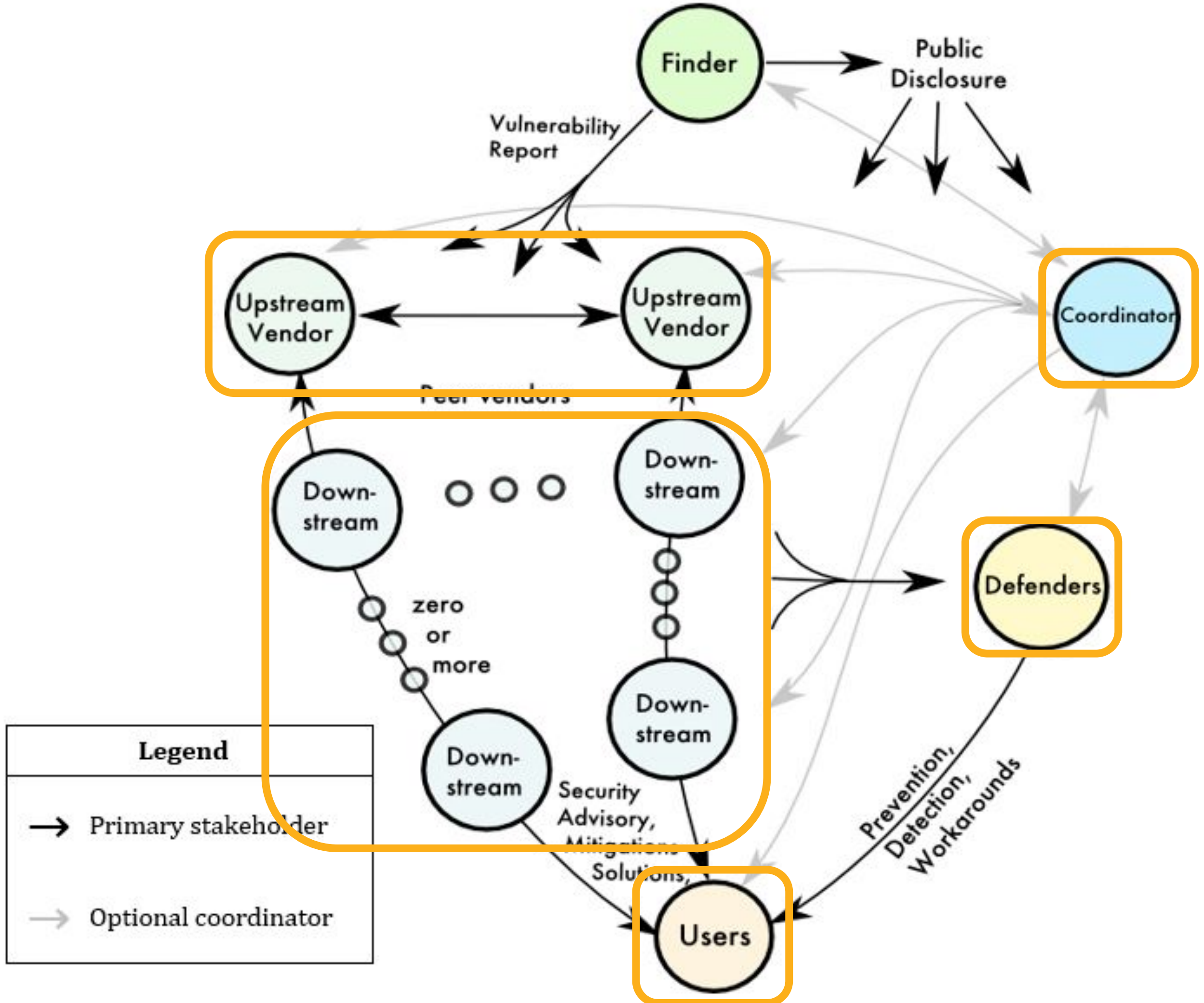
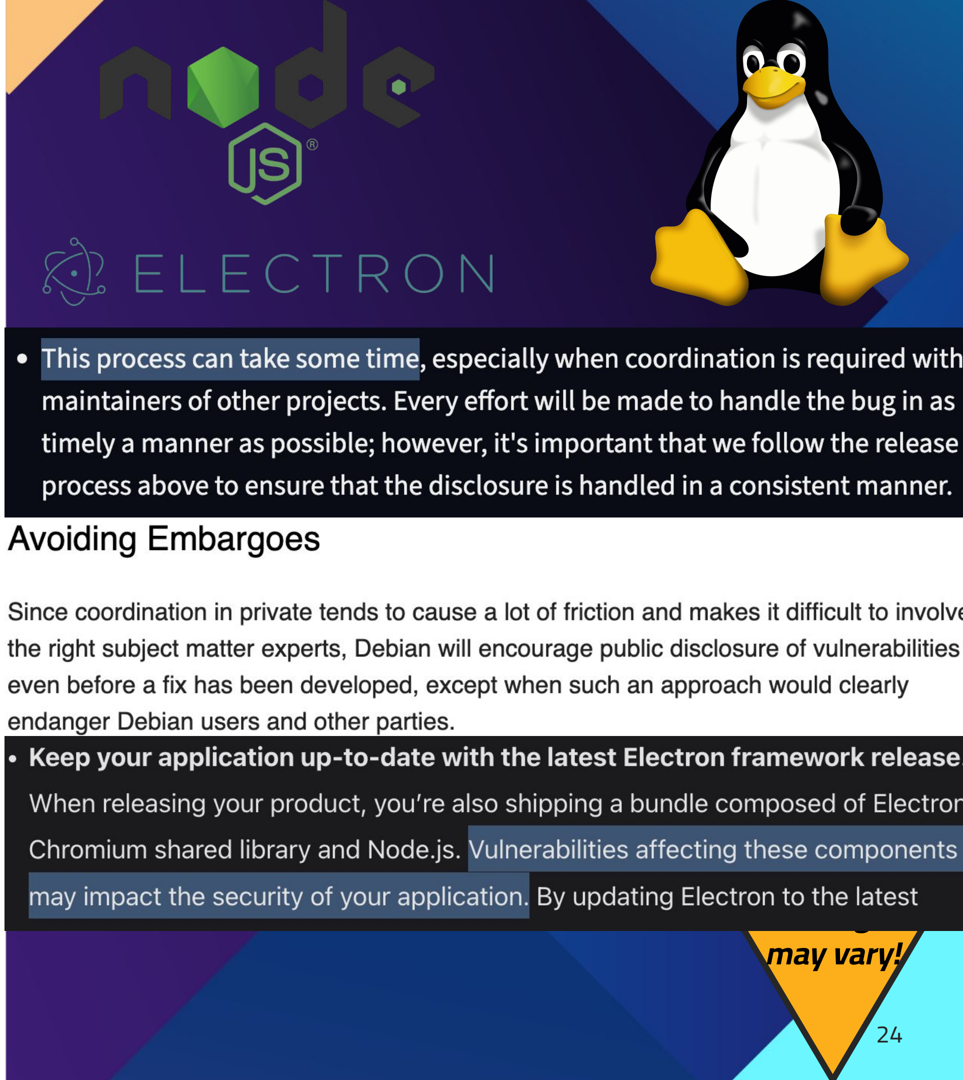


Image Source: [FIRST Guidelines & Practices for Multi-Party Vulnerability Coordination & Disclosure]

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

What can this look like?

- **Challenge:** Determining *who* + *how* to contact downstream
  - Establish relationships beforehand (e.g. runtimes + frameworks)
  - Establish bare minimum + notify all at PD (Kernel)
- **Challenge:** speed, timelines and notifications
  - Elongated embargo timelines (NodeJS)
  - Avoiding embargo timelines (Debian)
- **Challenge**: project needs resources
  - Include upstream patches downstream (Electron)
  - Open security handling up to volunteers (NodeJS)



- This process can take some time, especially when coordination is required with maintainers of other projects. Every effort will be made to handle the bug in as timely a manner as possible; however, it's important that we follow the release process above to ensure that the disclosure is handled in a consistent manner.

Avoiding Embargoes

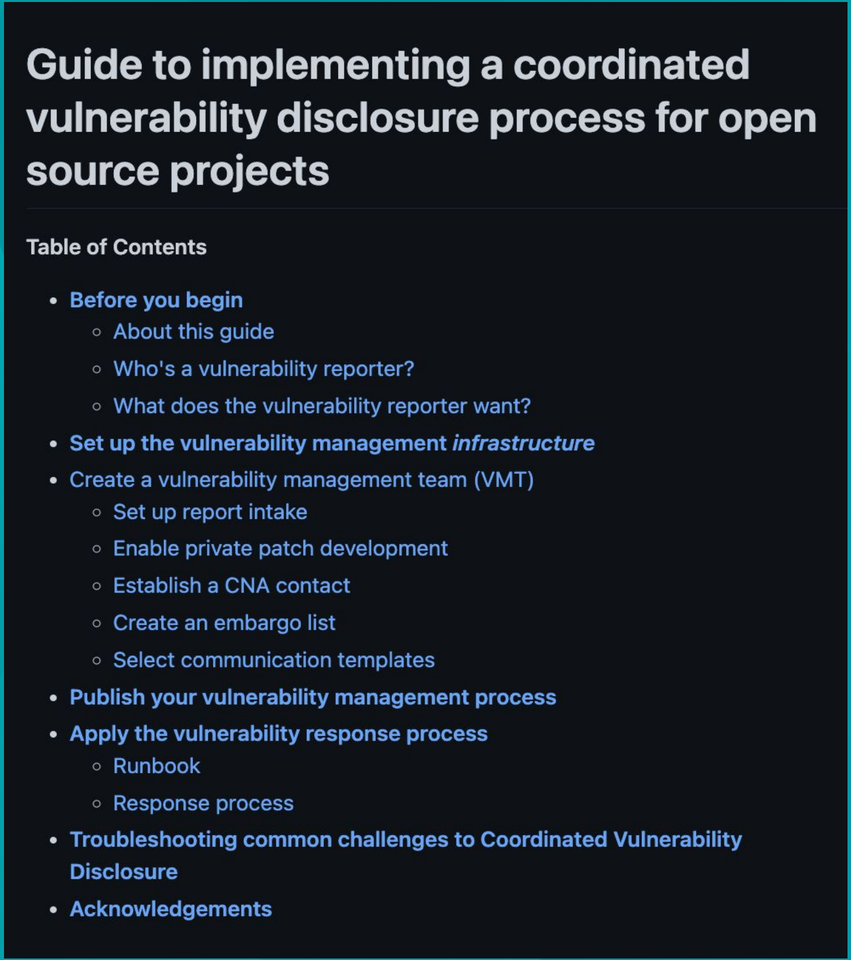
Since coordination in private tends to cause a lot of friction and makes it difficult to involve the right subject matter experts, Debian will encourage public disclosure of vulnerabilities even before a fix has been developed, except when such an approach would clearly endanger Debian users and other parties.

- **Keep your application up-to-date with the latest Electron framework release**
  When releasing your product, you're also shipping a bundle composed of Electron Chromium shared library and Node.js. Vulnerabilities affecting these components may impact the security of your application. By updating Electron to the latest

The OpenSSF's Vulnerability Disclosure Working Group focuses on these problems.

Collectively the group represents developers, suppliers, security researchers, incident responders, coordinators, and vulnerability management practitioners from around the globe.

The group supports and maintains tooling, templates, and best practices guides to help ALL parties involved in CVD within OSS.

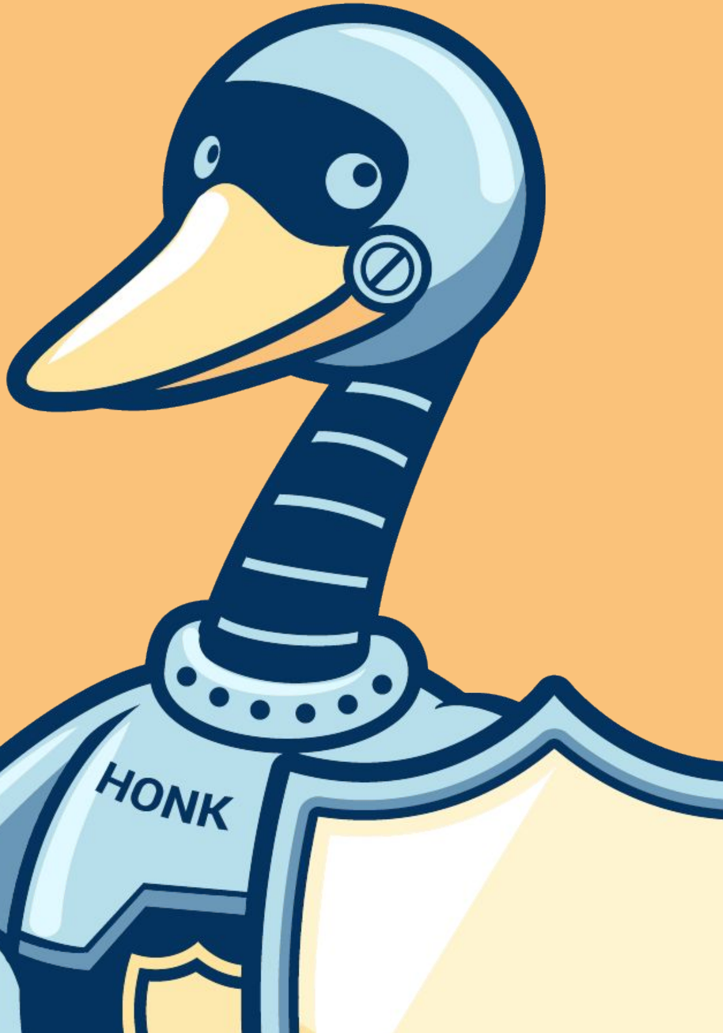


Guide to implementing a coordinated vulnerability disclosure process for open source projects

Table of Contents

https://github.com/ossf/oss-vulnerability-guide/blob/main/maintainer-guide.md
https://github.com/ossf/oss-vulnerability-guide/blob/main/finder-guide.md

Here are some things you to do to set up your project for CVD-success!



Upstream: Publish your vuln mgmt process/security policy

- Each project operates differently and has different needs.
- Tell people how you want to handle security reports and how they will be managed.

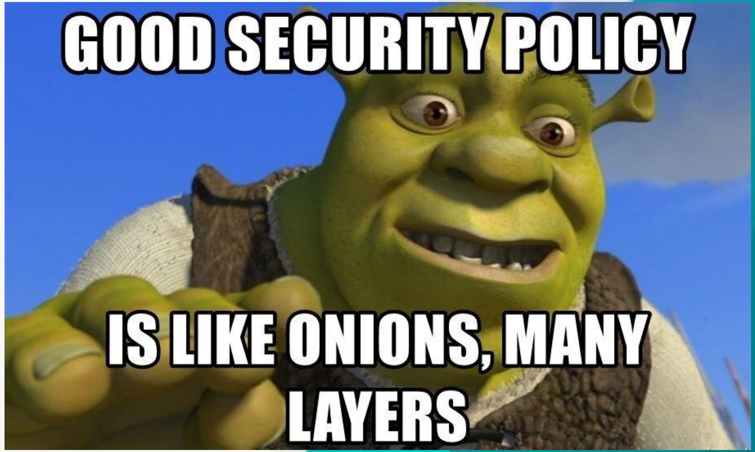


Image Source

Downstream: Look for "security@project.com" addresses, "security.md" and/or "security.txt" files. Use tools like Security Insights to learn about how the project does security

https://docs.github.com/en/code-security/getting-started/adding-a-security-policy-to-your-repository
https://gitlab.com/gitlab-org/gitlab-foss/-/blob/read-template-from-repository/doc/release/security.md

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Upstream: Establish your Security “team”

- Not every developer is a securityologist.
- Identify people in your project that might have these skills or find some security friends that can help in times of need!

Downstream: Understand the capabilities of the upstream software you consume. Use tools like Scorecard or Allstar to learn about critical practices your software may be using.

Image Source

Upstream: Establish a CNA contact (or other means of vuln id disclosure)

- It is important that as vulns are found and fixed that your downstream is told about it so they can take action.
- A CVE Numbering Authority (CNA) is a party authorized to issue CVE IDs for a particular scope of hardware/software, and is the most common way organizations communicate about vulnerabilities.
- There are other ID methods such as GHSA, OSV or GSD that are also CVE-compatible and OSS-workflow-friendly.
- Consider becoming a CNA to better manage your intake of vulns : Guide for OSS Projects to become a CNA

Image Source

Downstream: Know where to get alerts for CVEs discovered in the software you consume

Upstream: Setup a means for private report intake, private patch development, private testing & embargo lists

- Establishing a private way that a Finder can share details or reproducers with the project helps ensure bad actors don't learn about the problem before the project or the users.
- Like private reporting, it is important that patches that address the vulnerability be kept out of mainline code branches until after they have been tested and are ready for public disclosure.
- Bad actors monitor source code repositories for "interesting" (i.e. security-related) PRs and commits.
- Depending on the size and scale of the project, you may need to have a pre-authorized list of people/projects that either contribute to your project or are vital to your supply chain.

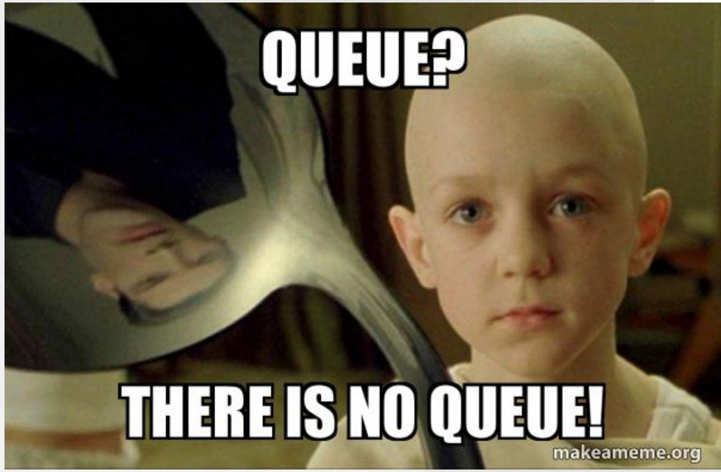


Image Source

Downstream: Not every project has this capability, monitor projects you consume that do not to avoid being "0-day'ed"

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Determine how you will communicate the disclosure

- Mailing list, blog, commit comment, VEX statement, full-on security advisory - there are many ways to tell your downstream that there was vulnerability and how to fix it.
- Consider to disclose to places like *oss-security* at PD for broader visibility.

Image Source

Downstream: Subscribe to project release or security mailing lists or blogs, monitor oss-security mailing list, use OSV to track oss vulns and malicious packages

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Be an informed OSS Consumer

Scorecard - project metrics

allstar - project metrics

GAUC - dependency graphs

SLSA - Supply Chain capabilities framework

sigstore - code signing

BP Badge - SDLC verification/attestation

OpenVEX/VEX - statements on the affectedness of software to vulns

OSV - OSS Vuln tracking

SCM Security BP Guide - Configure you source code repo securly

Concise Guide to Software Consumption - understand what to look at when ingesting 3rd-party sft

Consider giving back to projects that are important to YOUR business

https://github.com/ossf/

What questions do you have?

What additional resources from the WG would you find helpful?

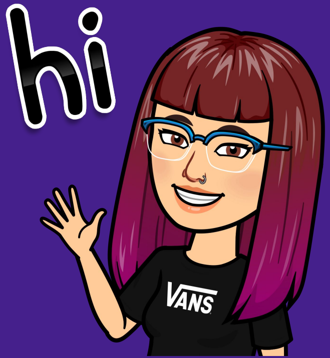
Image Source

OpenSSF
OPEN SOURCE SECURITY FOUNDATION

Ways to Participate

- Join the OpenSSF Mailing List
- Follow us on Twitter
- Follow us on LinkedIn
- Follow us on Mastodon
- Follow us on Facebook
- Subscribe to our YouTube Channel
- Join a Working Group/Project
- Access the Public Meetings Calendar
- Participate on Slack
- Follow OpenSSF on GitHub
- Become an Organizational Member

Thank You



@taladrane

@taladrane@fosstodon.org

https://github.com/taladrane

https://www.linkedin.com/in/madisonoliver24

CRob_at_Intel_dot_com

@SecurityCRob

@SecurityCRob@infosec.exchange

https://github.com/SecurityCRob

The Security Unhappy Hour,
Chips & Salsa

https://www.linkedin.com/in/darthcrob/